

Testes - Exercício Semana 3

Contexto	1
Exercício 1 (aquecimento)	2
Exercício 2	2
Exercício 3	2
Entregas	3
Primeira entrega	3
Segunda entrega	3
Conclusão	3

Contexto

Nesse exercício, vocês estão recebendo um servidor, que se encontra na pasta `serverTests`, ele deve ser usado como base para o exercício dessa semana.

Esse servidor segue a ideia da Onion Architecture, que vocês podem ler mais [clikando aqui](#), onde nós temos:

- Repositories são os *Domain Model* do documento
- Services são os *DomainServices* do documento
- Controllors são os *ApplicationS Services* do documento
- Providers são os arquivos responsáveis por fazer a injeção de dependência dos códigos.
- Models são os models do mongoose para se comunicar com o Mongo

Tendo isso em mente, gaste um tempo entendendo o que cada parte da pasta `serverTests/src` significa e seu papel no servidor como um todo. Tente entender o que cada parte está fazendo, é bem simples.

Esse servidor não foi feito para ser rodado live, eu não gastei tempo garantindo que ele seja um executável, mas ele tem tudo o que se precisaria para isso, se não acontecer é por um ou outro problema técnico, mas ele está pronto para o nosso foco: **testar um possível caso real**.

No mesmo nível da pasta **src** nós temos a pasta **test**, nela iremos concentrar todos os nossos testes. Ao abri-la você encontrará duas pastas:

1. fixtures

- a. Responsável por guardar todos os valores mockados do sistema, de forma organizada e com os arquivos separados por contexto

2. unit

- a. Responsável por guardar todos os nossos testes unitários. A estrutura de pastas deve espelhar a pasta **src**.
- b. Dentro dessa pasta vocês encontrarão vários testes já feitos para o servidor.

Para rodar os testes, simplesmente vá para a pasta `serverTests/server` e rode **yarn test** (depois de ter dado `yarn install`). Verá que já existem 5 testes no total.

Exercício 1 (aquecimento)

O primeiro exercício é bem simples, iremos testar o arquivo que se encontra em `serverTests/server/src/Utils/Controller/UserControllerUtils.js`. Ele tem apenas uma função e ela deve ser testada com a chamada do comando **yarn test**, junto com os outros testes.

Esse exercício é bem simples, faça quantos testes achar necessário. Pelo menos um é obrigatório. Use esse teste para entender como o mocha está achando seu teste, e como ele tá executando.

Exercício 2

Há pelo menos um teste faltando na função `createUserAction` dentro do `UserController`, você deve identificar qual é e fazê-la. Caso ache mais de um possível teste que sinta que ainda precisa ser feito, sintase livre para fazer, mas pense bem: **em um cenário real você não sairia testando tudo**, portanto pese bem o que acha que deve ou não ser feito.

O novo teste deve ter no seu texto explicativo as palavras “NOVO TESTE”.

Exercício 3

O terceiro exercício consiste em fazer um novo endpoint, que será usado para **update do usuário**. Faça os tratamentos que achar necessário para esse endpoint, adicionando ou

não validações, o que é obrigatório é dividir seu código entre as três camadas: **controller**, **service** e **repository**.

Os testes tem que ser unitários, portanto:

- não podemos chamar o *service* ao testar o *controller*
- não podemos chamar o *repository* ao testar o *service*
- não podemos chamar o *banco (mongo)* ao testar o *repository*
- caso a gente construa qualquer código de utils, ele deve ser inserido como dependência e não pode ser chamado

Os únicos códigos que devem ser chamados são da função testada e daquele contexto.

Para evitar a chamada das funções reais, façam uso do *stub* do *sinon*, para vocês mockarem (fingirem) a chamada das funções. Mais sobre a doc do *sinon.stub*, [clique aqui](#).

Entregas

Os exercícios serão entregues em duas partes. Ambos devem ser enviados para o meu email: **victor.heclis@cursoenfase.com.br**

Primeira entrega

A primeira entrega consiste do **exercício 1 e 2**, juntos. Eles devem ser entregues até o dia **15/12**.

Segunda entrega

A segunda entrega é do **exercício 3**, somente, sendo entregue até dia **22/12**.

Conclusão

Assim, terão 2 semanas para fazer todos esses testes. Recomendo fazer o mais rápido possível a primeira entrega, pois ao final dela iremos juntar para conversar sobre ela e tirar possíveis dúvidas, principalmente sobre o **exercício 2**, que podem ajudar na resolução da segunda entrega (exercício 3).

Sendo assim, a conversa fica para, no máximo, dia **16/12**, onde iremos tirar as dúvidas que surgirem daqueles que conseguirem entregar no prazo. Caso **todo o time entregue antes, adiantamos a conversa**.

Os prazos são um pouco apertados mas é para conseguirmos fazer antes de boa parte do time entrar de férias, o prazo de entrega da segunda entrega pode ser estendido até o ano que vem caso vejamos a necessidade, mas quem quiser entregar antes já poderá receber a correção.