

## Criando nosso e2e

### Transcrição

[0:00] Entendemos a configuração do Protractor, entendemos também tudo das nossas instalações e do próprio projeto.

[0:08] O que a gente vai fazer agora é criar o nosso primeiro teste com o Protractor.

[0:14] Então eu vou abrir o visual estúdio e a gente vê dentro da pasta SRC, a pasta searce, que ele já tem dois arquivos aqui como a gente viu na aula de Protractor.config.

[0:30] Só que eu não quero utilizar esses app configs.

[0:33] Ou melhor, eu não quero utilizar esses app.po.ts, nem esse e2e-spec.

[0:40] Eu quero criar o meu próprio.

[0:44] No caso aqui, eu vou deixar eles de lado, eu não vou nem deletar ele a princípio, eu vou deixar eles de lado mesmo só para que a gente às vezes tome como base alguma coisa ou outra, e vamos criar o nosso próprio.

[0:55] No caso, a nossa primeira tela que o usuário vai acessar vai ser qual?

[0:59] Vai ser a tela de login.

[1:02] Então o que a gente precisa ver?

[1:06] Nós precisamos ver o nome do componente que está dentro da pasta searce para a gente sempre ter a mesma nomenclatura em relação ao nome que foi dado na criação do projeto, no desenvolvimento, e o nome que a gente vai dar na criação dos nossos testes, nos arquivos dos nossos testes.

[1:28] A gente tem a pasta app aqui, tem a pasta home e dentro dessa pasta home aqui a gente tem esse signin, S I G N I N, que aparentemente tem esse loginForm e é esse componente mesmo que a gente vai precisar utilizar.

[1:50] Então vamos lá no e2e, src na pasta searce e vamos criar, dou um botão direito dentro da pasta searce do e2e, e eu clico aqui no new file.

[2:05] New file, vou digitar signin, S I G N I N, e vou utilizar a mesma terminologia que foi mostrado para a gente no protractor.config.js.

[2:21] Então é e2e-spec, spec.ts.

[2:31] Criamos aqui então o nosso arquivo signine2e.

[2:37] Esse spec aqui vai utilizar muita coisa do Jasmine.

[2:43] Na verdade, grande parte das coisas que vamos fazer aqui vai ser tudo com a base do Jasmine.

[2:51] Só para mostrar, por isso que eu deixei de lado esse app, appe2e, para que a gente veja um pouco da estrutura do Jasmine.

[3:01] Nós vamos utilizar três métodos que ele nos traz, que é qual?

[3:07] O describe, o beforeEach e o it, e spec também.

[3:14] Vamos utilizar esses daqui.

[3:18] Traduzindo ao pé da letra, esse describe vai descrever qual que é o componente que a gente quer testar, no caso aqui ele está dizendo workspace project App.

[3:31] Aqui ele tem o beforeEach, ou seja, antes de começar todos os nossos, digamos, its, ele vai declarar, vai executar, vai fazer tudo o que a gente colocar aqui para que ele faça.

[3:46] E esses its aqui, traduzindo ao pé da letra também essa mensagem, vai ser o nosso deve fazer alguma coisa, deve trazer alguma coisa, deve retornar algum parâmetro.

[4:02] Então essa é a estrutura que nós vamos utilizar.

[4:06] Agora, a gente não precisa mais dele, já entendemos o que ele faz, eu vou fechar ele aqui e vou criar o nosso próprio.

[4:13] Describe, esse describe recebe dois parâmetros como a gente viu.

[4:19] Um é string e outro é uma função.

[4:25] Criei a função aqui, uma arrow function.

[4:30] Nesse describe, nesse primeiro parâmetro do describe a gente vai dizer o que o nosso, digamos assim, escopo inteiro aqui vai ter que executar.

[4:45] De maneira mais macro, não tão micro quanto aqueles its que a gente viu no arquivo app.e2e.

[4:51] Nesse describe aqui, então, como estamos falando sobre o signin, sobre a parte de login a gente pode colocar ou testando componente da home, testando componente login, testando página signin, testando o que for mais intuitivo para você na hora que for executar o teste, e lembrando que a gente sempre faz o nosso código, sempre faz o nosso teste para que outras pessoas possam ler.

[5:22] Então nessas mensagens, lembre-se de deixar sempre o mais intuitivo possível.

[5:28] No caso desse describe, vamos colocar aqui testando tela.

[5:41] Como aqui ele está abaixo do home, seria interessante a gente colocar testando tela home, porque eu acredito que como a nossa estrutura do nosso projeto aqui está home, aí tem o signin, signup, que signup vai ser de criação de usuário, como está tudo abaixo de home é interessante a gente colocar aqui a home e a gente vai chamar os page objects que a gente vai criar de signin e de signup.

[6:11] Tanto de criação de usuário, tanto de login.

[6:17] Vamos colocar aqui o beforeEach.

[6:20] Se você estiver usando o Visual Studio Code, ele já mostra várias métodos que você pode utilizar.

[6:28] No caso, a gente vai utilizar o beforeEach e da mesma maneira, ele tem dois parâmetros, não, nesse caso aqui ele só tem um parâmetro, na verdade, que é uma função que vai ser passada e, no caso, no nosso app app.e2e, ele mostra que ele

foi declarado um page object.

[6:51] Como a gente não tem nenhum ainda, não precisamos nos preocupar com isso.

[6:57] E vamos criar o nosso primeiro it.

[7:00] Bom, o nosso primeiro it precisa ser, todos, na verdade, não só o primeiro, todos os its que a gente vai criar vai ser referente a alguma funcionalidade específica da página.

[7:12] Ou, não precisa ser unicamente uma função, não está limitado à quantidade de funções isso daqui.

[7:26] Aqui eu vou descrever o que esse it vai fazer, ou seja, ele deve retornar alguma coisa para a gente, ele deve esperar que o componente tal ou o imput tal tenha nomenclatura X.

[7:40] No caso, o que a gente precisa testar?

[7:42] A tela de login.

[7:44] Eu não tenho o meu page object ainda para ser chamado.

[7:49] Então esse it para a gente no momento talvez não faça sentido a gente digitar alguma coisa.

[7:56] O que a gente vai precisar fazer?

[7:57] Criar um page object aqui como o app.po tem.

[8:03] Ou melhor, como o app.e2e tem o page object específico dele.

[8:10] Vamos criar o nosso page object aqui do signin, signin.po.ts.

[8:22] E a gente já pode chamar ele aqui, import from, para dar o from a gente abre aspas, dá um ponto, traço e digita qual que é o arquivo que a gente quer chamar.

[8:42] No caso é o signin.po.ts, e dá um ponto e vírgula.

[8:54] Eu vou só aqui no PO dizer o export class, e aqui eu vou digitar signin.page, e abro a função.

[9:09] Esse page é importante que a gente declare ele, ou seja, deixe no sufixo porque o page indica que essa classe aqui é um page object.

[9:18] Por padrão de mercado, a gente utiliza sempre o sufixo page para indicar page objects.

[9:26] Criando aqui, vamos exportar ele dizer que esse signin.po recebe esse signin page.

[9:36] Voltei para o e2e, copieie e está tudo certo.