

Concluindo os testes

Transcrição

[0:00] Para finalizar agora, vamos criar a nossa e2e das fotos.

[0:06] Então dentro da pasta sources, da SRC, vamos dar new file e photo.e2e-spec.ts.

[0:20] E vamos criar a nossa mesma estrutura de e2e, que é no caso describe, Testando funcionalidade de Photos.

[0:35] Criamos a função aqui e um beforeEach.

[0:46] Vamos dar um import aqui dos nossos page objects, import from pageObjects photos e vou dar um import no photo-form PO, importei essa classe e abaixo dele vou fazer o import do outro page object.

[1:14] From pageObject photos, page object list.

[1:24] Importei agora e vou declará-los aqui dentro do describe, que é o quê?

[1:31] Let photoFormPage, que é do tipo PhotoFormPage e o PhotoForm, ou melhor, o PhotoListPage, que é do tipo PhotoListPage.

[1:52] E dentro do beforeEach vou declará-los como uma nova classe.

[2:02] E o PhotoFormPage é igual a new PhotoFormPage.

[2:09] E o PhotoListPage é igual a new PhotoListPage.

[2:20] Vou dar um CONTROL B aqui para aumentar a área aonde codificamos.

[2:27] E vou começar agora a criar os nossos its.

[2:29] Primeiro parâmetro o que ele vai fazer, e o segundo, a função do que ele vai executar como teste.

[2:37] Então o que primeiro precisamos fazer?

[2:38] Cadastrar uma imagem.

[2:42] Então deve, ou melhor, vamos navegar para a tela de cadastro.

[2:48] Então navegar, deve navegar para /p/add, que é a nossa rota, nosso login.

[3:03] PhotoForm, já completou sozinho para mim.

[3:07] PhotoFormPage navegar para photo.

[3:14] Aqui eu só vou criar as funções, eu não vou fazer nenhum comparativo apenas para irmos mais rápido com os nossos testes, mas o ideal é sempre você fazer um comparativo tipo expect to equal, que eu vou fazer agora também aqui.

[3:30] Aqui estava navegando, então aqui eu vou cadastrar.

[3:33] Deve cadastrar imagem.

[3:40] E crio a função.

[3:43] Essa função vai receber `photoForm.carregarImagem`, e depois ele vai no `photo form` fazer um pegar botão `upload`, e dentro dele vou dar um clique.

[4:04] Então até agora o que fizemos?

[4:07] Chamando `formPhoto`, navegamos até uma página e aqui depois disso, que ele carregou, navegou até aquela página, ele deve cadastrar uma imagem.

[4:17] Essa imagem deve conter um `carregar imagem`, eu apenas carrego a imagem e ele pega o botão de `upload` e dá um clique.

[4:30] Vamos fazer mais uma coisa aqui.

[4:33] Vamos lá nessa `photoFormPage`, e como precisamos fazer, digamos assim, uma pesquisa pelo nome do comentário, o que é o ideal a gente fazer?

[4:45] Nós criarmos, ou melhor, `photoPage`, vamos até a `photoFormPage.po` e vamos aqui para `carregar`, `comentar imagem`.

[5:05] Viemos aqui no `element by.CSS`, vamos dar um `CONTROL P` para procurar o nome daquela imagem, que no caso está dentro de `photo form.component.html`, e vamos registrar o comentário aqui com esse `formControlName`, que é a `description`.

[5:33] Voltamos para o `photo-form.po`, tiramos esse `CSS` aqui, `formControlName`, e colocamos o `description`.

[5:46] Esse `description` vai receber aquele mesmo comentário que vamos enviar para outra tela.

[5:54] Para outro `page object`.

[5:55] Então comentário, e ele recebe comentário.

[6:02] Dar um `return` aqui.

[6:07] Comentário.

[6:07] Voltamos para a nossa `photo.e2e-spec` e na chamada `dele`, vamos fazer o quê?

[6:16] `Photo form page.comentarImagem`.

[6:23] Nós passamos um comentário.

[6:25] O ideal é que a gente tenha esse comentário já cadastrado aqui nessa `photo.e2e`.

[6:32] Então o que vamos fazer?

[6:33] Aqui aonde cadastramos, vamos dar um `let` comentário igual a `photo bonita`.

[6:44] Um comentário bem legal.

[6:46] Vamos descer aqui para onde estanciamos esse comentário imagem, e vamos passar esse comentário como um parâmetro.

[6:55] Beleza até aí?

[6:58] Continuando, agora não focando mais no photoFormPage, vamos criar um outro it que vai testar o quê?

[7:06] Busca de imagem.

[7:13] Eu chamo outra função e nela eu vou chamar ao invés de photoForm, page, PhotoListPage.

[7:22] Que ele vai navegar para photoList, aí dentro desse photoListPage ele vai buscar a imagem que cadastramos o comentário agora e vamos passar o comentário para ela.

[7:36] Lembra que quando criamos esse método aqui, vou até dar um clique nele, ele vai receber esse comentário para fazer a busca por essa classe, ou melhor, vai fazer a busca por um elemento através do CSS dele para enviar esse comentário, e com ele vai trazer a imagem específica.

[7:58] Para isso que serviu essa declaração desse comentário aqui nesse photo.e2e.

[8:05] E agora que a gente já mandou o comentário, vamos terminar aqui com esse photoListPage invocando o buscar, ou melhor, que é o pegar imagem, que é esse outro método que criamos aqui.

[8:22] Então com esse código aqui já nós cadastramos a imagem e também fizemos a busca da mesma imagem que criamos.

[8:30] O que falta para a gente?

[8:32] Apenas testar realmente para ver se está tudo certo.

[8:36] Vou voltar aqui para a nossa hierarquia de pastas, e verificar só se a hierarquia está tudo certo.

[8:43] Só para vocês entenderem uma coisa, esse home e2e e esse photo e2e.

[8:47] Talvez você fique com dúvida pensando qual que vai executar primeiro.

[8:52] Porque pensando na estrutura desse cenário aqui, precisamos primeiro logar no sistema para depois fazer uma adição de uma photo ou uma busca de uma photo.

[9:02] Não se preocupa com isso porque aqui bateu certinho os nossos nomes, mas como é que o Protractor faz?

[9:09] Ele viu esses dois nomes aqui desses dois arquivos TS, e ele vê assim: "na ordem alfabética, qual que vem primeiro, o H ou o P?"

[9:21] Primeiro vem o H, então ele vai primeiro fazer esse teste aqui, executar esse e2e, vai percorrer ele e depois ele vem para esse outro teste.

[9:30] Como a gente fez esse teste aqui no home e2e criando um usuário e logando também, eu só vou fazer um ajuste nele para que ele não fique testando o cadastro de um usuário e depois ele logue, para que ele não fique cadastrando toda hora.

[9:46] Só vou fazer esse ajuste para que quando nós testarmos a nossa funcionalidade por completo, ele não fique criando vários usuários ao mesmo tempo.

[9:59] Então só comentar e vou navegar direto para a URL da home.

[10:02] Navegar para home, certo?

[10:08] E o que ele vai fazer aqui?

[10:10] Ele vai para a `signinPage.acessarHome`.

[10:19] Então vamos lá no nosso CMD, nosso Prompt, vou cancelar aqui o que eu estava executando, apenas o nosso projeto, e vou dar um `ng e2e`.

[10:35] Agora esperamos ele executar o projeto para ele executar também os nossos testes, que criamos para ver se está tudo bonitinho.

[10:45] Vamos ver aqui, executou o projeto.

[10:48] Vamos ver.

[10:49] Vai para a tela da home, logou, cadastrou, perfeito.

[10:55] Aqui, o que aconteceu?

[10:57] Ele navegou para a tela da home, ele verificou a URL, ele viu se precisava ou não fazer login e ele foi para essa página de adição de imagem, carregou a imagem e a busca foi feita.

[11:12] Então os nossos testes foram concluídos com sucesso.