

## Conclusão

### Transcrição

[0:00] Parabéns, você acaba de concluir o curso de Protractor.

[0:03] Não esqueça de concluir os exercícios, e qualquer dúvida, poste lá no fórum.

[0:09] Mas vamos rever o que aprendemos no curso de Protractor.

[0:12] Começando pelo Protractor.config.

[0:15] No Protractor.config, configuramos algumas informações relacionado até ao próprio teste automatizado, como, por exemplo, os capabilities.

[0:26] Os capabilities que informamos qual browser que ele ia utilizar, qual o destino do executar do browser e outras informações poderíamos colocar aqui também.

[0:36] Como, por exemplo, o Onprepare, é a função Onprepare que utilizamos para adicionar um locator customizado, como, por exemplo, o form control name que criamos aqui, para que, ao invés de repetirmos varias vezes um determinado código como, por exemplo, a busca de input, criamos apenas um form control name para que quando colocássemos o input, buscássemos apenas pelo próprio locator que criamos.

[1:06] Continuando aqui.

[1:08] A estrutura deles, criamos aqui, a estrutura de Design Pattern focado em page object que cada pagina, cada arquivo que criamos encapsulava uma pagina para que não criássemos demasiadas funções desorganizadas.

[1:24] E com isso, também, criamos uma pagina chamada e2e.

[1:31] Com a e2e utilizamos um pouquinho de Jasmine com as funções do describe, before each e os próprios it's para conectar junto ao Protractor.

[1:42] No Protractor, do Protractor em si, vimos o que?

[1:46] O browser, o browser.get que servia para navegação do nosso teste automatizado, ou seja, se eu quisesse navegar até uma pagina, poderíamos usar esse browser.get e também para encapsulamento ou busca de elementos usávamos o elemento.

[2:08] Esse element ele retornava uma variável ou um parâmetro só, que no caso, nós mesmos citávamos a informação ou qual elemento queríamos.

[2:17] Aqui nesse caso exemplificado utilizamos o by class name, ou seja, ele retornava um element que tinha esse nome de classe, e assim enviávamos a informação através do send Keys.

[2:31] Ou seja, independente do input que tivesse aqui através dessa classe que buscamos, qualquer parâmetro que enviássemos seria através do send Keys.

[2:41] E vimos também o element all que diferente do element ele retornava uma rede de elementos do (DOM), no caso, aqui buscamos no by.css e todos os arquivos que estivessem abaixo de no gutters a classe no gutters e também do col-4 ele

retornaria em forma de array para gente.

[3:05] E tudo isso foi visto no nosso curso de Protractor.

[3:07] Então, não deixe de comentar e avaliar o nosso curso. Obrigado e até a próxima.