

## Testes com Angular JS

Quando desenvolvemos aplicações que sofrem interação de usuários, temos dois itens importantes a assegurar sucesso: A funcionalidade e a interatividade com o usuário.

- Testes de comportamento (funcionalidade)  
Testar o comportamento da aplicação é útil para indicar que tudo corre de forma esperada, de maneira que futuras alterações não quebrem funcionalidades já existentes e testadas.
- Testes de interface (interação com usuário)  
Testar a interação com o usuário é essencial, afinal eles utilizarão a aplicação e, para eles, tudo deve ocorrer de forma transparente e clara.

### Nossa aplicação exemplo

Utilizaremos como exemplo a aplicação TodoMVC

<https://github.com/tastejs/todomvc/tree/gh-pages/examples/angularjs>

### Testes de interação com usuário (E2E)

Mesmo que escravize ratos topos para realizar seus testes, é importante definir um roteiro. Esta organização pode variar entre uma simples anotação até o modelo de testes proposto pelo Processo Unificado.

Mas o foco é: Como substituir os ratos topos?

Felizmente para nós, que utilizamos AngularJS, de 2013 para cá surgiu e evoluiu um tal de Protractor. Uma alternativa livre e grátis aos ratos topos, capaz de reduzir o custo com cereais e automatizar nossos testes de interação com a interface de usuário.

### Protractor

É uma ferramenta open-source que se baseia no WebDriver e que, basicamente, simula a interação do usuário com o navegador. A partir dele podemos controlar nossa interface, por exemplo:

```
browser.get( 'http://www.google.com.br' );
```

O trecho de código acima faria nosso browser ir até o site <http://www.google.com.br>, onde poderíamos ordenar novas atividades (clicar em um botão, passar o mouse sobre outro, escrever um texto...)

O foco aqui é testar a usabilidade e interação com o usuário, não importando se conseguiu-se realizar persistência de algo, comunicar-se com um webservice e afins... Para isto utilizamos o teste unitário (testaremos o comportamento de aplicação)

### **Testes de comportamento da aplicação (testando unidades)**

Conforme vimos acima, temos a capacidade de testar se a interface consegue agir de forma esperada, isto é bom. Mas veja bem: não adianta testar só a casca, tem de funcionar de verdade. Não é possível, através do teste de interface, assegurar-se de que um registro foi gravado, filtrado corretamente e validado.

Para realizar estes testes utilizaremos o teste unitário, que possui algumas opções interessantes para nós que, novamente, utilizamos AngularJS. Um mega combo muito conhecido é o Jasmine + Karma.

Jasmine é um framework que se propõe a testar comportamentos, enquanto o Karma é uma ferramenta mágica que disponibiliza ao desenvolvedor a opção de abrir diversos navegadores para testar seu código... De verdade!

Utilizar os dois juntos é uma mão na roda, afinal sabemos sobre a diversidade de navegadores e a constante luta para compatibilidade.