

08

Colocando em prática

Hora de praticar, implementando as novas funcionalidades apresentadas no capítulo. Como de costume, segue um resumo dos passos que devem ser executados:

- 1) Permita que o usuário filtre a lista digitando em um campo. Ele ficará imediatamente abaixo do `jumbotron`, como uma linha do grid do Bootstrap. Uma linha no grid do bootstrap é qualquer elemento container que tenha a classe `row`. Dúvida? Você pode consultar a explicação do capítulo e "colar" a estrutura, deixe para quebrar a cabeça com Angular apenas! Para filtrar, aplique um `filter` na diretiva `ng-repeat`, mas tome cuidado, pois tanto o campo de entrada quanto o valor de `filter` devem apontar para a mesma propriedade. A diferença é que o campo de entrada utilizará `two-way data binding` e a diretiva `ng-repeat` utilizará `filter`.
- 2) Agora que sua lista é filtrada, melhore ainda mais a experiência do usuário animando os elementos que são removidos da lista, aplicando uma espécie de fade, que na verdade realiza uma transição do tamanho atual do elemento para um tamanho menor. Para esta tarefa, você precisará do módulo `ngAnimate` que, para estar disponível, você precisa importar o script `angular-animate.min.js`, já disponibilizado na pasta `public/js/lib`.

Errata: No vídeo não é apresentado como declarar o módulo `ngAnimate` como dependência do módulo `alurapic`. Segue a linha de código que deve ser adicionado no arquivo `main.js`:

```
angular.module('alurapic', ['minhasDiretivas', 'ngAnimate']);
```

- 3) Ter o módulo `ngAnimate` carregado não é suficiente. Você deve criar seus estilos CSS pegando carona nas classes adicionadas dinamicamente pelo Angular, quando este módulo está presente. Existem uma série de classes, porém foco na `ng-leave-active`. Como este é um treinamento de Angular e não de CSS, sinta-se livre para consultar o CSS na explicação do capítulo, inclusive para saber onde gravar o CSS e como adicioná-lo na marcação.
- 4) Agora que sua lista é filtrada e elementos removidos sofrem animações, repare que a cada dígito o `two-way data binding` do Angular atualiza em `$scope` a propriedade que guarda o que o usuário digitou através da diretiva `ng-model`. Podemos dar um tempo antes que `ng-model` atualize a propriedade em `$scope`, que é usada por `ng-repeat` para filtrar a lista. Usamos para isso a diretiva `ng-model-options`. Lembre-se que ela trabalha com milissegundos, sendo assim, postergue em meio segundo a atualização.

Mãos à obra!