

Consolidando seu conhecimento

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

- 1) Temos duas classes, `MensagemView` e `NegociacoesView`, com um código bem parecido. Razão suficiente para usar herança! O primeiro passo é criar a classe mãe `View`, que recebe o `elemento` no construtor, e define os métodos `update` e `template` (que joga uma exceção):

```
// aluraframe/client/js/app/views/View.js

class View {

    constructor(elemento) {
        this._elemento = elemento;
    }

    update(model) {
        this._elemento.innerHTML = this.template(model);
    }

    template(model) {
        throw new Error('Você deve sobrescrever este método em seu template');
    }
}
```

- 2) Agora faça com que as classes `MensagemView` e `NegociacoesView` estendam a classe `View`, por exemplo:

```
class NegociacoesView extends View {
```

- 3) Em ambas as classes, `MensagemView` e `NegociacoesView`, apague o método `update`, já que ele foi herdado da classe `View`. Também garante que você remova o prefixo `_` do método `_template`.

- 4) Teste o seu código e fique atento ao console.

- 5) Crie a classe que representa uma mensagem, dentro de seu arquivo dedicado, com construtor que recebe uma valor padrão (''). Também já crie os métodos `get` e `set`.

- 6) Crie a classe `MensagemView` que também herdará de `View`. Lembre-se que você precisará reescrever o método `template` na classe, já que o template é específico de cada filho. O template será um simples parágrafo:

```
`<p class="alert alert-info">${model.texto}</p>`
```

Você pode olhar na classe `NegociacoesView` para ver como foi implementado.

- 7) Importe os dois arquivos no final do arquivo `index.html`.

8) No arquivo `index.html`, adicione a `div` abaixo do cabeçalho da página:

```
<div id="mensagemView"></div>
```

9) No construtor da classe `NegociacaoController`, instancie a `Mensagem` e `MensagemView`.

10) Ainda na classe `NegociacaoController`, altere o método `adiciona` para atualizar a mensagem e view.

11) Teste o código no navegador. Como o resultado deve aparecer uma mensagem na tela quando uma nova negociação for criada.