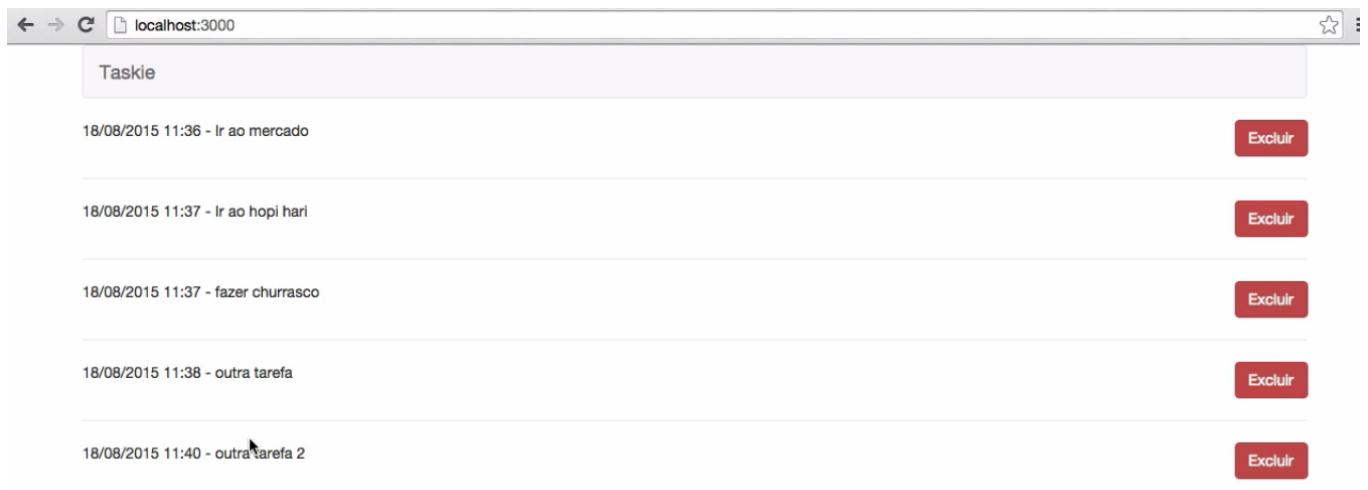


02

## Adicionando tarefas e lidando com eventos

Falta inserir um formulário para agregarmos novas tarefas a nossa lista, até agora já conseguimos que ela tenha a seguinte aparência:



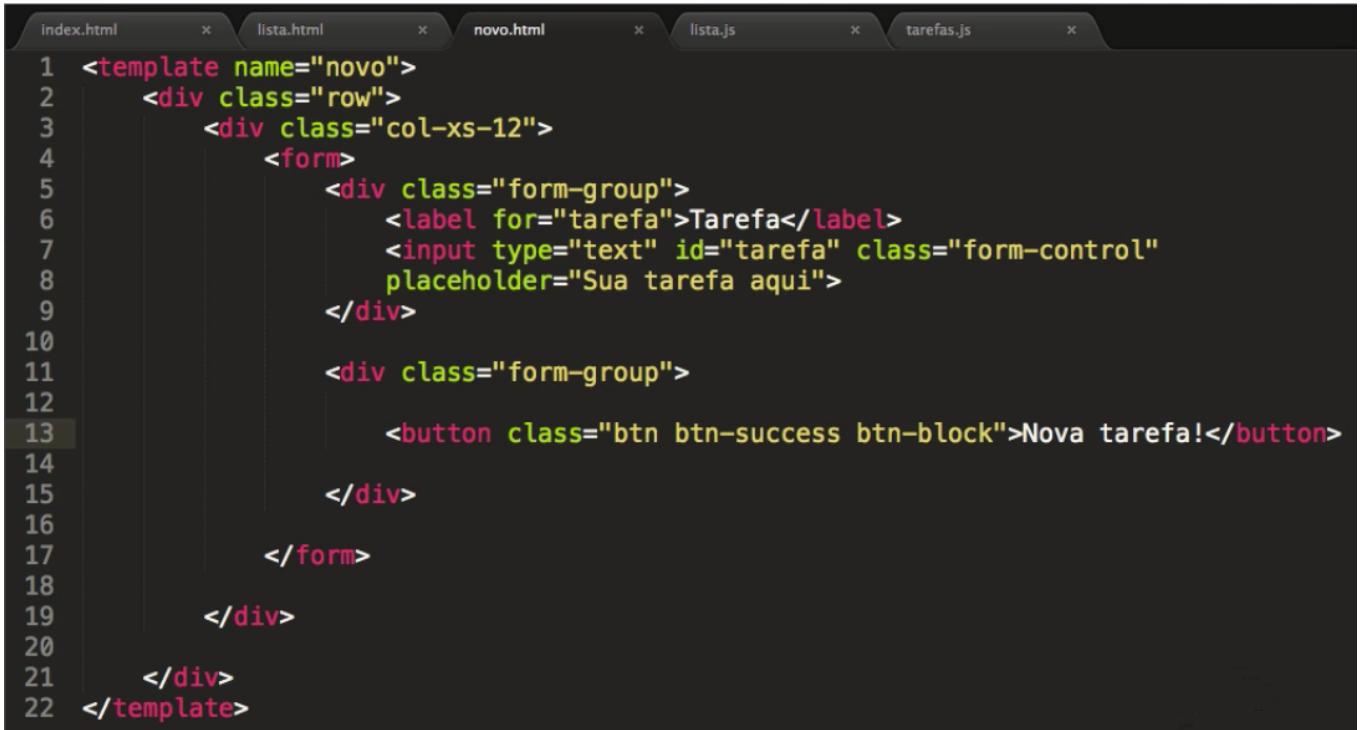
Task	Excluir
18/08/2015 11:36 - Ir ao mercado	Excluir
18/08/2015 11:37 - Ir ao hopi hari	Excluir
18/08/2015 11:37 - fazer churrasco	Excluir
18/08/2015 11:38 - outra tarefa	Excluir
18/08/2015 11:40 - outra tarefa 2	Excluir

Lembrando que, quando pensamos em uma *single page application*, temos que pensar em componentes.

Para inserir um formulário na nossa página temos que criar no *sublime* um arquivo novo, assim, separamos esse componente dos demais. Criamos um novo arquivo, na pasta "novo" e salvaremos como "novo.html". Digitaremos para inserir seu nome o seguinte: `template nome="novo"`. Lembrando de fechar o `template`.

Vamos acrescentar na linha de baixo do que acabamos de escrever as classes do *bootstrap*. Vamos inserir um código diferente em cada linha um `div class="row"`, `div class="col-sx-12"`, `form`, `div class="form-group"`, `label` `for="tarefa">Tarefa"`, `input type="text" id="tarefa" class="form-control"`, `placeholder="Sua tarefa aqui"`. E fecharemos esse formulário com o `div`.

Na próxima linha, criaremos um formulário para o botão, digitaremos `div class="form-group"` e acrescentaremos o botão através do `button class="btn btn-sucess btn-block">Novatarefa!` e fechamos o botão.



```

1 <template name="novo">
2     <div class="row">
3         <div class="col-xs-12">
4             <form>
5                 <div class="form-group">
6                     <label for="tarefa">Tarefa</label>
7                     <input type="text" id="tarefa" class="form-control" placeholder="Sua tarefa aqui">
8                 </div>
9
10            <div class="form-group">
11                <button class="btn btn-success btn-block">Nova tarefa!</button>
12            </div>
13        </form>
14
15    </div>
16
17    </div>
18
19    </div>
20
21 </div>
22 </template>

```

Temos que inserir nosso novo template na `index.html`, digitaremos `{{> novo}}`, na linha de cima da `{{> lista}}`.

Teremos:

```

23     <main class="container">
24
25     {{> novo}}
26
27     {{> lista}}
28
29     </main>

```

Se rodarmos a aplicação veremos que temos agora um campo para acrescentar a tarefa e também um botão, mas ele ainda não possui funcionalidade.

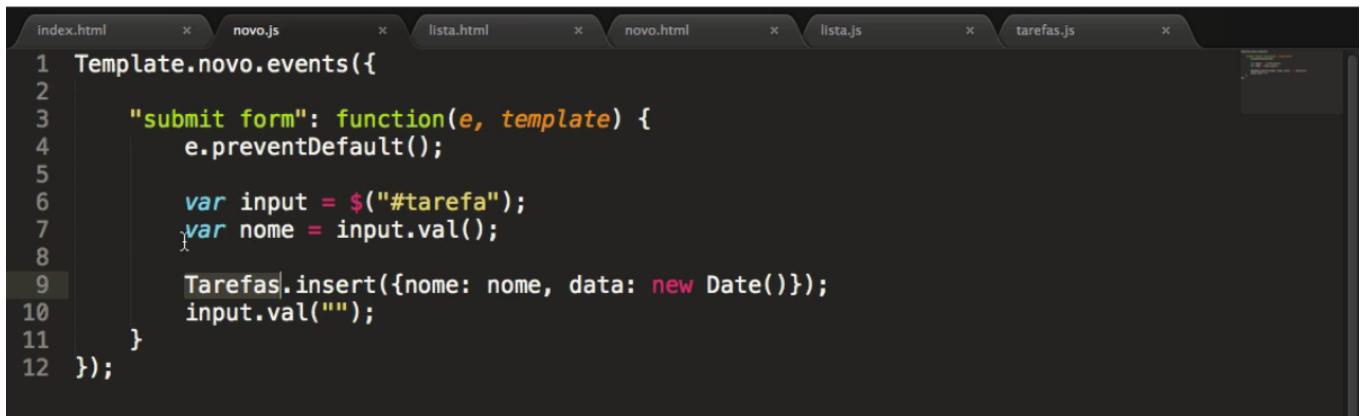
Todo `javascript` de um template, fica no `javascript`. Vamos no `sublime` para criar um novo arquivo, o "novo.js".

Como queremos lidar com os eventos do nosso template, vamos utilizar o `events` para tratar o uso do botão. Digitaremos `Template.novo.events`. Vamos passar para ele, na linha de baixo, a expressão do que ele tem que capturar, por exemplo, `"submit form"`, ou, poderia ser, `"clique do formulário"`. Isso é você quem decide. Digitaremos a função e essa função, por sua vez, recebe duas coisas, `"e"` que é disparado e o `"template"`, que é a classe. Teremos `"submit form: function(e, template)"`.

Acrescentamos também um `e.preventDefault`, do `javascript`, uma vez que o formulário não pode ser submetido de verdade, quem tem que tratar isso é o *Meteor*.

Na próxima linha acrescentamos um `var input = $("#tarefa")`, que servirá para pegar a tarefa. Agora, pegaremos o `"nome"`, `var nome = input.val()`, este é um código do *jQuery*.

Na próxima linha vamos providenciar para que isso seja inserido no banco. Vamos digitar um `insert` e passaremos o registro para ele. Teremos `Tarefas.insert ({nome: nome, data: new Date()})`. Por fim, acrescentaremos algo para limpar esse `input`, isto é, para que ele fique vazio, senão, o conteúdo não será alterado. Teremos, `input.val("")`.



```
1 Template.novo.events({
2
3     "submit form": function(e, template) {
4         e.preventDefault();
5
6         var input = $("#tarefa");
7         var nome = input.val();
8
9         Tarefas.insert({nome: nome, data: new Date()});
10        input.val("");
11    }
12});
```

Vamos pensar, rapidamente, o que aconteceu. O "tarefas.js" não está nem no "client", nem no "server". Ele está no "models". Isso quer dizer que ele pertence a ambos, então, quando pedimos para inserir no cliente, ele foi também no banco de servidores. E isso é extremamente prático.

Teremos o seguinte em nossa *single page*: