

Resumindo

Transcrição

Vamos revisar o que vimos até aqui: implementamos um mecanismo de View dentro da aplicação. Ou seja, nós temos uma tabela na qual exibimos os dados da negociação. Mas em vez das marcações estarem no arquivo HTML, estas foram colocadas em uma classe nova chamada `NegociacaoView.js`.

Como o código da tabela ficou no JavaScript, a View precisou encontrar alguma forma de se renderizar e aparecer no HTML. Por isso, nós criamos uma `<div>` que leva o `id` e indicou o ponto no qual o arquivo da tabela será inserido. Para realizar tal ação, a View recebeu um modelo - com qual “tampamos” as lacunas do template.

Nós ainda criamos o método `_template()`, utilizando a *template string*. Vimos que podemos gerar expressões mais "rebuscadas" para montar tags `<tr>` dinamicamente. Usamos novamente a função `map()` para transformar o conteúdo de um *array*. Além disso, utilizamos a função `join()` para poder concatenar todos os itens do *array* que equivalem às tags `<tr>` da tabela.