

Resumindo

Transcrição

Vamos revisar o que foi visto até aqui. Vimos que quando trabalhamos com data, precisamos transformar o formato da *string* para um objeto `Data`, com `dia-mês-ano`. É importante evitar a repetição do código sempre que for preciso usar a data no sistema, para isto, isolamos o trecho referente às conversões numa classe: o `DateHelper`. Em vez de usarmos concatenações nesta, optamos por usar uma *template string* que é criada com um ``` (*backtick*, em inglês). Quando usamos a crase no início e no fim, podemos colocar expressões dentro da *template string*, sem precisar das concatenações. Mais adiante, nos aprofundaremos no assunto.

Como a `Negociacao` está pronta, começamos a preparar a listagem de negociações para a exibição. Porém, nesta listagem não podemos incluir, remover ou alterar uma `Negociacao` - uma das regras de negócio. Então, nós criamos uma classe do modelo que recebeu o nome `ListaNegociacoes`. Depois, conseguimos adicionar e obter essas negociações, por meio do método `adiciona()` e do getter `negociacoes`. Porém, vimos que essa lista de negociações dentro do model não estava blindada. Qualquer um que chamasse o getter conseguiria apagar ou incluir a lista, por isso, lançamos mão da programação defensiva.

Caso o getter de `ListaNegociacoes` fosse chamado, o retorno seria um *array* original e independente de qualquer interferência de fora. Fizemos isto utilizando um *array* vazio, seguido da função `concat()`. Criamos também alguns métodos auxiliares "privados" na Controller, além de brincarmos um pouco com as expressões regulares para validarmos o texto passado para conversão de data.