

## Consolidando seu conhecimento

Chegou a hora do exercício do capítulo. Para fazê-lo, é importante que você tenha feito o último exercício do capítulo anterior ou tenha baixado o stage com o código completo do capítulo anterior no texto explicativo deste capítulo. Pronto para a linhas gerais do que deve ser feito? Vamos lá!

- 1 - No lugar de deixarmos espalhado pelo nosso sistema a lógica que converte um texto para a data e uma objeto data para texto vamos criar uma classe com essa responsabilidade. Dessa forma, toda vez que precisarmos lidar com a data pediremos para esta classe. Crie o arquivo `aluraframe/client/js/app/helpers/DateHelper.js` e defina a classe `DateHelper` com os métodos `textoParaData` e `dataParaTexto` (envolve o uso de template string). Uma exigência é que ambos sejam estáticos e que haja uma validação no construtor e nos métodos. Ainda lembra como criamos métodos estáticos?
- 2 - Não esqueça de importar o script da classe em `index.html` e alterar `NegociacaoController` para utilizá-la.
- 3 - Crie o arquivo `aluraframe/client/js/app/models/ListaNegociacoes.js` e defina a classe `ListaNegociacoes` com os métodos `adiciona` e `getter negociacoes`, não se esqueça de usar programação defensiva no `getter`. Importe o script em `index.html`.
- 4 - Agora, em `NegociacaoController`, adicione uma propriedade que é uma instância de `ListaNegociacoes`. Crie um método privado chamado `criaNegociacao`, que isolará aquela parte do código que cria a negociação. Adicione cada nova negociação à lista de negociações.