

Resumindo

Transcrição

Vimos que o código das Views `NegociacoesView` e `MensagemView` tinham trechos em comum. Nós isolamos tais partes dentro de uma classe, juntamente com o construtor que recebeu o elemento e o método `update()`. Depois, fizemos com que as duas Views herdassem da classe `View`, assim, não repetimos o código em comum. Mas coube às classes filhas implementarem o método `template()`.

Criamos ainda uma "armadilha" para evitar a possibilidade de que o desenvolvedor se esquecesse de incluir o método, incluindo uma mensagem de erro no Console. Lembrando que um método da classe filha sobrescreve métodos da classe pai.

Depois, adicionamos o `constructor()` nas classes filhas com o `super`, que passava o parâmetro para a classe pai. Fizemos também um pequeno ajuste, retirando o prefixo `_` do método `template()`, que anteriormente era privado. A alteração foi necessária porque os métodos `template()` de `NegociacoesView` e `MensagemView` precisavam sobrescrever o método em `View`.

Aguardamos você na segunda parte do curso.

Você pode continuar os seus estudos com o curso de [JavaScript Avançado II](https://cursos.alura.com.br/course/javascript-es6-orientacao-a-objetos-parte-2) (<https://cursos.alura.com.br/course/javascript-es6-orientacao-a-objetos-parte-2>) da Alura.