

06

## Blindando as negociações dentro da lista

### Transcrição

Colocamos em cheque a questão da imutabilidade da lista de negociações. A classe `ListaNegociacoes` será a única que poderá ter acesso a esta. A seguir, vamos incluir a seguinte linha no `adiciona()` do `NegociacaoController` :

```
this._listaNegociacoes.negociacoes.length = 0;
```

Com isso, o trecho ficou assim:

```
adiciona(event) {
    event.preventDefault();

    this._listaNegociacoes.adiciona(this._criaNegociacao());
    this._limpaFormulario();

    this._listaNegociacoes.negociacoes.length = 0;
    console.log(this._listaNegociacoes.negociacoes);
}
```

Ao definirmos que o `length` é igual a `0`, estamos esvaziando o `array`. Quando imprimimos no Console, devemos ver quantas `negociacoes` são exibidas. Se conseguimos esvaziar o `array`, ou seja, se alterarmos a lista, isto significa que as negociações não foram blindadas:



Você pode perguntar "mas não colocamos em getter?". A resposta é: sim, mas `negociacoes` é um `array`. Trata-se de uma referência. Quando usamos o `length`, alteramos a propriedade do objeto... Da mesma forma, não temos uma única forma de adicionar uma negociação. Não usamos apenas o método `adiciona()`. Por exemplo, podemos usar o `negociacoes.push()` :

```
adiciona(event) {
    event.preventDefault();

    this._listaNegociacoes.adiciona(this._criaNegociacao());
```

```

    this._listaNegociacoes.negociacoes.push(this._criaNegociacao());
    this._limpaFormulario();
    console.log(this._listaNegociacoes.negociacoes);
}

```

O método `adiciona()` deveria ser o único caminho para adicionarmos uma negociação, mas se executarmos o código, veremos que conseguimos salvar duas negociações. Se preenchermos o formulário uma vez, veremos que temos duas `Negociacao`. Alguém conseguiu adicionar fora do método `adiciona()`. Resolveremos o problema com programação defensiva. Quando alguém pedir uma lista de negociações, devolveremos uma nova lista - uma cópia da que tenho dentro da `ListaNegociacoes`. Mesmo que adicionarmos o `length = 0` ou inserir um elemento, como a lista está separada, não será possível adicionar outra negociação.

Em `ListaNegociacoes.js`, usaremos um truque: daremos um `return` com `array` vazio, seguido pelo método `concat()`:

```

get negociacoes() {
    return [].concat(this._negociacoes);
}

```

Ao passarmos o `this._negociacoes` dentro do `concat()`, o retorno será uma nova lista, um novo `array`. Agora se tentarmos usar o `push()` do `NegociacaoController`, só conseguiremos fazer a alteração na cópia da lista, mas não na original. Se fizermos um teste no formulário e preenchermos os campos **uma** vez, só deverá ter **uma** negociação.



Só é exibida uma `Negociacao`. O `push()` só funcionou na cópia da lista. Podemos fazer pequenos ajustes no método `adiciona`, que ficará assim:

```

adiciona(event) {
    event.preventDefault();
    this._listaNegociacoes.adiciona(this._criaNegociacao());
    this._limpaFormulario();
}

```

Conseguimos incluir negociações dentro da lista que só pode ser alterada por intermédio do método `adiciona()`.

