11

E tem espaço para self?

Neste capítulo você aprendeu gradativamente como resolver o problema do contexto de invocação da nossa armadilha passada para nosso modelo. A primeira solução foi passar o contexto de invocação como parâmetro e usar a mágica de `Reflect.apply` para alterar o contexto de invocação:

```
class NegociacaoController {

    constructor() {

        // código anterior omitido

        this._listaNegociacoes = new ListaNegociacoes(this, function(model) {

            this._negociacoesView.update(model);
        });
    }

    // código posterior omitido
}
```

Depois, aprendemos que o uso de uma arrow function resolvia essa questão, porque o valor de seu this é definido no local onde ela é declarada em nosso código e seu valor não muda (diferente do this de uma function que é dinâmico). Sendo assim, o `this` da nossa armadilha será a instância de `NegociacaoController`:

```
class NegociacaoController {

    constructor() {

        // código anterior omitido

        this._listaNegociacoes = new ListaNegociacoes(model =>
            this._negociacoesView.update(model));
    }

    // código posterior omitido
}
```

Essa solução evitou passarmos mais um parâmetro em nosso modelo, o contexto de invocação e ficou mais compacta e elegante.

Contudo, há ainda uma terceira solução, mas envolve a declaração de uma variável extra. Podemos guardar uma referência para a instância de `NegociacaoController` em uma variável. Geralmente essa variável é chamada de `self`:

```
class NegociacaoController {

    constructor() {

        // a variável self guarda uma referência para this, instância de NegociacaoController
```

```
            let self = this;

            // aqui usei uma function tradicional, mas poderia ser uma arrow function também

            this._listaNegociacoes = new ListaNegociacoes(function(model) {
                self._negociacoesView.update(model);
            });
        }

        // código posterior omitido
    }
```

Quando nossa armadilha for executada, o `self` será nosso `NegociacaoController`.