



## Para saber mais: Padrões de código

Para facilitar e simplificar nosso código, criamos uma classe de Serializador com alguns métodos principais e apenas estendemos essa classe para outra mais específica, como `SerializadorFornecedor`, para conseguir aproveitar os métodos da classe `Serializador` de uma forma específica para cada caso.

Esse tipo de técnica em código é um padrão de design de código (também conhecido em inglês como `design patterns`), o padrão que usamos se chama `Template Method`, no qual estabelecemos em uma classe propriedades e métodos que são usados em casos semelhantes e estendemos essa classe, para aproveitar seus métodos e propriedades em casos mais específicos.

Em vídeo, usamos a classe `Serializador` com um método `serializar()`, que usa uma propriedade chamada `this.contentType`. Essa propriedade não existe na classe `Serializador`, mas é criada nas classes que a estendem, como a classe `SerializadorFornecedor`.

No link a seguir, você pode ler mais sobre [como usar alguns padrões de código em JavaScript](https://imasters.com.br/devsecops/design-patterns-com-javascript-typescript) (<https://imasters.com.br/devsecops/design-patterns-com-javascript-typescript>). Também tem essa [lista sobre padrões de código em inglês](https://roadmap.sh/guides/design-patterns-for-humans) (<https://roadmap.sh/guides/design-patterns-for-humans>).