**Padrão de projeto - state**

Nesse modelo o objeto tende a mudar de estado assim que seu estado interno mude. Esse estado tem um número finito de estados que um programa possa estar. Existe a possibilidade do programa poder ou não trocar para outros estados através das regras de troca, que são chamadas transições. Elas também são finitas e pré-determinadas.

A seguir temos um exemplo de como seria essas mudanças de estado.

Diagrama de estados da máquina de refrigerantes

Figura 1. Fonte: https://www.thiengo.com.br/padrao-de-projeto-state-estado

“*Note que a máquina de refrigerantes têm até o momento exatos quatro estados, são eles:*

* *Sem moeda;*
* *Com moeda;*
* *Venda;*
* *e Sem refrigerantes*.”

Outro exemplo que podemos citar é aluguel de carro. Em uma classe AluguelCarro. Um carro pode estar em um desses três estados: Disponível, Vitrine e Alugado. O método Alugar funciona assim em cada estado:

* No **Disponível**, ele move o carro para a vitrine.
* Na **vitrine** ele torna o carro público, mas só poderá ser alugado se o usuário atual é um cliente cadastrado
* No **Alugado** ele não faz nada.

Fontes:

https://refactoring.guru/pt-br/design-patterns/state

<https://www.devmedia.com.br/design-patterns-state-parte-4/16783>

<https://www.thiengo.com.br/padrao-de-projeto-state-estado>

https://pt.wikipedia.org/wiki/State