



Universidade Federal de São Carlos
MBA em Machine Learning in Production
Turma C - Time B
Padrões de Projeto

Atividade: 2

Sinergia entre Padrões

Alunos:

Herbert Dias de Macedo
João Felipe Moura
Paulo Daniel Nobre Ferreira
Ricardo José Campelo Arruda Júnior
Rodolfo Arruda
Valter Alberto Melgarejo Martins

Atividade:

Orientações:

1) Analise abaixo dois trechos do documento de requisitos de um novo sistema inteligente de monitoramento de edifícios:

Requisito Funcional:

Um novo empreendimento imobiliário está com intenção de desenvolver um sistema inteligente de monitoramento de elevadores. Você foi incumbido de projetar uma solução computacional que gerencia o controle de movimentação dos vários elevadores do edifício. Os elevadores são convencionais, podendo subir, descer e parar nos andares solicitados. A inteligência do sistema está no fato de que o edifício deve monitorar o funcionamento dos elevadores para proporcionar mais segurança e também uma experiência de uso mais agradável para os moradores. As ações que o sistema pode tomar variam desde ações de segurança até mesmo a ações de entretenimento, como reprodução de músicas dentro do elevador.

Exemplos:

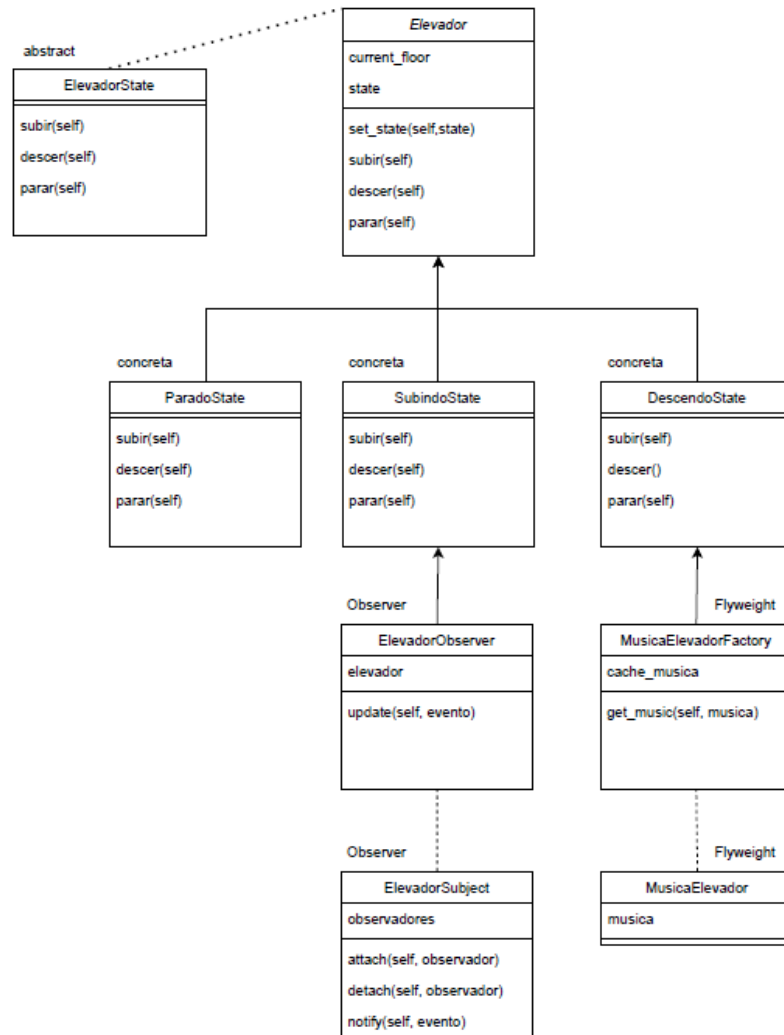
- Se um determinado elevador emperrar durante seu funcionamento, o sistema deve perceber e automaticamente invocar a equipe de manutenção e segurança. Enquanto isso uma música agradável pode começar a ser reproduzida naquele elevador específico para acalmar o usuário.
- Se um determinado elevador estiver em manutenção, o sistema pode, entre outras ações, interferir na velocidade de deslocamento dos outros elevadores.

Requisitos Não-Funcionais:

- O código cliente que manipula os elevadores deve ser o mais inconsciente possível dos objetos concretos existentes
- Nesta primeira versão do sistema, um elevador nunca estará em dois estados ao mesmo tempo.
- Embora não haja limitação de memória na infraestrutura computacional usada, deve-se prezar para que não haja criação desnecessária de objetos em memória.

Atividade:

Para esta atividade, foi desenvolvido o seguinte diagrama:



O código da tarefa pode ser visto no seguinte Git:

https://github.com/vaaaltin/MLProd/blob/main/1o%20Semestre/ESBD%20-%20202/Atividade%202.2/state_obs_fly.py

A explicação do modelo:

- Classe **Elevador** é a classe principal, nela contém as informações principais do elevador, como o andar atual e o estado atual
- Classe **ElevadorState** é uma classe abstrata, onde são definidos os comportamentos do elevador para diferentes estados: Subindo, Descendo e Parado
- Classes **SubindoState** e **DescendoState** são classes concretas que herdam de **ElevadorState** e implementam os métodos *subir*, *descer* e *parar*, para seus respectivos estados

- Classe ***ParadoState*** é uma classe concreta que também herda de ***ElevadorState***, e implementa os métodos *subir*, *descer* e *parar* para o estado parado. Estes que são únicos em relação a ***SubindoState*** e ***DescendoState***
- Classes ***ElevatorObserver*** e ***ElevatorSubject*** implementam o padrão observer para observar os eventos do elevador, como manutenção e problemas
- As classes ***MusicaElevador*** e ***MusicaElevadorFactory*** implementam o padrão Flyweight para ter uma maior eficiência de memória, para tocar as músicas no elevador