

Padrões de Projeto

Equipe: Rogério Araújo, Fernanda Silva

1. Descrição do domínio

Deve ser desenvolvido um software que faça reservas de assentos para shows/peças de teatros. Os requisitos funcionais estão descritos abaixo.

2. Lista de Requisitos Funcionais

- RF.01** - Listar atrações - Concluído
- RF.02** - Realizar reserva de assento - Concluído.
- RF.03** - Configurar notificações - Não concluído
- RF.04** - Listar assentos reservados - Concluído
- RF.05** - Cadastrar atrações - Concluído
- RF.06** - Gerar relatórios - Não concluído

3. Lista de Padrões utilizado

3.1 - Injeção de Dependência:

Utilizado para criação de configuração e conexão com o banco de dados. Com o propósito de implementar uma arquitetura fracamente acoplada para obter um código melhor testável, atualizável e extensível.

`DatabaseConfiguration` é injetado e `DatabaseConnection` terá tudo para criar uma conexão, precisando do `$config`.

Sem DI, a configuração seria criada diretamente `DatabaseConnection`, o que não é muito bom para testar e estender.

Fig. 01 - Diagrama representando o uso do padrão na aplicação



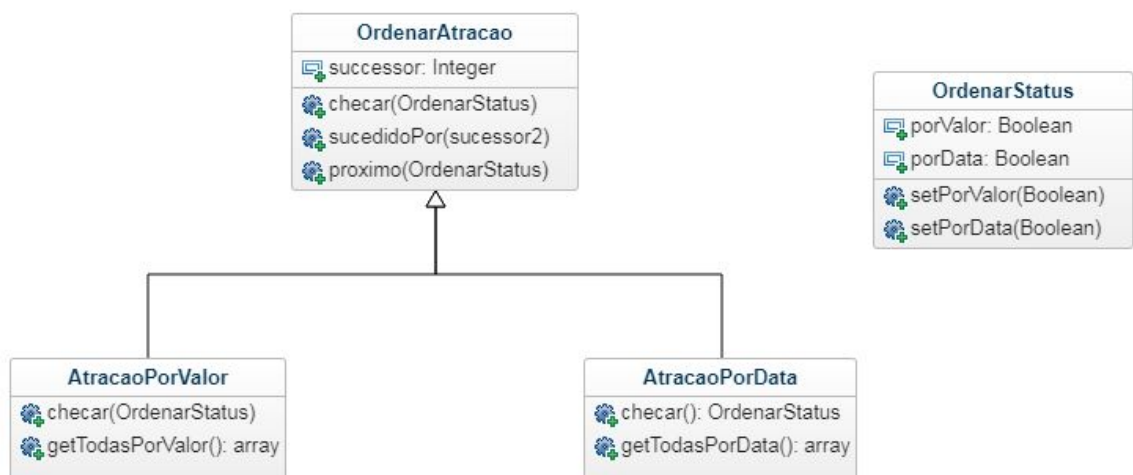
Powered by **genMyModel.com**

3.2 - Chain of Responsibility:

Utilizado para construir uma cadeia de objetos a fim de manipular uma chamada em ordem sequencial. Se um objeto não pode manipular a chamada, ele delega a chamada para o próximo na cadeia e assim por diante.

A aplicação permite listar atrações por data ou valor de ingresso, para isso foi adotado este padrão, ao solicitar a listagem de atrações por data, por exemplo, uma cadeia de objetos é criada e as chamadas vão sendo repassadas em sequência até encontrar a classe responsável que poderá manipular a chamada e retornar a lista.

Fig. 02 - Diagrama representando o uso do padrão na aplicação



Powered by **genMyModel.com**