



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E  
TECNOLOGIA DA PARAÍBA - CAMPUS CAJAZEIRAS  
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS  
PADRÕES DE PROJETO  
AMANDA G PEDROZA  
ERICLYS MOREIRA  
RAFAELA P .RAMOS

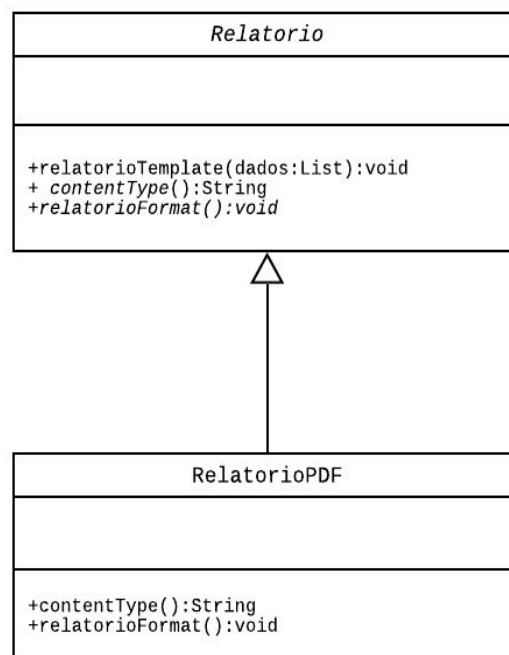
**PROJETO OAUGE ATRAÇÕES**

CAJAZEIRAS -PB  
ABRIL , 2019

## 1.0 Template method

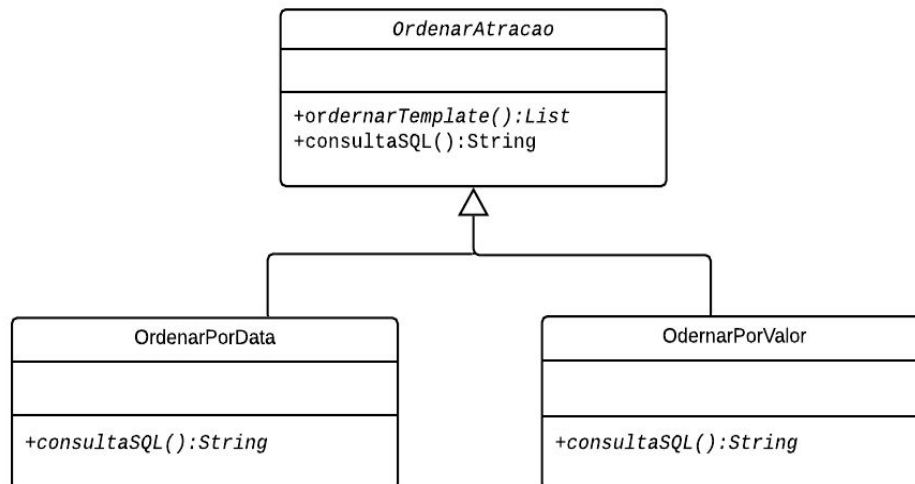
Tem como propósito definir o esqueleto do algoritmo para que passos variantes desse algoritmo possam ser redefinidos por subclasses. No desenvolvimento do nosso projeto optamos por utilizar esse padrão em duas funcionalidades: geração de relatório e na ordenação de atrações .

A escolha de se utilizar template method nas classes de geração de relatório teve sua motivação inicial decorrente da utilização framework JasperReport para a geração do relatório, pois o processo de geração contém algoritmos comuns e sua parte variante só ocorre no momento de exportar o arquivo para o formato específico e a utilização do jasper se tornou essencial decorrente do requisito do projeto onde era solicitado que a aplicação estivesse aberta para implementação de novos formatos. Com base nisso a não utilização do padrão template method iria ocasionar repetição de código desnecessária além de não proporcionar flexibilidade para adicionar novos formatos com facilidade, como foi proposto para adicionar outra opção de formato de relatório basta apenas criar uma classe que estenda da classe abstrata Relatório e implemente os métodos abstratos herdados utilizando o método de exportação para o novo formato desejado oferecido pelo jasper .



Como já foi citado também utilizamos template method para a ordenação das atrações, a escolha teve como motivação o aproveitamento de código, pois o que varia de uma ordenação para outra era apenas a consulta sql para listar as atrações, o que também ocasionou a flexibilidade para a implementação de novos tipos de ordenação como por

exemplo por ordem alfabética, bastaria apenas criar uma nova classe que estendesse da classe abstrata OrdenarEvento e implementar o método herdado retornando a String da consulta sql.

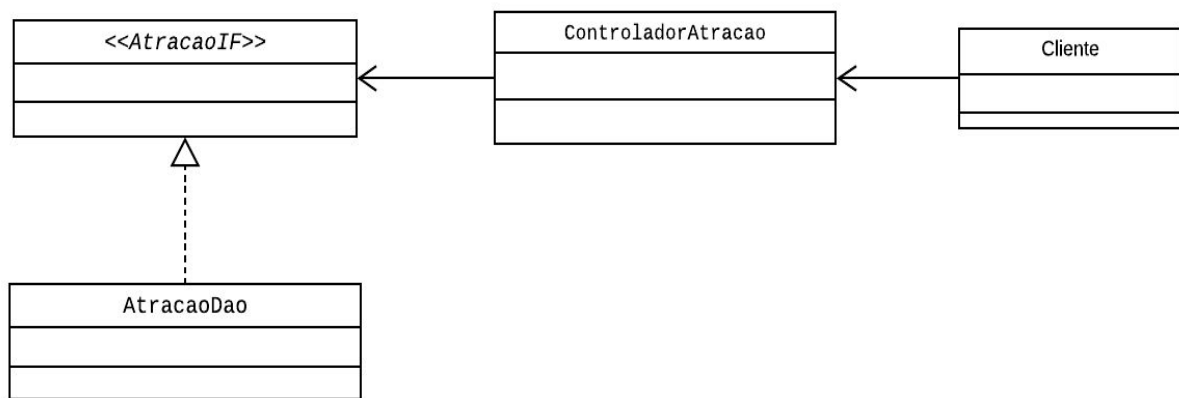


Utilizando template method além dos benefícios já descritos como aproveitamento de código e flexibilidade para adicionar novas opções de ordenação e formatos de relatórios, também conseguimos manter o princípio de aberto-fechado (Open/closed), pois permite que estendamos o comportamento das classes sem modificá-las.

## 2.0 Injeção de dependência

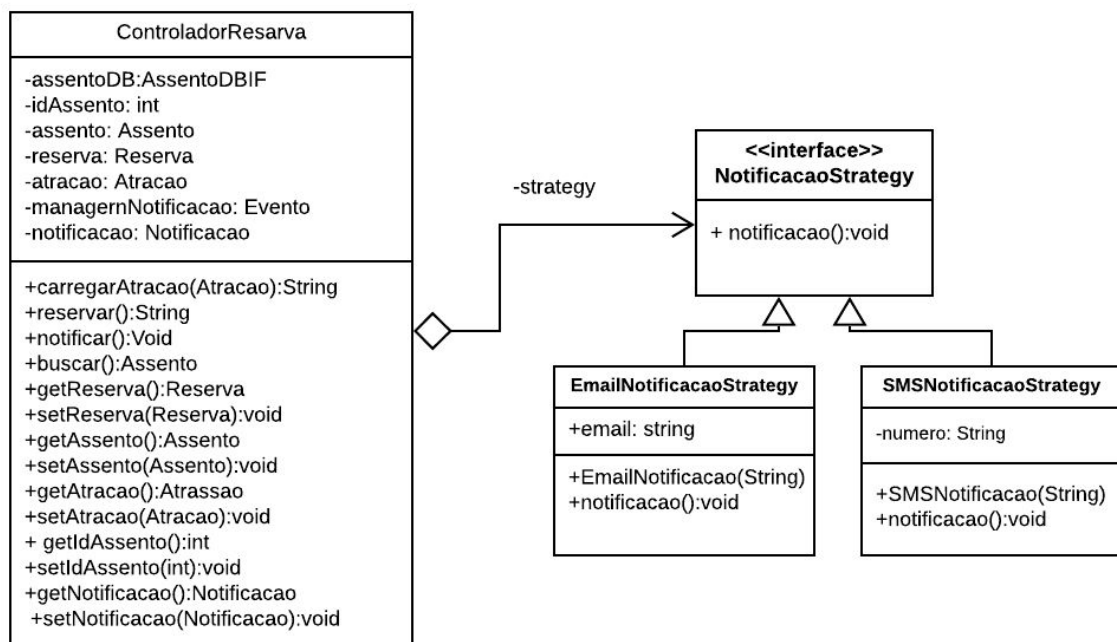
É um padrão de projeto utilizado para manter o baixo acoplamento entre classes ou módulos do sistema. O objetivo principal é fazer com que uma classe não tenha conhecimento de como instanciar um objeto de um tipo do qual é dependente.

No nosso projeto utilizamos a anotação @EJB para injetar as classes de persistência através das interfaces sendo o container EJB responsável por instanciar as classes, de forma que mantemos o baixo acoplamento entre os módulos do projeto e diminuimos a dependência entre a camada de persistência e controladores, mantendo o princípio de inversão de dependência onde diz que devemos depender de abstrações e não de implementações.



### 3.0 Padrão strategy

O padrão de estratégia é útil quando temos vários algoritmos para tarefas específicas e queremos que nossa aplicação seja flexível para escolher qualquer algoritmo em tempo de execução para uma tarefa específica.



Para o nosso projeto, tivemos que implementar uma opção na tela inicial do sistema para configurar as notificações que serão enviadas quando um assento for reservado na aplicação, no qual teremos uma estratégia de notificação que é a via

e-mail, contudo o sistema precisa estar preparado para novas formas de notificações como notificações push, SMS. O que torna esse padrão apropriado para essa implementação.