**Relatório Técnicas de Programação**

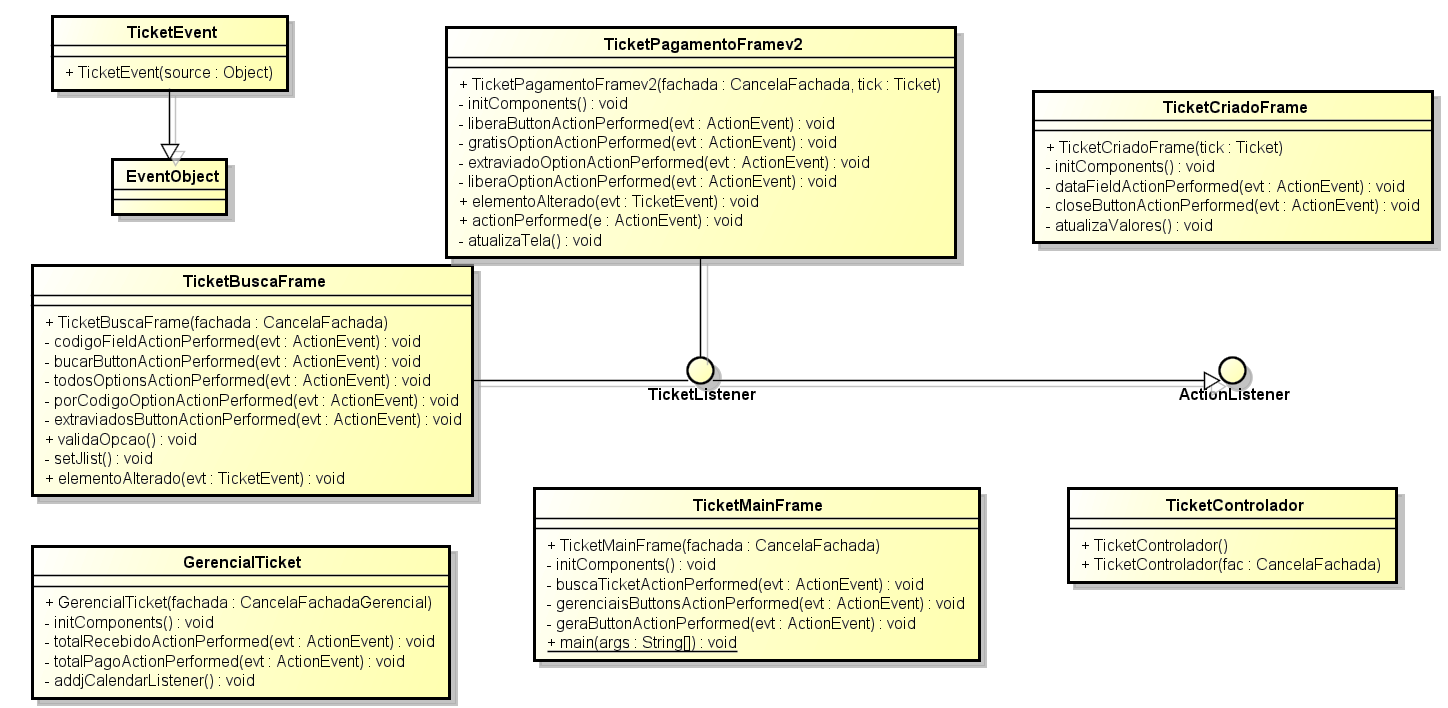
Bruno Feres Klein, Italo Lobato Qualisoni

[*bruno.klein@acad.pucrs.br*](mailto:bruno.klein@acad.pucrs.br)*,* [*italo.qualisone@acad.pucrs.br*](mailto:italo.qualisone@acad.pucrs.br)

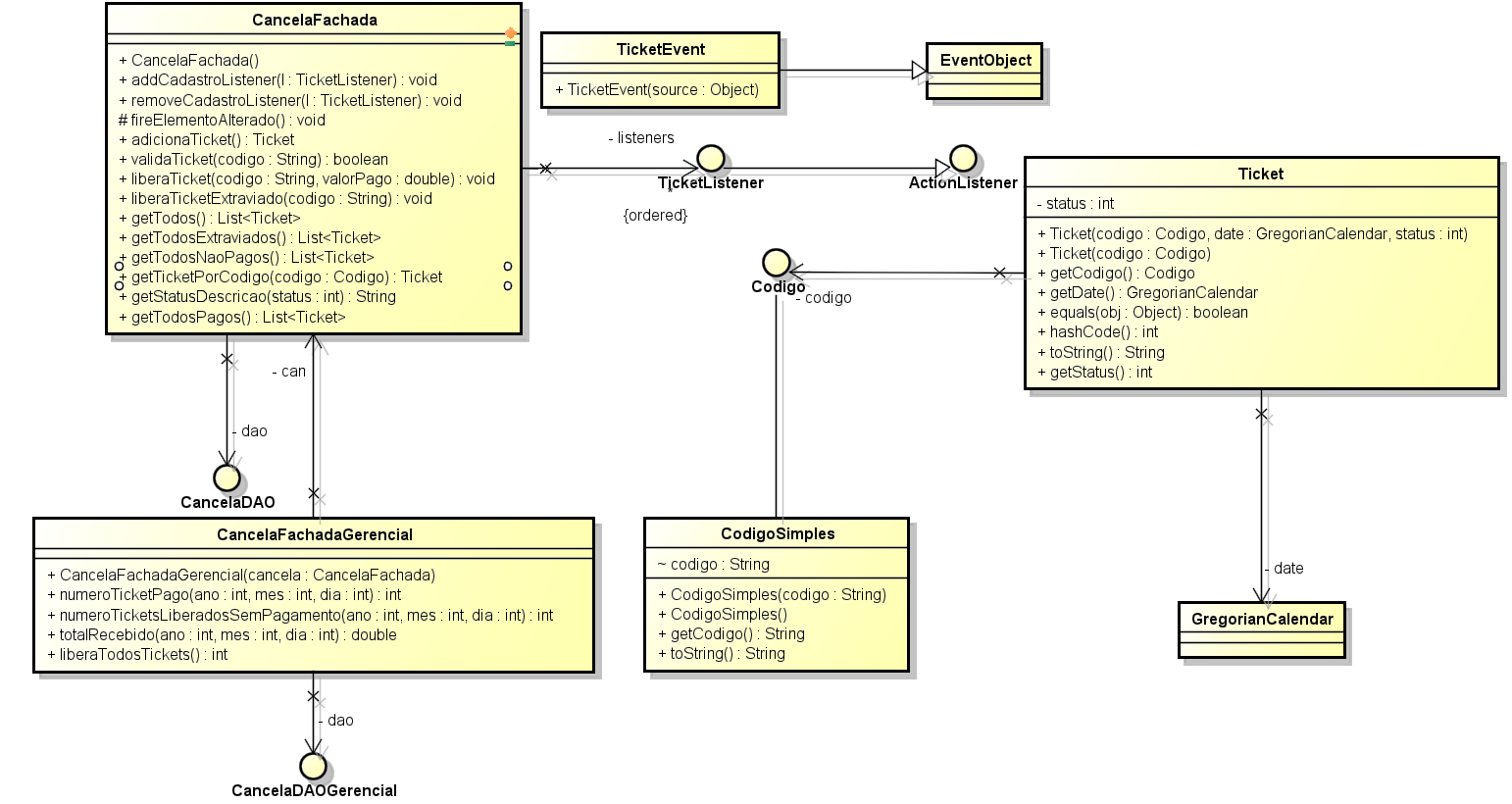
**Padrões utilizados**

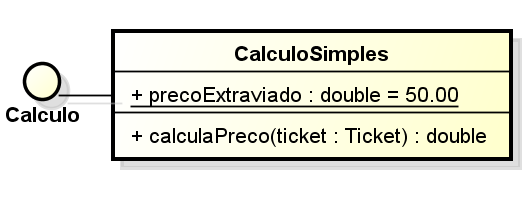
Foram utilizados os padrões, Facade, o padrão DAO na camada de persistência e o padrão Domain Model na camada de domínio que foram requisitados no enunciado do trabalho, além dos padrões Strategist e Observer que julgamos necessários para a elaboração de um trabalho mais consistente.

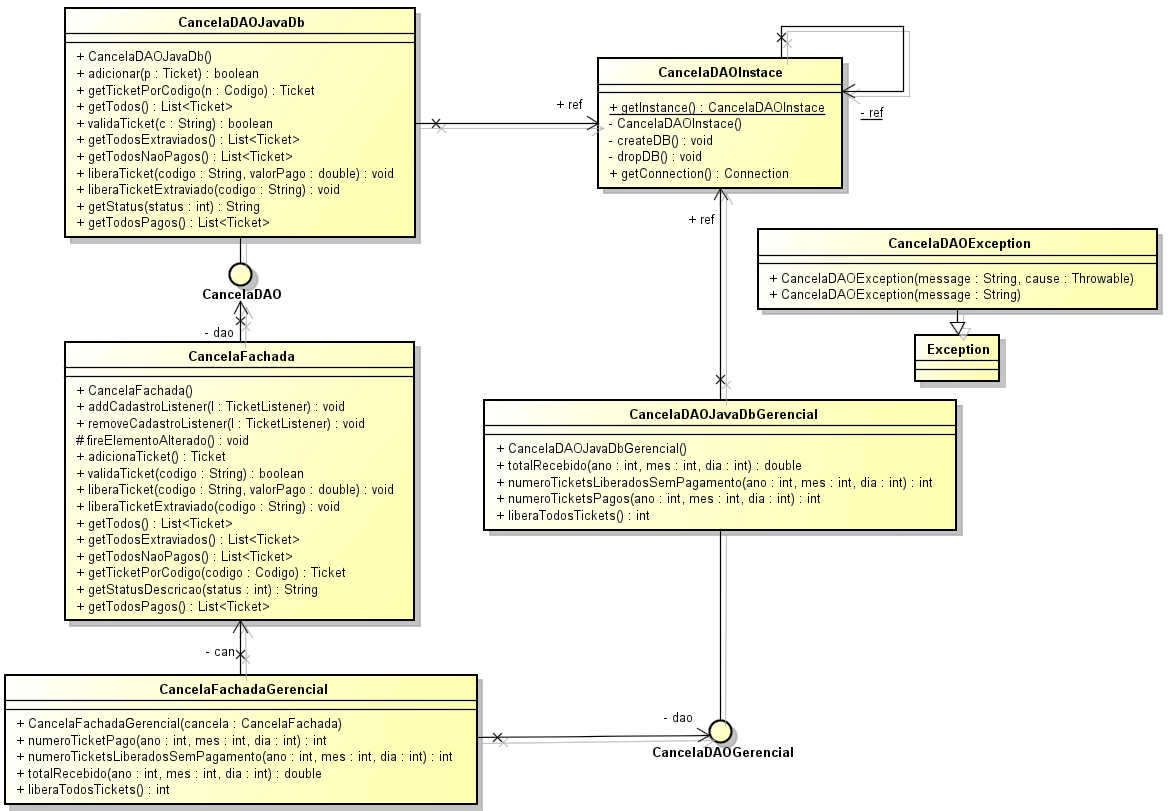
**Camada de Apresentação**



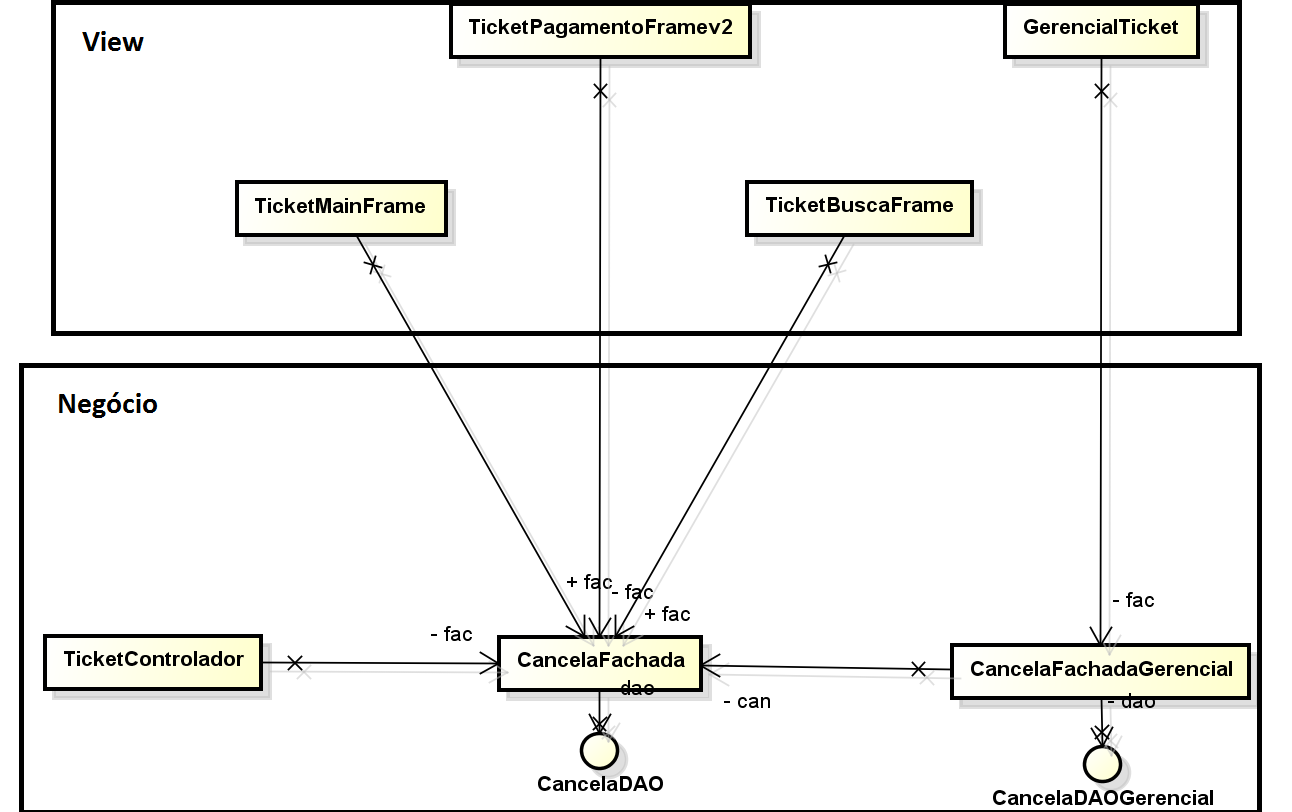
**Camada de Negócio**





**Camada de Persistência**

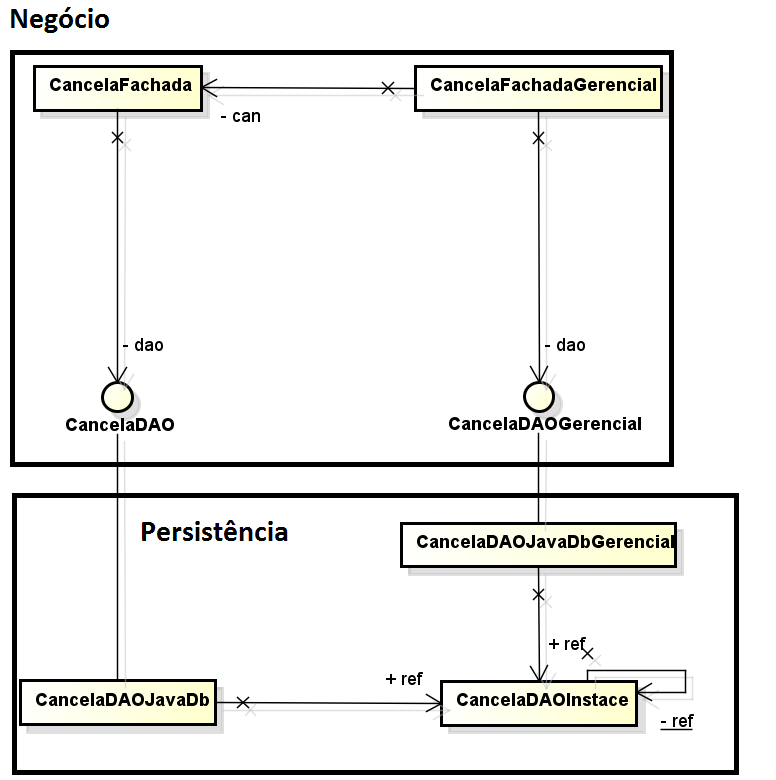
**Comunicação entre a Camada de Apresentação e Camada de Negócio**



Nesta figura está sendo mostrada a forma de comunicação entre as Camadas de Apresentação e de Negócio pelo padrão de projeto Facade. Como podemos ver, na camada de negócios há duas fachadas, uma normal que é responsável diversas funcionalidades do sistema e outra gerencial que é voltada para a parte administrativa do sistema.

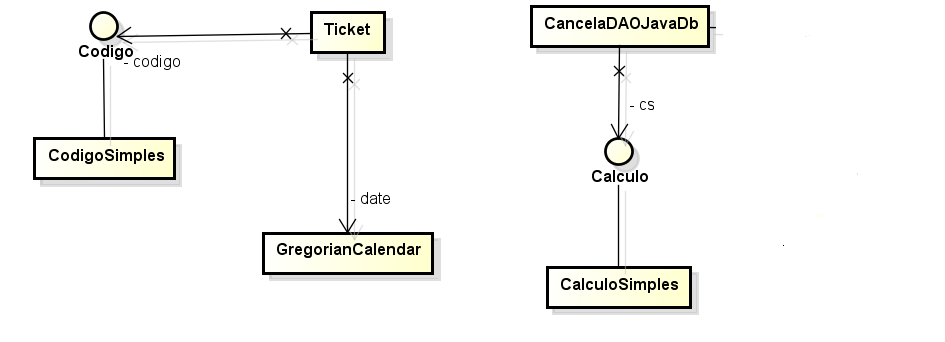
**Comunicação entre a Camada de Negócio e Camada Persistência**

A camada de Negócio se comunica com a camada de persistência através de duas interfaces da camada de Negócio que são implementadas na camada de persistência.

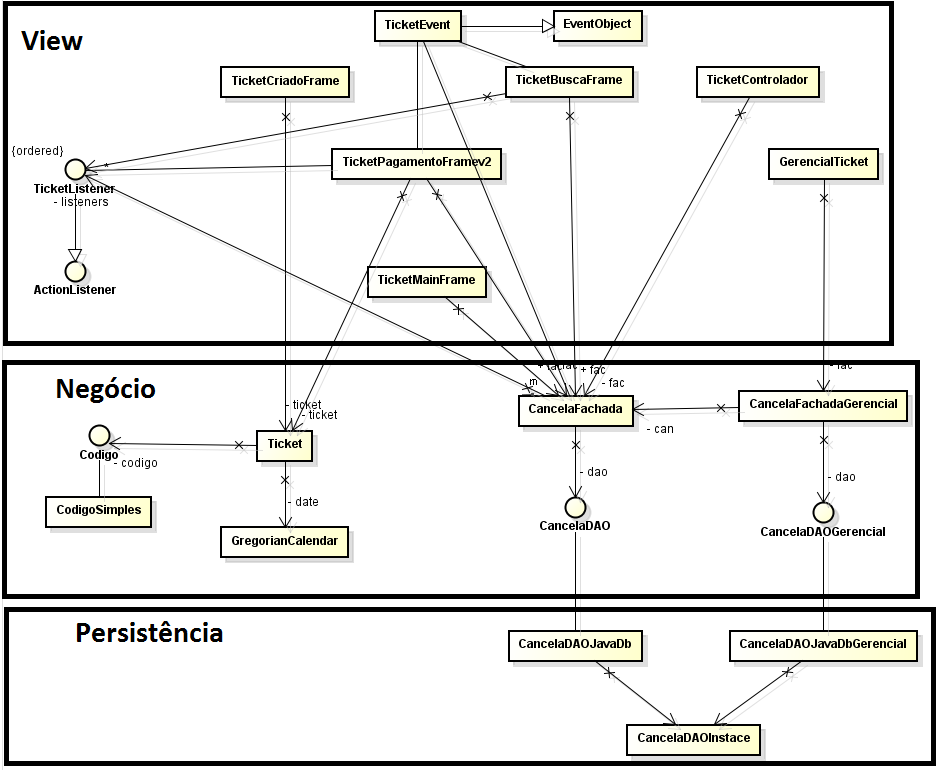


**Padrão Strategist**

Foi utilizado na Camada de Negócio, para o encapsulamento de dois algoritmos. Foi encapsulado o algoritmo Calcular() como interface, e este é implementado pela classe CalculoSimples. Também encapsulamos um algoritmo, que gera códigos aleatórios, como a interface Codigo e a partir dela definimos a classe CodigoSimples.



**Sistema**



**Padrão Observer**

As classes que tem participação neste padrão são: TicketPagamentoFramev2, TicketBuscaFrame, TicketListener(Interface Observer) e CancelaFachada, sendo que a CancelaFachada é o o Subject contendo uma coleção de objetos de TicketListener. Sempre que há ocorrência de um evento o Subject vai avisar as duas outras classes que estão observando, que neste caso são TicketPagamentoFravev2 e TicketBuscarFrame(estas implementam TicketListener).

