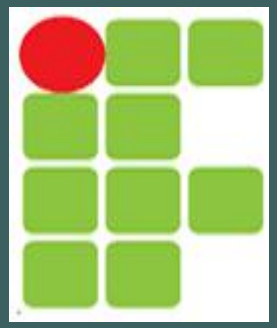


Aula 04

Diretrizes para projeto de objetos: padrões GRASP (2)

Análise e Projeto Arquitetural de Software

Prof. Thiago

Pós-Graduação em Gestão de Sistemas de Informação

Objetivos

- Após esta aula, você deverá ser capaz de:
  - Aplicar os padrões GRASP remanescentes para atribuição de responsabilidades
    - Invenção Pura
    - Indireção
    - Variações Protegidas
    - Controlador

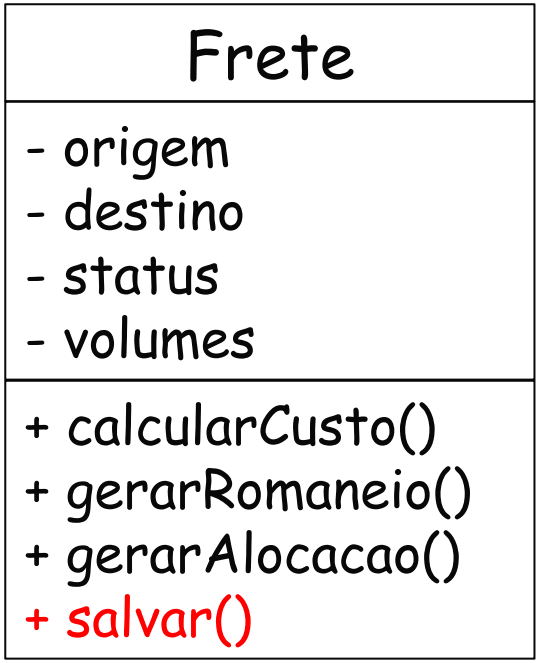
Invenção Pura

Problema: Que objeto deve ter a responsabilidade quando não queremos violar Coesão e Baixo Acoplamento, mas as soluções oferecidas por Especialista na Informação não são suficientes?

Solução: Atribuir um conjunto coeso de responsabilidades a uma classe artificial ou de conveniência, que não represente um conceito no domínio do problema

Invenção Pura - exemplo

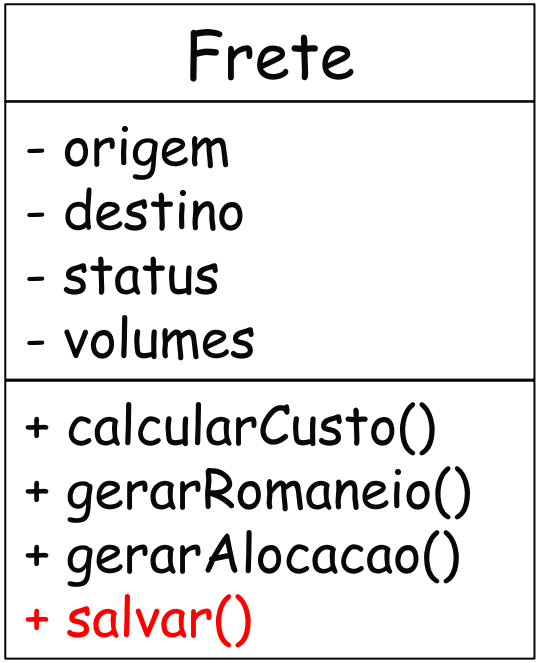
- Os dados de um *Frete* da transportadora deveriam ser persistidos no banco de dados
- Segundo o padrão *Especialista na Informação*, o Frete possui os dados a serem persistidos
- Logo, deveria ter um método para salvar os dados em um BD



Invenção Pura - exemplo

- Porém, em larga escala, isso significa acoplar à classe a interface de acesso ao banco de dados relacional

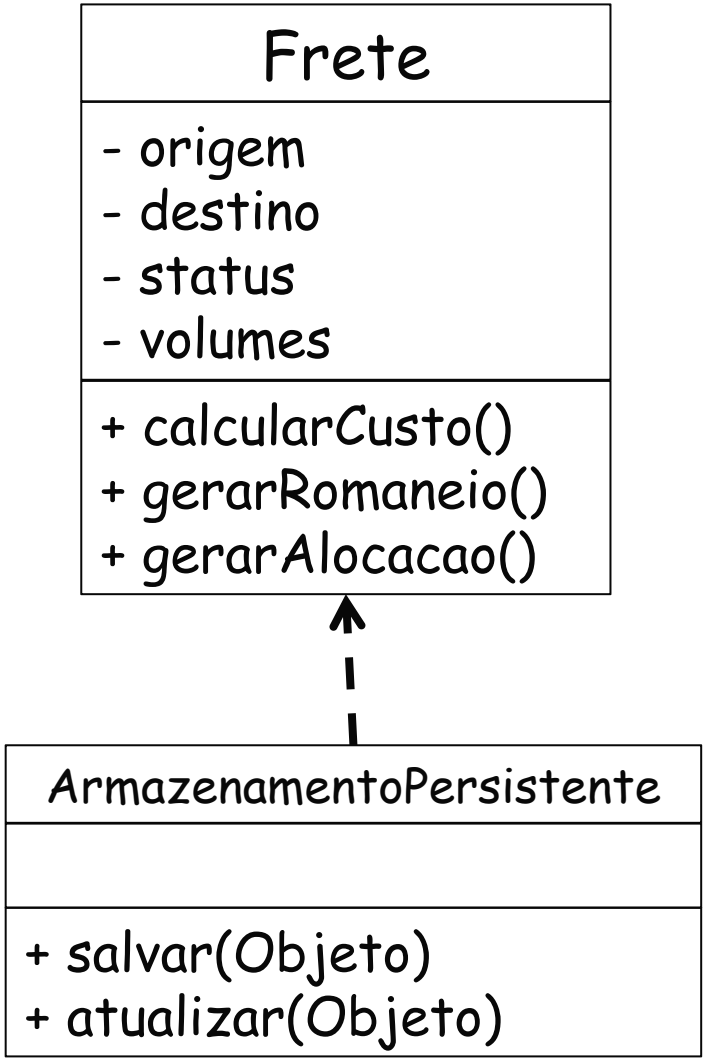
- Em uma ampliação da escala, toda classe de domínio teria esse acoplamento



Invenção Pura - exemplo

- Atribuímos, então, a responsabilidade de persistir os dados para a classe *ArmazenamentoPersistente*

- Esse conceito é uma Invenção Pura, pois não está presente no domínio da aplicação



Indireção

Problema: Como evitar o acoplamento entre dois (ou mais) tipos de objetos, mantendo o potencial de reúso?

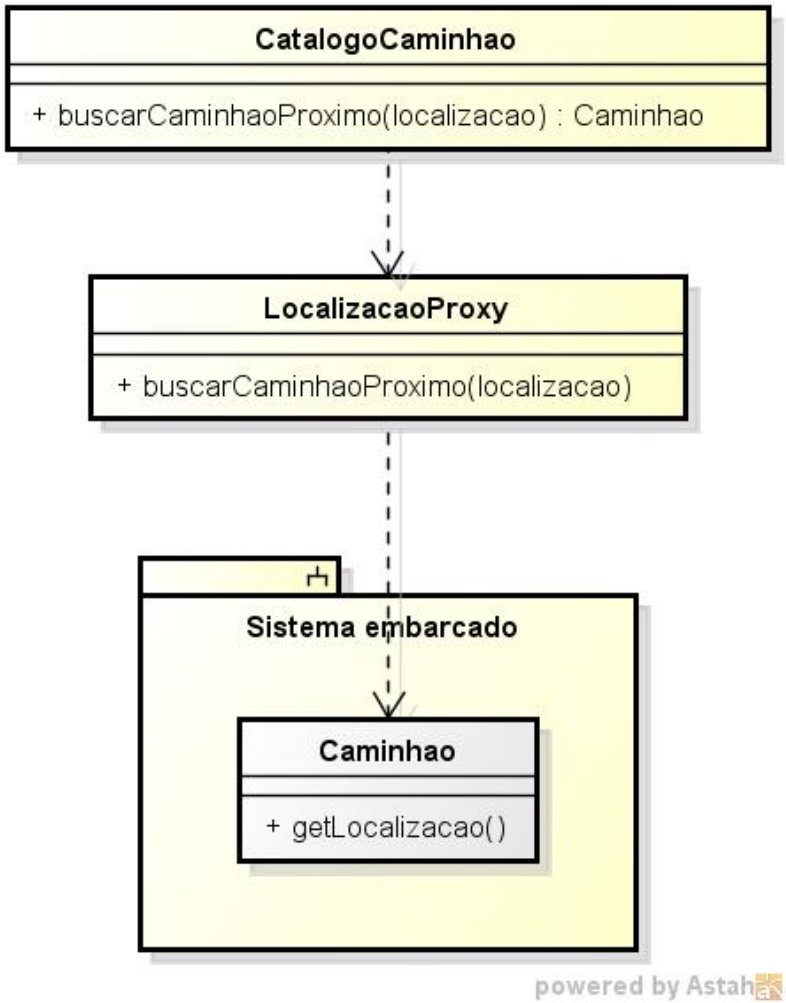
Solução: Criar um objeto intermediário para atuar como mediador entre dois ou mais tipos de objetos.

Indireção - exemplo

- A transportadora passa a ter GPS com sistema operacional embarcado em cada caminhão
- Para poder operar em escala nacional de maneira eficaz, é necessário que o Frete tenha alocado a si o Caminhão que está mais próximo da origem do Frete e que esteja livre
- Parte do sistema fica embarcada no GPS do Caminhão, que informa regularmente a posição do Caminhão

Indireção - exemplo

- Objetos da classe Caminhão “rodam” no sistema embarcado
- O CatalogoCaminhao não deve ter acesso direto ao sistema embarcado, pois identificar o caminhão mais próximo exige a intercomunicação com o sistema embarcado
- A classe LocalizacaoProxy cria um nível adicional de indireção, consultando cada Caminhão e obtendo sua localização



Variações Protegidas

Problema: Como projetar objetos, subsistemas e sistemas de forma que as variações ou instabilidade nesses elementos não tenham impacto negativo sobre outros elementos?

Solução: Identificar pontos de variação ou instabilidade previsível; atribuir responsabilidades para criar uma interface estável em torno deles

Variações Protegidas - exemplos

- O romaneio digital, do padrão *Polimorfismo*

  Uma interface "esconde" o tipo de romaneio do chamador

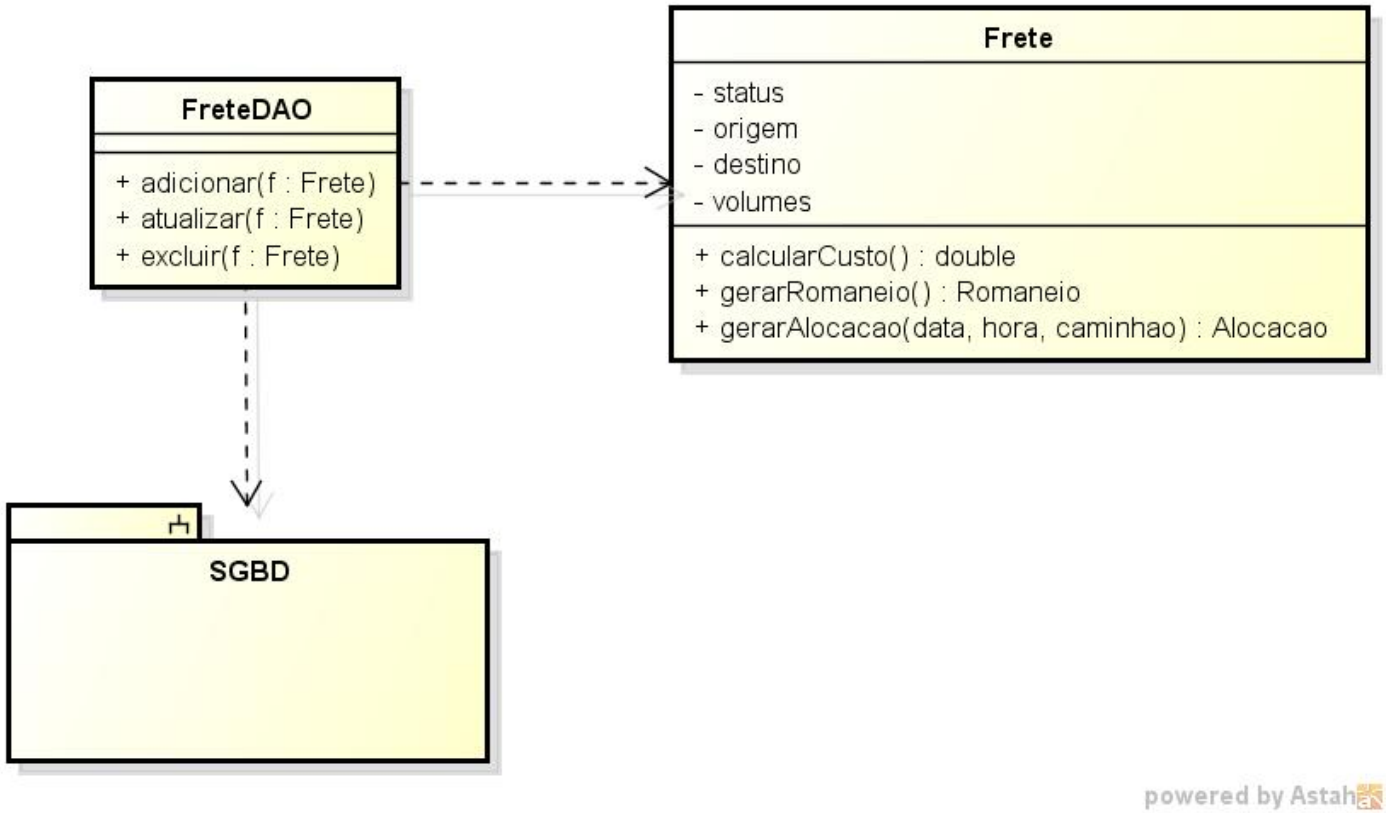
- LocalizacaoProxy do padrão Indireção

  Qualquer classe que aceite a chamada getLocalizacao poderia ser "localizada" pelo proxy

Variações Protegidas - exemplos

- O padrão DAO (Data Access Object)

  Ao desacoplar as particularidades de um banco de dados das classes cujos dados serão persistidos



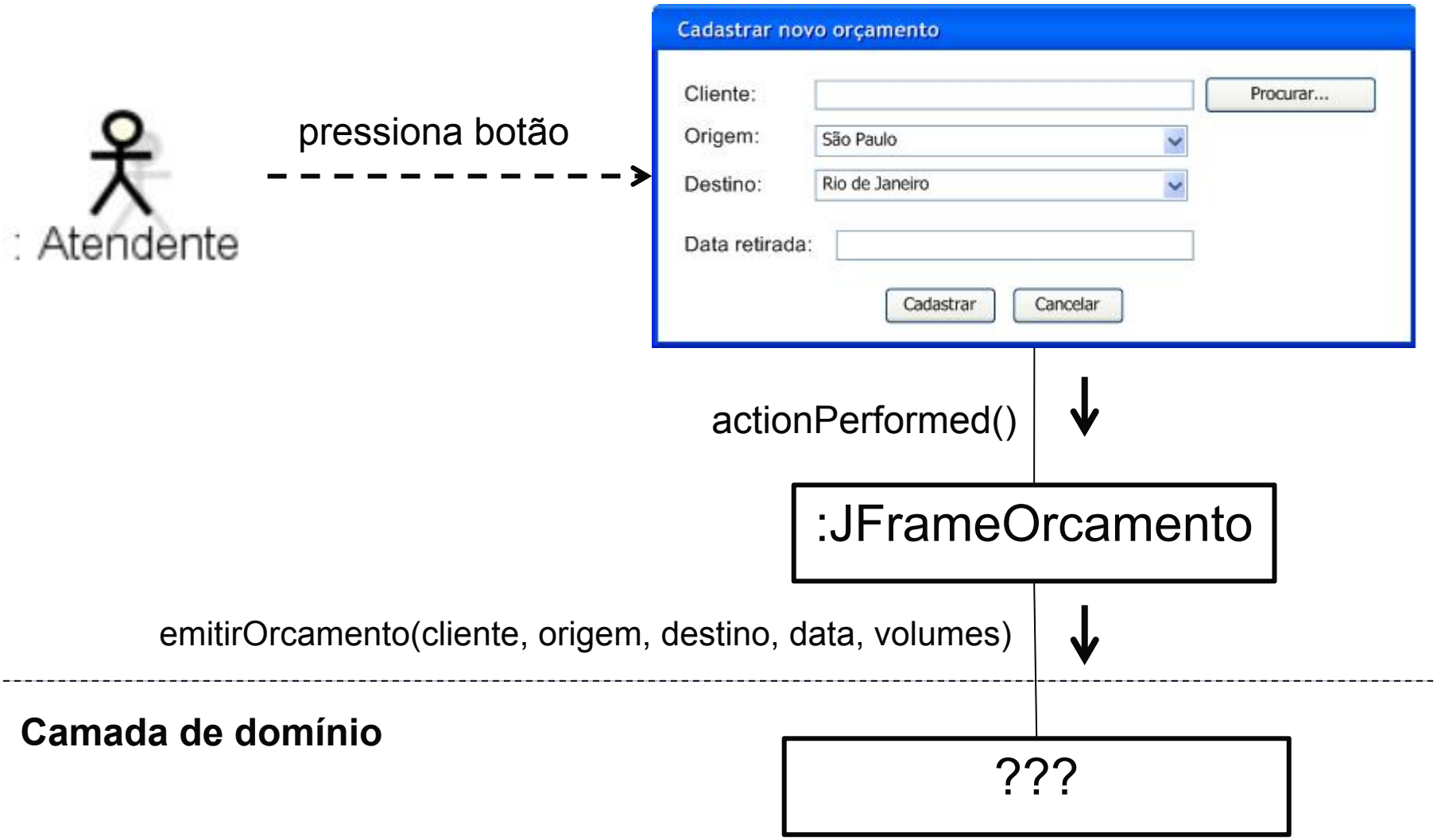
Controlador

Problema: Qual é o primeiro objeto fora da camada de interface com o usuário que recebe e coordena uma operação do sistema?

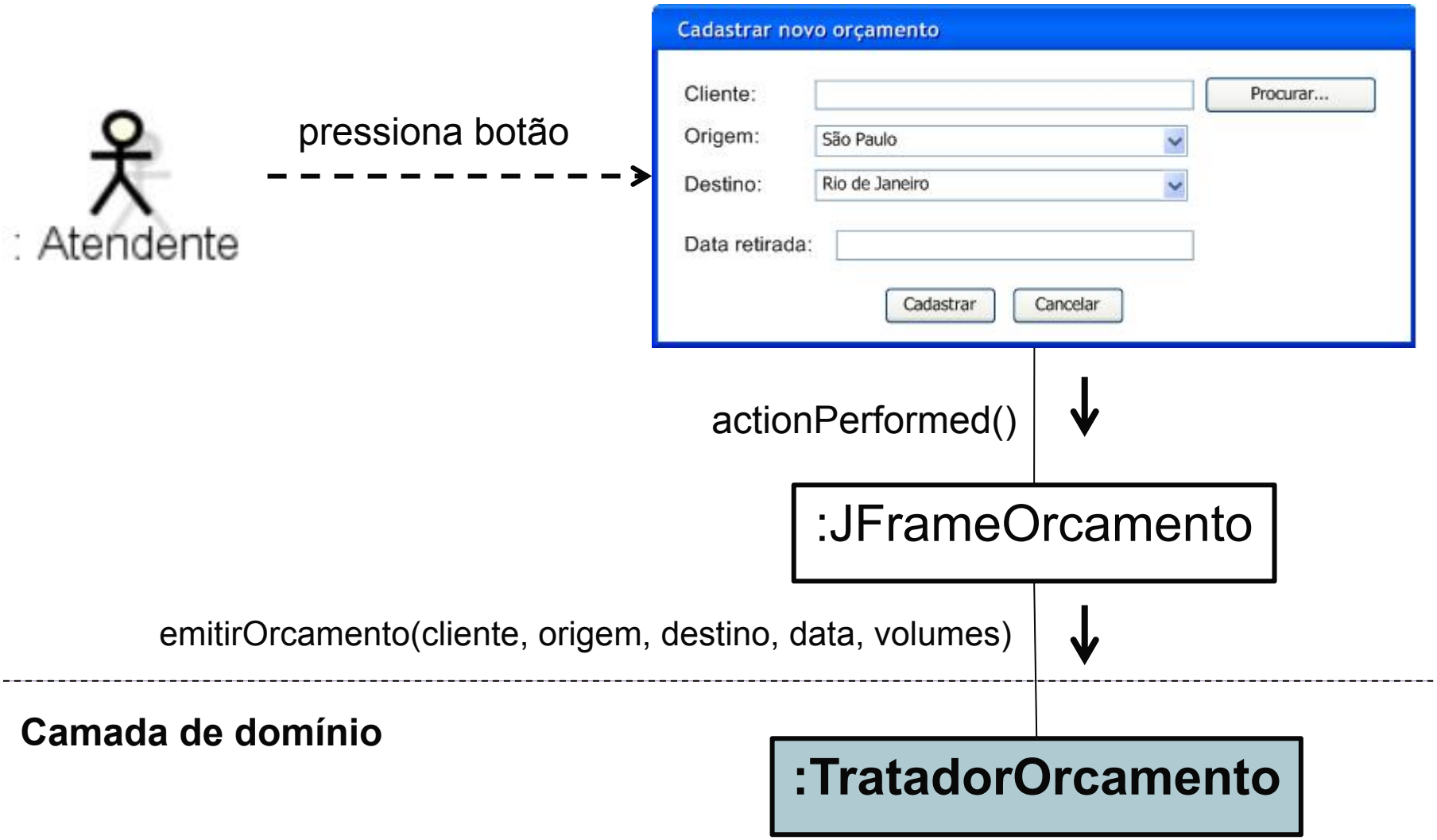
Solução: Atribua a responsabilidade a uma classe que representa:

- O “sistema global”, ou um “objeto-raiz” de um subsistema (controlador fachada)
- Um cenário de um caso de uso dentro do qual ocorre o evento do sistema

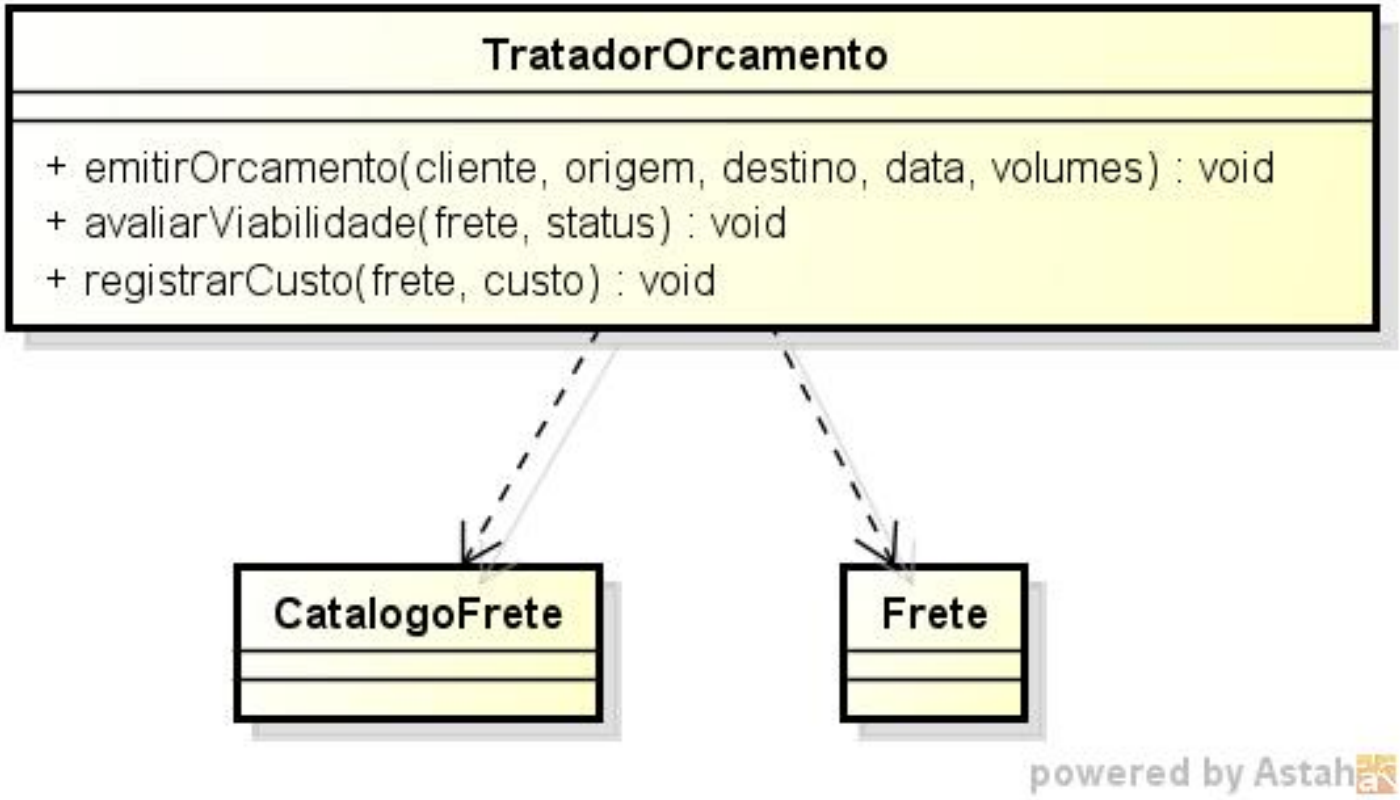
Controlador - exemplo



Controlador - exemplo



Controlador - exemplo



Controlador

Vantagens

- **Aumento das possibilidade de reutilização e de interfaces "plugáveis"**

  Como a lógica da aplicação não é tratada na camada de interface, a sua substituição é facilitada

- **O estado do caso de uso pode ser mantido com maior facilidade**

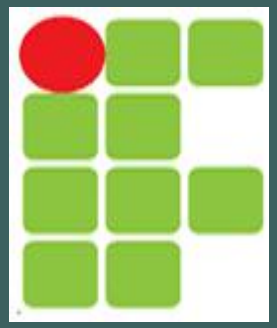
  Se o caso de uso exige uma sequência correta de operações, manter esse estado pode ser tarefa para o controlador

Controlador

Problemas

- **Sobrecarregar o controlador com todos os eventos do sistema**
- **O controlador executa as operações ao invés de delegá-las para as classes do domínio**

  O controlador por si só executa pouco trabalho



Obrigado!

tsbarcelos@ifsp.edu.br