



# Pós-Graduação Engenharia de Software

## Arquitetura de Software e Padrões de Projeto

### *Aula 05*

*Prof. Msc Rogério Augusto Rondini*  
*[rarondini.paradygma@gmail.com](mailto:rarondini.paradygma@gmail.com)*



# Conteúdo

---

- Padrões de Projeto – parte 02

- Strategy

- Adapter

- Integração de Sistemas

- Troca de Arquivos

- Mensagem

- Web Services



# Strategy Pattern

---

- Em muitas situações é necessário alterar o comportamento da aplicação dependendo do contexto em que se está executando
- Supondo o exemplo a seguir

Se tipoPagamento = "dinheiro"

Desconto de 20%

Senão

Desconto de 15%

Fim se



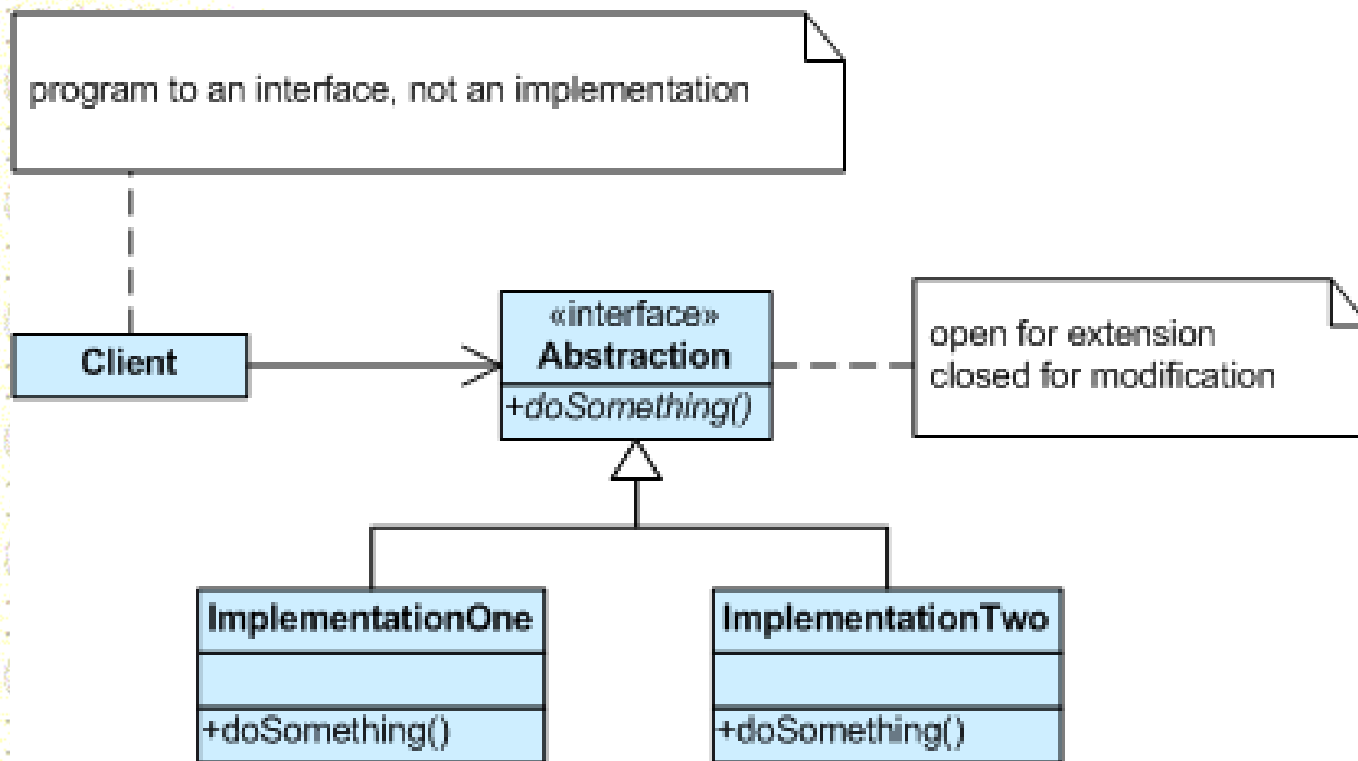
# Strategy Pattern

---

- O padrão Strategy permite a definição de uma família de algoritmos (estratégias) que podem ser executadas de acordo com o contexto
- Em geral, define-se uma abstração através de classes abstratas ou interfaces, e um conjunto de implementações



# Strategy Pattern



- Adicionalmente, pode-se utilizar o Factory para decidir qual estratégia será utilizada





# Adapter/Wrapper

---

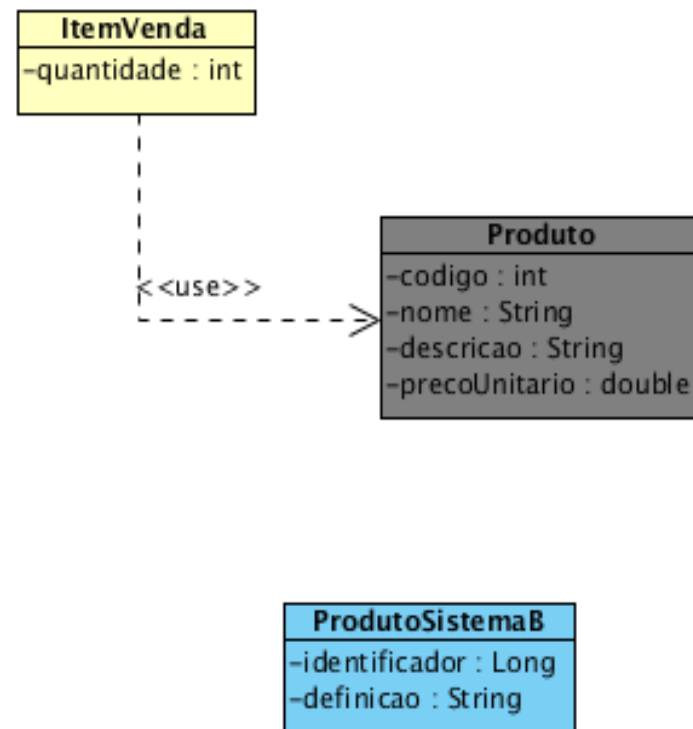
- Padrão utilizado para integrar objetos com interfaces distintas, porém, com abstração semelhante





# Adapter

- Supondo uma classe ItemVenda que tenha dependência (ou uma associação) com um objeto Produto
- Porém, o objeto Produto pode vir de diferentes sistemas, cada um com formato diferente





# Adapter

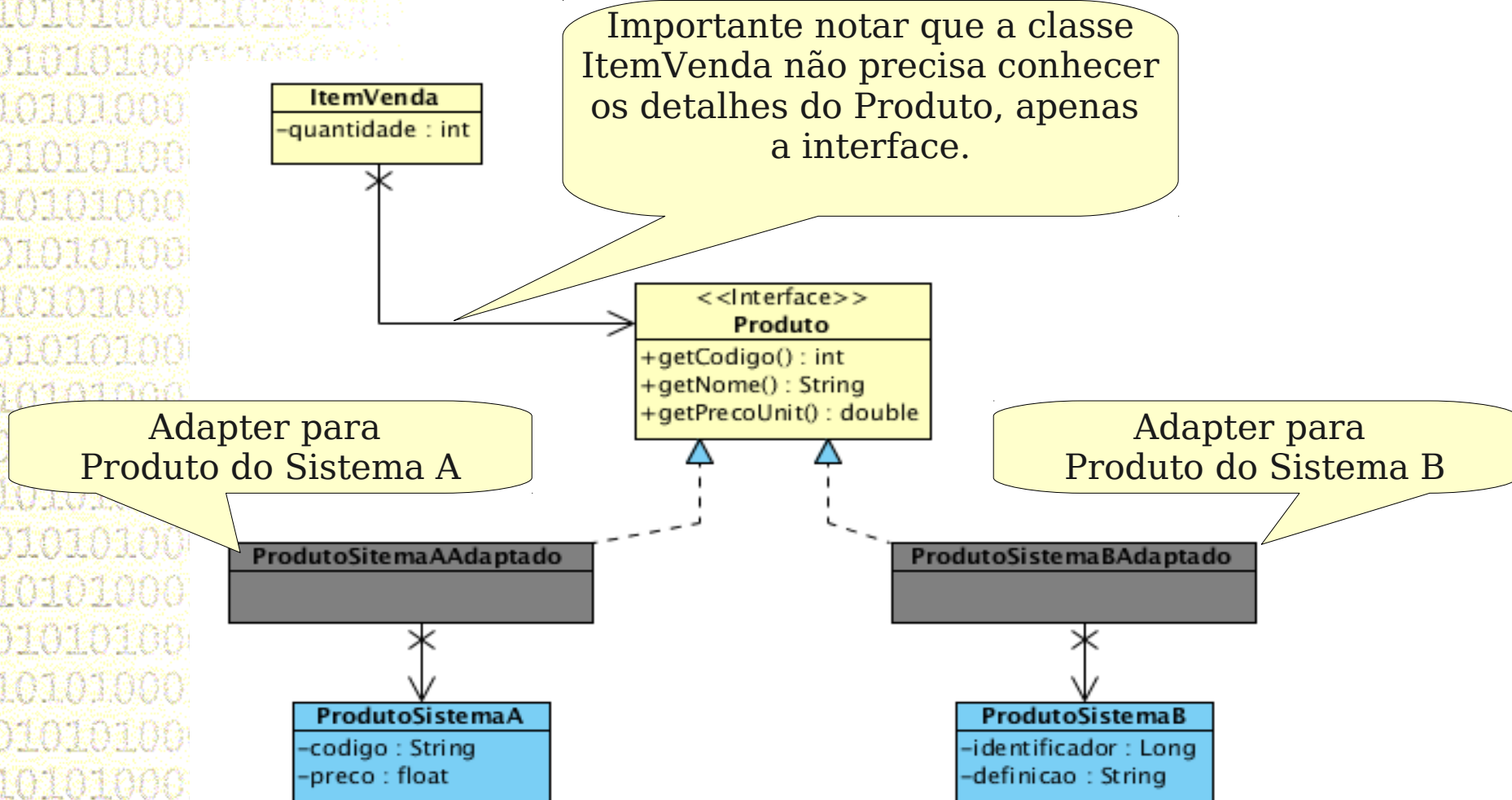
---

- Solução

- Promover a classe Produto para uma interface
- Criar adaptadores para cada produto específico



# Adapter





# Integração de Sistemas

- Atualmente, aplicações corporativas raramente resolvem problemas de forma isolada.
- Cada aplicação é responsável por resolver algo específico, que no final precisam se integrar para fornecer uma solução de negócio completa.
- Adicionalmente, nem sempre tais aplicações são escritas na mesma plataforma



# Integração de Sistemas

---

- Definir uma arquitetura de integração não é tarefa simples.
- Existem diversas soluções para um mesmo problema
- E muitas vezes, saber se a escolha foi boa ou ruim levará algum tempo :-)
- Três das principais abordagens
  - Troca de Arquivos (ainda funciona...)
  - Web Services
  - Mensagens



# Troca de Arquivos

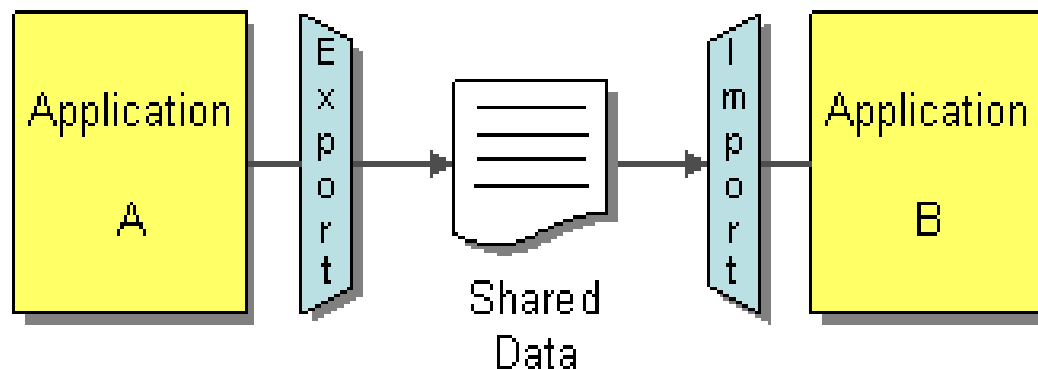
---

- Apesar de ser uma das soluções de integração mais antigas, ainda é muito utilizada
- Pode gerar alguns problemas, como a manipulação indevida de arquivos, mas normalmente o custo de implantação é baixo



# Troca de Arquivos

- Uma aplicação deve produzir dados que possam ser consumidos por outras aplicações
  - A integração deve se capaz de transformar dados em diferentes formatos
  - Dados devem ser produzidos com certa frequência
  - Formato do arquivo deve ser comum





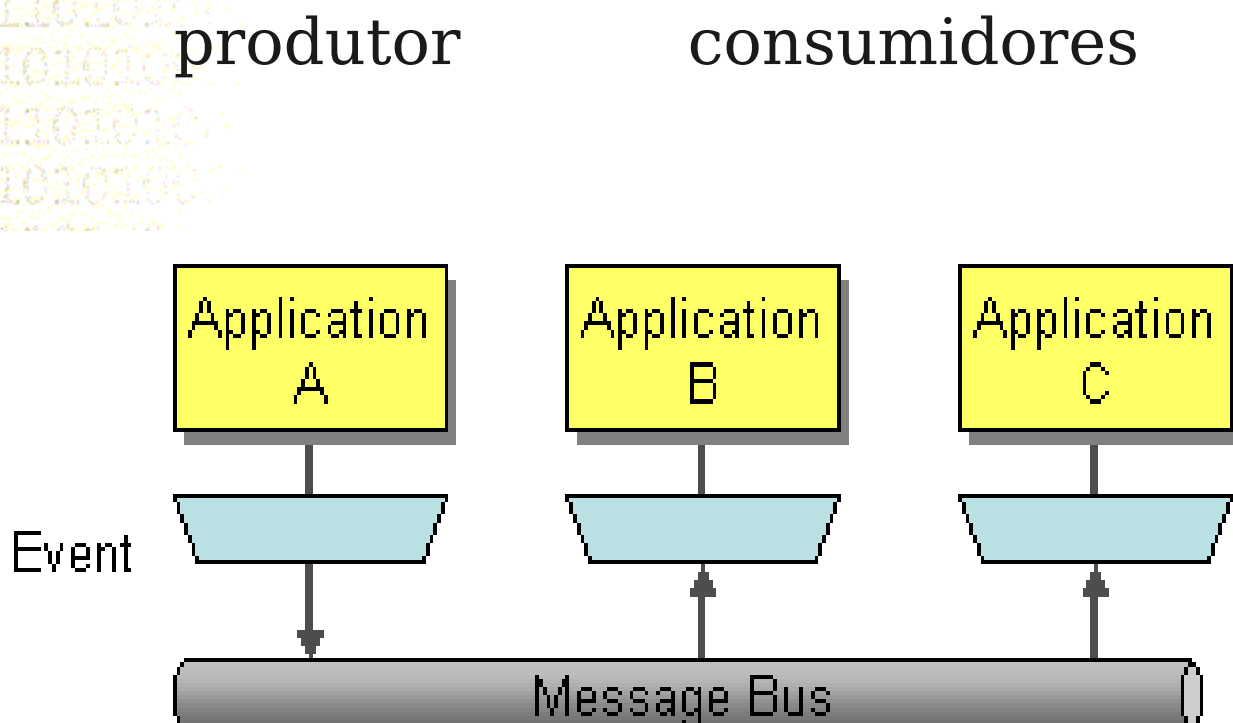


# Mensagem

---

- De certa forma, é um pouco semelhante à troca de arquivos, porém, os dados são transmitidos através de mensagens, diretamente de uma aplicação para outra
  - Não existe interferência do usuário, tornando o processo mais confiável
  - Conceito de Produtor/Consumidor (pattern)
  - Modelo Assíncrono
  - Surge também a figura do MESSAGE BUS (barramento)

# Mensagem





# Web Services

---

- Conforme já mencionado, realiza chamada direta de operações disponibilizadas por outras aplicações
- Normalmente as chamadas são síncronas
- A Figura a seguir ilustra um cenário complexo entre um cliente que executa uma operação em um sistema e este faz acesso a outros cinco bancos e uma agência de crédito



# Web Services

