



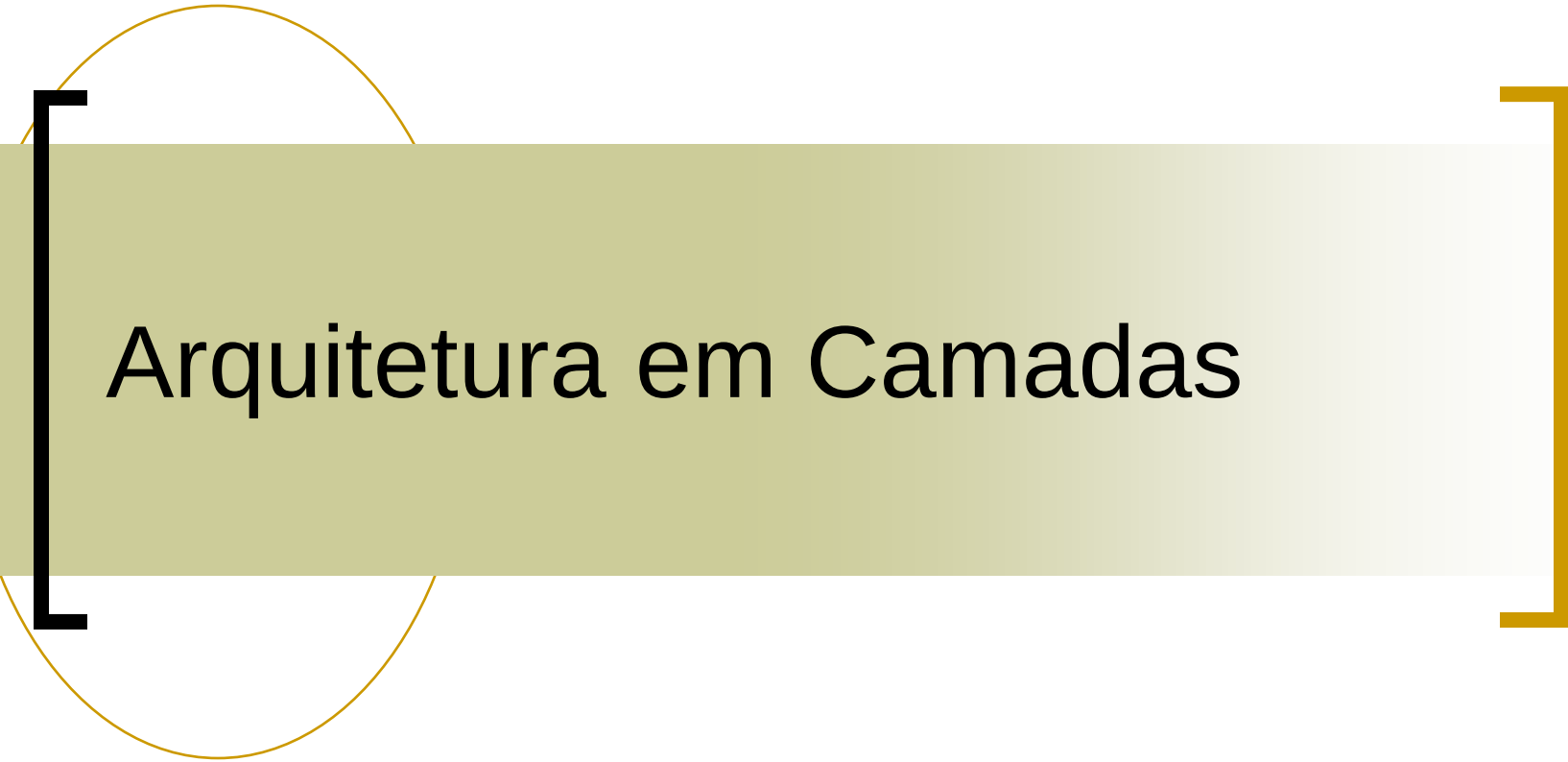
Padrões Arquiteturais: Da Desordem a Estrutura

Eduardo Figueiredo

<http://www.dcc.ufmg.br/~figueiredo>

[Da Desordem a Estrutura]

- Layered Architecture
 - Arquitetura em Camadas
- Blackboard
 - Arquitetura de Repositório
- Pipes and Filters
 - Duto e Filtro



Arquitetura em Camadas

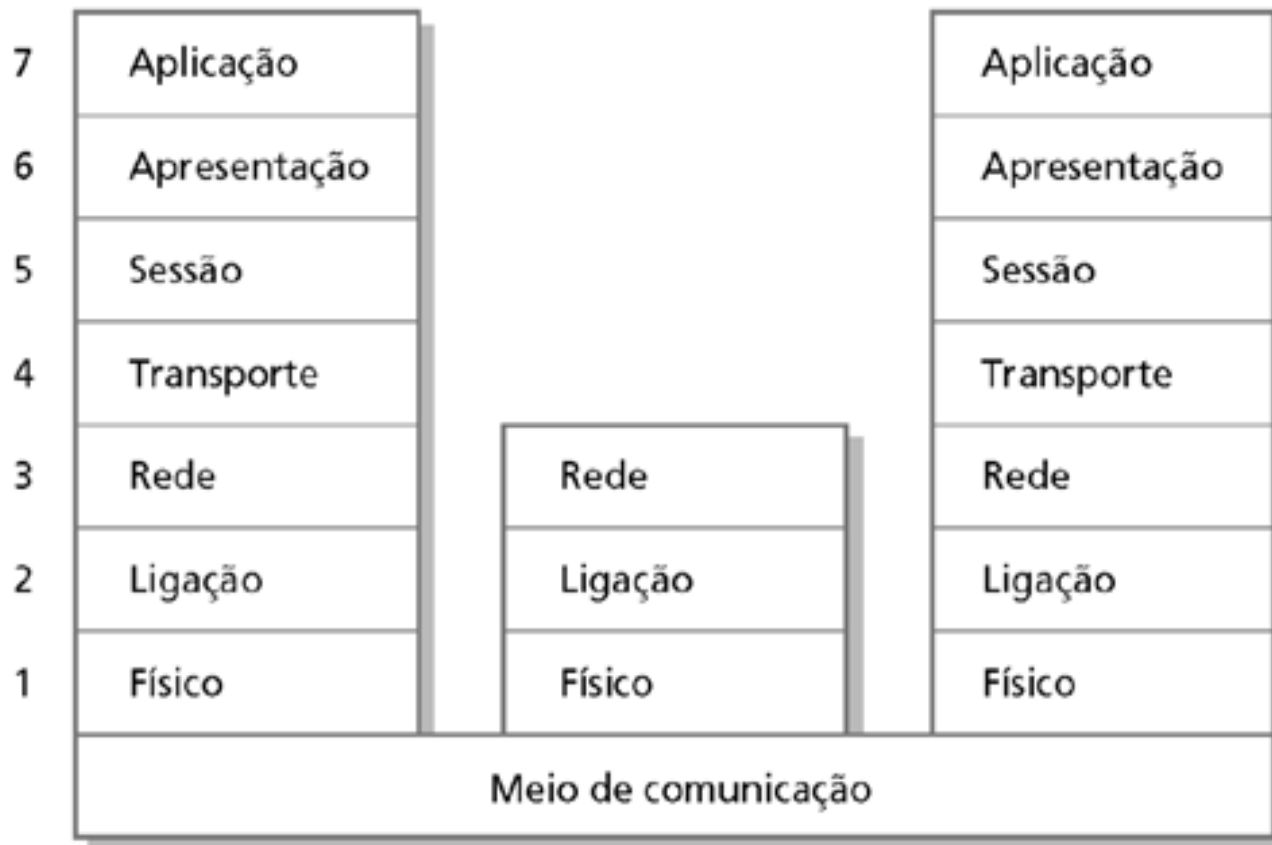
[Arquitetura em Camadas]

- Organiza o sistema em um conjunto de camadas
 - Cada camada oferece um conjunto de serviços
- Uma camada somente
 - Solicita serviços da camada inferior
 - Fornece serviços para a camada superior



[Exemplo 1: Protocolos OSI]

Modelo de camadas para sistemas de comunicação



[Exemplo 2: Três Camadas]



[Vantagens]

- Favorece o modelo de desenvolvimento incremental
- As camadas podem ser facilmente substituídas por equivalentes
 - Requer interfaces estáveis
- Mudanças em uma camada teoricamente só impacta a camada superior
- Camadas superiores podem ser independentes de plataforma/hardware



[Desvantagens]



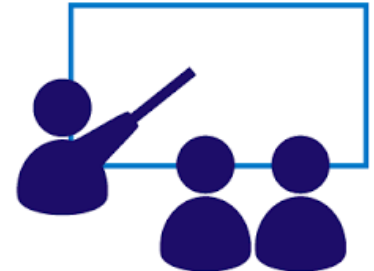
- Estruturar o sistema em camadas não é trivial
 - Pode ser difícil identificar quais os serviços elementares das camadas inferiores
- Muitas camadas podem comprometer o desempenho do sistema
 - A requisição tem que trafegar pelas várias camadas até ser atendida



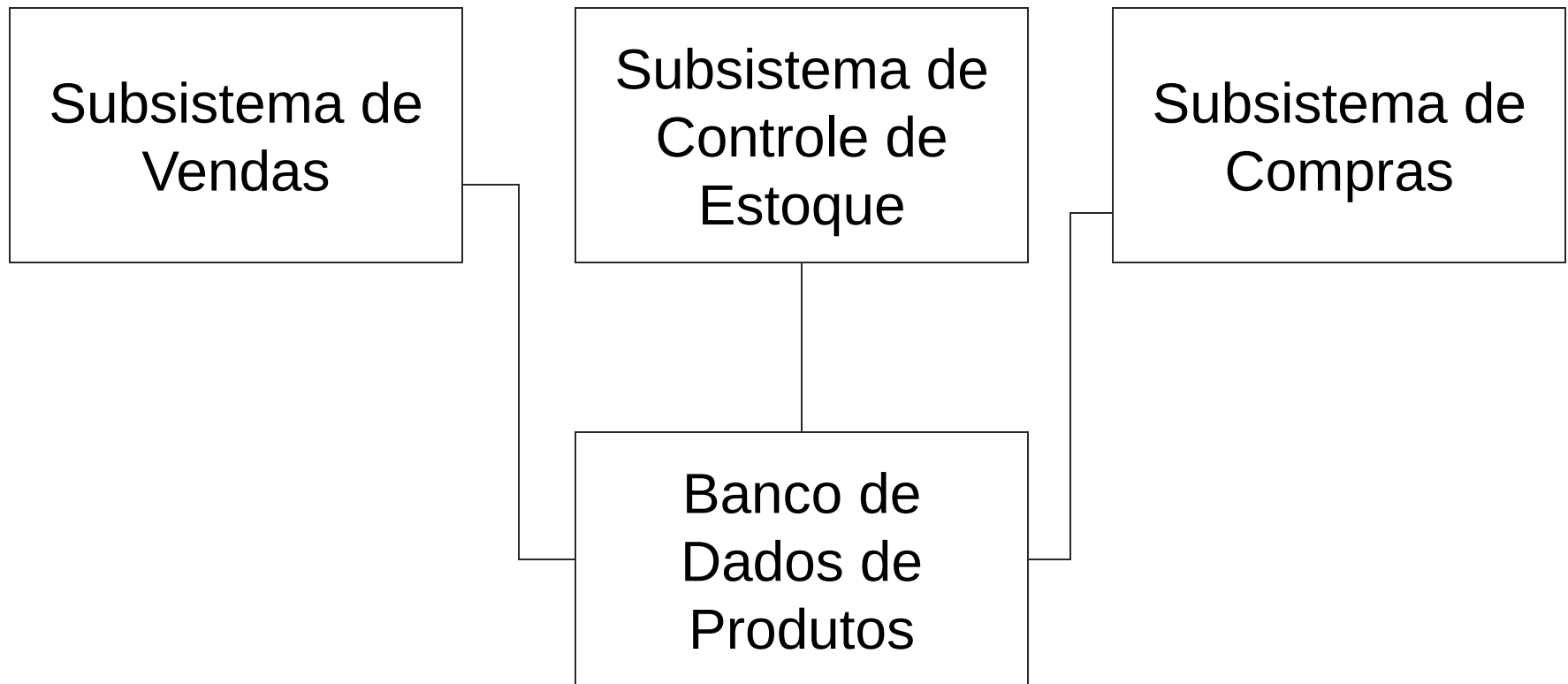
Arquitetura de Repositório

[Arquitetura de Repositório]

- Também conhecido como *Blackboard*
- Os subsistemas manipulam a mesma base de dados
 - Um (ou mais) subsistema gera os dados
 - Vários subsistemas leem os dados
- Adotado principalmente quando dados são compartilhados em grandes quantidades



[Exemplo de Repositório]



[Vantagens]



- Componentes podem ser independentes
 - Os subsistemas que gravam dados não necessitam saber quem os usa
- Maneira eficiente de compartilhar dados e backup centralizado
- Formas de proteção dos dados podem ser implementadas
- Fácil integrar novos subsistemas

[Desvantagens]

- Ponto único de falhas no sistema
 - Problemas no repositório afetam o sistema inteiro
- Ineficiência na organização e comunicação dos componentes
 - Subsistemas diferentes podem ter requisitos diferentes
- Dificuldade para distribuir os dados
 - Dados redundantes ou inconsistentes





Duto e Filtro

[Duto e Filtro]

- Padrão de organização da dinâmica de um sistema
- Dois papéis principais
 - **Dutos:** componentes que conduzem ou distribuem os dados
 - **Filtros:** componentes que transformam os dados
- Usado principalmente em aplicações de processamento de dados



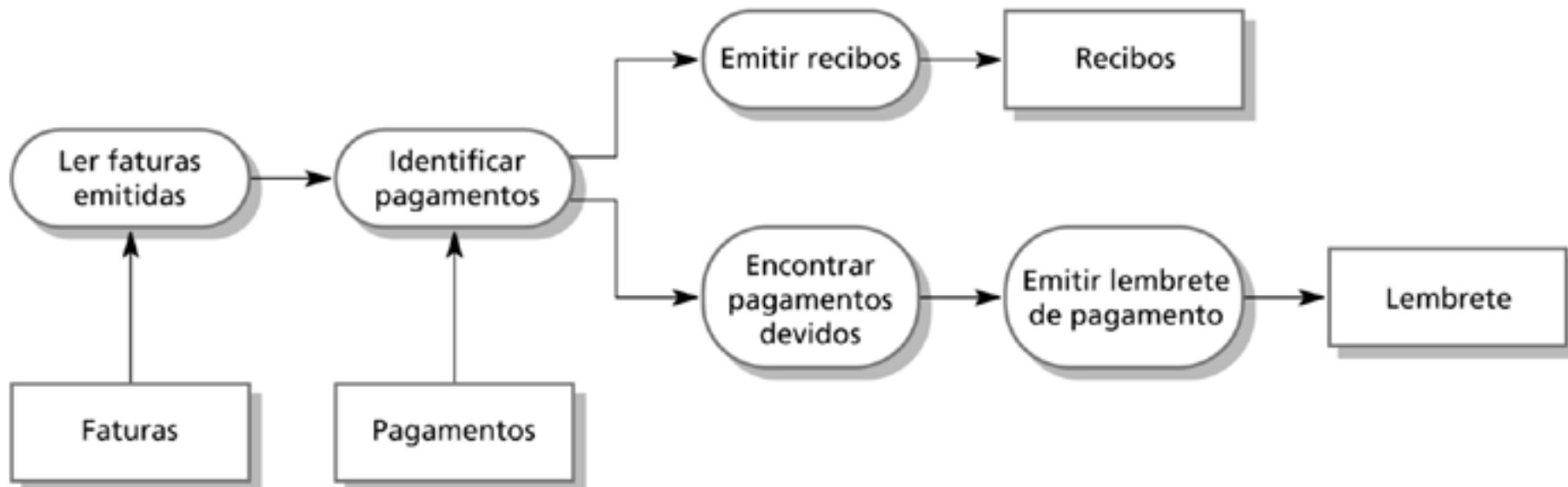
[Dinâmica do Padrão]

- Os dados de entrada se movem pelos **duto**s
- Os dados são transformados pelos **filtros** até serem convertidos em dados de saída
 - As transformações podem ocorrer em sequência ou em paralelo



[Exemplo de Dutos e Filtros]

- Entradas: Faturas e Pagamentos
- Saídas: Recibos e Lembretes



[Vantagens]



- O módulo de transformação (filtro) é bem modular
 - Facilmente reusável e substituível
- O estilo de workflow é aderente a muitos processos de negócios
- É simples evoluir o sistema pela adição de filtros
- Se aplica tanto a sistemas sequenciais quanto a sistemas concorrentes

[Desvantagens]

- O formato dos dados trafegados deve ser acordado entre os componentes
- Pode haver um *overhead* causado pela padronização dos dados
- Incompatibilidade no formato dos dados pode dificultar a reutilização de filtros



[Bibliografia]

- Ian Sommerville. **Engenharia de Software**, 10a. Edição. 2019.
 - Cap. 6 (Seção 6.3)
- F. Buschmann et al. **Pattern-Oriented Software Architecture: A System of Patterns**. John Wiley & Sons, 1996.
 - Cap. 2 Architectural Patterns