Projeto de Arquitetura

Objetivos

- Apresentar projeto de arquitetura e discutir sua importância
- Explicar as decisões de projeto de arquitetura que têm de ser feitas
- Apresentar três estilos complementares de arquitetura que abrangem a organização, decomposição e controle
- Discutir como as arquiteturas de referência são usadas para comunicar e comparar arquiteturas

Tópicos abordados

- Decisões de projeto de arquitetura
- Organização de sistema
- Estilos de decomposição modular
- Modelos de controle
- Arquiteturas de referência

Arquitetura de software

- O processo de projeto para identificar os subsistemas que constituem um sistema e o *framework* para controle e comunicação de subsistema é denominado projeto de arquitetura.
- A saída desse processo de projeto é uma descrição da arquitetura de software.

Projeto de arquitetura

- É o primeiro estágio do processo de projeto de sistema.
- Representa a ligação entre os processos de especificação e de projeto.
- É freqüentemente conduzido em paralelo com algumas atividades de especificação.
- Envolve a identificação dos componentes principais do sistema e suas comunicações.

Vantagens da arquitetura explícita

- Comunicação de *stakeholder*
  - A arquitetura pode ser usada como um foco de discussão pelos *stakeholders* do sistema.
- Análise de sistema
  - Se há possibilidade de o sistema atender a seus requisitos não funcionais.
- Reuso em larga escala
  - A arquitetura pode ser reusável em uma variedade de sistemas.

Características de arquitetura e de sistema

- Desempenho
  - Localizar operações críticas e minimizar comunicações. Usar componentes de alta ao invés de baixa granularidade.
- Proteção
  - Usar uma arquitetura em camadas com itens críticos nas camadas mais internas.
- Segurança
  - Localizar características críticas de segurança em um pequeno número de subsistemas.
- Disponibilidade
  - Incluir componentes redundantes e mecanismos para tolerância à falhas.
- Facilidade de manutenção
  - Usar componentes substituíveis e de baixa granulariade.

Conflitos de arquitetura

- O uso de componentes de alta granularidade aprimora o desempenho mas diminui a facilidade de manutenção.
- A introdução de dados redundantes aprimora a disponibilidade, mas torna a proteção mais difícil.
- Ao localizar características relacionadas à segurança, geralmente significa maior comunicação e, por essa razão, o desempenho é degradado.

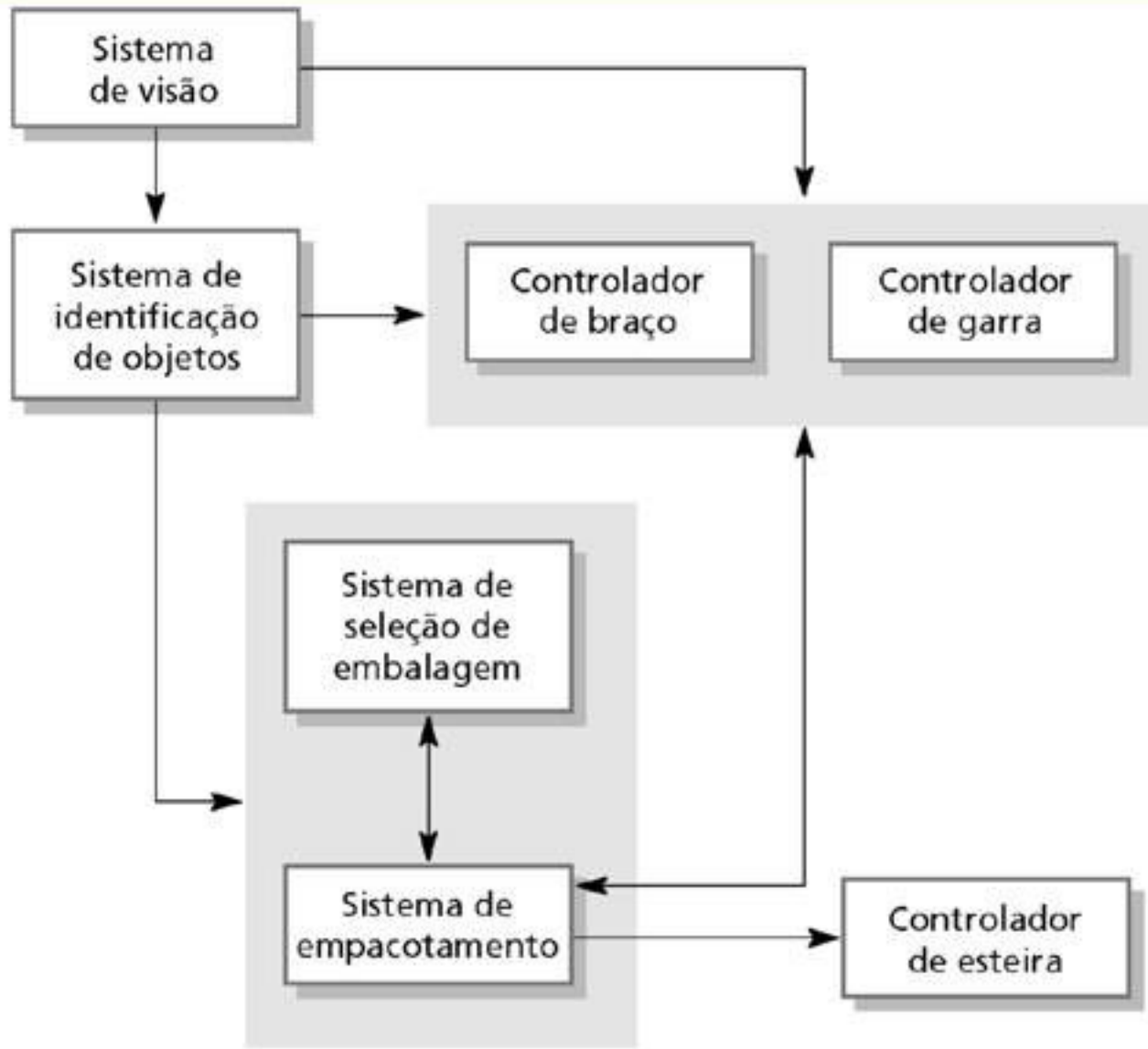
Estruturação de sistema

- Está relacionado à decomposição do sistema em subsistemas que interagem.
- O projeto de arquitetura é normalmente expresso como um diagrama de blocos que apresentam uma visão geral da estrutura do sistema.
- Modelos mais específicos que mostram como os subsistemas compartilham dados, como são distribuídos e como interfaceiam uns com os outros, também podem ser desenvolvidos.

Sistema de controle robotizado de empacotamento

Figura 11.1

Diagrama de blocos de um sistema de controle robotizado de empacotamento.



Diagramas caixa e linha

- Muito abstrato – não mostram a natureza dos relacionamento de componentes, nem as propriedades externamente visíveis dos subsistemas.
- Contudo, são úteis para comunicação com os stakeholders e para planejamento de projeto.

Decisões de projeto de arquitetura

- Projeto de arquitetura é um processo criativo cujas atividades diferem radicalmente dependendo do tipo de sistema que está sendo desenvolvido.
- Contudo, uma série de decisões comuns afetam todos os processos de projeto.

Decisões de projeto de arquitetura

- Existe uma arquitetura genérica de aplicação que possa ser usada?
- Como o sistema será distribuído?
- Quais estilos de arquitetura são apropriados?
- Qual será a abordagem fundamental usada para estruturar o sistema?
- Como o sistema será decomposto em módulos?
- Qual estratégia deve ser usada?
- Como o projeto de arquitetura será avaliado?
- Como a arquitetura do sistema deve ser documentada?

Reuso de arquitetura

- Sistemas do mesmo domínio freqüentemente têm arquiteturas similares que refletem os conceitos de domínio.
- As linhas do produto de aplicação são construídas em torno de um núcleo de arquitetura com variantes que satisfazem requisitos específicos de clientes.
- As arquiteturas de aplicação são abordadas no Capítulo 13, e linhas de produto no Capítulo 18.

Estilos de arquitetura

- O modelo de arquitetura de um sistema pode estar de acordo com um modelo ou com um estilo genérico de arquitetura.
- A consciência desses estilos pode simplificar o problema de definição de arquiteturas de sistema.
- Contudo, a maioria dos sistemas de grande porte são heterogêneos e não seguem um único estilo de arquitetura.

Modelos de arquitetura

- São usados para documentar um projeto de arquitetura.
- odelos estáticos de estrutura que mostram os principais componentes do sistema.
- Um modelo dinâmico de processo que mostra a estrutura de processo do sistema.
- Um modelo de interface que define as interfaces de subsistemas.
- Modelos de relacionamentos, tal como um modelo de fluxo de dados, que mostra os relacionamentos dos subsistemas.
- Um modelo de distribuição que mostra como subsistemas são distribuídos pelos computadores.

Organização de sistema

- Reflete a estratégia básica que é usada para estruturar um sistema.
- Três estilos de organizações são amplamente usados:
  - O estilo de repositório de dados compartilhados;
  - Estilo de serviços e servidores compartilhados;
  - Estilo de máquina abstrata ou em camadas.

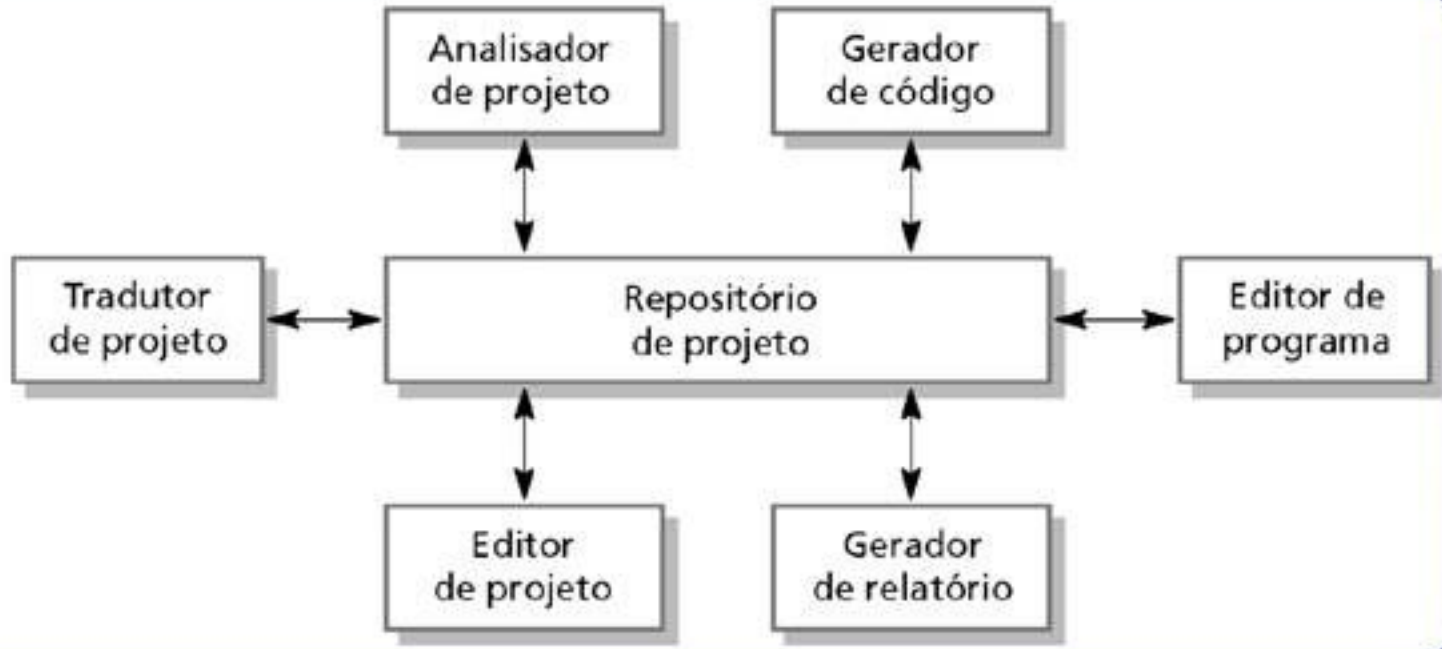
Modelo de repositório

- Os subsistemas devem trocar dados. Isso pode ser feito de duas maneiras:
  - Os dados compartilhados são mantidos em um banco de dados central ou repositório e podem ser acessados por todos os subsistemas;
  - Cada subsistema mantém seu próprio banco de dados e passa dados explicitamente para outros subsistemas.
- Quando grandes quantidades de dados são compartilhadas, o modelo de repositório de compartilhamento é o mais usado.

Arquitetura de conjunto de ferramentas CASE

Figura 11.2

Arquitetura de um conjunto de ferramentas CASE integradas.



Características de modelo de repositório

- Vantagens
  - É uma maneira eficiente de compartilhar grandes quantidades de dados;
  - Os subsistemas não necessitam saber como os dados são produzidos pelo gerenciamento centralizado, por exemplo, *backup*, proteção, etc.
  - Um modelo de compartilhamento é publicado como o esquema de repositório.
- Desvantagens
  - Os subsistemas devem estar de acordo com um modelo de dados do repositório. É, inevitavelmente, um compromisso;
  - A evolução de dados é difícil e dispendiosa;
  - Não há escopo para políticas específicas de gerenciamento;
  - Dificuldade para distribuirde forma eficiente.

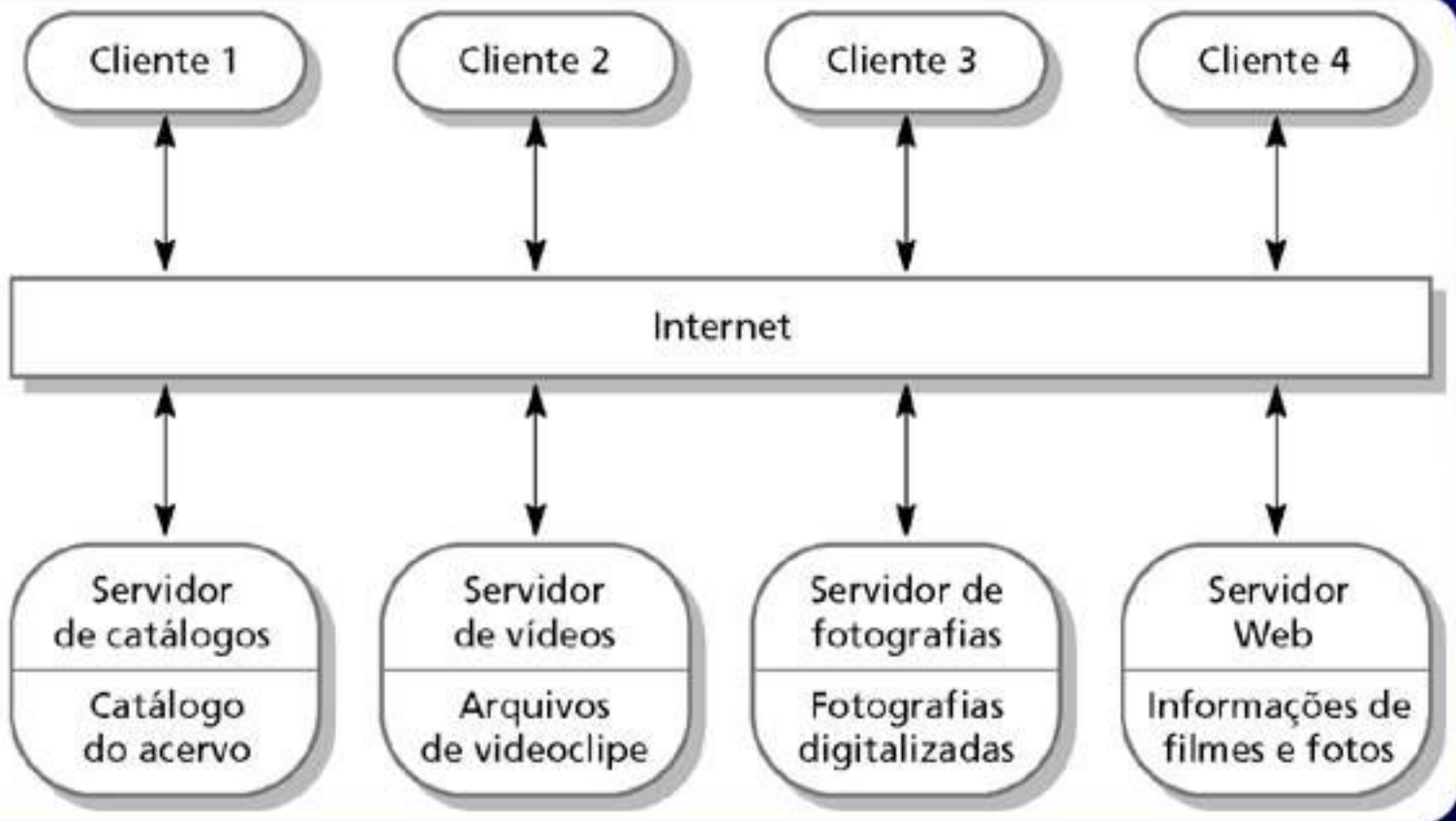
Modelo cliente-servidor

- É um modelo distribuído de sistema que mostra como dado e processamento são distribuídos por uma variedade de componentes.
- Estabelece servidores independentes que fornecem serviços específicos, tais como impressão, gerenciamento de dados, etc.
- Estabelece clientes que acessam esses serviços.
- É uma rede que permite aos clientes acessar os servidores.

Biblioteca de filmes e fotografias

Figura 11.3

Arquitetura de um sistema de acervo de filmes e fotografias.



Características de cliente-servidor

- Vantagens
  - A distribuição de dados é direta;
  - Faz uso efetivo dos sistemas em rede. Pode solicitar hardware mais barato;
  - É fácil adicionar novos servidores ou atualizar servidores existentes.
- Desvantagens
  - Nenhum modelo de dados compartilhado e, dessa forma, os subsistemas usam diferentes organizações de dados; por isso, a troca de dados pode ser ineficiente.
  - Gerenciamento redundante em cada servidor;
  - Nenhum registro central de nomes e serviços – pode ser difícil descobrir quais servidores e serviços estão disponíveis.

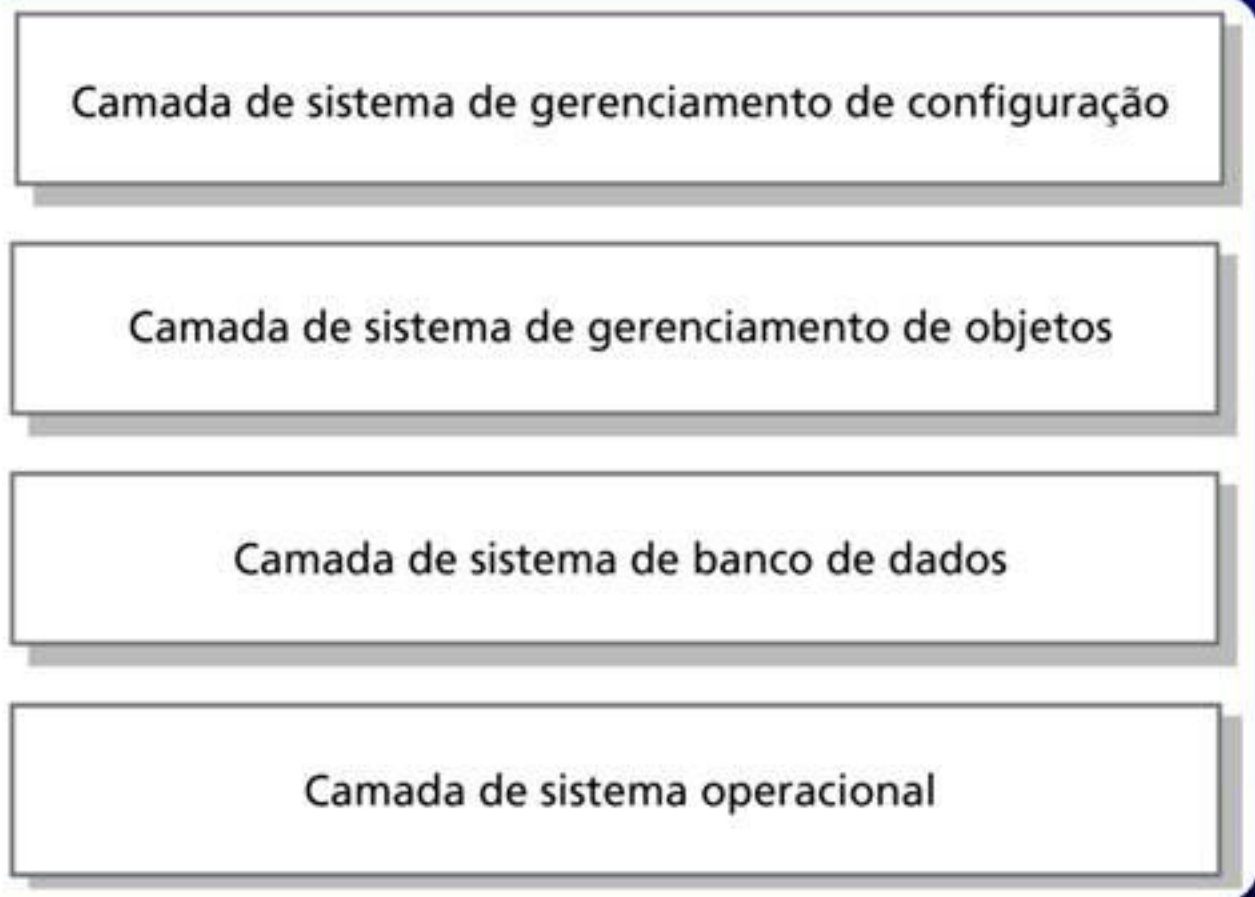
Modelo de máquina abstrata (em camadas)

- Usado para modelar o interfaceamento dos subsistemas.
- Organiza o sistema em um conjunto de camadas (ou máquinas abstratas), cada uma dss quais fornecenso um conjunto de serviços.
- Apóia o desenvolvimento incremental dos subsistemas em camadas diferentes. Quando uma camada de interface muda, somente a camada adjacente é afetada.
- Contudo, é freqüentemente artificial estruturar sistemas dessa maneira.

Sistema de gerenciamento de versões

Figura 11.4

Modelo em camadas de um sistema de gerenciamento de versões.



Estilos de decomposição modular

- Estilos de decomposição de subsistemas em módulos.
- Não há distinção rígida entre organização de sistema e decomposição modular.

Subsistemas e módulos

- Um subsistema é um sistema em si cuja operação não depende dos serviços fornecidos por outros subsistemas.
- Um módulo é um componente de sistema que fornece serviços para outros módulos; não é normalmente considerado um sistema separado.

Decomposição modular

- É um outro nível de estrutura onde os subsistemas são decompostos em módulos.
- Dois modelos de decomposição modular podem ser usados
  - Um modelo de objeto onde o sistema é decomposto em objetos que se comunicam;
  - Um modelo de *pipeline* ou fluxo de dados onde o sistema é decomposto em módulos funcionais que transformam entradas em saídas.
- Se possível, decisões sobre concorrência devem ser postergadas até que os módulos estejam implementados.

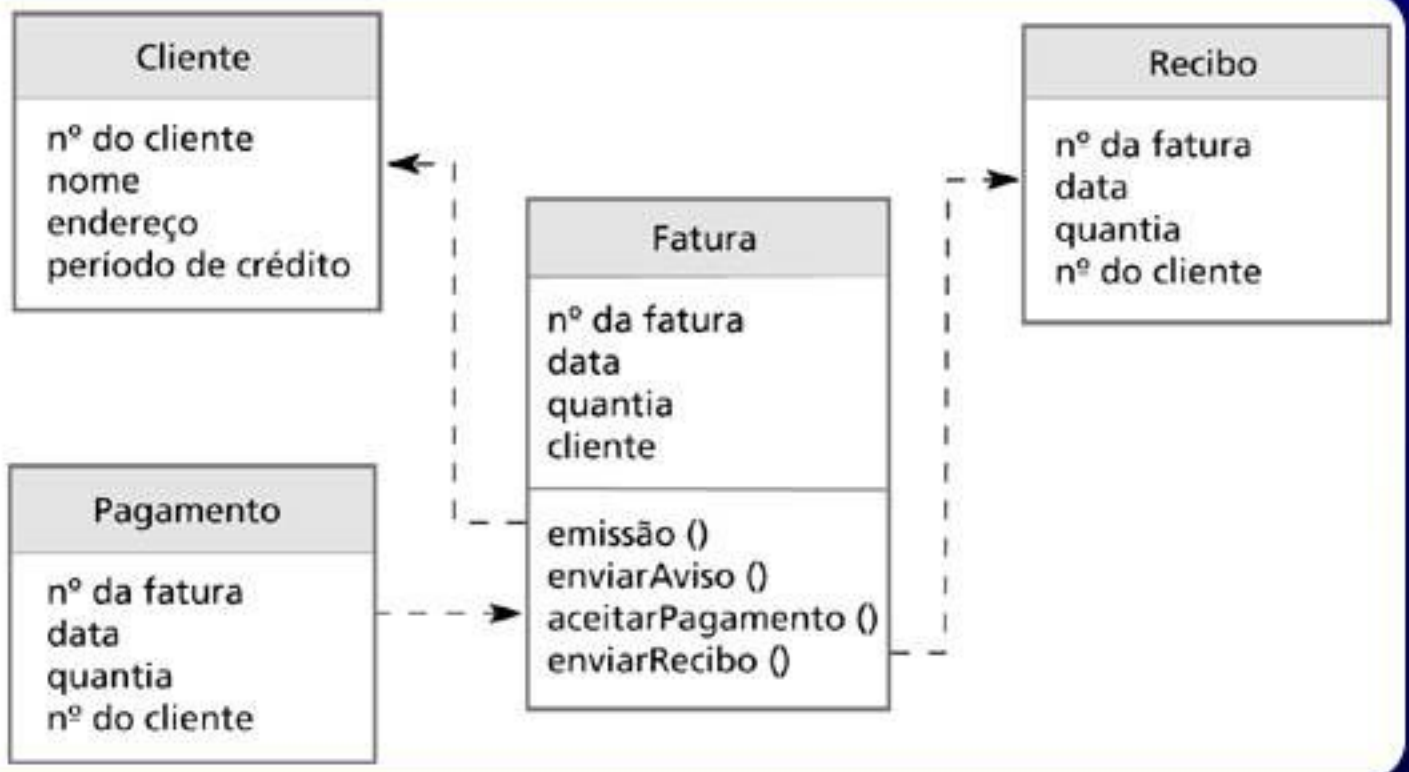
Modelos de objetos

- Estruturar o sistema em um conjunto de objetos fracamente acoplados com interfaces bem definidas.
- A decomposição orientada a objetos está relacionada à identificação de classes de objetos, aos seus atributos e às suas operações.
- Quando implementados, os objetos são criados a partir dessas classes e um tipo de controle é usado para coordenar as operações de objetos.

Sistema de processamento de faturas

Figura 11.5

Modelo de objeto de um sistema de processamento de faturas.



Vantagens do modelo de objetos

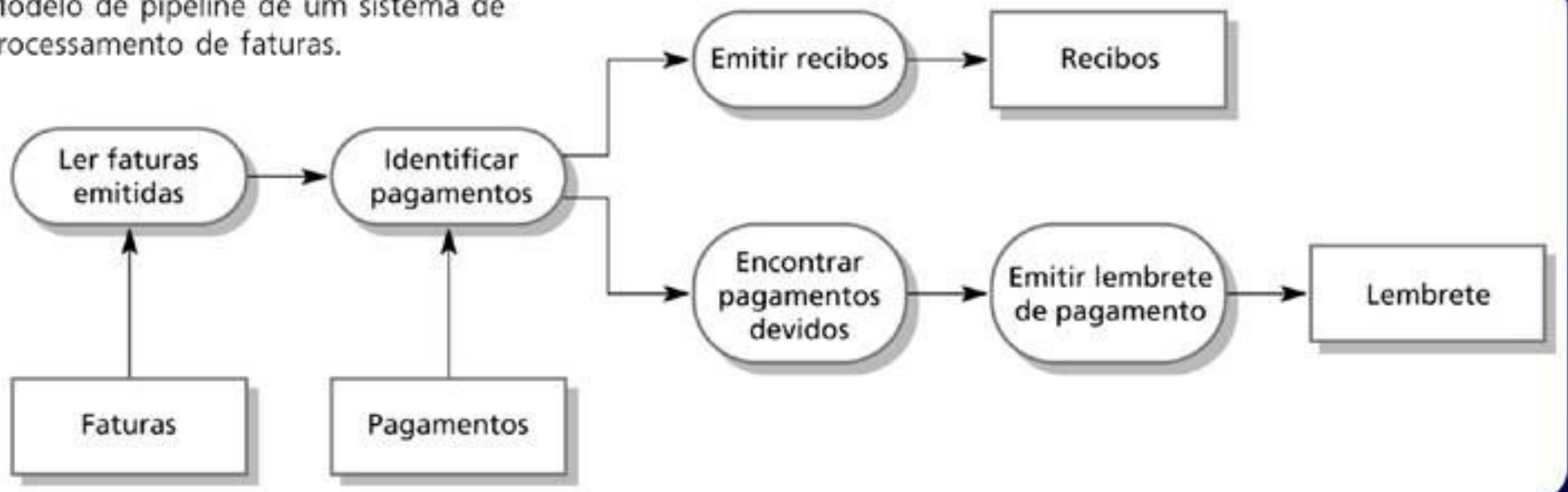
- Objetos não são firmemente acoplados e, desse modo, sua implementação pode ser modificada sem afetar outros objetos.
- Os objetos podem refletir entidades do mundo real.
- Linguagens de implementação orientada a objeto são amplamente usadas.
- Contudo, mudanças de interface de objeto podem causar problemas e entidades complexas podem ser difíceis de representar como objetos.

Pipelining orientado a funções

- Transformações funcionais processam suas entradas para produzir saídas.
- Pode ser chamado de estilo de duto (pipe) e filtro (como no *shell* UNIX)
- Variações dessa abordagem são muito comuns. Quando as transformações são seqüenciais, isso é um modelo seqüencil em lotes, que é extensivamente usado em sistemas de processamento de dados.
- Não é realmente adequado para sistemas interativos.

Sistema de processamento de faturas

Figura 11.6 Modelo de pipeline de um sistema de processamento de faturas.



Vantagens do modelo de *pipeline*

- Apóia reuso de transformações.
- Organização intuitiva para comunicação com stakeholders.
- É fácil adicionar novas transformações.
- É relativamente simples implementar como sistema concorrente ou seqüencial.
- Contudo, requer um formato comum para a transferência de dados ao longo do pipeline e o apoio a interações baseadas em eventos é difícil.

Modelos de controle

- Diferente do modelo de decomposição de sistema, os modelos de controle estão relacionados ao fluxo de controle entre subsistemas
- Controle centralizado
  - Um subsistema tem responsabilidade global pelo controle, e inicia e pára outros sistemas.
- Controle baseado em eventos
  - Cada subsistema pode responder a eventos gerados externamente a partir de outros subsistemas ou do ambiente do sistema.

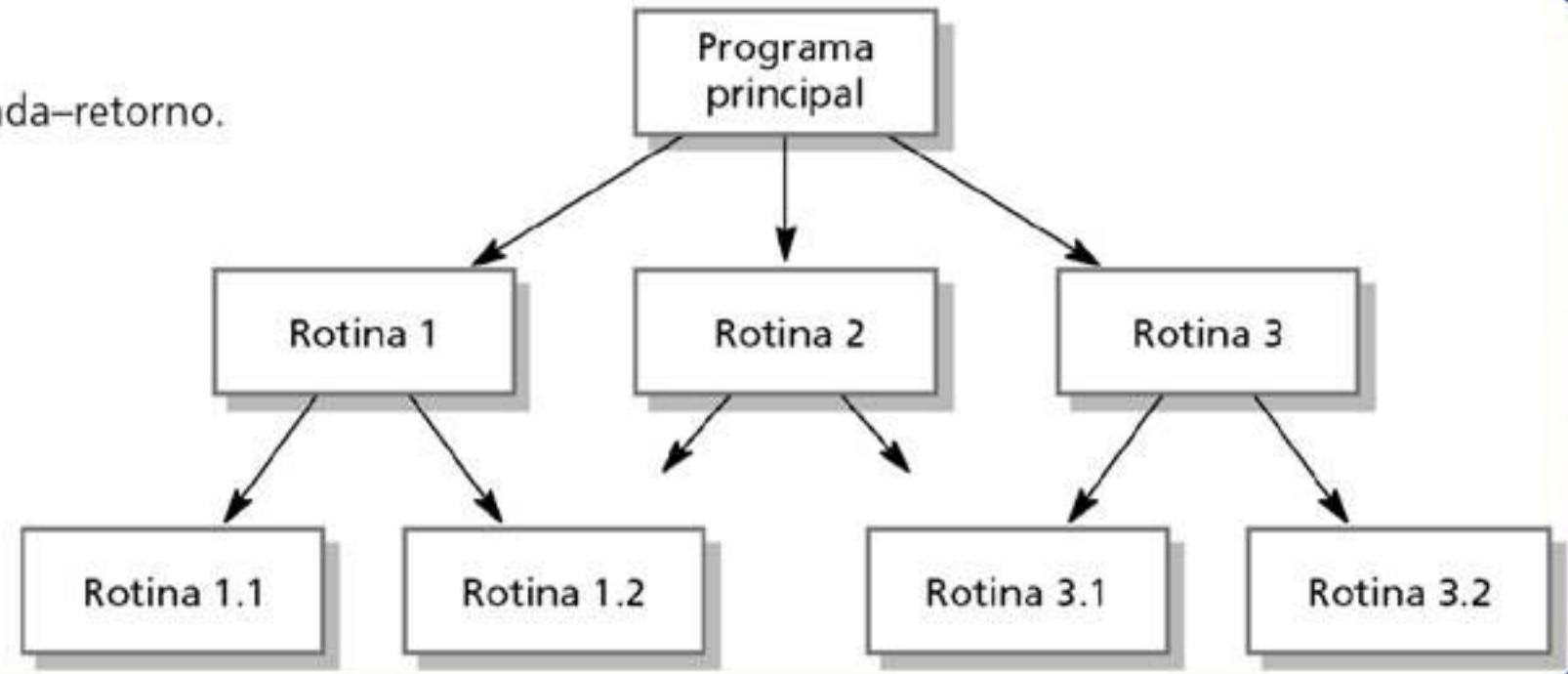
Controle centralizado

- Um subsistema de controle é responsável pelo gerenciamento da execução de outros subsistemas.
- Modelo chamada-retorno
  - É o modelo de subrotina *top-down* onde o controle inicia no topo de uma hierarquia de subrotina, e se move para baixo da hierarquia. É aplicável a sistemas seqüenciais.
- Modelo de gerenciador
  - É aplicável a sistemas concorrentes. Um componente de sistema controla a parada, o início e a coordenação de outros processos de sistema. Pode ser implementado em sistemas seqüenciais como uma declaração 'Case'.

Modelo chamada-retorno

Figura 11.7

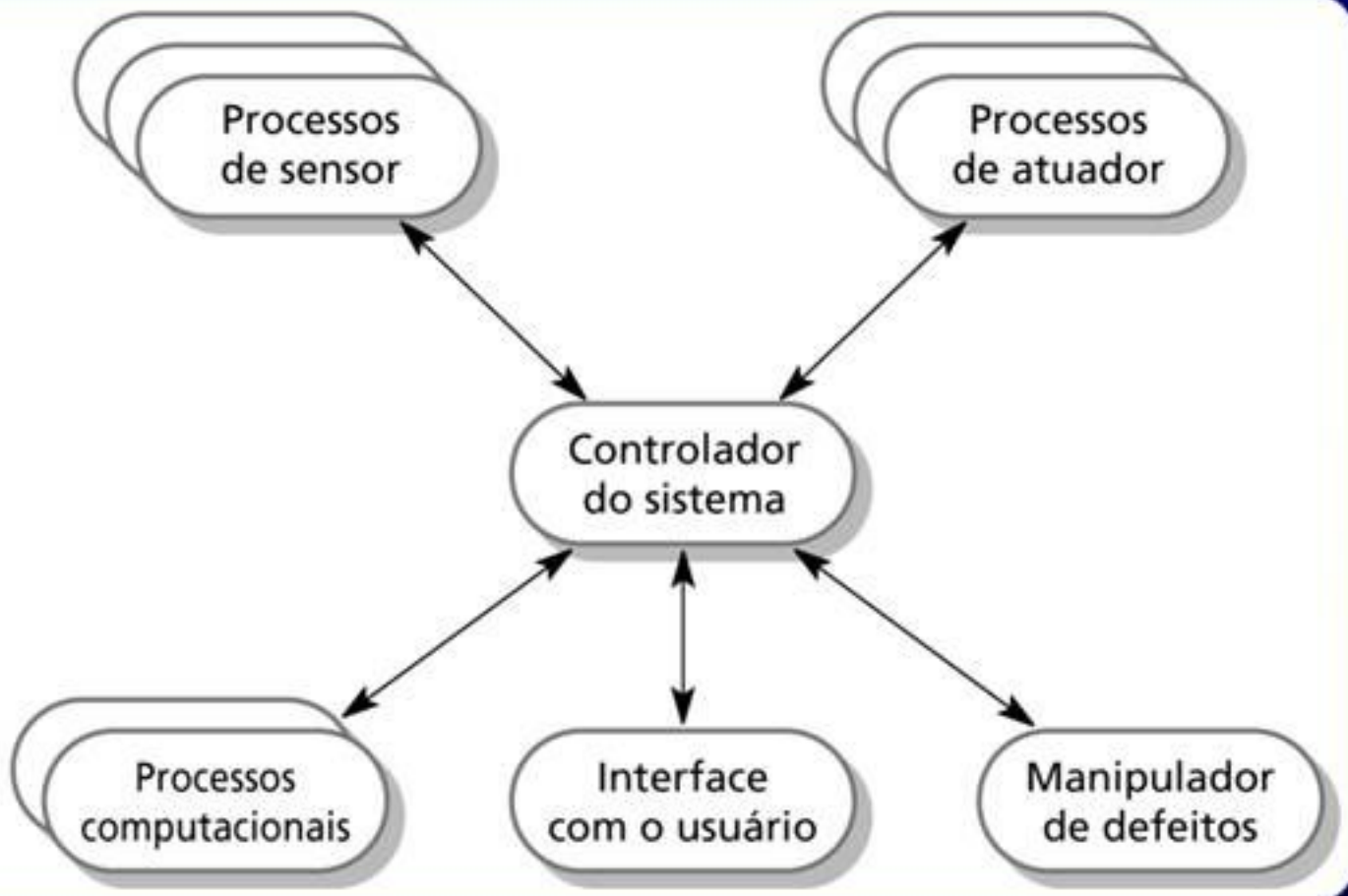
Modelo de controle chamada–retorno.



Controle de sistema tempo real

Figura 11.8

Modelo de controle centralizado para um sistema de tempo real.



Sistemas orientados a eventos

- Dirigidos por eventos gerados externamente onde o *timing* dos eventos está fora do controle dos subsistemas que processam o evento.
- Dois modelos dirigidos a eventos principais
  - Modelos de broadcast. Um evento é transmitido a todos os subsistemas. Qualquer subsistema programado para manipular esse evento pode responder a ele.
  - Modelos orientados a interrupções. Usado em sistemas de tempo real onde as interrupções são detectadas por um tratador de interrupções e passadas por algum outro componente para processamento.
- Outros modelos dirigidos a eventos incluem sistemas de planilhas e de produção.

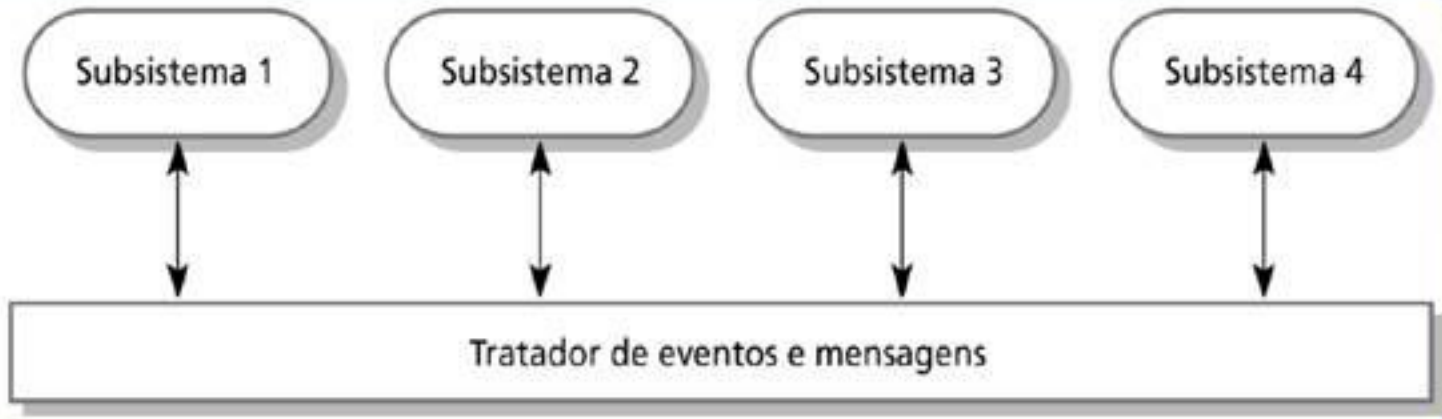
Modelo de *broadcast*

- É efetivo na integração de subsistemas em computadores diferentes em uma rede.
- Subsistemas registram um interesse em eventos específicos. Quando estes ocorrem, o controle é transferido para o subsistema que pode tratar o evento.
- A política de controle não é embutida no tratador de eventos e mensagens. Os subsistemas decidem sobre os eventos de seu interesse.
- Contudo, os subsistemas não sabem se um evento será tratado e nem quando será.

Broadcasting seletivo

Figura 11.9

Modelo de controle baseado em broadcast seletivo.



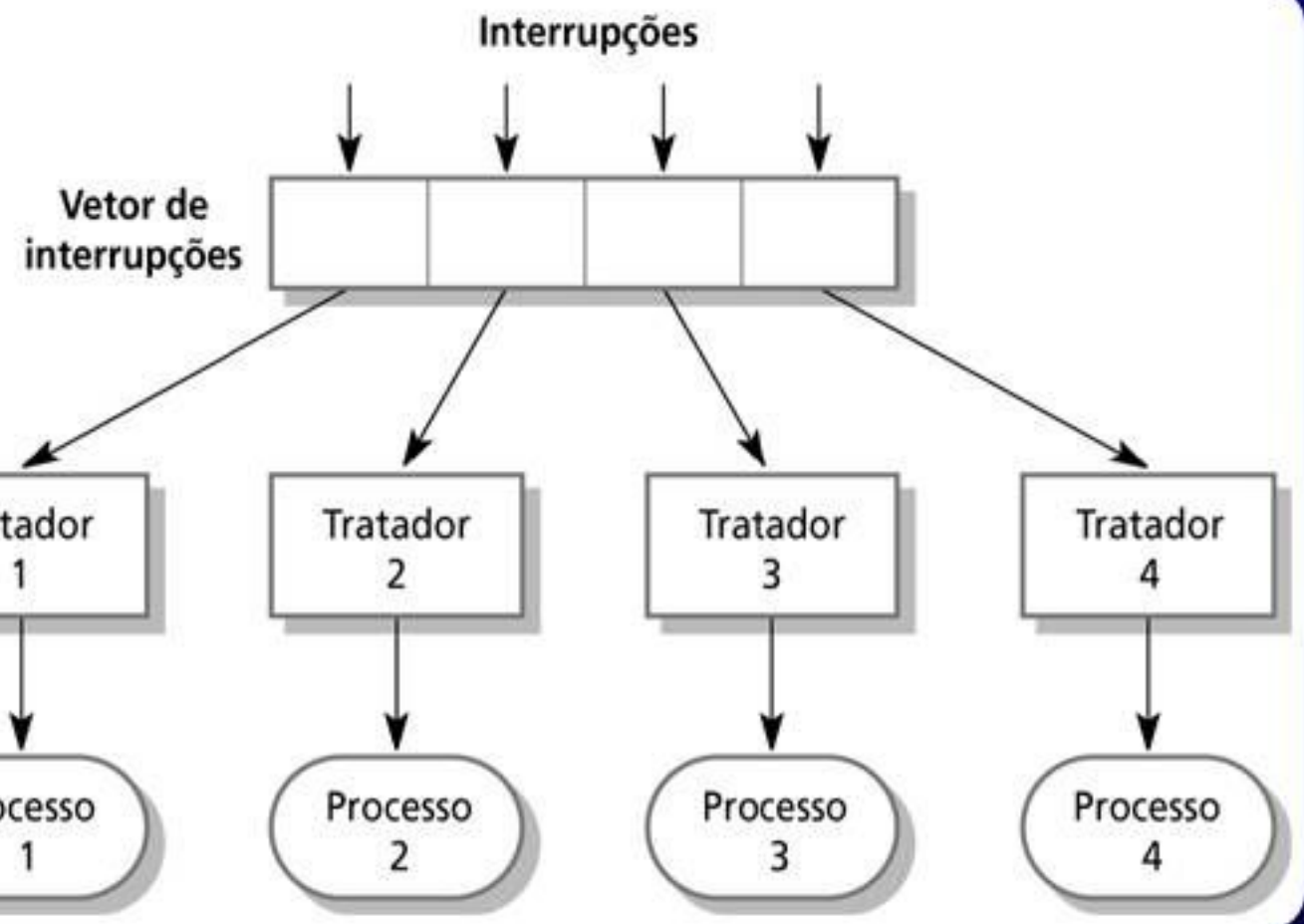
Sistemas dirigidos a interrupções

- É usado em sistemas de tempo real onde a resposta rápida para um evento é essencial.
- Existem tipos de interrupções conhecidos com um tratador definido para cada tipo.
- Cada tipo é associado à uma localização da memória, e uma chave de hardware causa a transferência de controle para seu tratador.
- Permite respostas rápidas, mas é complexo para programar e difícil de validar.

Controle dirigido a interrupções

Figura 11.10

Modelo de controle orientado a interrupções.



Arquiteturas de referência

- Os modelos de arquitetura podem ser específicos para algum domínio de aplicação.
- Existe dois tipos de modelos de domínio específico
  - Modelos genéricos que são abstrações de uma série de sistemas reais que englobam as caracterísiticas principais desses sistemas. Abordados no Capítulo 13.
  - Modelos de referência são mais abstratos, é um modelo idealizado. Fornece um meio de informação sobre essa classe de sistema e sobre comparação de arquiteturas diferentes.
- Os modelos genéricos são geralmente modelos *bottom-up*. Os modelos de referência são modelos *top-down*.

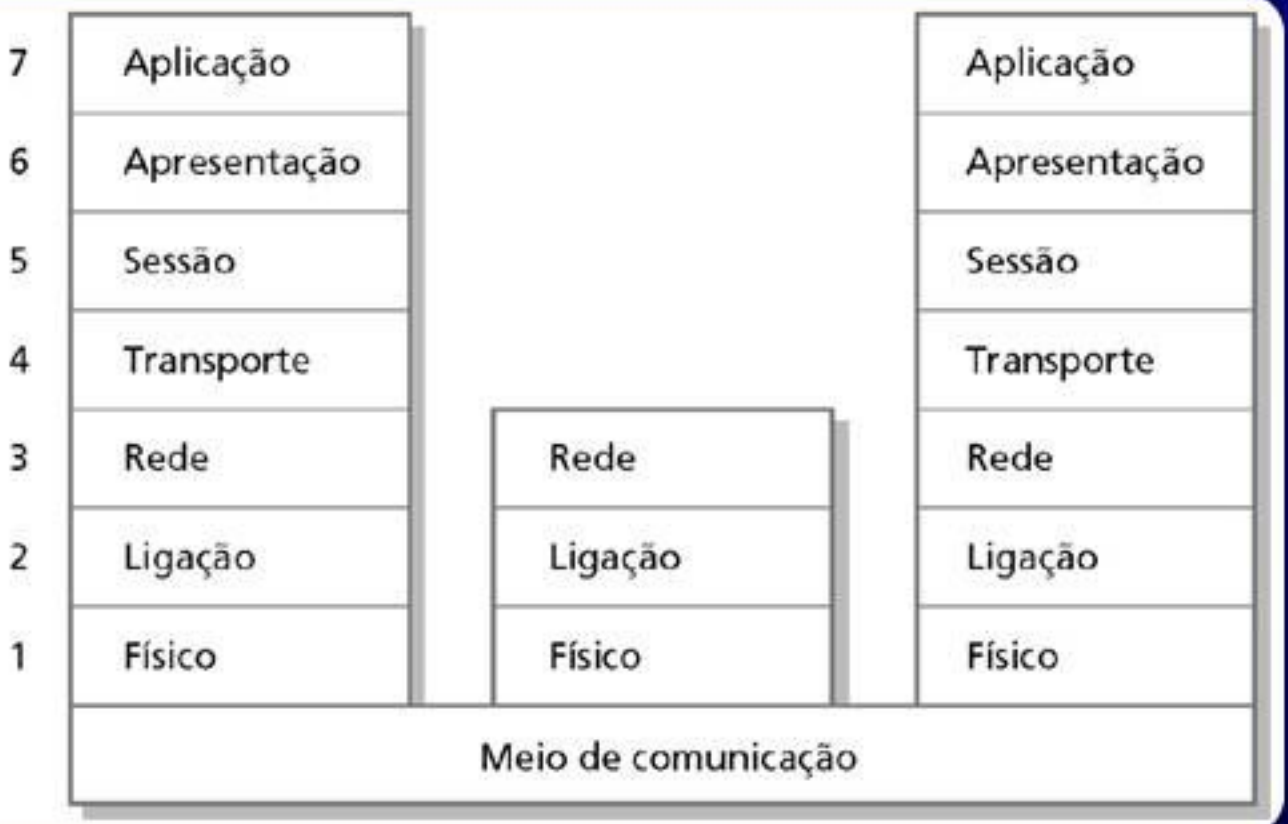
Arquiteturas de referência

- Os modelos de referência são derivados de um estudo de domínio de aplicação ao invés de sistemas existentes.
- Pode ser usado como base para a implementação de sistemas, ou comparar sistemas diferentes. Atua como um padrão contra o qual os sistemas podem ser avaliados.
- O modelo OSI é um modelo de camadas para sistemas de comunicação.

Modelo de referência OSI

Figura 11.11

Arquitetura de modelo de referência OSI.



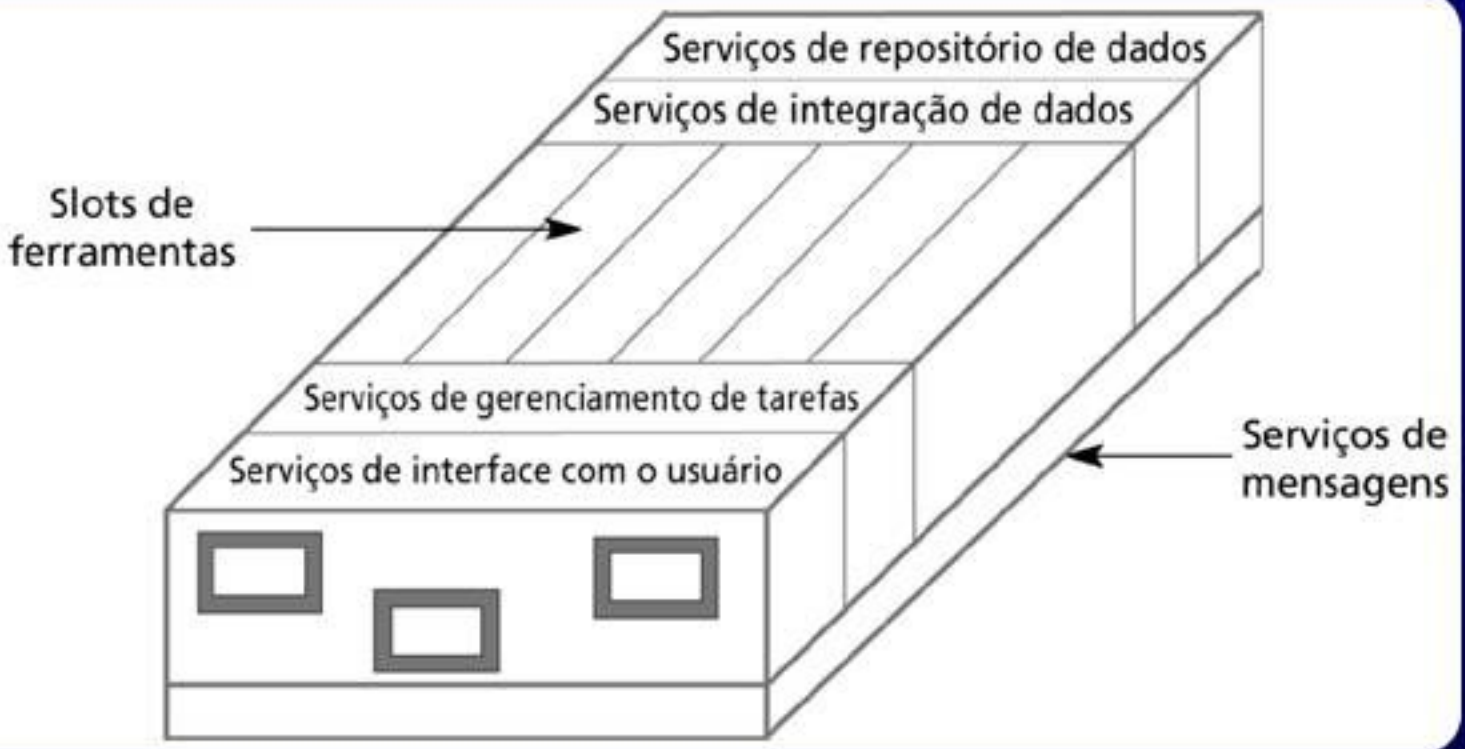
Modelo CASE de referência

- Serviços de repositório de dados
  - Armazenamento e gerenciamento de itens de dados.
- Serviços de integração de dados
  - Gerenciamento de grupos de entidades.
- Serviços de gerenciamento de tarefas
  - Definição e aprovação de modelos de processo.
- Serviços de mensagens
  - Comunicação ferramenta-ferramenta e ferramenta-ambiente.
- Serviços de interface de usuário
  - Desenvolvimento de interface de usuário.

O modelo de referência ECMA

Figura 11.12

Arquitetura de referência ECMA para ambientes CASE.



Pontos-chave

- A arquitetura de software é o *framework* fundamental para a estruturação de sistema.
- Decisões de projeto de arquitetura incluem decisões sobre o tipo de aplicação, a distribuição e os estilos de arquitetura a serem usados.
- Modelos diferentes de arquitetura, tais como um modelo de estrutura, um modelo de controle e um modelo de decomposição podem ser desenvolvidos.
- Modelos organizacionais de um sistema incluem modelos de repositório, modelos cliente-servidor e modelos de máquina abstrata.

Pontos-chave

- Modelos de decomposição modular incluem modelos de objetos e modelos de pipelining.
- Modelos de controle incluem modelos de controle centralizado e dirigidos a eventos.
- Arquiteturas de referência podem ser usadas para comunicar arquiteturas de domínio específico, avaliar e comparar projetos de arquitetura.

Modelos de arquitetura

- Modelos diferentes de arquitetura podem ser produzidos durante o processo de projeto.
- Cada modelo apresenta perspectivas diferentes sobre a arquitetura.

Atributos de arquitetura

- Desempenho
  - Localizar operações críticas e minimizar comunicações.
- Proteção
  - Usar uma arquitetura em camadas com itens críticos nas camadas mais internas.
- Segurança
  - Isolar componentes críticos de segurança
- Disponibilidade
  - Incluir componentes redundantes na arquitetura.
- Facilidade de manutenção
  - Usar componentes substituíveis e de baixa granulariade.