



Sistemas Distribuidos

Prof. Walter Gima



Arquitecturas Sistemas Distribuídos



Agenda

- Arquiteturas
 - Estilos Arquitetônicos
 - Camadas
 - Objetos
 - Eventos
 - Dados

Conceito de Arquitetura de Software

- é a estrutura de um sistema, que consiste de é a estrutura de um sistema, que consiste de componentes de software, das propriedades externamente visíveis desses componentes e dos relacionamentos entre eles.
- é a estrutura ou organização dos mais é a estrutura ou organização dos mais significativos componentes do sistema e suas interações.

Conceito de Arquitetura de Software

Componentes:

- é uma unidade modular com interfaces requeridas e fornecidas bem definidas que é substituível dentro do seu ambiente.
- são elementos de uma arquitetura que são elementos de uma arquitetura que geralmente implementam: processamento (funcionalidade ou comportamento); estado (informação ou dados); interação (interconexão, comunicação, coordenação e mediação).

Conceito de Arquitetura de Software

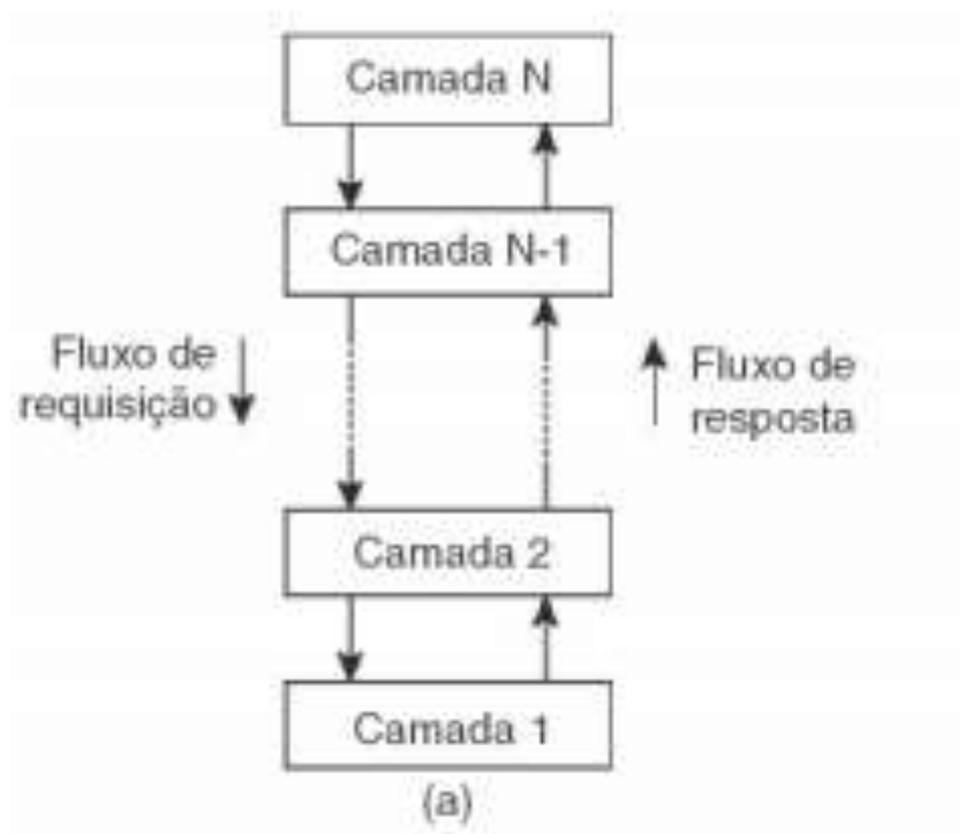
Conectores

- tipo de componente responsável pela interação tipo de componente responsável pela interação entre componentes.
- em sistemas desktop convencionais os conectores são geralmente representados por simples chamadas de procedimento (procedure call) ou acesso a dados compartilhados.
- em sistemas complexos eles passam a ter em sistemas complexos eles passam a ter identidades, papéis e artefatos de implementação único.

Estilos arquiteturais:

- Camadas
- Objetos
- Eventos
- Dados

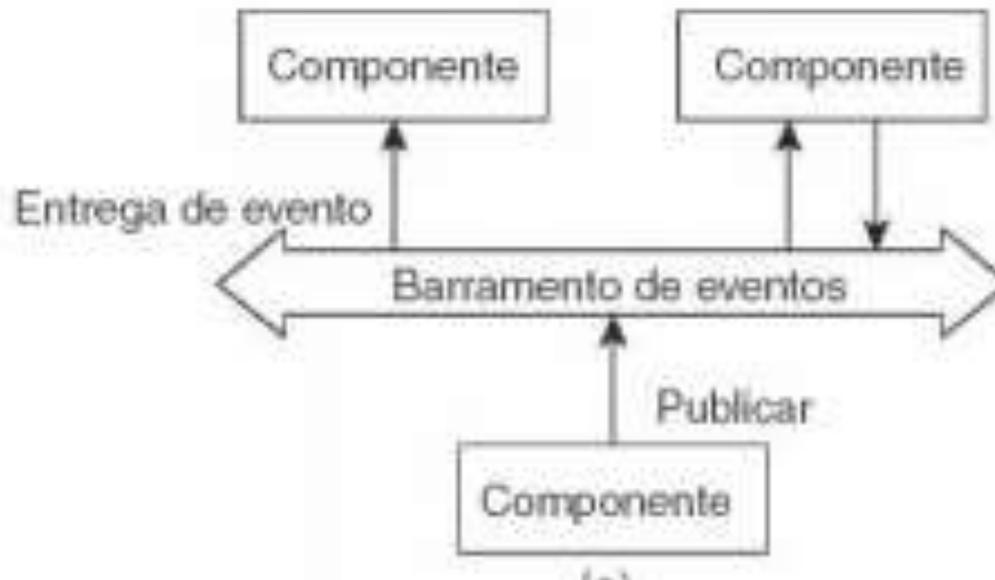
Estilo Camadas:



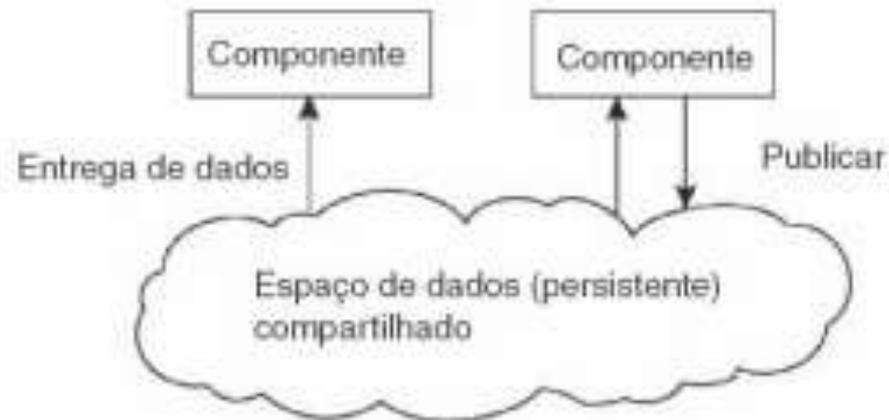
Estilo Objetos:



Estilo Eventos:



Estilo Dados compartilhado:



Arquiteturas de Sistemas

Arquiteturas de Sistemas

Camadas

Ex:

- Camada de interface do usuário
- Camada de processamento
- Camada de dado

Arquiteturas de Sistemas

Camadas

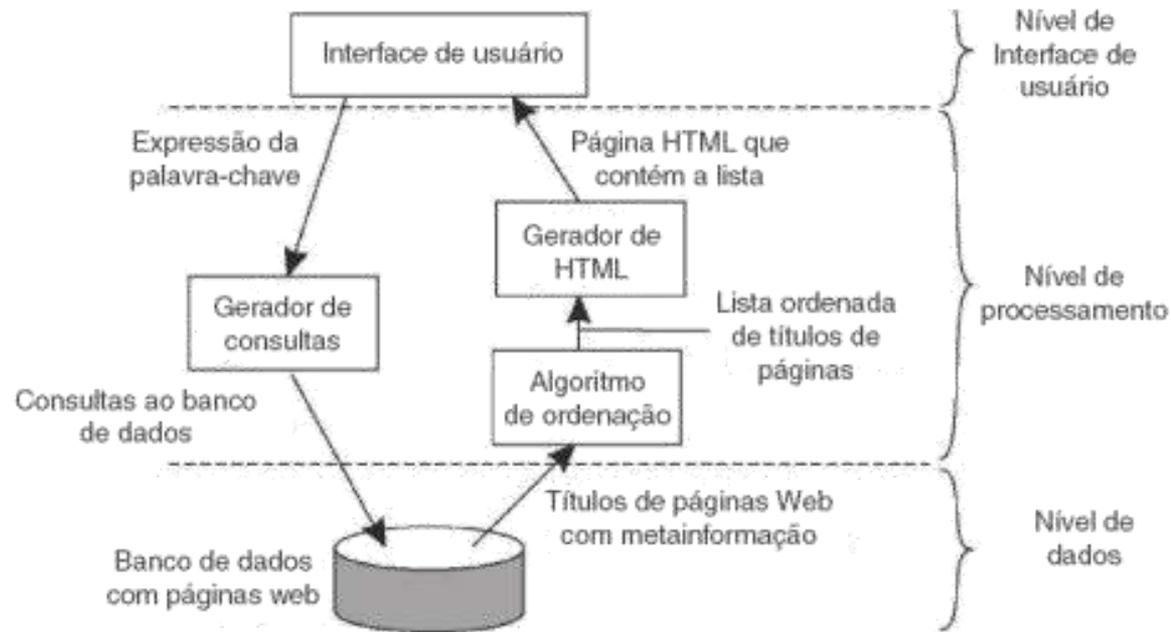


Figura 2.4 Organização simplificada de um mecanismo de busca da Internet em três camadas diferentes.

Arquiteturas de Sistemas

Camadas MVC

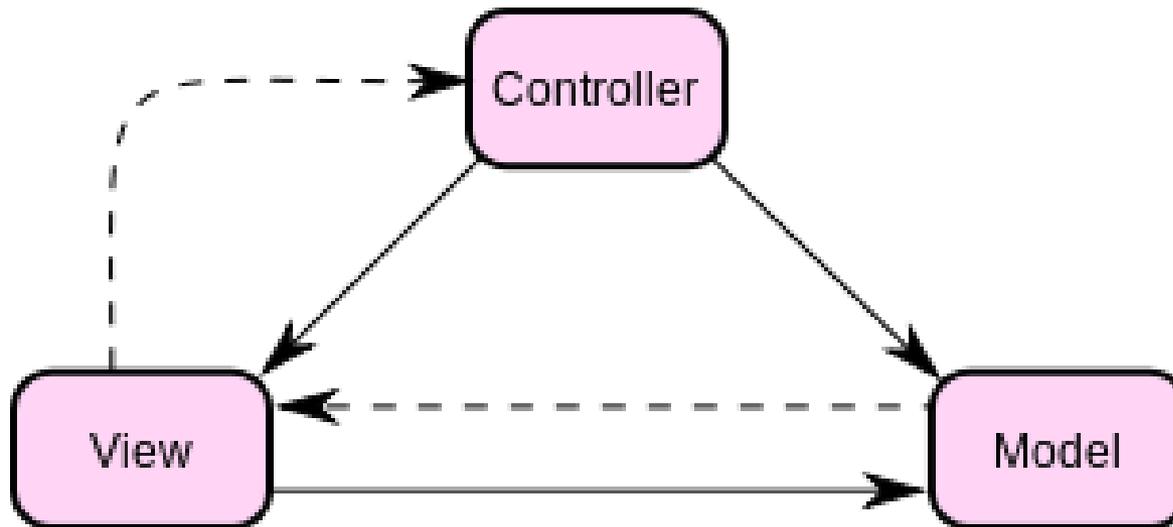
Um **controlador (controller)** envia comandos para o modelo para atualizar o seu estado (por exemplo, editando um documento). O controlador também pode enviar comandos para a visão associada para alterar a apresentação da visão do modelo (por exemplo, percorrendo um documento).

Um **modelo (model)** armazena dados e notifica suas visões e controladores associados quando há uma mudança em seu estado. Estas notificações permitem que as visões produzam saídas atualizadas e que os controladores alterem o conjunto de comandos disponíveis. Uma implementação *passiva* do MVC monta estas notificações, devido a aplicação não necessitar delas ou a plataforma de software não suportá-las.

A **visão (view)** Gera uma representação (Visão) dos dados presentes no modelo solicita do modelo.

Arquiteturas de Sistemas

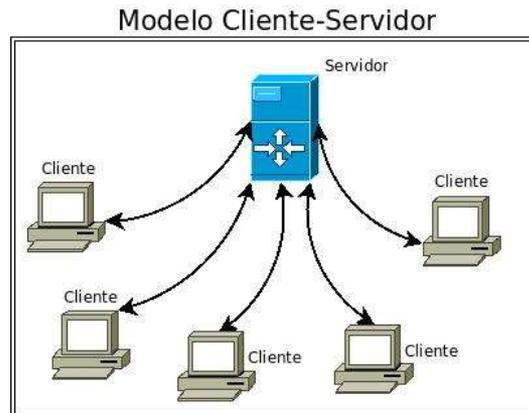
Camadas MVC



Arquiteturas de Sistemas

Cliente - Servidor

A característica do modelo cliente-servidor, descreve a relação de programas numa aplicação. O componente de servidor fornece uma função ou serviço a um ou mais clientes, que iniciam os pedidos de serviço.



- Tanenbaum, Andrew S; Steen, Maarten Van. Sistemas Distribuídos: Princípios e Paradigmas, 2ª ed. São Paulo: Pearson – Prentice Hall, 2007.
- Coulouris, George. Sistemas Distribuídos: Conceitos e Projetos, 4ª ed. Rio de Janeiro: Bookman, 2007.
- fonte: <https://pt.wikipedia.org/wiki/MVC>



Anhanguera

Dúvidas ?

walter.gima@anhanguera.com