BCC
Engenharia de Software
2018.2

Professor Rodrigo Andrade

Aula Passada

Quais são alguns exemplos de Software como Serviço?

Duas vantagens de Cloud Computing?

Qual a diferença entre Verificação e Validação?

Quais os quatro mecanismos fundamentais da Produtividade?

Arquitetura de software

- Descreve como subsistemas estão interconectados
- Pode se referir à documentação que descreve os componentes de um software
- Ajuda desenvolvedores e clientes a entenderem um sistema
- Análogo a arquitetura na engenharia civil

Padrão arquitetural

- Descreve a arquitetura geral de uma solução para uma família de problemas similares
- Tenta facilitar a aplicação de uma solução para problemas conhecidos

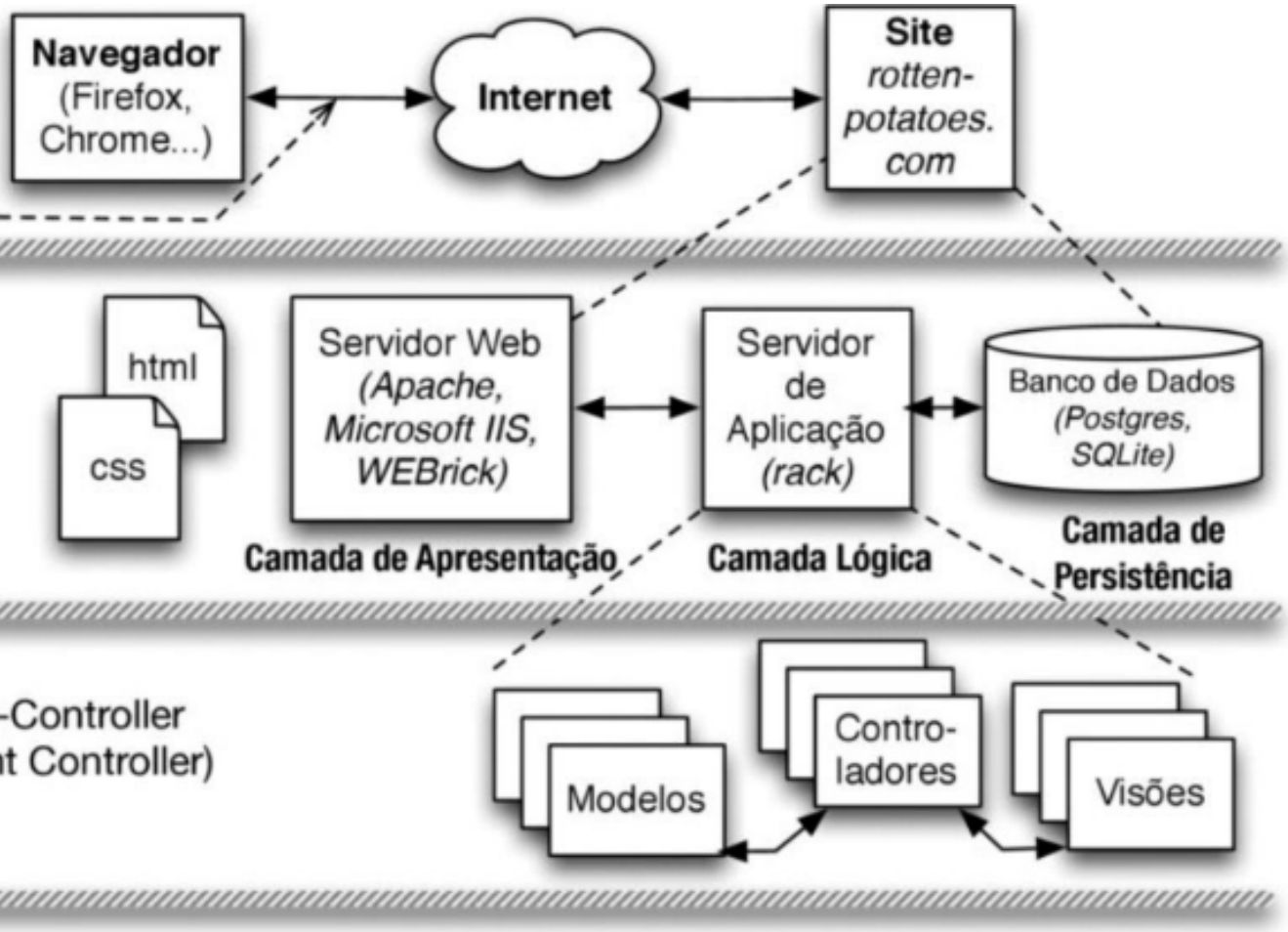
§2.1 100.000 pés
• Cliente-servidor (vs. P2P)

§2.2 50.000 pés
• HTTP e URIs

§2.3 10.000 pés
• XHTML e CSS

§2.4 5.000 pés
• Arquitetura de 3 camadas
• Escalabilidade horizontal

§2.5 1.000 pés—Model-View-Controller
(vs. Page Controller, Front Controller)



§2.6 500 pés: Modelos Active Record (vs. Data Mapper)
§2.7 500 pés: Controladores REST (*Representational State Transfer* para ações auto-contidas)
§2.8 500 pés: Template View (vs. Transform View)

• **Active Record** • **REST** • **Template View**
• Data Mapper • Transform View

Padrões arquiteturais

Cliente-servidor

Arquitetura em 3 camadas

Model-View-Controller

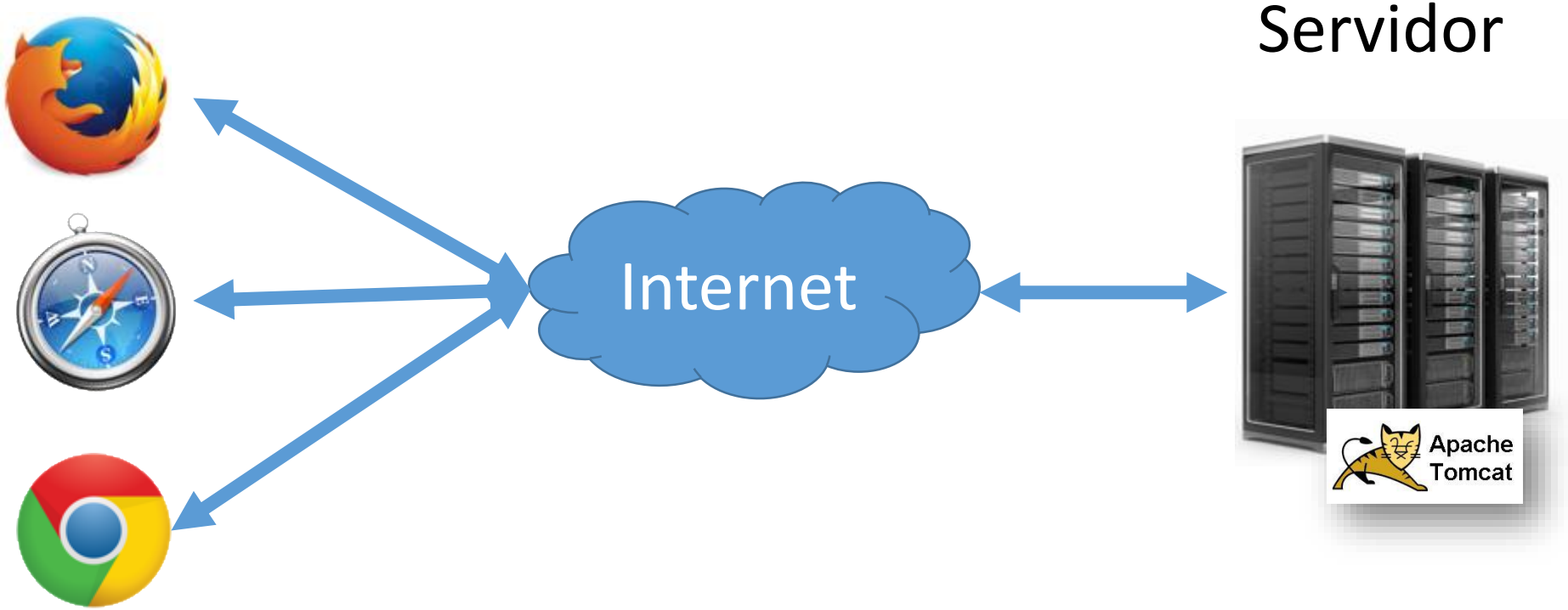
Cliente - Servidor

Definição de Cliente-Servidor

- Padrão de projeto
  - Estrutura reutilizável
  - Comportamento ou estratégia que captura uma solução comprovada para problemas conhecidos
- Separação lógica entre Cliente e servidor permanece inalterada

Clientes

Servidor



Clientes

Servidor

Internet

Como as requisições são organizadas?

Protocolo HTTP

GET https://google.com:80/search?q=software&lang=pt

Método HTTP

Protocolo

Hostname

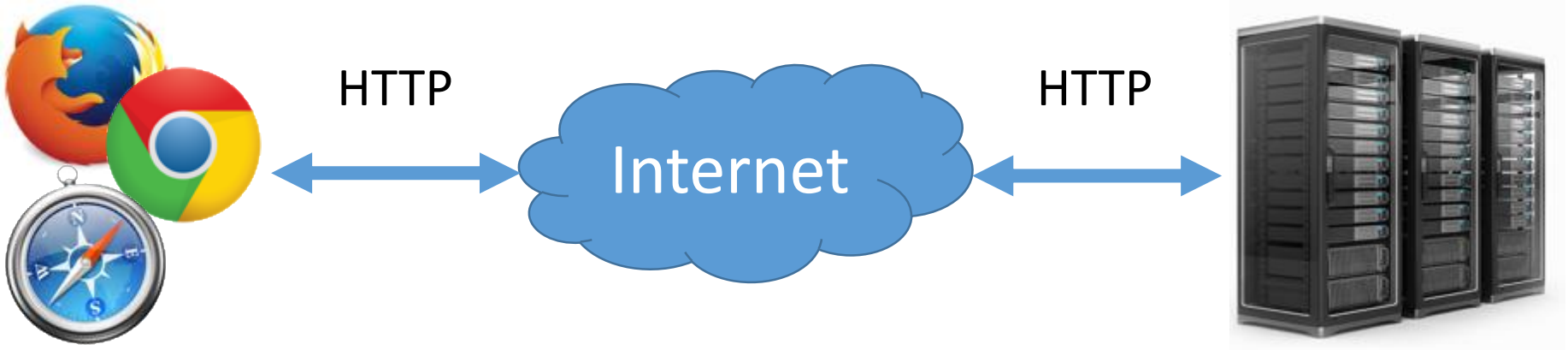
Porta

Caminho do recurso

Argumentos

Protocolos de comunicação

GET http://uag.com:80/uag/aluno/search?nome=Fulano
POST http://localhost:8080/uag/aluno/show/1



HTTP depende do TCP/IP

Qual aplicação popular funciona como cliente e como servidor ao mesmo tempo?

Redes peer-to-peer

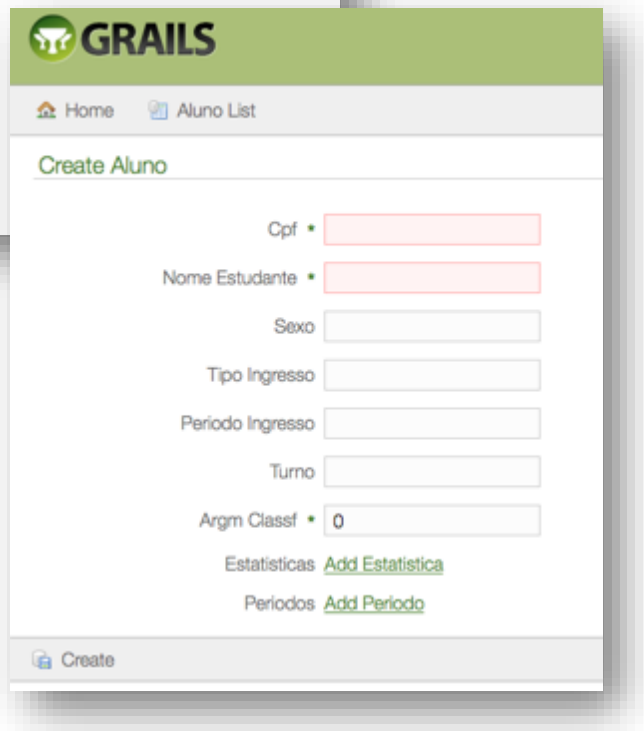
Exibição do conteúdo para os clientes

HTML

- Linguagem de marcação universal
- Misturam rótulos (tags) e conteúdo útil aos usuários

HTML para requisição a um serviço

```
<div id="create-aluno" class="content scaffold-create" role="main">
  <h1>Create Aluno</h1>
  <form method="post" action="/uaq/aluno/save">
    <fieldset class="form"></fieldset>
    <fieldset class="buttons"></fieldset>
  </form>
</div>
```



Representa a requisição da ação de criar um novo aluno

HTML para exibir resultado da requisição

```
<div id="show-aluno" class="content scaffold-show" role="main">
  <h1>Show Aluno</h1>
  <ol class="property-list aluno">
    <li class="fieldcontain">
      <span id="cpf-label" class="property-label">Cpf</span>
      <span class="property-value" aria-labelledby="cpf-label">058.012.192-56</span>
    </li>
```

Arquitetura em 3 camadas

Camadas

Camada de Apresentação

Camada lógica

Camada de persistência

Camada de apresentação

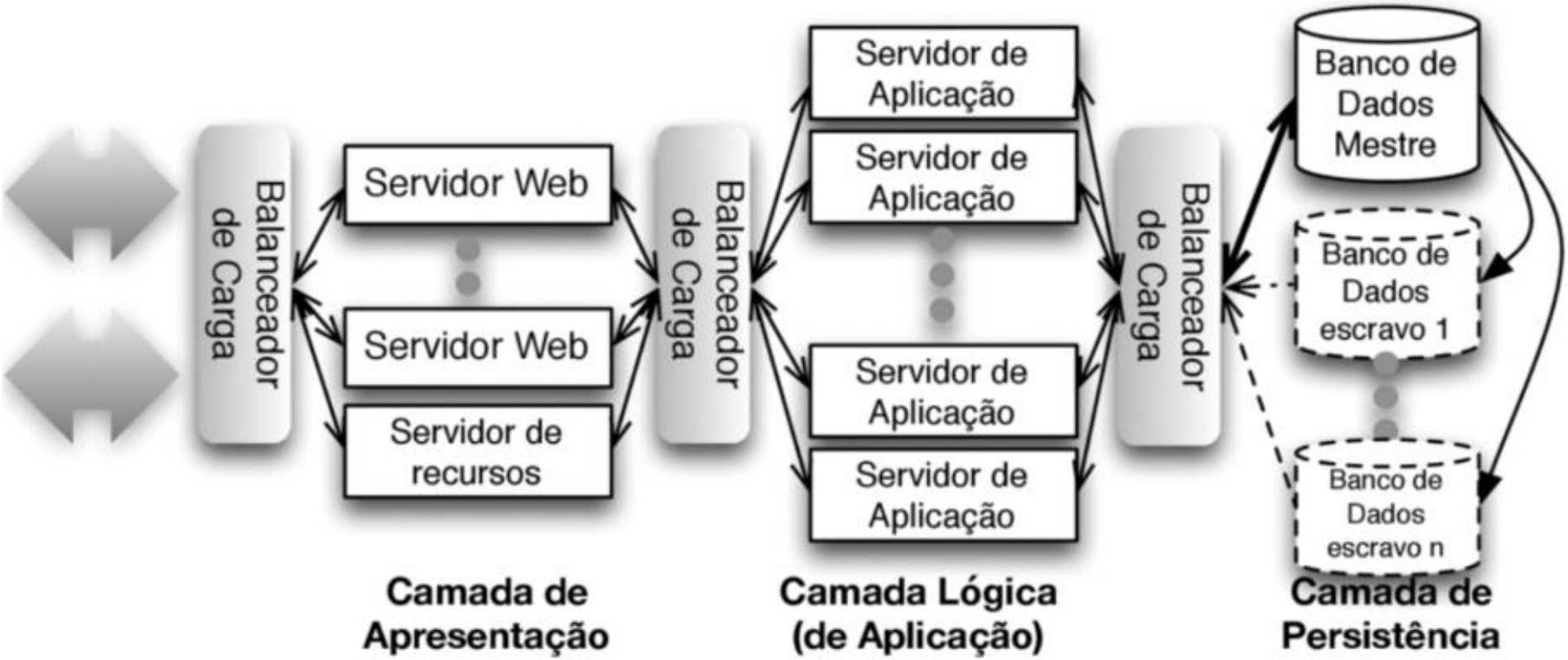
- Consiste em um servidor web (HTTP)
  - Aceita requisições de clientes
  - Responde itens estáticos
- Rails utiliza o servidor Puma como padrão

Camada lógica

- Conta com um servidor de aplicação
  - Esconde operações HTTP de baixo nível
  - Recebe requisições HTTP e encaminha para o trecho de código mais adequado
- Rails utiliza o servidor Puma como padrão

Camada de persistência

- Armazena dados da aplicação durante requisições HTTP
  - Exemplo: Dados da sessão, login e informações de perfil
- Rails utiliza o SQLite3 como padrão


Balanceador de Carga
Servidor Web
Servidor Web
Servidor de recursos
Balanceador de Carga
Servidor de Aplicação
Servidor de Aplicação
Servidor de Aplicação
Servidor de Aplicação
Balanceador de Carga
Banco de Dados Mestre
Banco de Dados escravo 1
Banco de Dados escravo n
Camada de Apresentação
Camada Lógica (de Aplicação)
Camada de Persistência

HTTP

Internet

HTTP

Servidor Web

Servidor de aplicação

Banco de dados

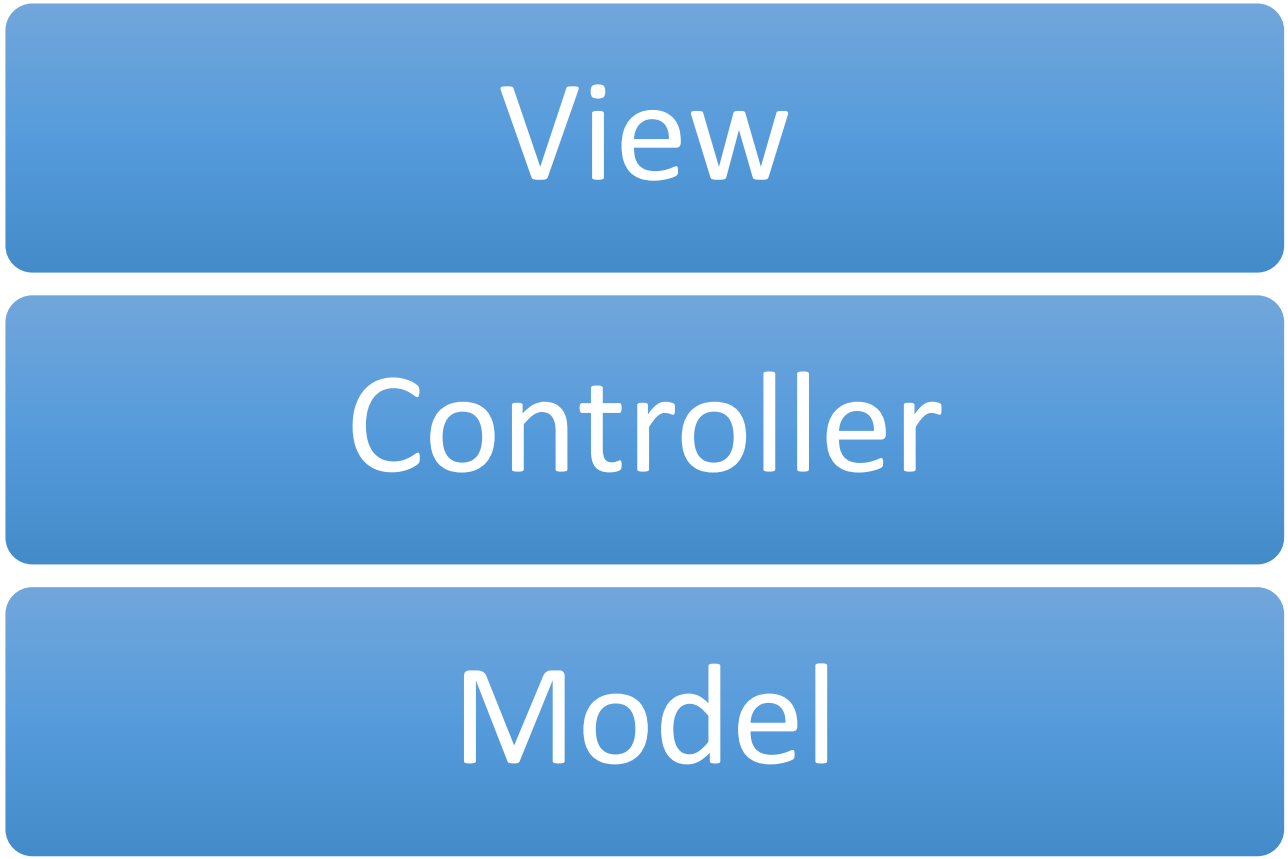
Apresentação

Lógica

Persistência

Arquitetura Model-View-Controller (MVC)

Tipos diferentes de código



Model

- Faz parte da Camada Lógica
- Manipulam diretamente dados da aplicação
  - Armazenamento
  - Alteração
  - Remoção
- Exemplo:
  - Classes que representam entidades da aplicação e usam Hibernate ou JDBC

Controller

- Implementa a lógica do negócio
- Recebe requisições dos usuários (views)
- Pede ao Model para recuperar ou modificar informações
- Decide qual visão será exibida como resultado
- Um Controller associado a cada Model

View

- Interfaces gráficas para os usuários
- Apresenta informações sobre os modelos com os quais os usuários podem interagir
- Interface entre usuários de um sistema e seus dados
- Exemplo
  - Documentos html

Resumo da aula

- Diferentes padrões, em níveis diferentes:
  - Cliente-servidor
  - MVC
  - Arquitetura três camadas

Referência

Capítulo 2