

SCC0265 – Sistemas Interativos Web



Desenvolvimento de Aplicações Hiperfídia na Web

Renata Pontin M. Fortes

(renata@icmc.usp.br)

PAE: Willian Watanabe (watinha@gmail.com)

Instituto de Ciências Matemáticas e de Computação - ICMC



Sumário



- A World Wide Web
- Engenharia de Web
- Papel da Modelagem, Processo e Arquitetura de Aplicações

A World Wide Web - Site Web x Aplicação Web

◆ Arquitetura de **site Web**

◆ **Servidor Web**, **conexão de rede** e **browsers** (clientes)

◆ **Aplicações Web**

◆ Desenvolvidas a partir de um sistema Web para adicionar regras de negócio

◆ “...uma aplicação Web é um sistema Web que permite ao usuário executar lógica de negócio usando o *browser*...”

◆ As entradas de dados e a navegação do usuário afetam o **conteúdo do site**

Arquitetura: além da arquitetura de site Web, há também um **servidor de aplicações**

Engenharia de Web

(Web Engineering)



■ DINAMISMO

- Tecnologias de ativação (*executam no servidor*)
 - CGI e Java Servlet
 - ASP, PHP, JSP, etc

- Clientes dinâmicos (*executam no cliente*)
 - JavaScript, Java Applets, ActiveX, Flash

Engenharia de Web

(Web Engineering)

- ◆ Técnicas para desenvolvimento Web
 - ◆ Engenharia de Web – *Web Engineering*
- ◆ **Processo** usado para criar aplicações Web de alta qualidade
 - ◆ Utiliza conceitos e princípios da ES tradicional
 - ◆ Ênfase em atividades técnicas e gerenciais (navegação, interface...)

Por que Web Engineering é importante?

Necessidade de construir sistemas

Confiáveis, Usáveis e Adaptáveis.

Engenharia de Web

(Web Engineering)



- Quais são os passos?
 - (1) Formulação do problema;
 - (2) Planejamento e análise de requisitos;
 - (3) Projeto arquitetural, navegacional e da interface;
 - (4) Implementação;
 - (5) Testes.
- Mecanismos para **controle de configuração** e **garantia de qualidade** são MUITO necessários.

Engenharia de Web (*Web Engineering*)

Como saber se as etapas foram cumpridas corretamente?

- Aplicar práticas de SQA convencionais: revisão técnica formal, avaliação de usabilidade, funcionalidade, etc.
- Para evitar *websites* “emaranhados” e obter sucesso no desenvolvimento de aplicações web complexas



uso de abordagens disciplinadas de *Web Engineering* e novos métodos e ferramentas de desenvolvimento, disponibilização e avaliação de aplicações

Engenharia de Web

(*Web Engineering*)



produto

- Propriedades que diferenciam **aplicações web** das aplicações de software tradicionais:
 - **Uso intensivo da rede**
 - aplicações estão (residem) na rede (internet, intranet) e devem servir a diversas comunidades de clientes;
 - **Direcionadas a conteúdo**
 - uso da hipermídia para apresentar texto, gráfico, áudio e vídeo aos usuários;
 - **Evolução contínua**
 - aplicações web evoluem muito rapidamente.

Engenharia de Web

(Web Engineering)



- Características que direcionam o **processo** de desenvolvimento:
 - **Urgência**
 - Prazo curto para desenvolvimento (poucos dias ou semanas).
 - Uso de métodos adaptados ao desenvolvimento rápido;
 - **Segurança**
 - Medidas severas de segurança têm que ser implementadas;
 - **Cuidados estéticos**
 - Diretamente relacionados ao sucesso da aplicação web.

Engenharia de Web

(*Web Engineering*)

- Desenvolvimento baseado em componentes:
 - CORBA, COM/DCOM e **JavaBeans**
 - componentes que comunicam-se uns com os outros e com outros serviços
- Segurança:
 - medidas de segurança como *firewalls* e criptografia
- Padrões Internet:
 - uso intensivo de HTML na última década;
 - crescimento (tamanho e complexidade) das aplicações web
→ novo padrão : **XML**
 - desenvolvedores definem suas próprias tags.

Engenharia de Web

(*Web Engineering*)



- Manutenção de Aplicações Web
 - Similar à ES, onde **80%** dos custos são voltados à manutenção.
 - *Web Engineering* é o processo de “**projetar para mudar**”.
- Flexibilidade de Aplicações Web → fácil de estender e manter.
 - Capacidade de integrar novos requisitos funcionais sem grandes modificações no sistema
 - Layout gráfico e a aparência

Modelagem, Processo e Arquitetura

■ Entendimento

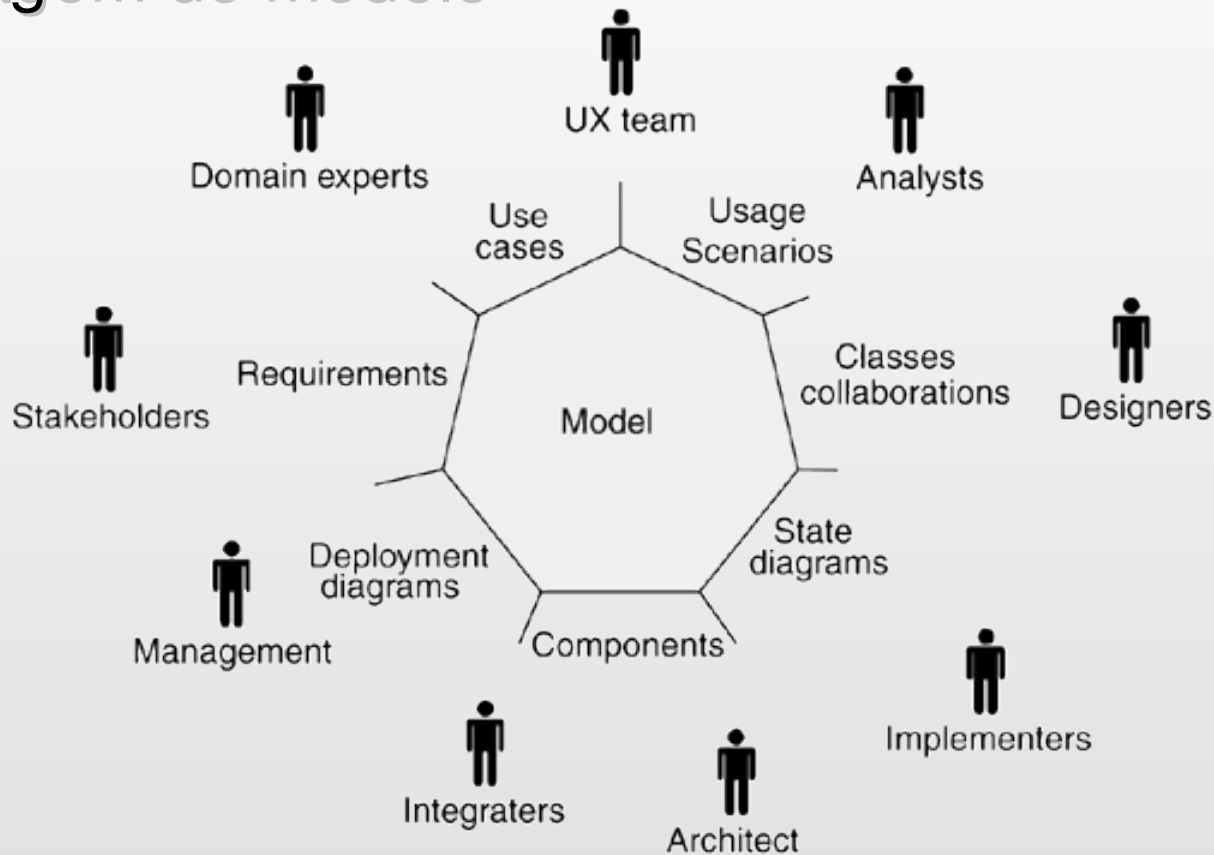
- O que **se deseja** construir, o que **está sendo** construído e o que **foi** construído
- Descrição da realidade (níveis de abstração)
 - Mais abstratos: modelos mais longe da realidade e mais simples
 - Menos abstratos: modelos mais próximos da realidade e mais complexos
- Modelos devem expor o que é importante para o entendimento do sistema

■ Comunicação

- Divisão do problema em partes menores
- Facilidade para explicar o sistema para outras pessoas (usuários, arquitetos, desenvolvedores, etc)

Modelagem, Processo e Arquitetura

Vantagem do modelo



Modelagem, Processo e Arquitetura

- Ferramentas CASE (*Computer Aided Software Engineering*)
 - Geração de código a partir de modelos (*model driven*) e geração de modelos a partir de código-fonte (engenharia reversa)
 - Auxiliam no processo de desenvolvimento

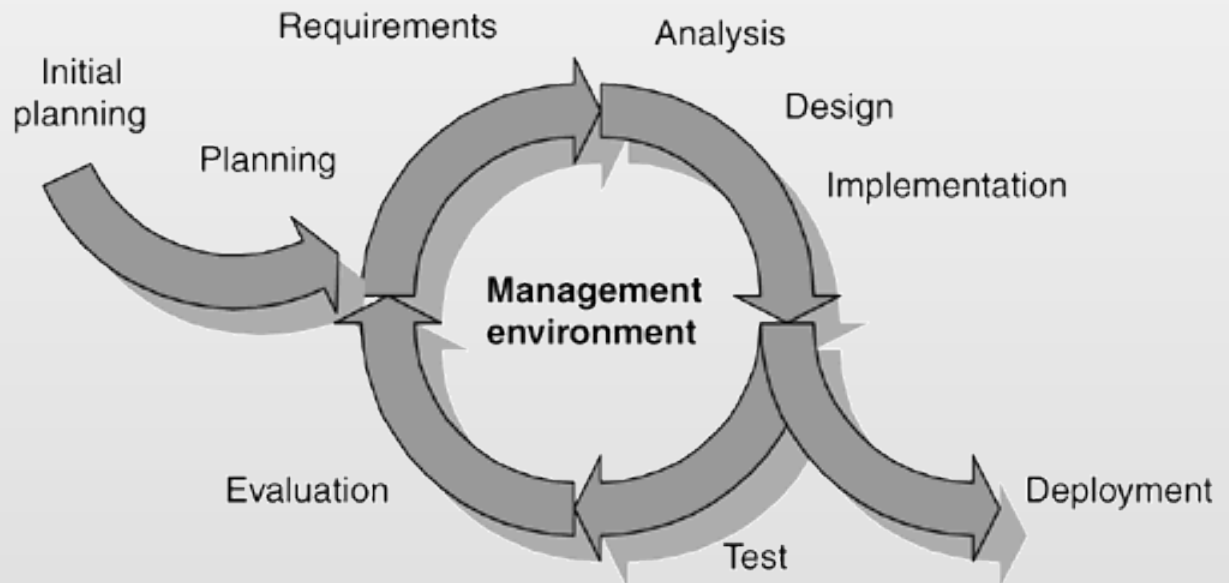
OBS.: o papel da modelagem não é produzir código através da automação ou produzir documentos por meio da engenharia reversa (produtos secundários das ferramentas CASE)

- **Valor real da modelagem:** capacidade de ver uma simplificação do sistema através de um ponto de vista específico por onde o sistema se torna mais fácil de entender.
- Modelos muito complexos → modelagem perde o sentido

Modelagem, Processo e Arquitetura

processo de desenvolvimento

- Organizar o desenvolvimento do software:
 - Acelerar o desenvolvimento e melhorar a qualidade
 - Produzir artefatos: mais importante: modelos
- Depende da empresa, aplicação, equipe, prioridades do projeto, etc...



Modelagem, Processo e Arquitetura

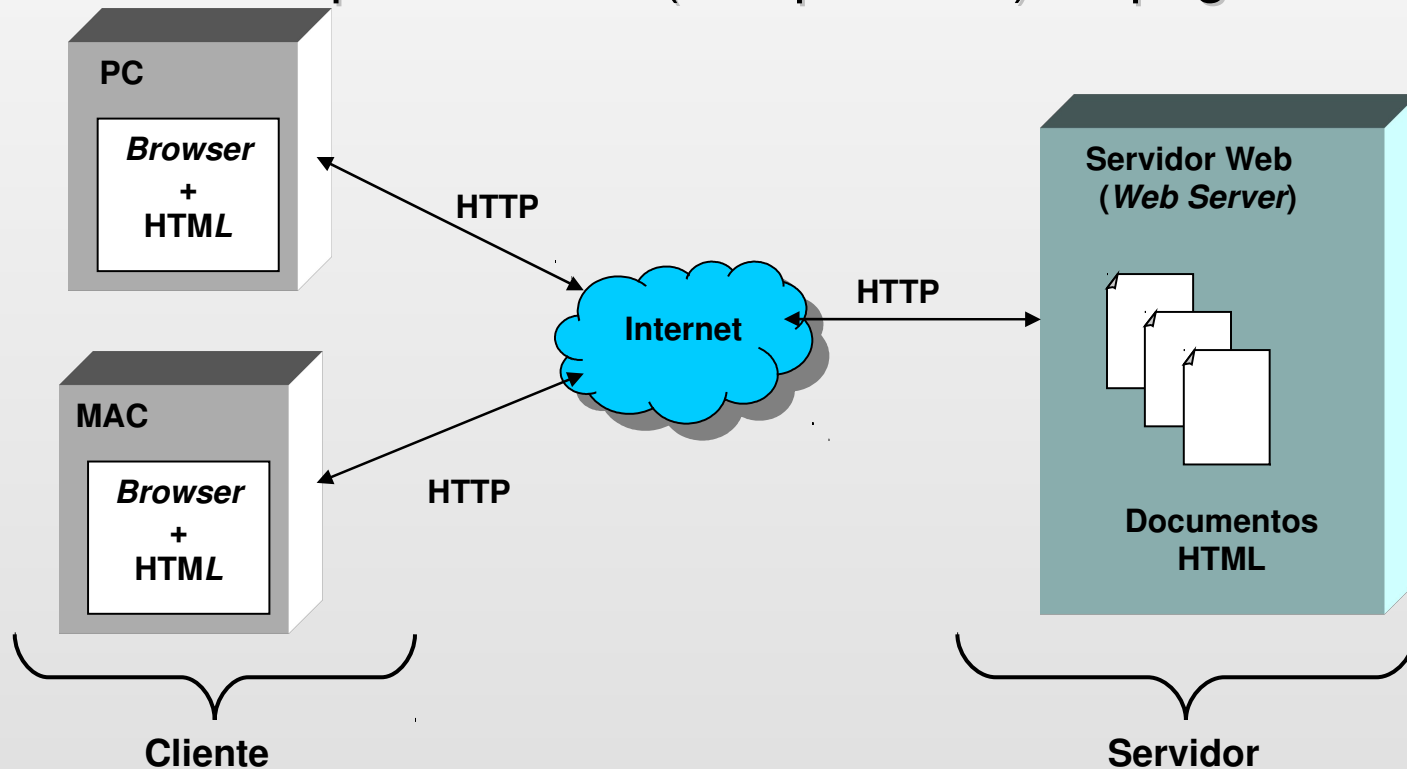
- O papel da Arquitetura
 - Influência no processo de desenvolvimento e no produto final
 - Define regras para a construção do software
 - como o software deve ser “pensado” ?
 - Arquitetura da Web - **cliente/servidor** diferenciado
 - Servidor não controla o cliente / Interação iniciada pelo cliente
 - Baseado no paradigma estímulo / resposta (transação)
 - Se esse comportamento não é o esperado, pode-se acrescentar outros recursos à arquitetura (mais complexidade)

Mais uso, mais experiência, evolução

- Padrões arquiteturais: *Façades, Page Composition, Template Pages, etc*
- Arquiteturas relativamente complexas

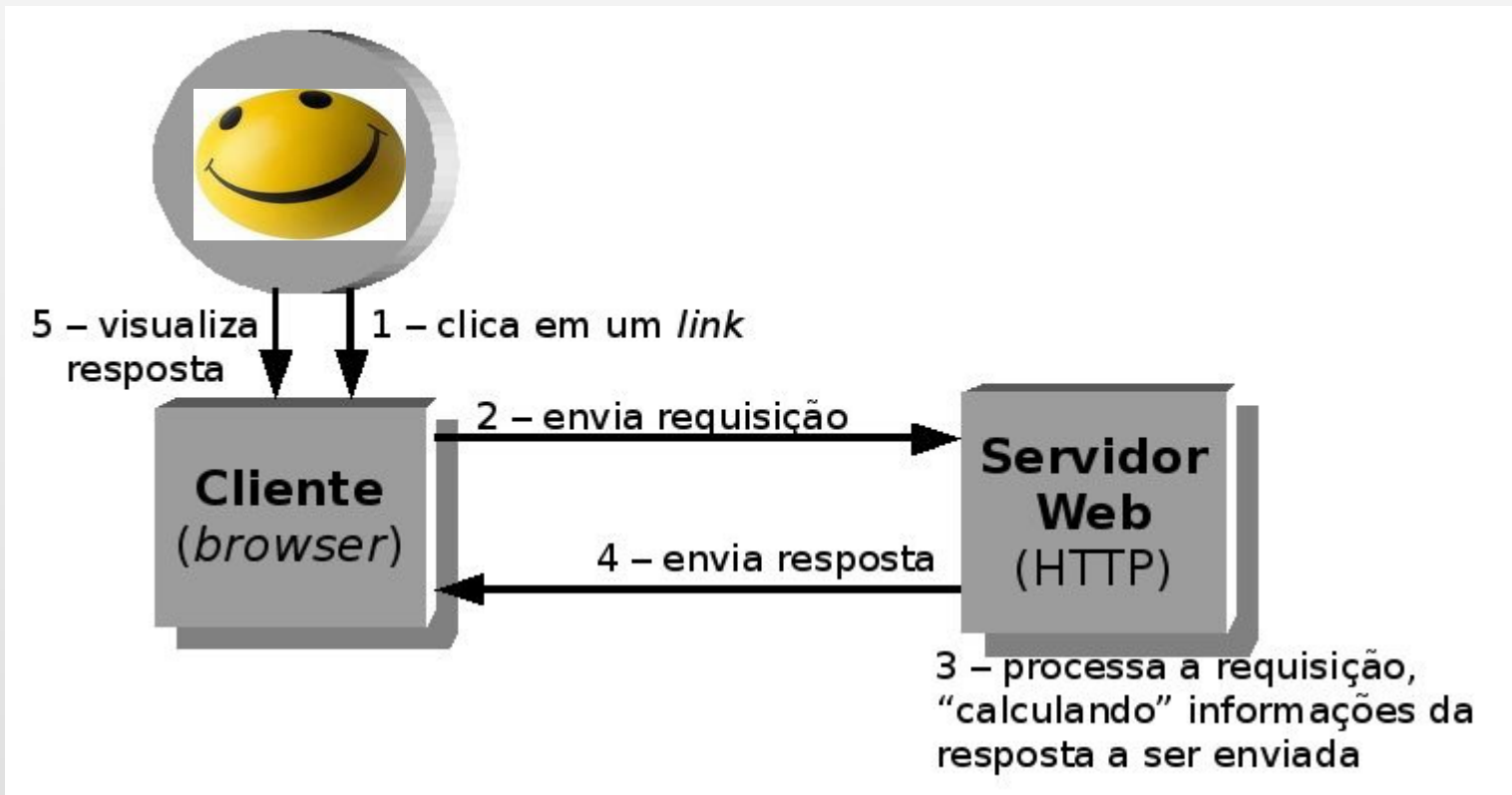
Arquiteturas da Web

- A Era do Hipertexto
 - Transferência de documentos HTML multimídia (estáticos)
 - *Browser* : apresentador (interpretador) de páginas HTML



Arquiteturas da Web

- A Era do Hipertexto
 - Comunicação via Protocolo HTTP



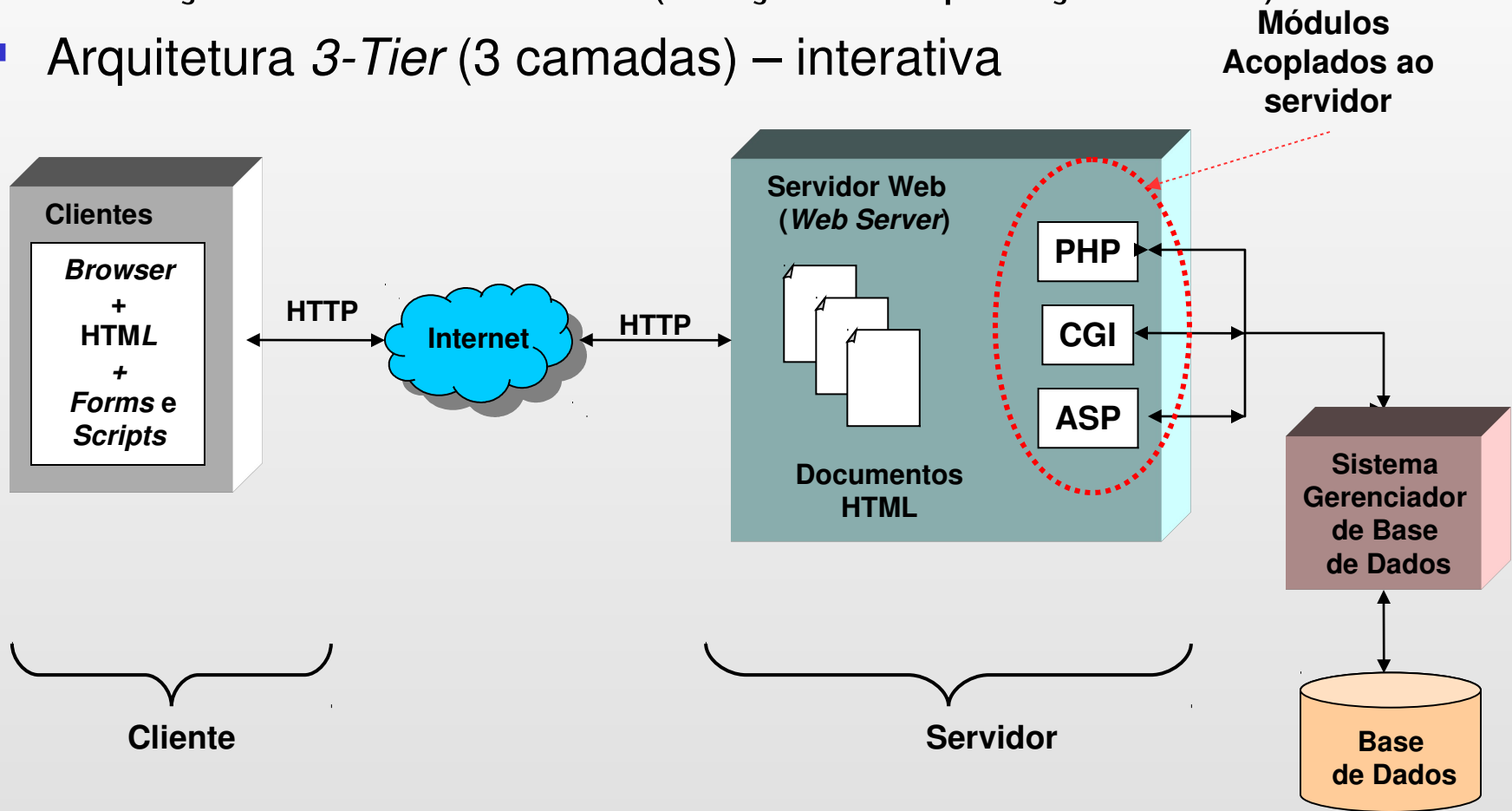
Arquiteturas da Web



- A Era Interativa
 - Conceito: Aplicação Web
 - No **cliente**: funcionalidades de interação no navegador
 - **Formulários** - para construir interfaces de interação com a aplicação (botões, caixas de texto, caixas de seleção)
 - **Scripts** (JavaScript) - para controlar eventos sobre os elementos do formulário
 - **Programas cliente** - Applets Java – processamento de dados no cliente
 - No **servidor**: documentos HTML estáticos + páginas dinâmicas - PHP, ASP, CGI, JSP, Servlet

Arquiteturas da Web

- Interação Cliente/Servidor (criação de aplicações web)
- Arquitetura *3-Tier* (3 camadas) – interativa



HTML da Web interativa

```
<TITLE>Questionario</TITLE>
<H1>Exemplo de Questionario</H1>

<P>Responda:
<FORM METHOD=post ACTION="http://www.icmc.usp.br/exemplo">
<P>Seu nome: <input name="name" size="48">
<P><input name="masc" type=radio> Homem
<P><input name="fem" type=radio> Mulher
  Ordem na familia: <input name="familia" type=int>

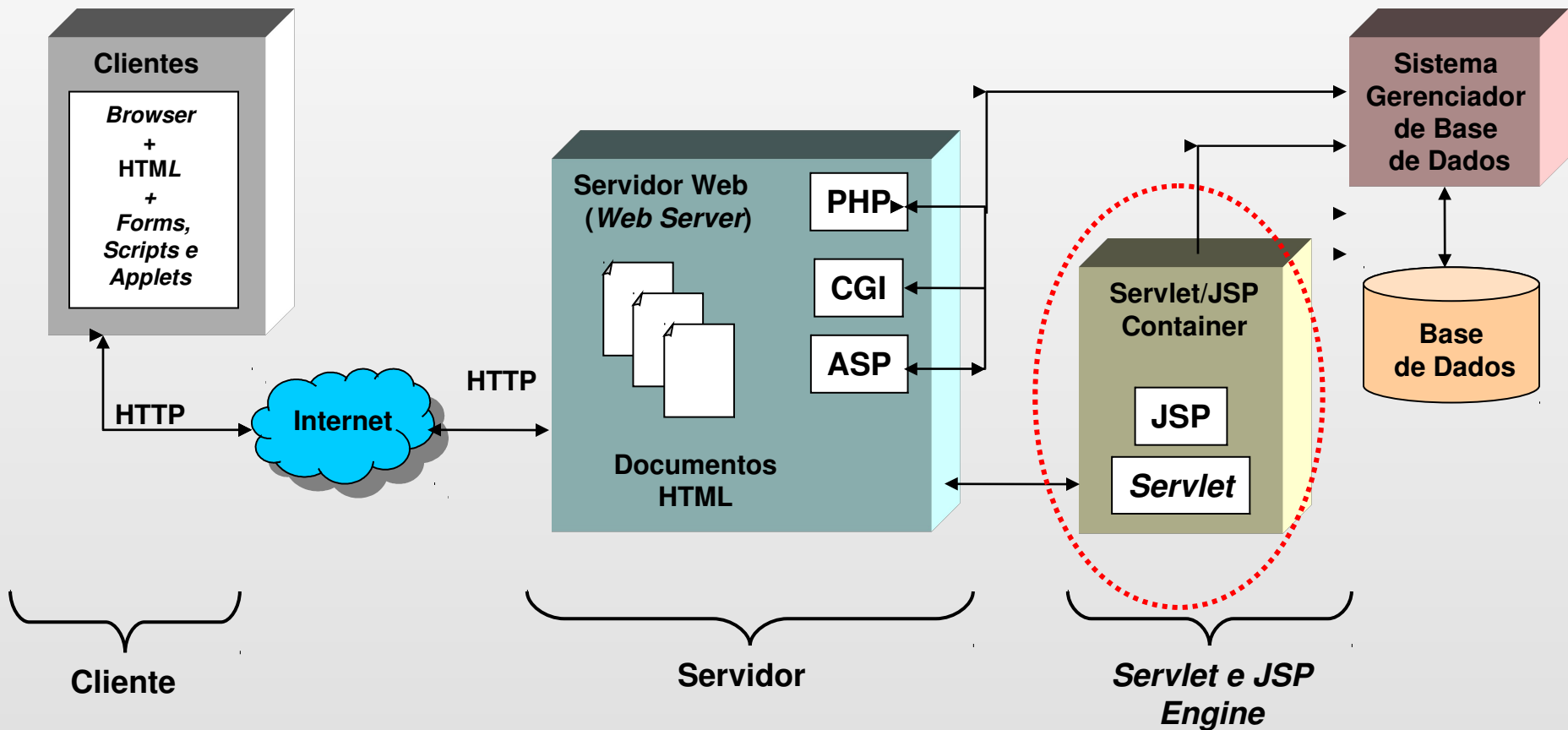
<P>Cidades em que possui residencia:
<UL PLAIN>
<LI><input name="cidade" type=checkbox value="spaulo"> S.Paulo
<LI><input name="cidade" type=checkbox value="scarlos"> S.Carlos
<LI>Others <textarea name="outro" cols=48 rows=4></textarea>
</UL>

<P>Apelido: <INPUT NAME="apelido" size ="42">

<P>Obrigada por suas respostas!
<P><INPUT TYPE=SUBMIT> <INPUT TYPE=RESET>
</FORM>
```

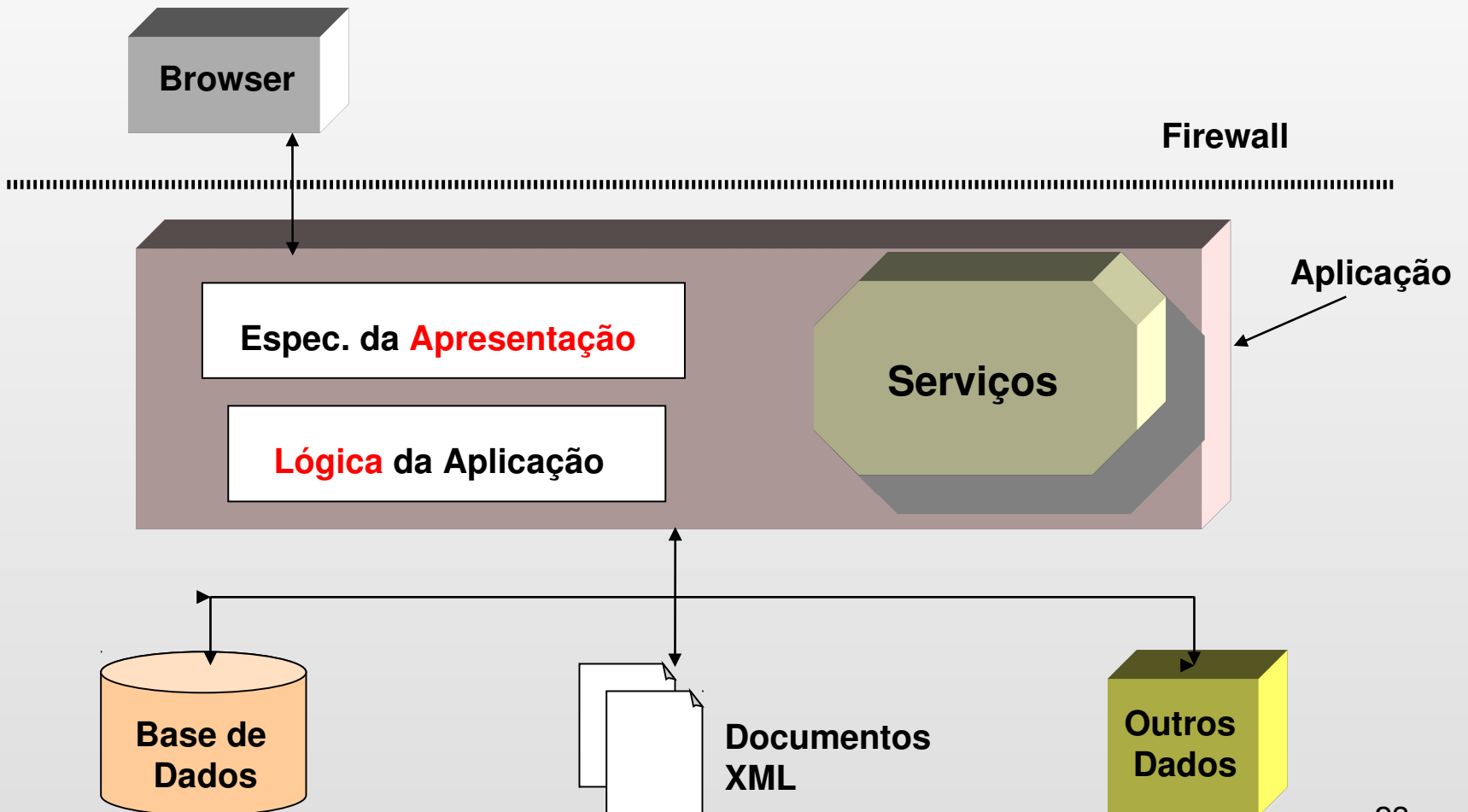
Arquiteturas da Web

- Arquitetura *N-Tier* (n camadas) – interativa



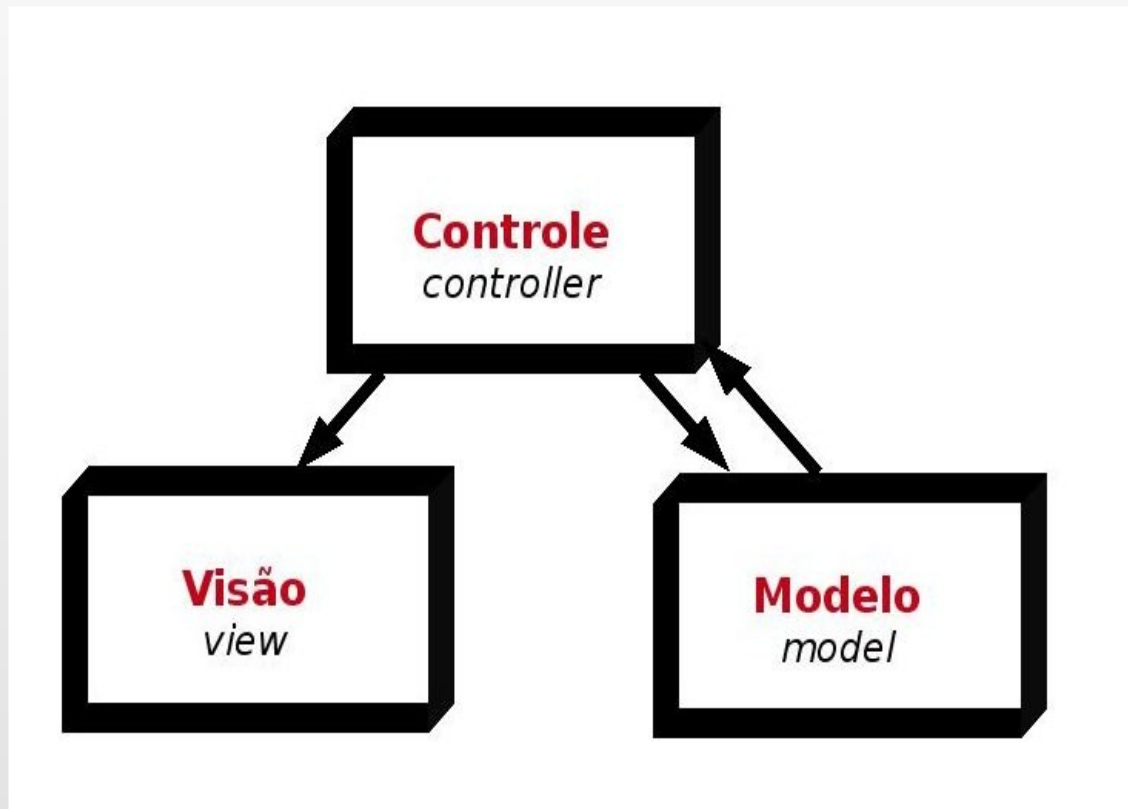
Arquiteturas da Web

Separação apresentação, lógica e dados da aplicação



Padrão de Arquitetura de aplicação Web

MVC = *Model View Controller*



Abordagens para Modelagem Web



- Facilitar o entendimento, especificação, documentação, visualização, comunicação e construção de aplicações Web
- Problema: *gap* em termos de objetivos
 - Modelagem de Informação (sites Web)
 - Modelagem funcional (aplicações Web)
- **OOHDM (Object Oriented Hypermedia Design Method)**
 - método de autoria para sistemas hipermídia
 - Conjunto de modelos (com respectivas notações) e um método de projeto
 - Ênfase para navegação e interface com usuário
- **WebML (Web Modeling Language)**
 - Conjunto de modelos para modelagem de web-sites
 - Pouca preocupação com aspectos funcionais (boa visão estrutural, de composição e navegação, de apresentação)

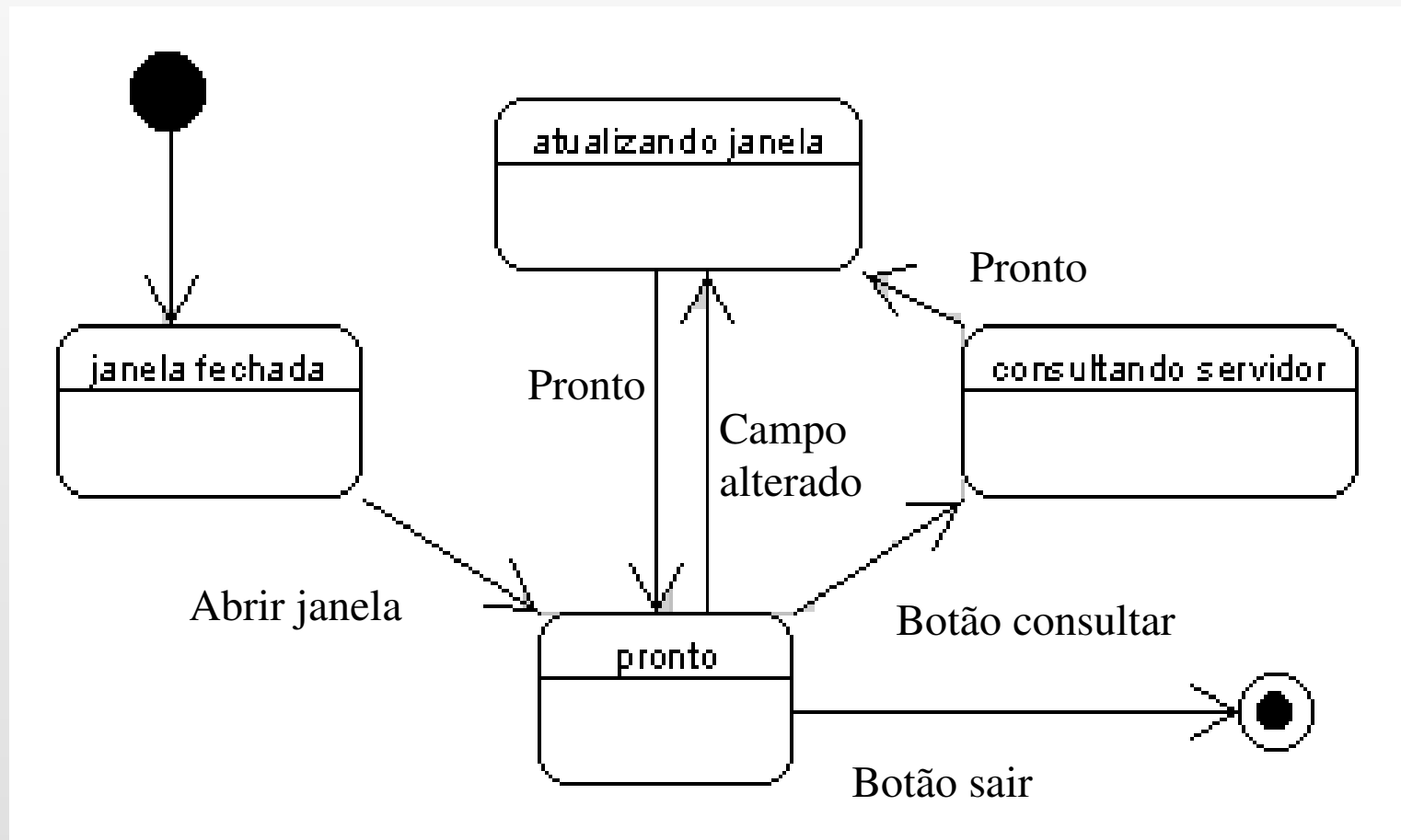
Abordagens para Modelagem Web



- **W2000**
 - Extensão dos modelos UML para modelagem Web
 - Ênfase na modelagem Hipermídia (informação/navegação) e preocupação com modelo funcional
 - Mapeamento do modelo funcional para sistema ainda confuso
- **UWE (UML-based Web Engineering)**
 - Extensão da UML para modelagem Web + ferramenta de suporte (ArgoUWE) com geração semi-automática de código e modelo.
 - Ênfase na modelagem conceitual, navegacional e de apresentação.
- **WAE (Web Application Extension) (Conallen)**
 - Extensão dos diagramas da UML para aplicações Web
 - Ênfase na modelagem funcional e no mapeamento dos modelos para tecnologias de desenvolvimento
 - Pouca (mas, presente) preocupação com modelagem de informação (informação/navegação)

Modelagem com UML

- Diagrama de Estado representando Interface do Usuário



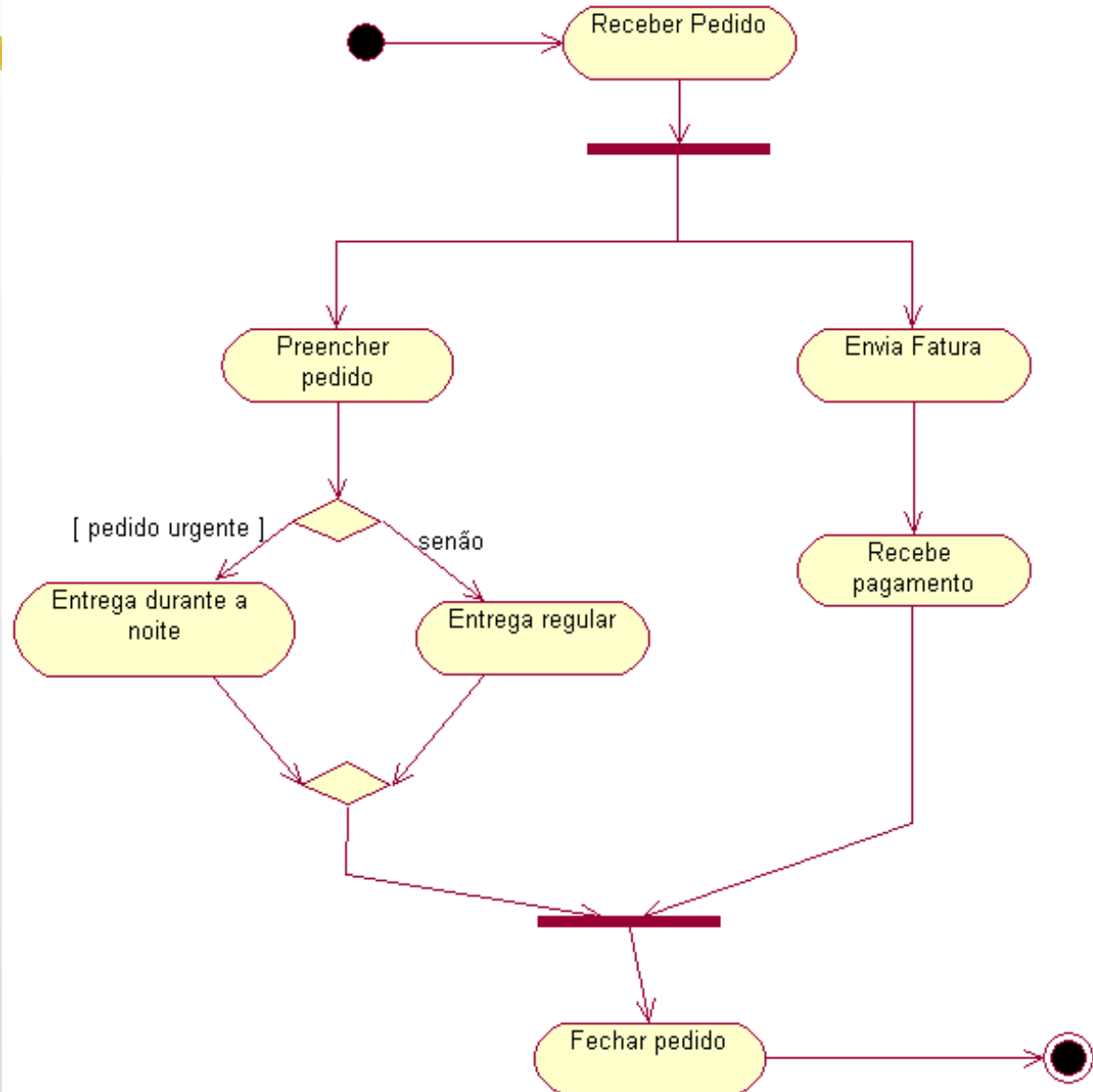
Modelagem com UML

■ Diagrama de Atividades

- Comportamento de uma trilha de execução em particular
 - Variante do diagrama de estados
- Descreve uma seqüência de atividades com suporte ao comportamento condicional e paralelo
- Comportamento condicional
 - Desvios (**branch**)
 - Única entrada e várias saídas (mutuamente exclusivos)
 - Intercalações (**merges**)
 - Várias entradas e única saída (fim do desvio)
- Comportamento paralelo
 - Separação (**fork**)
 - Única transição de entrada e várias transições de saída (a ordem de execução não é relevante)
 - Junção (**join**)
 - Fechamento da separação (próxima atividade só é executada quando todas as atividades separadas terminarem)

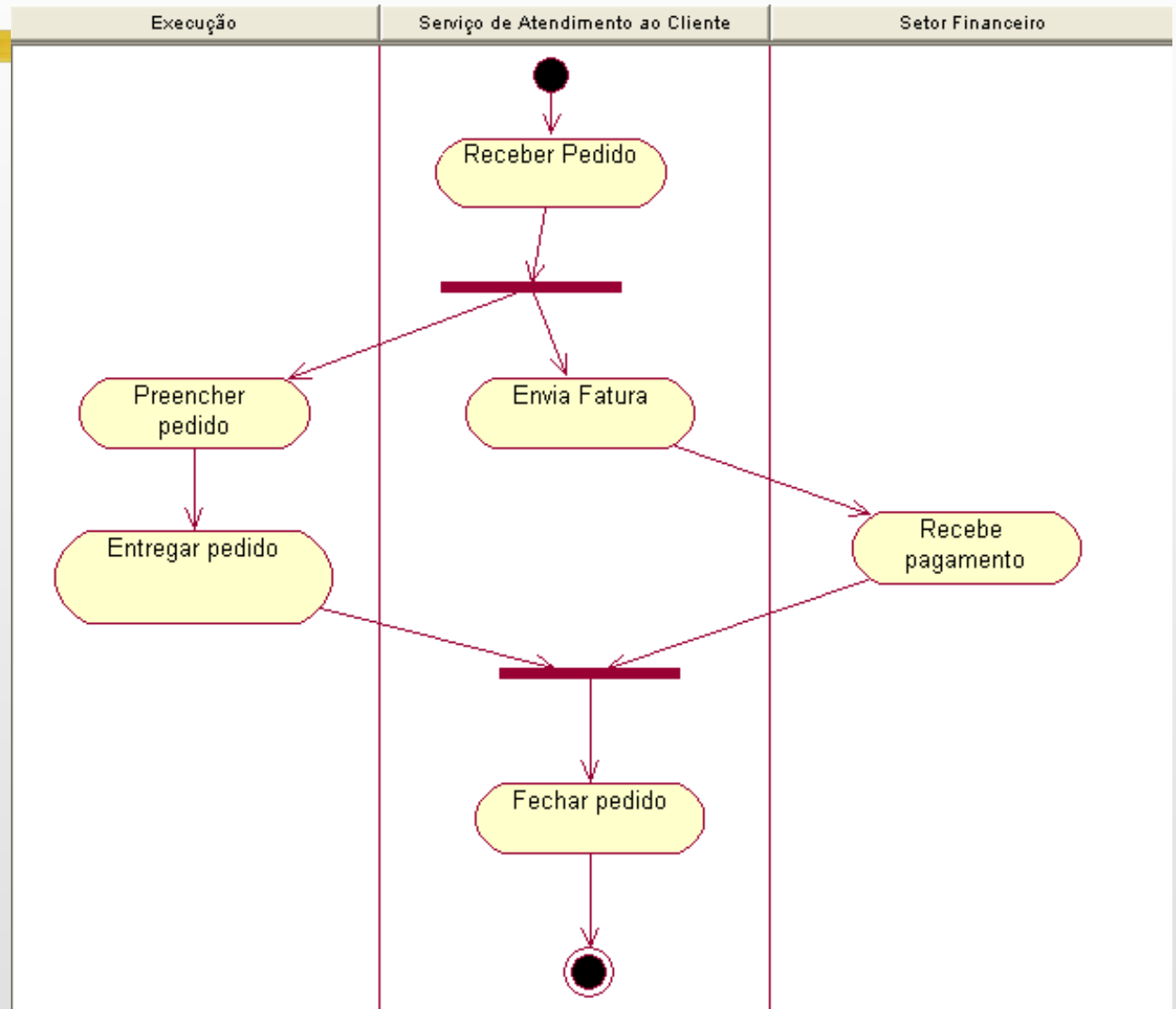
Modelagem com UML

Exemplo:



Modelagem com UML

Exemplo (2)



Bibliografia



- Conallen, JIM: *Desenvolvendo Aplicações Web com UML*, 2a ed, Ed. Campus, 2003.
- Fowler, Martin; Scott, Kendall, *UML Essencial*, 2a ed, Ed. Bookman, 2000.
- Page-Jones, Meilir, *Fundamentos do Desenho Orientado a Objeto com UML*, Ed. Person Education, 2001.
- Pressman, Roger S: *Engenharia de Software*, 5a ed., Ed. Makron Books, 2002.
- Hickson, Rosangela: *Projeto de Sistemas Web Orientados a Interface*, Ed. Campus, 2003.

Agradecimentos:

- *Elaine Quintino Silva*, por boa parte dos slides