

Desenvolvimento de Aplicações WEB

Renata Pontin de Mattos Fortes

ICMC-USP S.Carlos, 2011

Tópicos

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

A World Wide Web

- **Internet**

- Rede formada por outras redes (*inter-network*) que permite a comunicação entre os computadores interligados (surgimento nos anos 60)

- **World Wide Web**

- Serviço da Internet
 - Ferramentas e protocolos para compartilhar informação dos servidores da Internet
- Primeiros conceitos em 1989; primeiro *browser* gráfico divulgado em 1993 ([Mosaic](#))
 - [Tim Berners-Lee](#) criou os primeiros *web sites*
 - Propósito de divulgação de pesquisa científica
 - servidor de arquivos baseado em URL (*Universal Resource Location*)
 - Os documentos eram simplesmente apresentados no *browser*

A World Wide Web

- **World Wide Web**

- Em 1990, uso comercial da rede
- Protocolo de transferência: **HTTP** (*Hypertext Transfer Protocol*)
 - Permite a troca de informação entre os clientes e servidores Web padronizando o “diálogo”
 - “É a linguagem de comunicação”
 - O servidor Web é também chamado “Servidor HTTP”
- Sistema de Tim Berners-Lee → Sistema Hipermídia (Hipertexto + Multimídia)
 - **Nó** – unidade de informação (página, frame, formulário...)
 - **Link** – representa uma associação entre dois nós
 - **Âncora** – link para o próprio documento
- Web (teia) → Interligação entre recursos (**navegação**)

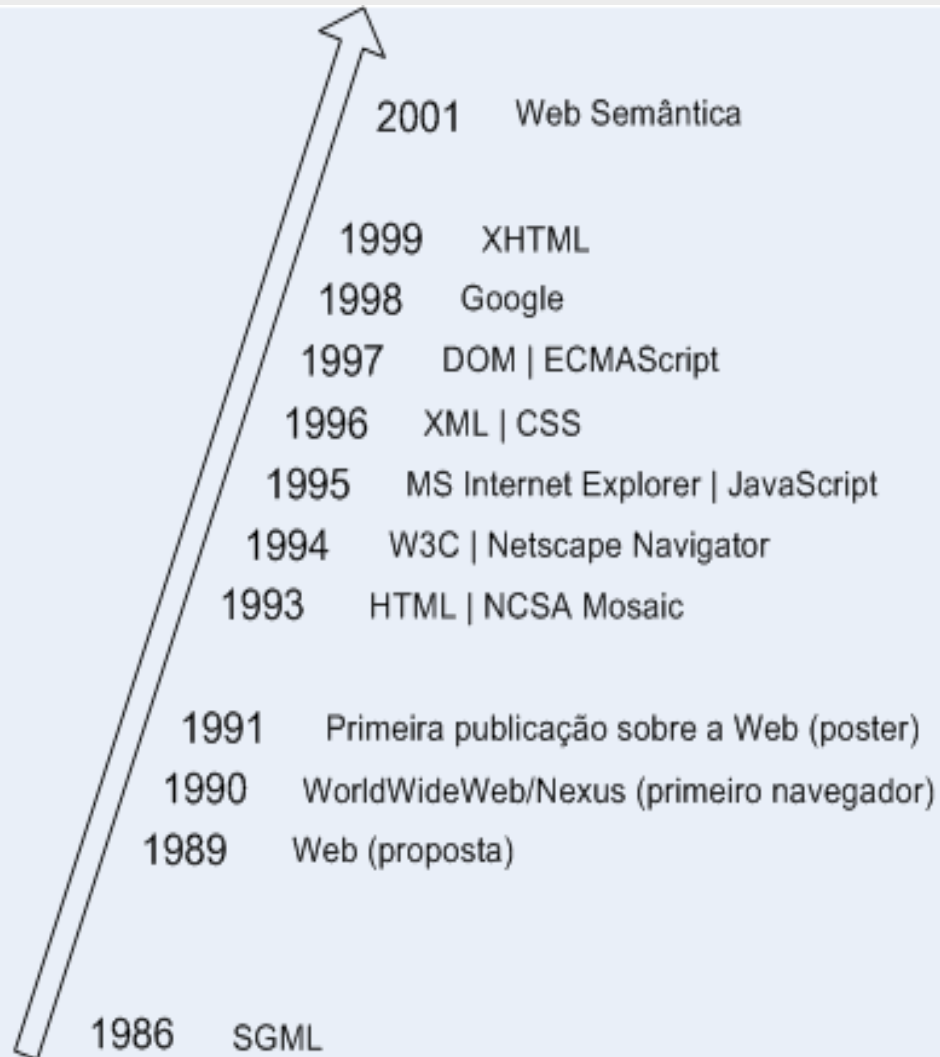
A World Wide Web

- Interesse maior: especificação do **HTML 2.0 em 1996**
 - **Formulários HTML** ⇒ permitiram a criação de aplicações mais interativas
 - botões, caixas de seleção, caixas de texto,...

A World Wide Web

- Site Web x Aplicação Web
- Arquitetura de site Web
 - **servidor Web**, **conexão de rede** e **browsers** (clientes)
- Aplicações Web
 - *Desenvolvida 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**

A World Wide Web



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, Active Client Pages

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.

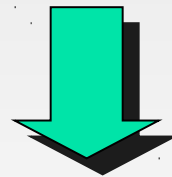
Engenharia de Web (Web Engineering)

- 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 extremamente 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

O papel da modelagem

■ Entendimento

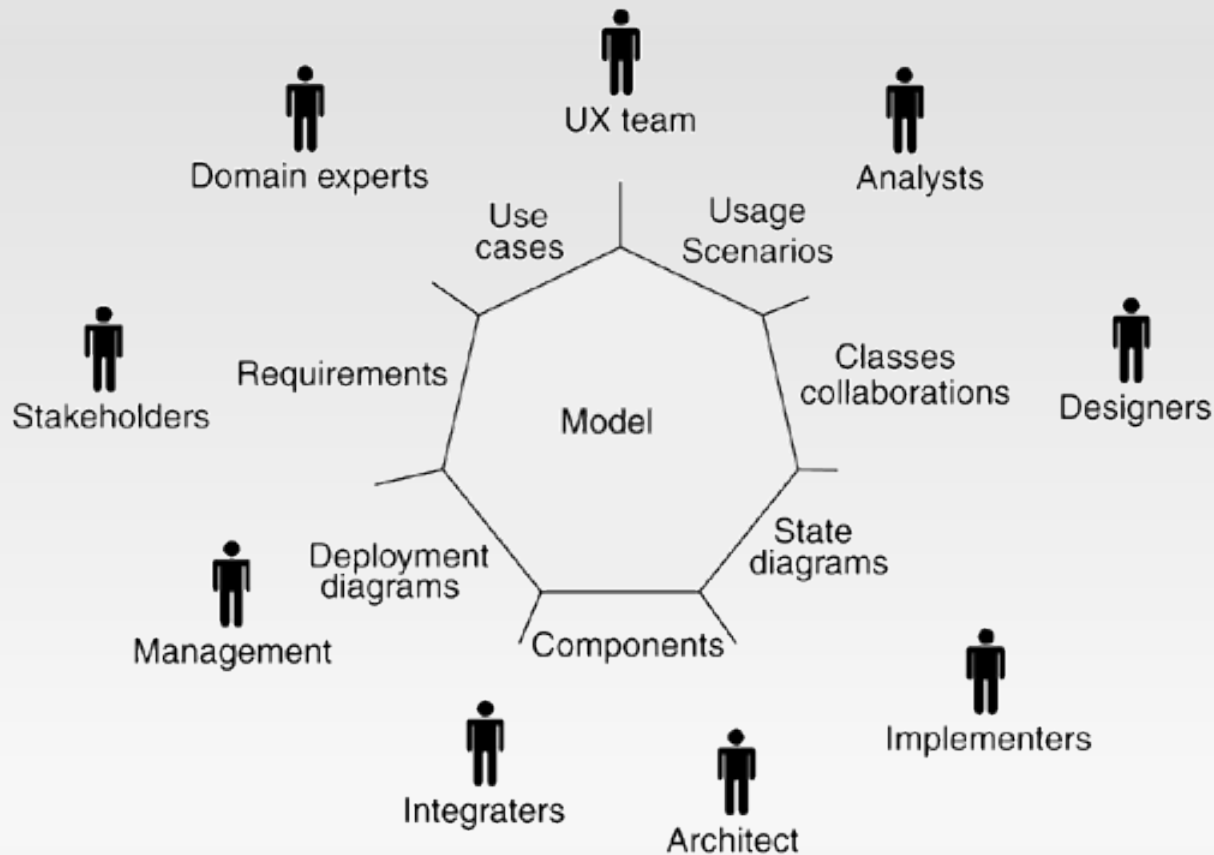
- O que **se deseja** construir, o que **está sendo** construindo e o que **foi** construído
- Descrição da realidade (níveis de abstração)
 - Mais abstratos: modelos mais longe da realidade e mais simples
 - Menos abstratos: modelos mais próximos da realidade e mais complexos
- Modelos devem expor o 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)
 - Auxilia no processo de desenvolvimento

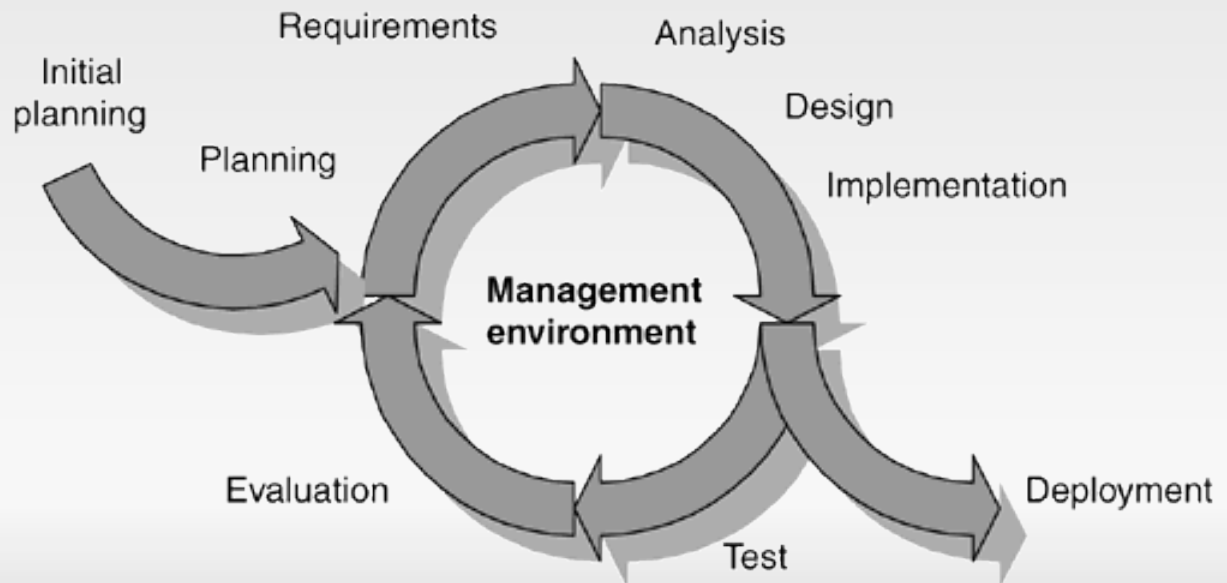
OBS.: o papel da modelagem não é produzir código através da automação ou produzir documentos por meio da engenharia reversa (produtos secundários das ferramentas CASE)

- 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

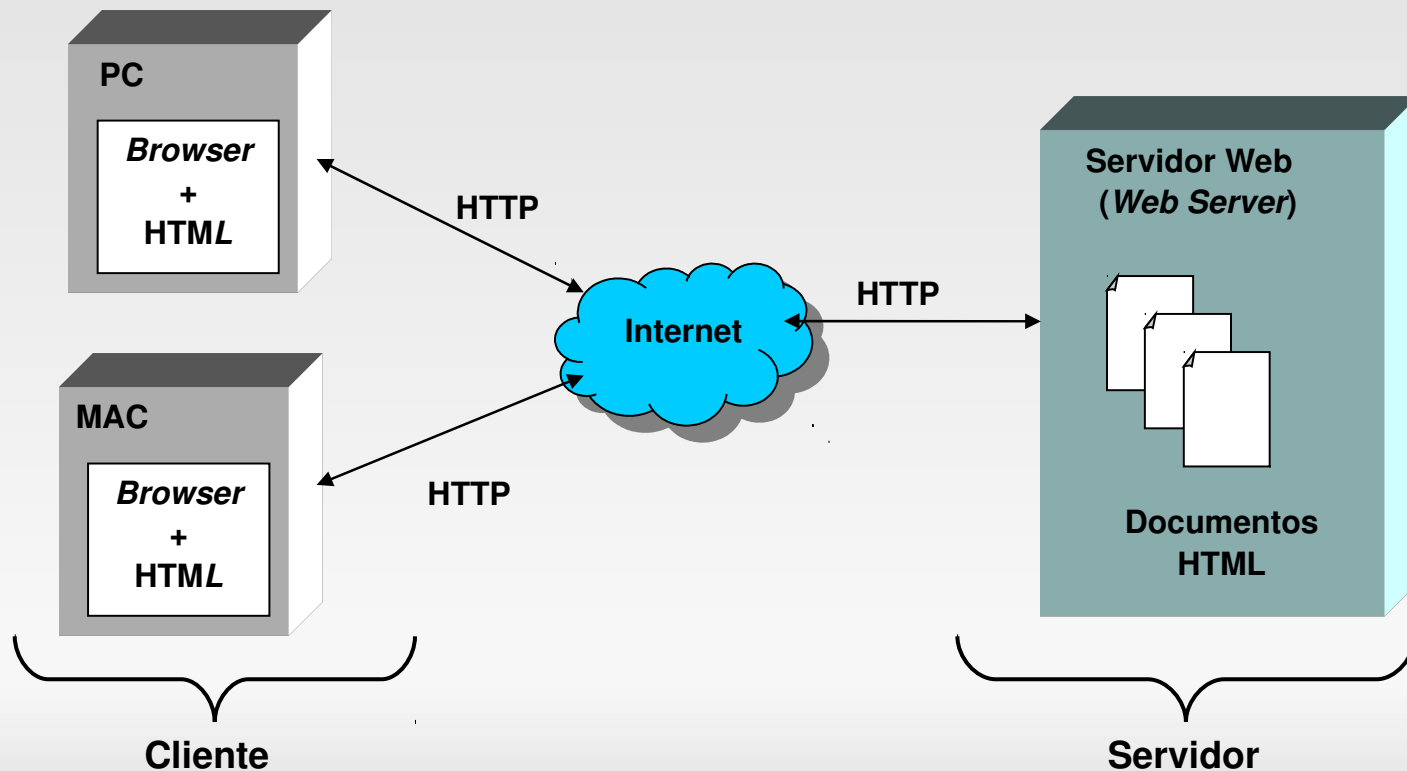
- O papel da Arquitetura
 - Influência no processo de desenvolvimento e no produto final
 - Define regras para a construção do software
 - como o software deve ser “pensado” ?
 - Arquitetura da Web - cliente/servidor diferenciado
 - Servidor não controla o cliente / Interação iniciada pelo cliente
 - 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

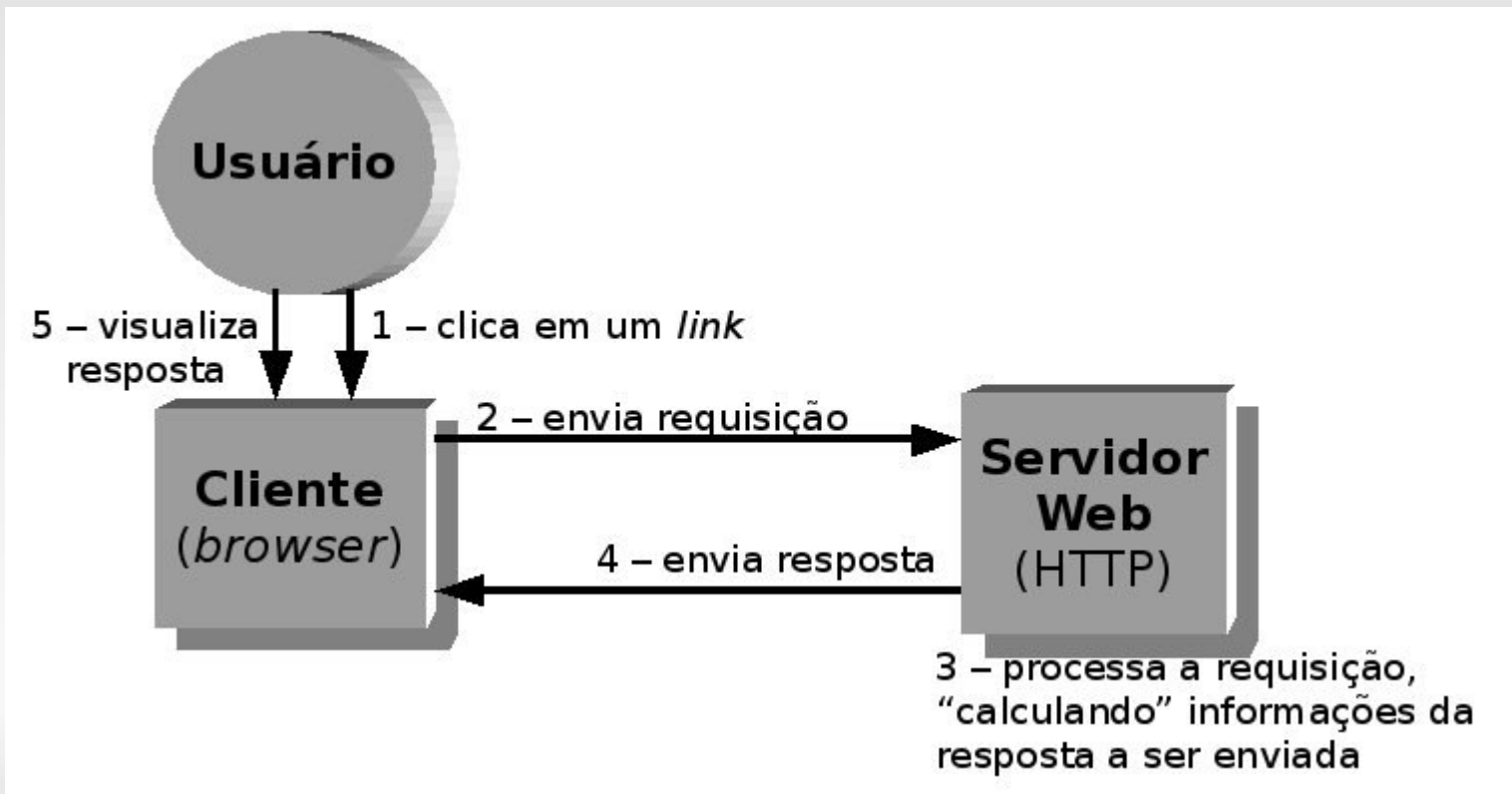
Arquiteturas da Web

- 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

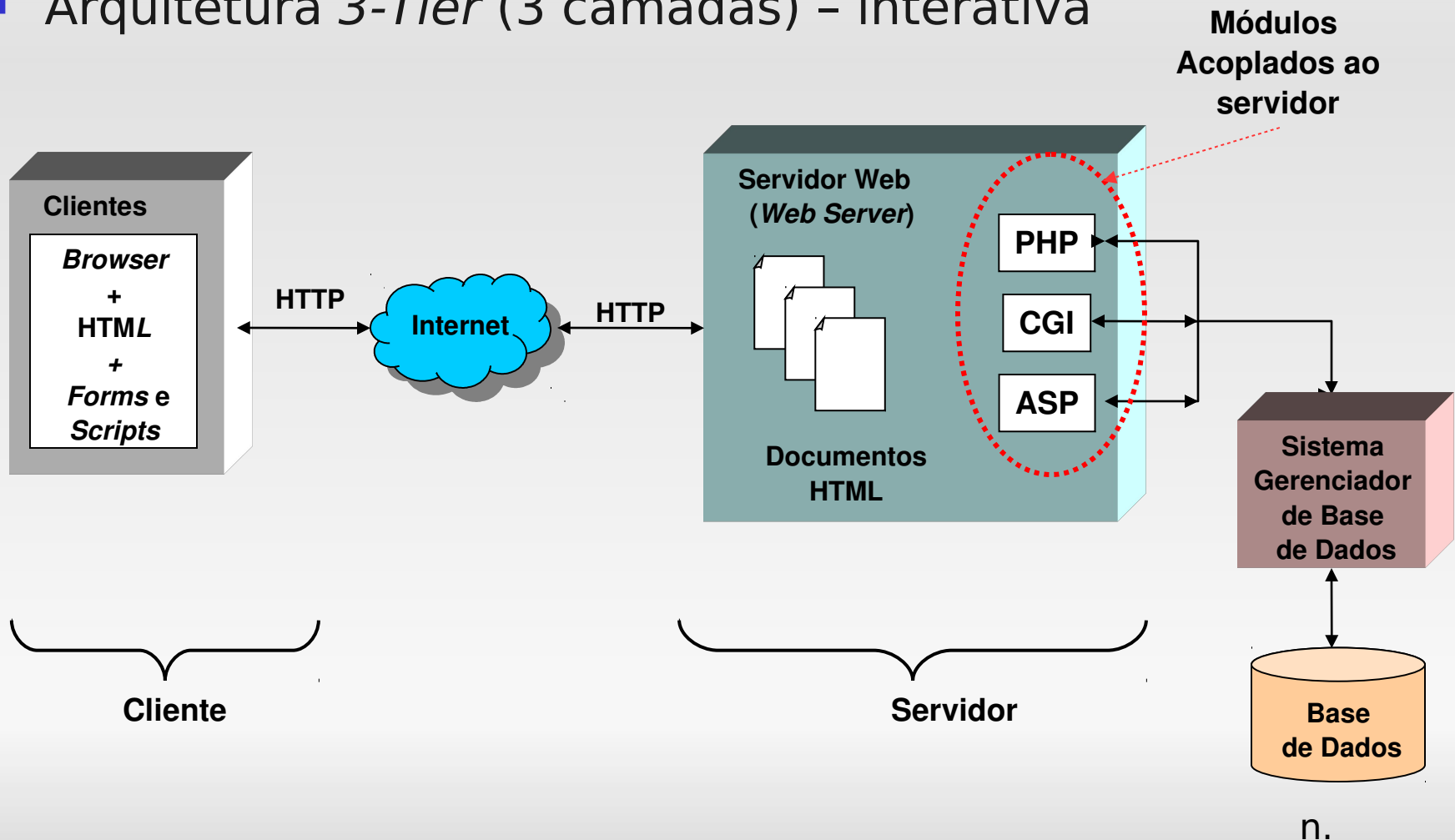


Arquiteturas da Web

- 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>

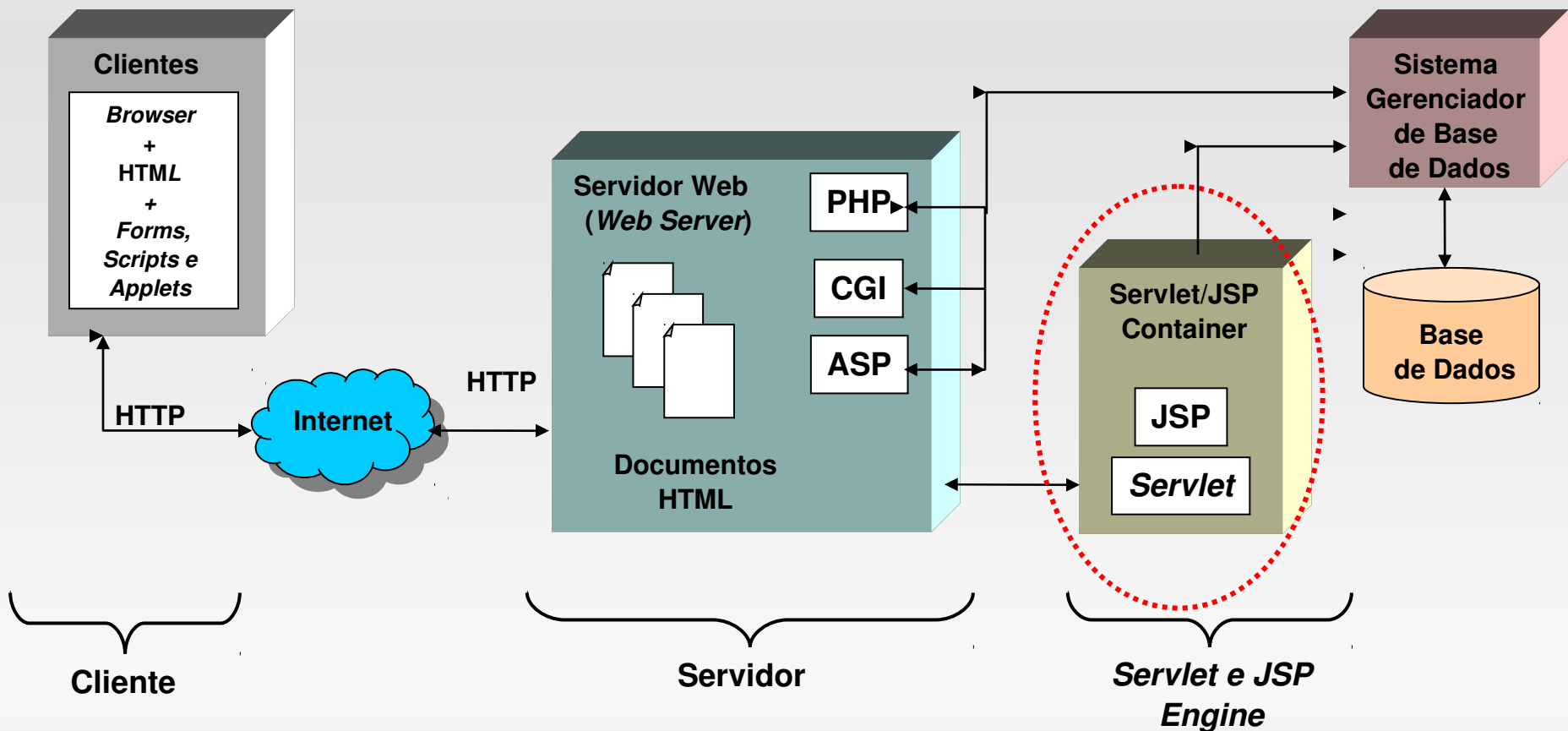
<P>Cidades em que possui residencia:
<UL PLAIN>
<LI><input name="cidade" type=checkbox value="spaulo"> S.Paulo
<LI><input name="cidade" type=checkbox value="scarlos"> S.Carlos
<LI>0thers <textarea name="outro" cols=48 rows=4></textarea>
</UL>

<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



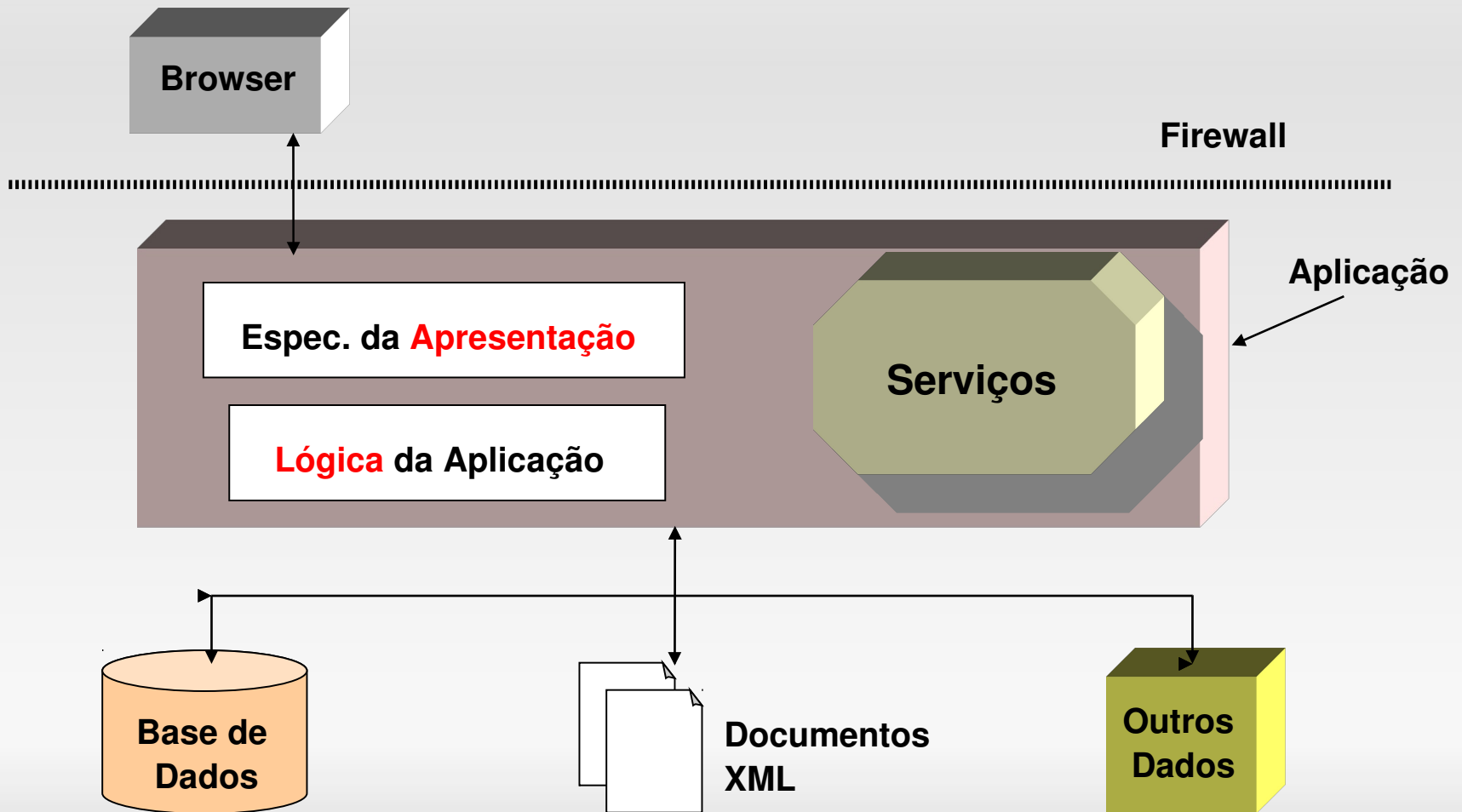
Arquiteturas da Web

■ **Servidores de aplicação:**

- Sistemas que fornecem a infraestrutura de serviços para a execução de aplicações distribuídas
- Permitem a integração das aplicações com infraestrutura já existente (banco de dados, sistemas legados,...) de forma transparente
- Promovem a separação entre a **apresentação, lógica e dados** da aplicação:
 - Atualizações da aplicação tornam-se mais fáceis
 - O processamento da aplicação é todo feito no servidor
- Servidores de aplicação podem ser:
 - Servidores de Web, Servidores de Email, Servidores de base de dados, ...

Arquiteturas da Web

Separação apresentação, lógica e dados da aplicação

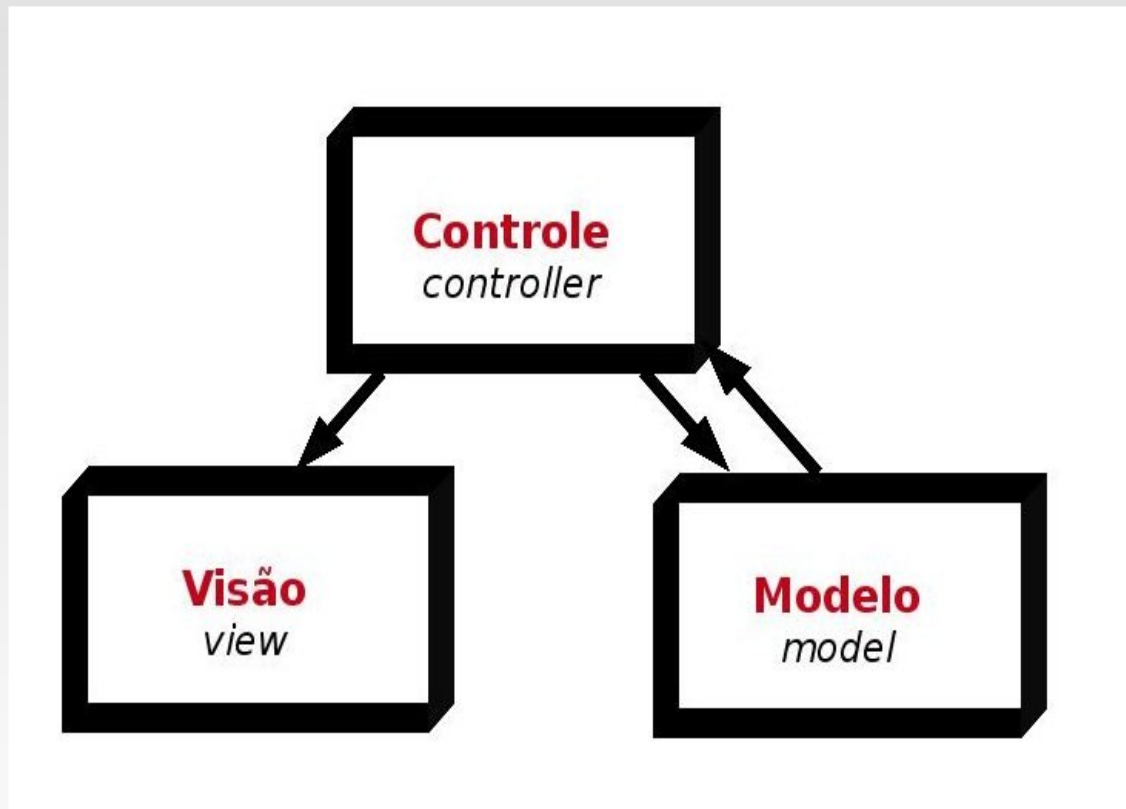


Arquiteturas da Web

- Vantagens dos Servidores de Aplicação:
 - maior concentração do desenvolvedor na lógica de negócio (**lógica da aplicação**) – *business logic*
- Outras características:
 - Tolerância a falhas
 - Balanceamento de carga
 - Gerenciamento de transações (integridade da transação)
 - Segurança
 - Persistência
- Conceitos importantes:
 - objetos distribuídos, componentes,
 - J2EE (Java 2 Enterprise Edition) e ASP (Application Service Providers)

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

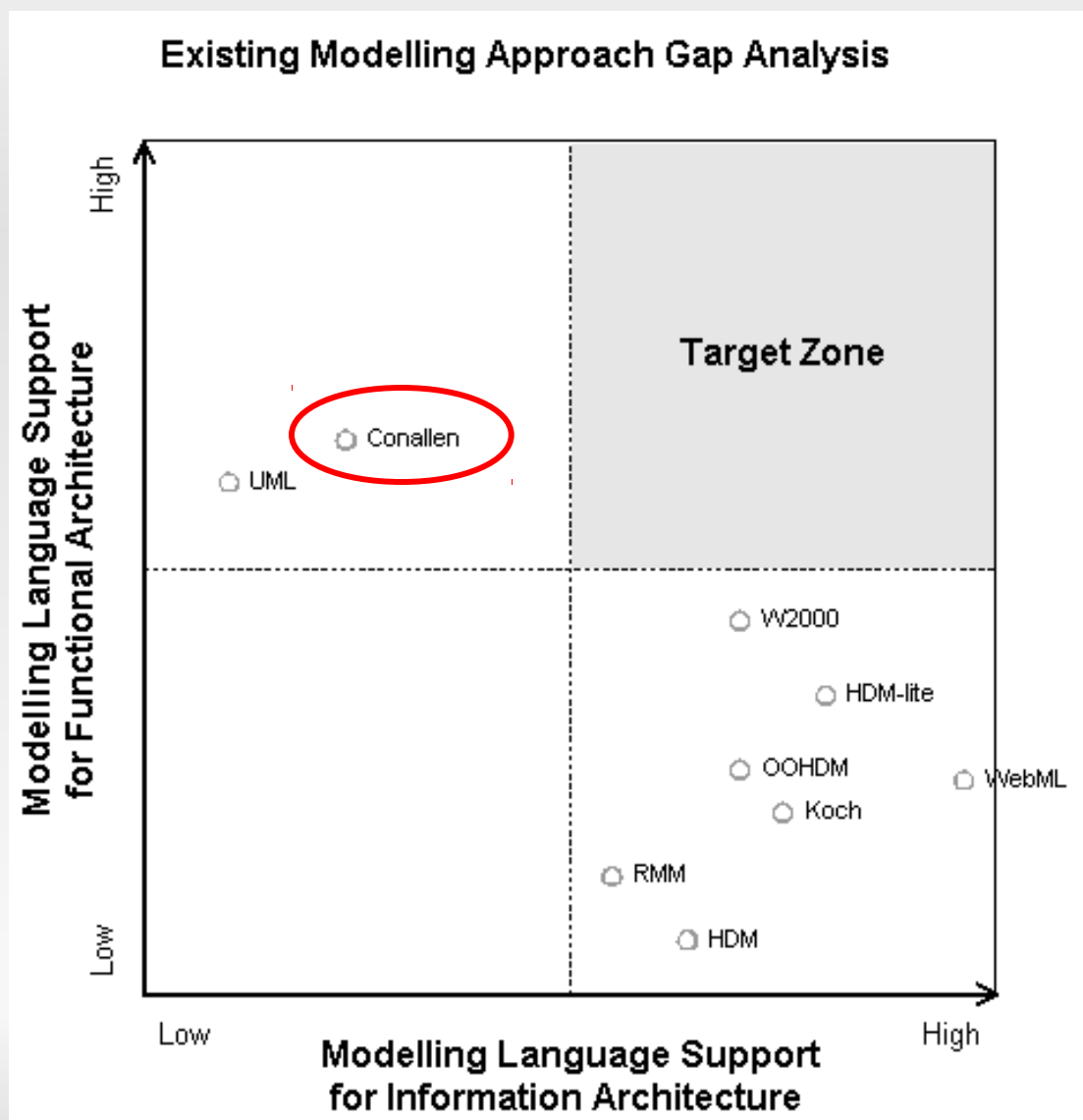
■ UWE (UML-based Web Engineering)

- 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)

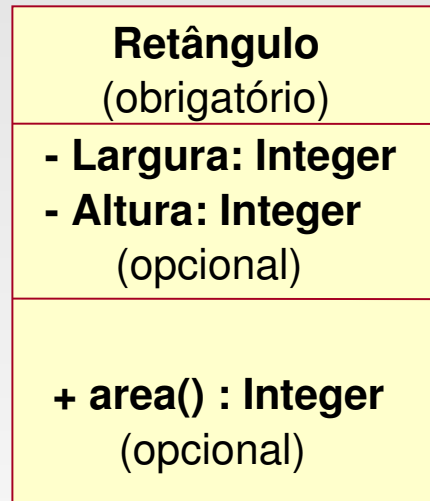
- Extensão dos diagramas da UML para aplicações Web
- Ênfase na modelagem funcional e no mapeamento dos modelos para tecnologias de desenvolvimento
- Pouca (mas, presente) preocupação com modelagem de informação (informação/navegação)

Abordagens para Modelagem Web



Modelagem com UML

■ Notação:



Restrição

{area <= 600}

Sintaxe de atributos

visibilidade nome : tipo = valor inicial

visibilidade é um dos seguintes símbolos:

- + visibilidade pública
- # visibilidade protegida
- visibilidade privada

Sintaxe de operações

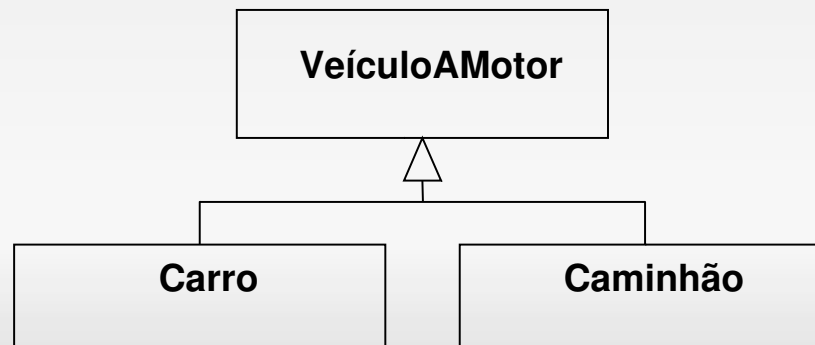
visibilidade nome (parâmetros) : tipo de retorno

parâmetros é uma lista de parâmetros separados por vírgula e especificados segundo a sintaxe:

nome : tipo = valor default

Modelagem com UML

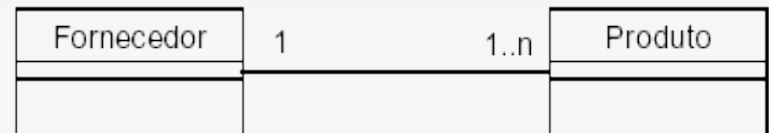
- Tipos de relacionamento entre classes
 - Generalização
 - Associação
 - Composição/Aggregação
- Generalização
 - Caracteriza Herança (superclasse/subclasse)



Modelagem com UML

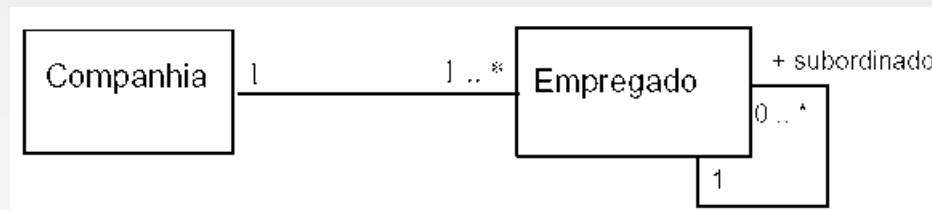
■ Associação

- Relação entre ocorrência de classes
 - Conceitual → relações conceituais
 - Especificação → responsabilidades
 - Implementação → composição interna das classes
- Duas pontas de associação
 - Multiplicidade (número de objetos que podem participar de um relacionamento)
 - 0..1 (opcional- 0 ou mais)
 - 1..n (de 1 a n)
 - 1 (exatamente 1)
 - 0..* (muitos- 0 ou mais)
 - 1..* (1 ou mais)
 - n..m (de n a m)



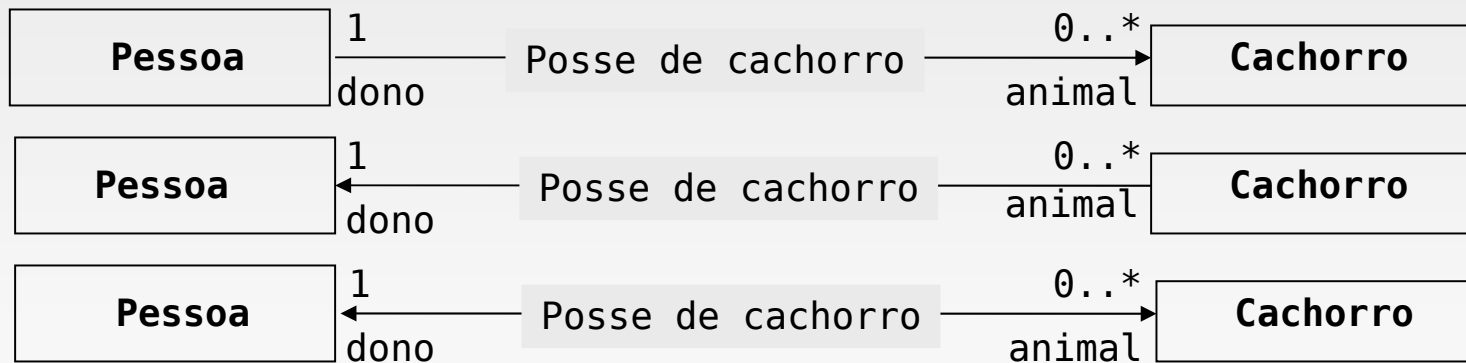
Modelagem com UML

- Associação com papéis
 - Cada ponta de associação pode ser rotulada (papel)
 - Se a ponta não possui papel, o papel é o nome da classe alvo



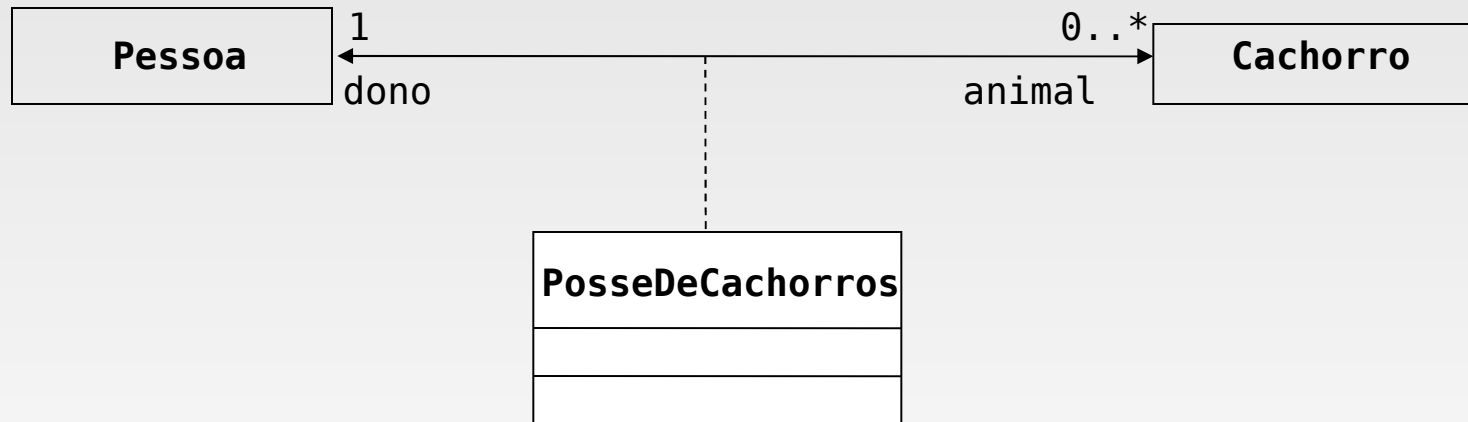
Modelagem com UML

- Associação com nome e navegabilidade (implementação)
 - Nome → quando o nome ajuda no entendimento
 - Navegabilidade → implementação
 - Unidirecional
 - Bidirecional



Modelagem com UML

- Classe de associação
 - só existe quando há o relacionamento entre duas classes



Modelagem com UML

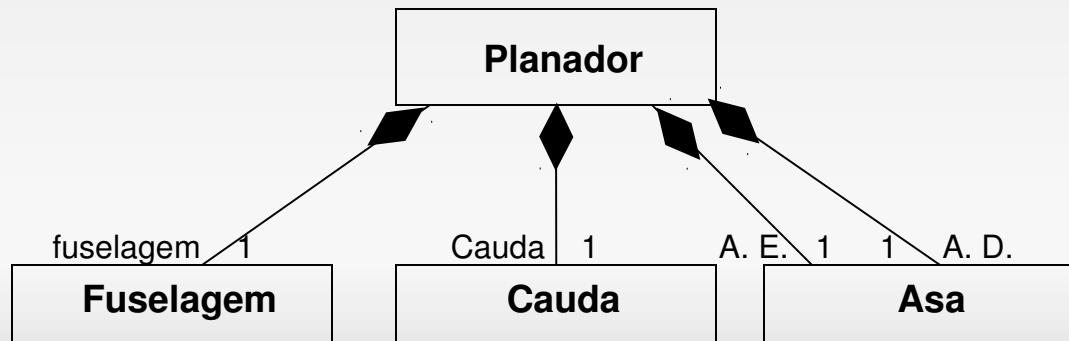
■ Agregação

- Indica que uma classe é **parte-de** outra classe



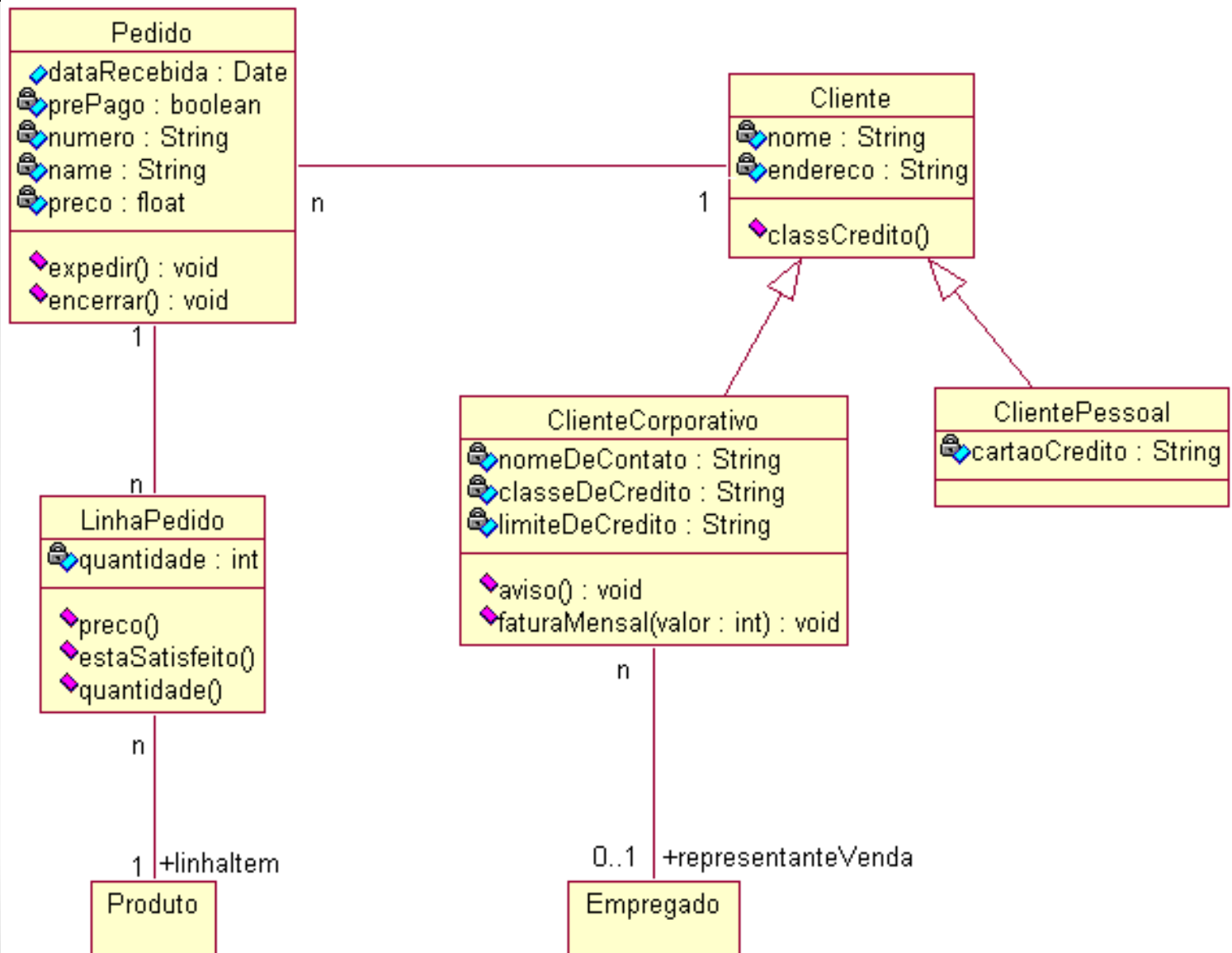
■ Composição

- Indica que uma classe é **composta** por outras
 - Objeto composto não existe sem seus componentes
 - Se a classe composta é removida, todos as suas partes também o são



Modelagem com UML

Exemplo 1: Conceitual



Modelagem com UML

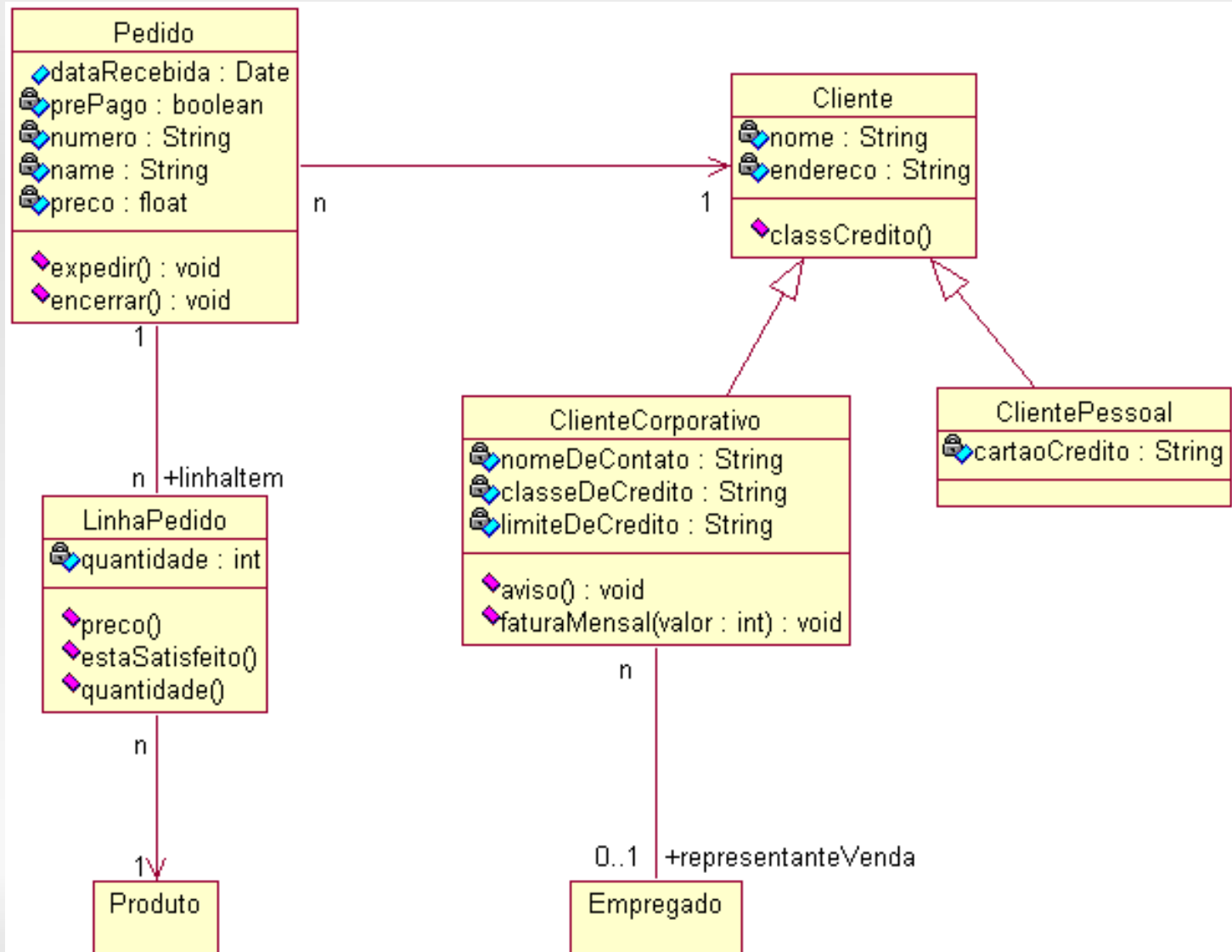
■ Código (Exemplo 1)

```
public class Pedido {  
    public Date dataRecebida;  
    private boolean prePago;  
    private String numero;  
    private String name;  
    private float preco;  
    public LinhaPedido linhaItem[];  
    public Cliente theCliente;  
  
    public Pedido(){  
    }  
  
    public void expedir(){  
    }  
  
    public void encerrar(){  
    }  
}
```

```
public class Cliente {  
    private String nome;  
    private String endereco;  
    public Pedido thePedido[];  
  
    public Cliente(){  
    }  
  
    public String classCredito(){  
    }  
}
```

Modelagem com UML

Exemplo 2: Especificação/ Implementação



Modelagem com UML

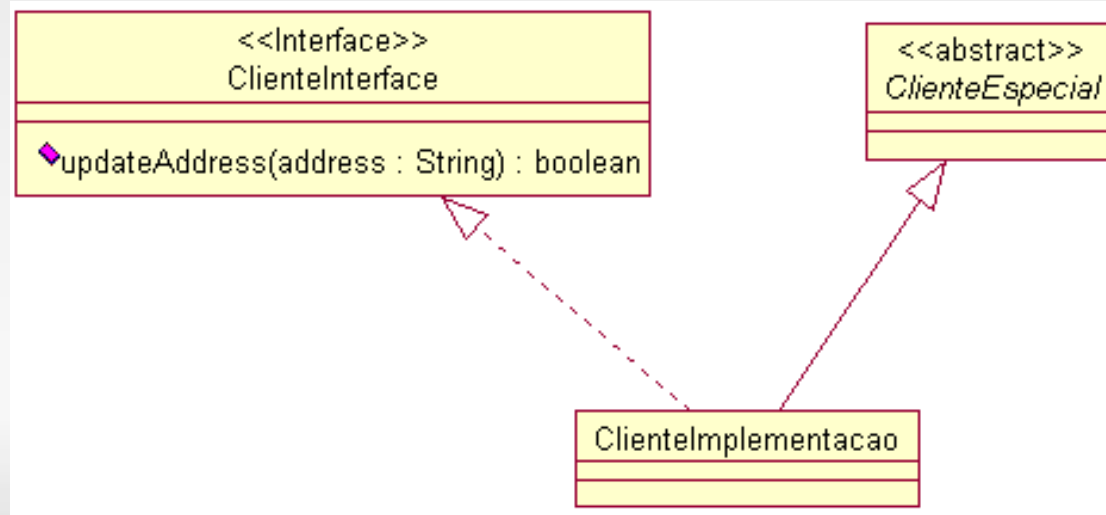
■ Código (Exemplo 2)

```
public class Pedido {  
    public Date dataRecebida;  
    private boolean prePago;  
    private String numero;  
    private String name;  
    private float preco;  
    public LinhaPedido linhaItem[];  
    public Cliente theCliente;  
  
    public Pedido(){  
    }  
  
    public void expedir(){  
    }  
  
    public void encerrar(){  
    }  
}
```

```
public class Cliente {  
    private String nome;  
    private String endereco;  
  
    public Cliente(){  
    }  
  
    public String classCredito(){  
    }  
}
```

Modelagem com UML

- Estereótipos (*Stereotypes*)
 - Mecanismo de extensão da UML que permite estender o meta-modelo para suprir necessidades que não se encontram entre os meta-elementos disponíveis.
 - Estereótipos podem ser aplicados a qualquer objeto do diagrama (classes, atributos, operações, associações e generalizações, etc).
 - Ex: <<abstract>>, <<metaclass>>, << interface>>



Modelagem com UML

- Código das classes estereotipadas

```
public interface ClienteInterface{  
    public boolean updateAddress(String address);  
}
```

```
public abstract class ClienteEspecial{  
    public ClienteEspecial(){  
    }  
}
```

```
public class ClienteImplementacao extends ClienteEspecial  
implements ClienteInterface {  
    public ClienteImplementacao(){  
    }  
    public boolean updateAddress(String address){  
    }  
}
```

Modelagem com UML

- Restrições
 - Condições ou proposições que devem ser mantidas verdadeiras
 - É mostrada como uma cadeia entre chaves { } (*OCL – Object Constraint Language*)
- Notas
 - Comentário inserido no diagrama

Modelagem com UML

- **Comportamento do Sistema**

- **Diagrama de Interações**

- Relata como grupos de objetos se relacionam (se comportam) dentro de um único caso de uso.
 - Como os objetos colaboram em um caso de uso
 - Diagramas : de Seqüência e de Colaboração

- **Diagrama de Estados**

- Relata o comportamento de um único objeto através de muitos casos de uso
 - Ciclo de vida de um objeto em um caso de uso

- **Diagrama de Atividades**

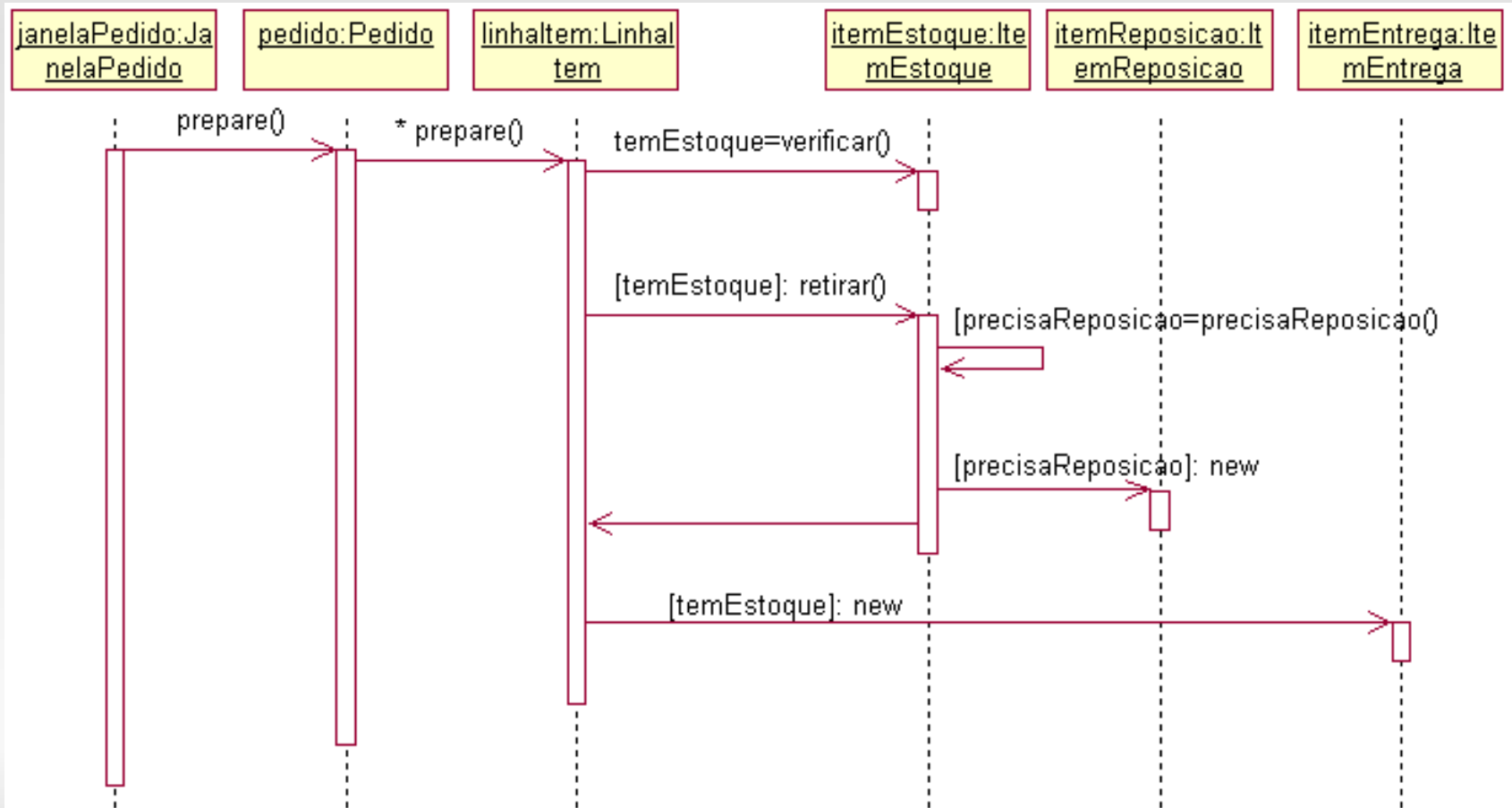
- Relata o comportamento de vários casos de uso ou trilhas de execução
 - Compreender o comportamento geral do sistema

Modelagem com UML

- Diagrama de Seqüência (Diagramas de Interação)
 - Colaboração entre objetos organizada em um seqüência de tempo
 - Abordagens:
 - Cenários entre ator e sistema
 - Mostra a interação do ator com o sistema (como caixa-preta)
 - Fase: análise
 - Cenários entre objetos do sistema
 - Mostra a interação entre os vários objetos do sistema para a execução de um caso de uso
 - Fase: projeto
 - Duas dimensões
 - Vertical: representa o tempo
 - Horizontal: representa os diferentes objetos

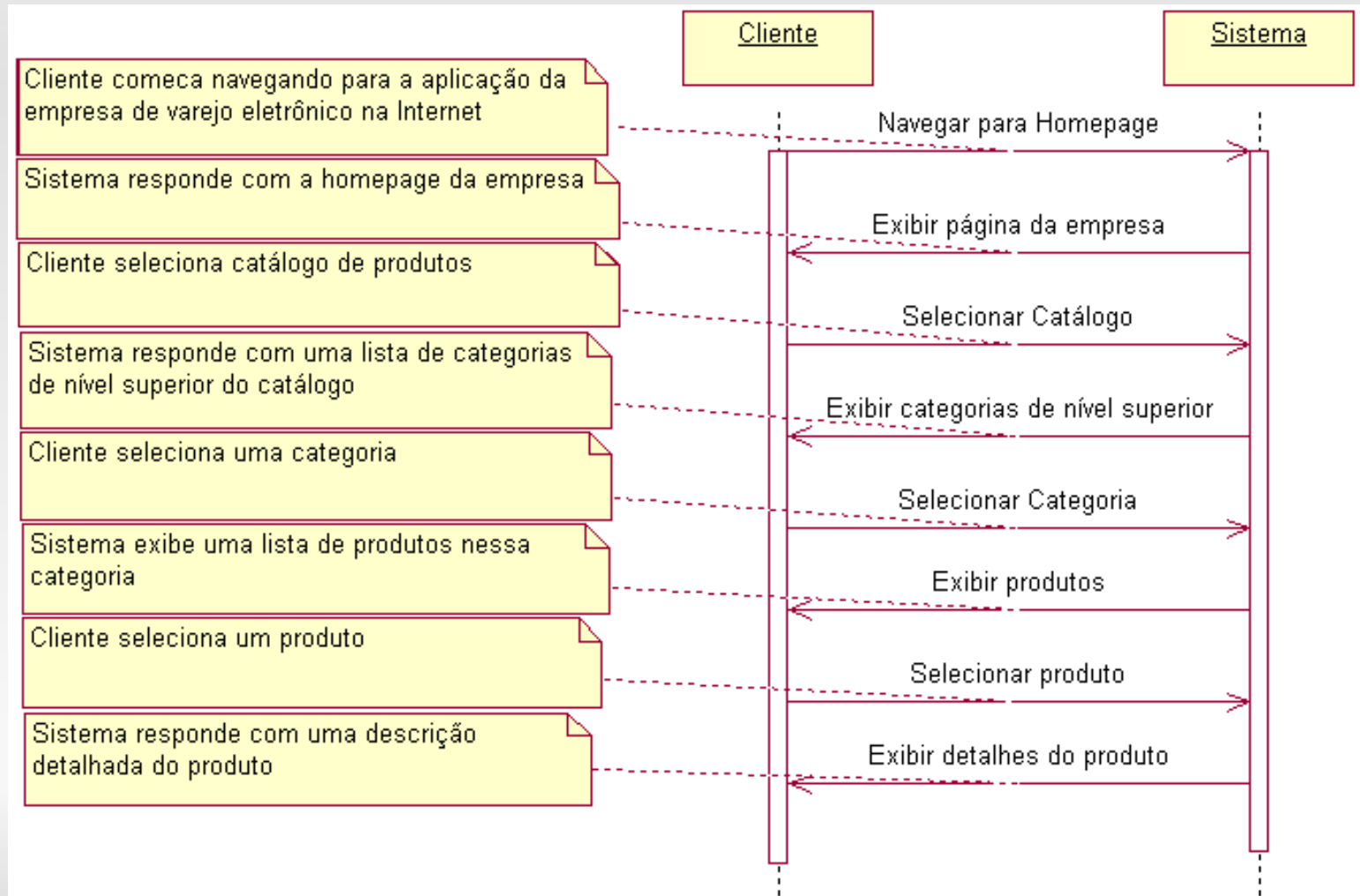
Modelagem com UML

- Exemplo (interação entre objetos):



Modelagem com UML

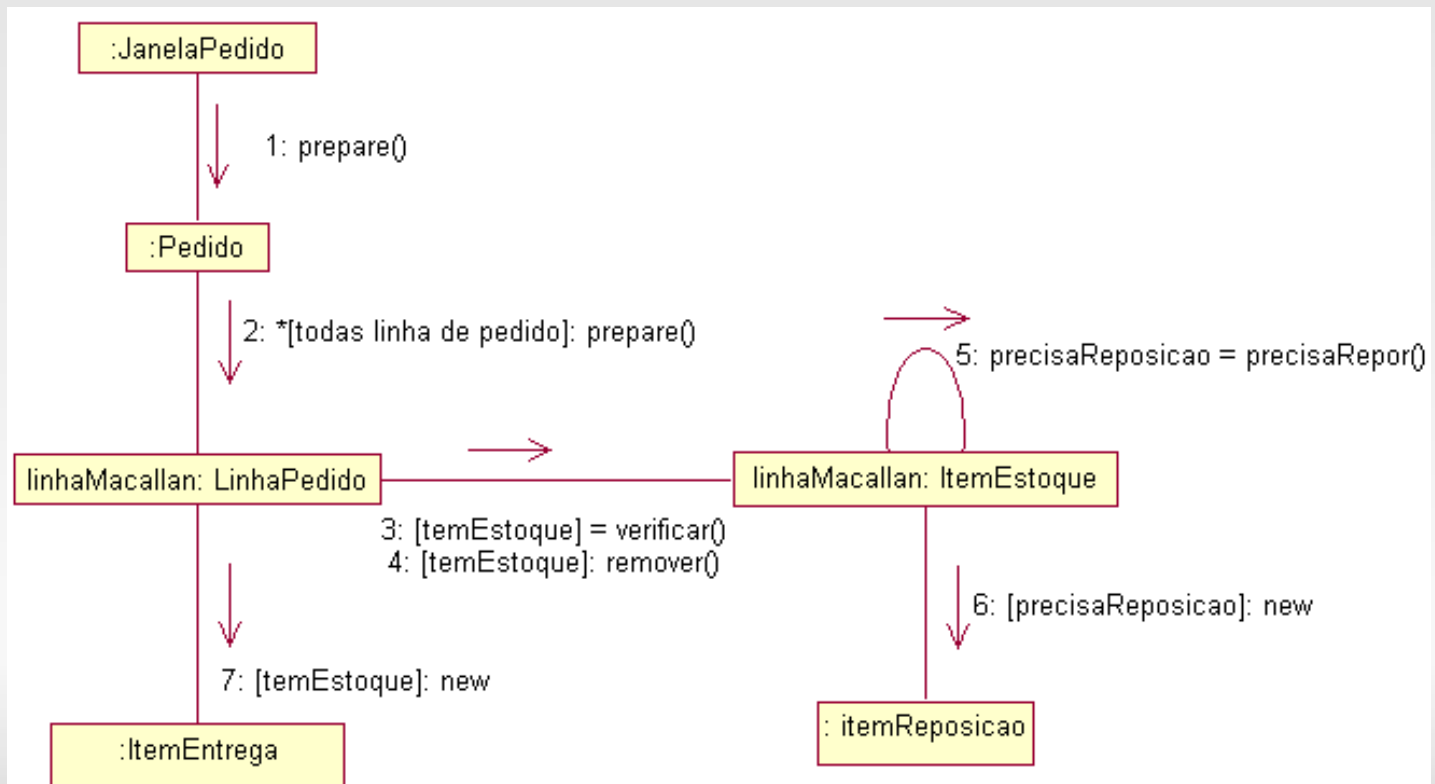
- Exemplo (interação entre ator e sistema)



Modelagem com UML

■ Diagrama de colaboração

- Não mostra o tempo e interação com atores
 - seqüência de mensagens é determinada usando-se números em seqüência e os objetos mostrados como ícones



Modelagem com UML

■ Diagrama de Estados

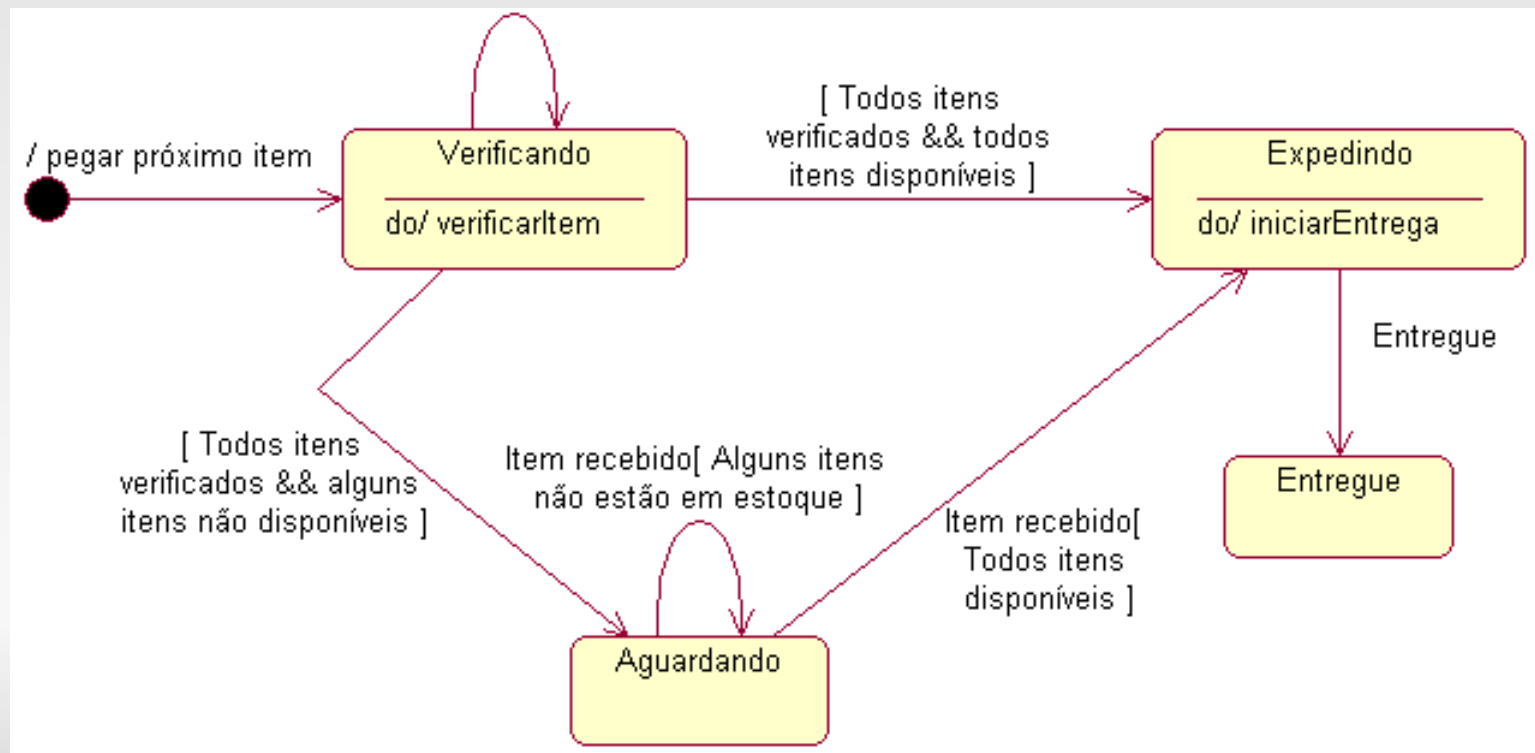
- Representa o **ciclo de vida** de um objeto ou interfaces com usuário
- Para uma classe, mostra os estados que um objeto desta classe pode assumir e as transições que ele pode fazer de estado para estado em função de algum evento
- Útil quando algum objeto apresenta comportamento diferente durante o sistema
- Evento acontece quando
 - Há recebimento de uma mensagem
 - Uma condição se torna verdadeira ou
 - Há a passagem de um período de tempo designado após um determinado evento.

Modelagem com UML

- A sintaxe do rótulo de uma transição é (deve-se ter pelo menos 1 das informações):

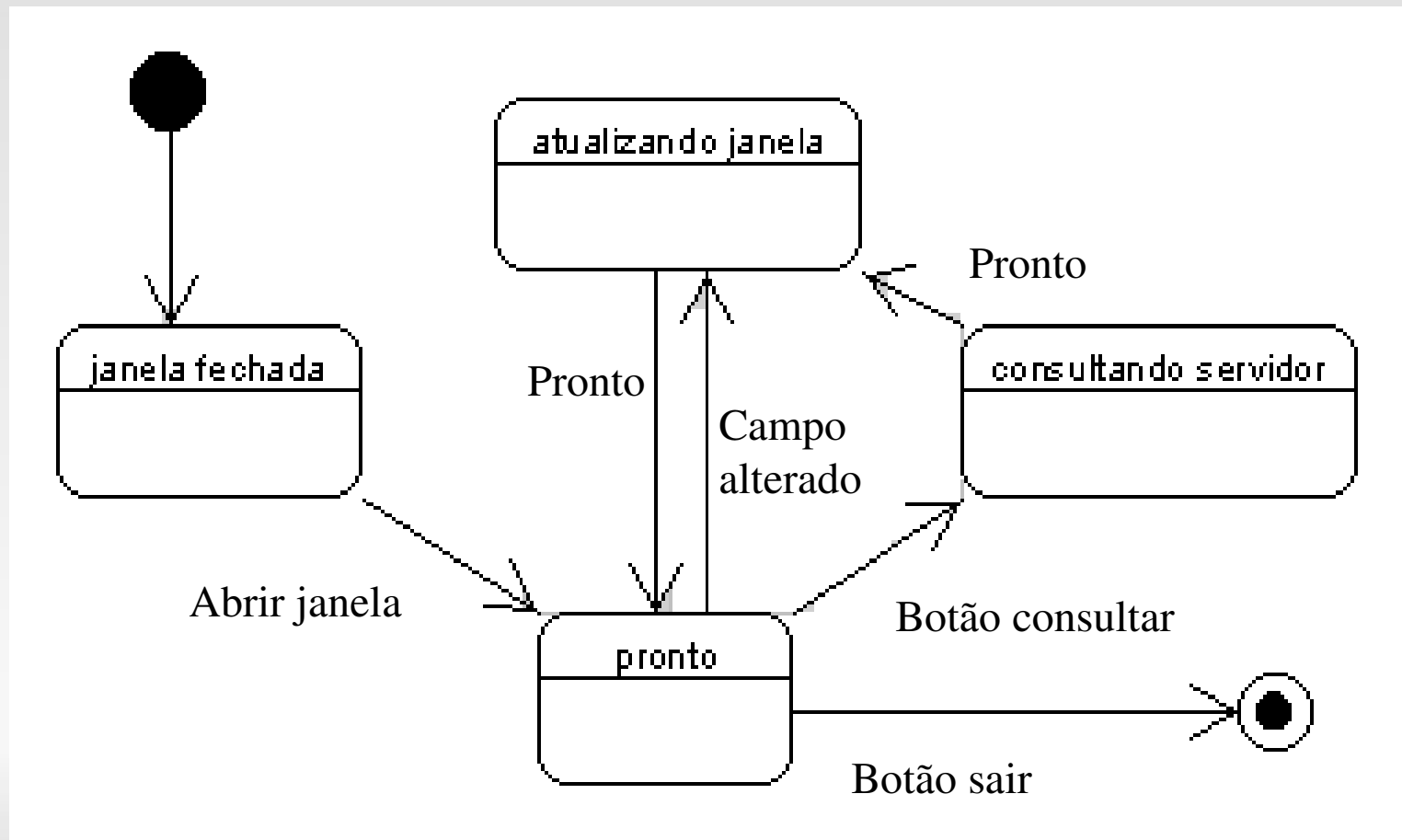
evento [condições] / ação

Exemplo



Modelagem com UML

- Diagrama de Estado representando Interface do Usuário



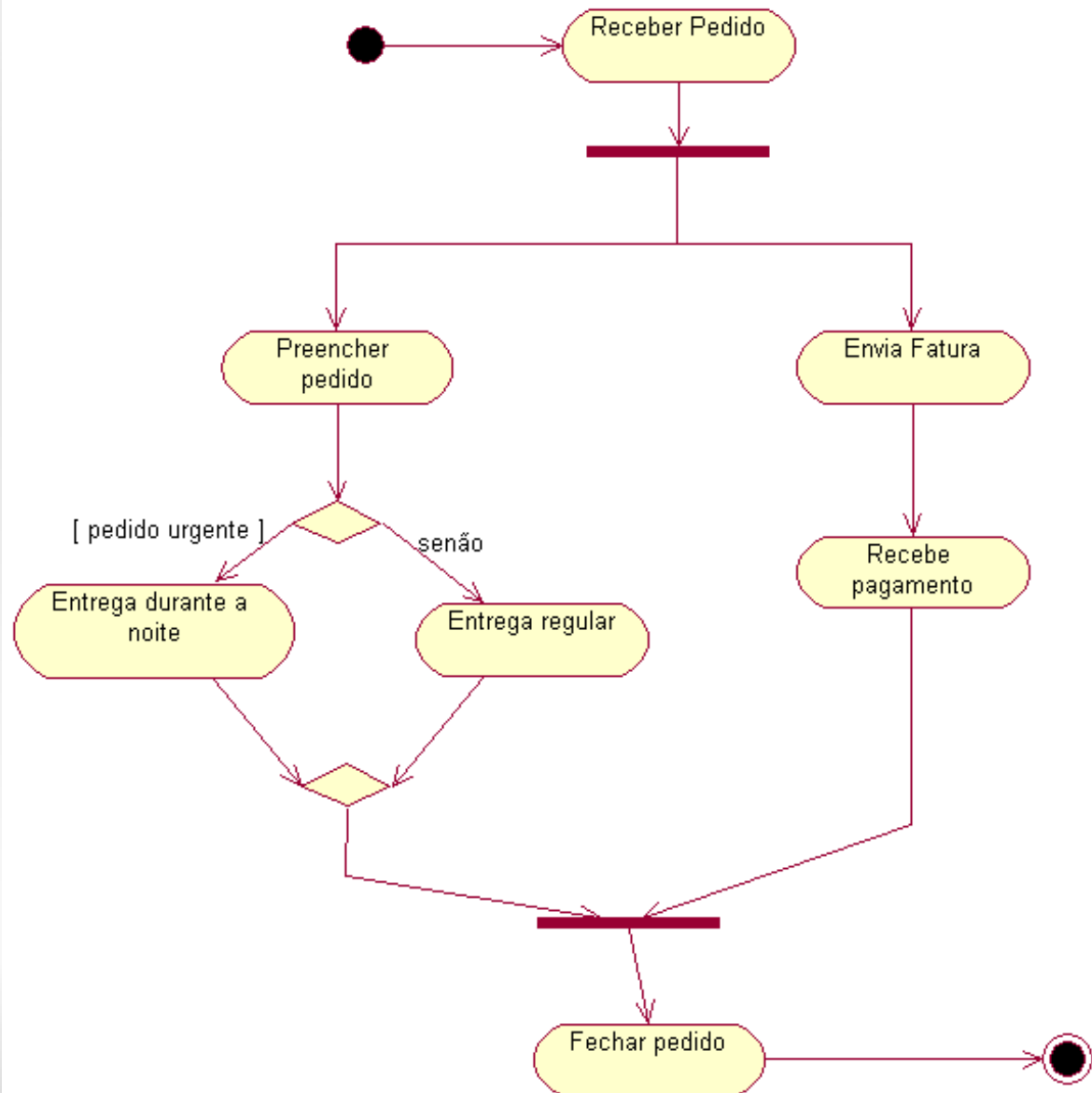
Modelagem com UML

■ 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

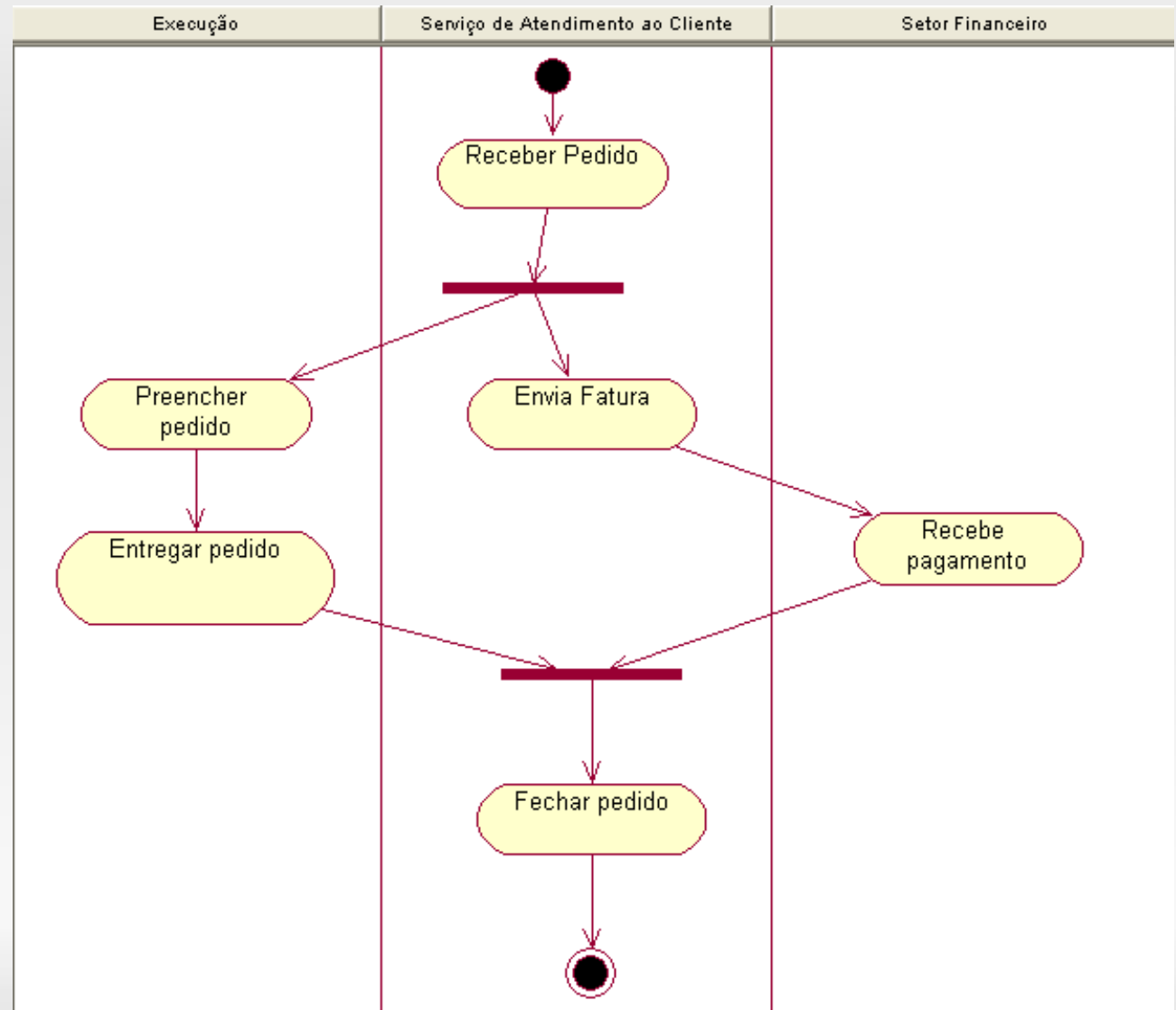
Modelagem com UML

Exemplo:



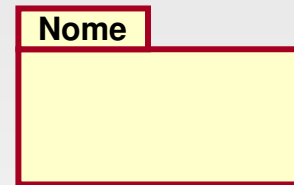
Modelagem com UML

Exemplo (2)

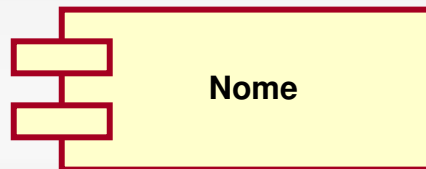


Modelagem com UML

- Organização dos artefatos do sistema
 - Pacotes
 - Organização das classes do sistema
 - Há dependência entre pacotes quando há dependência entre as classes que compõem o pacote



- Componentes



- Representam os componentes físicos do sistema

Bibliografia e Referências

■ Livros:

- 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

To Do

1. Construa uma página 'news-by-X' a ser disponibilizada na Web para apresentar notícias de jornais eletrônicos
 - Utilize folha de estilo apropriadamente na página construída, e apresente-se como sendo a pessoa X que indica / recomenda a leitura dessas informações e defina a navegação entre as fontes dos jornais originais.
 - atenção para a definição das informações sobre as notícias, tais como: título, data de publicação, origem
2. Desenvolva uma aplicação web que permita que os leitores de sua página 'news-by-X' possam adicionar comentários e palavras-chave, via browser, às notícias lidas. Esses adicionais devem ser devidamente identificados e disponibilizados para os posteriores leitores.