

---

# Modelos de Sistema

---

# Objetivos

---

- Explicar por que o contexto de um sistema deve ser modelado como parte do processo de RE
- Descrever modelagem de comportamento, modelagem de dados e modelagem de objetos
- Apresentar algumas das notações usadas na Unified Modeling Language (UML)
- Mostrar como *workbenches* CASE apóiam a modelagem de sistema

# Tópicos cobertos

---

- Modelos de contexto
- Modelos de comportamento
- Modelos de dados
- Modelos de objetos
- Métodos estruturados

# Modelagem de sistema

---

- A modelagem de sistema auxilia o analista a entender a funcionalidade do sistema e os modelos são usados para se comunicar com os clientes.
- Modelos diferentes apresentam o sistema a partir de perspectivas diferentes
  - Perspectiva externa que mostra o contexto ou ambiente do sistema;
  - Perspectiva comportamental que mostra o comportamento do sistema;
  - Perspectiva estrutural que mostra a arquitetura de sistema ou de dados.

# Tipos de modelos

---

- Modelo de fluxo de dados que mostra como os dados são processados em estágios diferentes.
- Modelo de composição que mostra como as entidades são compostas de outras entidades.
- Modelo de arquitetura que mostra os subsistemas principais.
- Modelo de classificação que mostra como as entidades têm características comuns.
- Modelo estímulo-resposta que mostra a reação do sistema aos eventos.

# Modelos de contexto

---

- Modelos de contexto são usados para ilustrar o contexto operacional de um sistema – eles mostram o que se encontra nos limites do sistema.
- Assuntos sociais e organizacionais podem afetar a decisão sobre onde posicionar os limites de sistema.
- Modelos de arquitetura mostram o sistema e seu relacionamento com outros sistemas.

# O contexto de um sistema caixa eletrônico

**Figura 8.1**

Contexto de um sistema de caixa eletrônico.



# Modelos de processo

---

- Modelos de processo mostram o processo geral e os processos que são apoiados pelo sistema.
- Modelos de fluxo de dados podem ser usados para mostrar o processamento e o fluxo de informações de um processo para um outro.





# Modelos de comportamento

---

- Modelos de comportamento são usados para descrever o comportamento geral de um sistema.
- Dois tipos de modelo de comportamento são:
  - Modelos de fluxo de dados que mostram como os dados são processados conforme se movem através do sistema;
  - Modelos de máquina de estado que mostram as respostas do sistema aos eventos.
- Estes modelos mostram perspectivas diferentes e, dessa forma, ambos são requeridos para descrever o comportamento do sistema.

# Modelos de fluxo de dados

---

- Diagramas de fluxo de dados (DFD) podem ser usados para modelar o processamento de dados do sistema.
- Esses modelos mostram os passos do processamento quando os dados fluem através de um sistema.
- Os DFDs são uma parte intrínseca de muitos métodos de análise.
- Notação simples e intuitiva para que os clientes possam entender.
- Mostram o processamento fim-a-fim de dados.

# DFD de processamento de pedido



**Figura 8.3**

Diagrama de fluxo de dados do processamento de um pedido.

# Diagramas de fluxo de dados

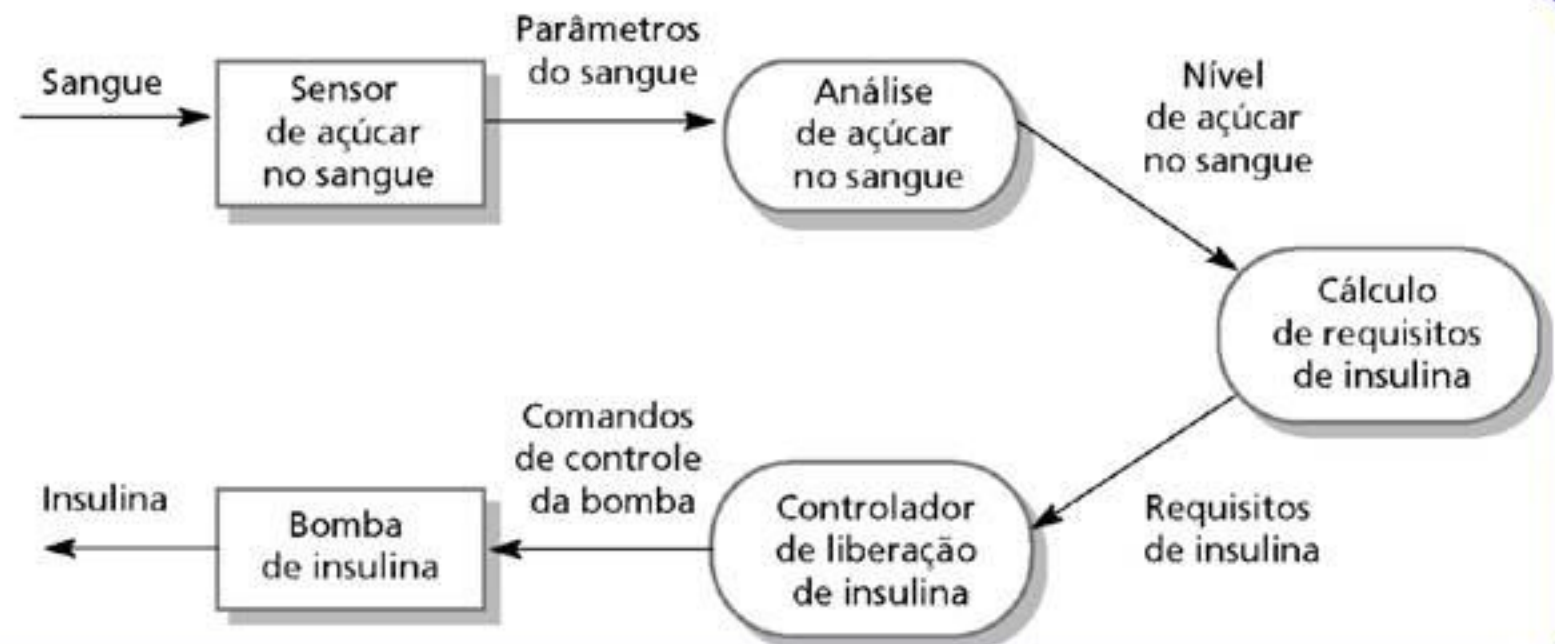
---

- Os DFDs modelam o sistema a partir de uma perspectiva funcional.
- Acompanhamento e documentação de como os dados associados um processo são úteis para desenvolver uma compreensão geral do sistema.
- Diagramas de fluxo de dados podem também ser usados na apresentação de troca de dados entre um sistema e outros em seu ambiente.

# DFD de bomba de insulina

**Figura 8.4**

Diagrama de fluxo de dados de uma bomba de insulina.



# Modelos de máquina de estado

---

- Modelam o comportamento do sistema em resposta aos eventos externos e internos.
- Mostram as respostas do sistema aos estímulos e, assim, são freqüentemente usados para modelagem de sistemas de tempo real.
- Modelos de máquina de estado mostram os estados do sistema como nós, e os eventos como arcos entre estes nós. Quando um evento ocorre, o sistema muda de um estado para um outro.
- Statecharts são uma parte integral da UML e são usados para representar os modelos de máquina de estados.

# Statecharts

---

- Permitem a decomposição de um modelo em submodelos (ver o seguinte slide)
- Uma breve descrição das ações é incluída seguindo o 'faça' em cada estado.
- Podem ser complementados por tabelas que descrevem os estados e os estímulos.



# Modelo de forno de microondas

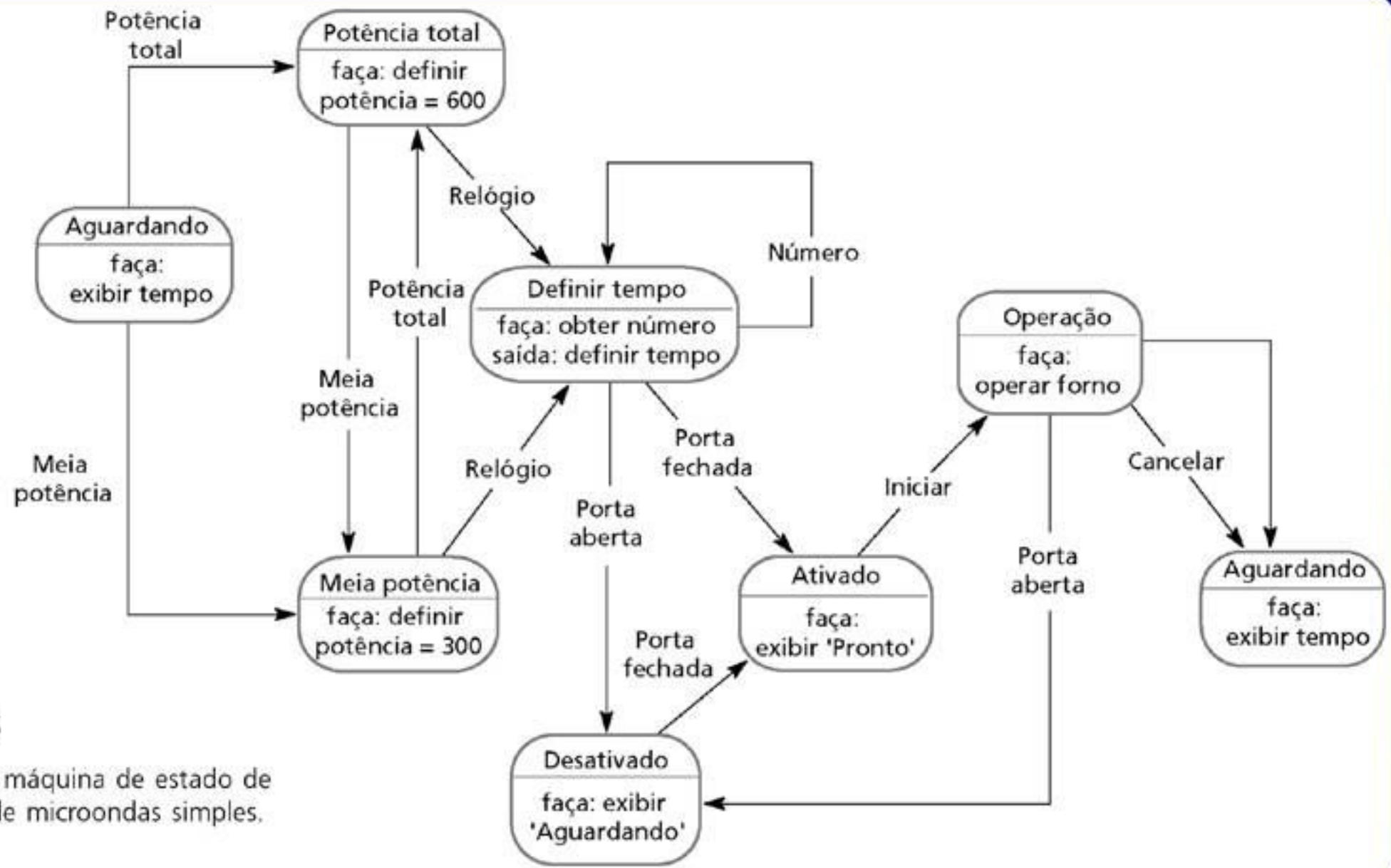


Figura 8.5

Modelo de máquina de estado de um forno de microondas simples.

# Descrição de estados de forno de microondas

**Tabela 8.1** Descrição de estados e estímulos do forno de microondas

Estado	Descrição
Aguardando	O forno está aguardando uma entrada. O display apresenta a hora atual.
Meia potência	A potência do forno é definida como 300 watts. O display apresenta 'Meia potência'.
Potência total	A potência do forno é definida como 600 watts. O display apresenta 'Potência total'.
Definir tempo	O tempo de cozimento é definido como o valor fornecido pelo usuário. O display apresenta o tempo de cozimento selecionado e é atualizado quando o tempo for definido.
Desativado	A operação do forno é desativada por questões de segurança. A luz interna do forno é ligada. O display apresenta 'Não está pronto'.
Ativado	A operação do forno é ativada. A luz interna do forno é apagada. O display apresenta 'Pronto para cozinhar'.
Operação	Forno em operação. A luz interna do forno é ligada. O display apresenta a contagem regressiva do relógio. Ao término do cozimento, o alarme soa por 5 segundos. A luz do forno é ligada. O display apresenta 'Cozimento concluído' enquanto o alarme soa.

Estímulo	Descrição
Meia potência	O usuário pressionou o botão de meia potência.
Potência total	O usuário pressionou o botão de potência total.
Relógio	O usuário pressionou um dos botões do relógio.
Número	O usuário pressionou uma tecla numérica.
Porta aberta	A trava da porta do forno não está fechada.
Porta fechada	A trava da porta do forno está fechada.
Iniciar	O usuário pressionou o botão Iniciar.
Cancelar	O usuário pressionou o botão Cancelar.

© 2007 by Pearson Education

# Operação de forno de microondas

**Figura 8.6**

Operação do forno de microondas.



# Modelos semânticos de dados

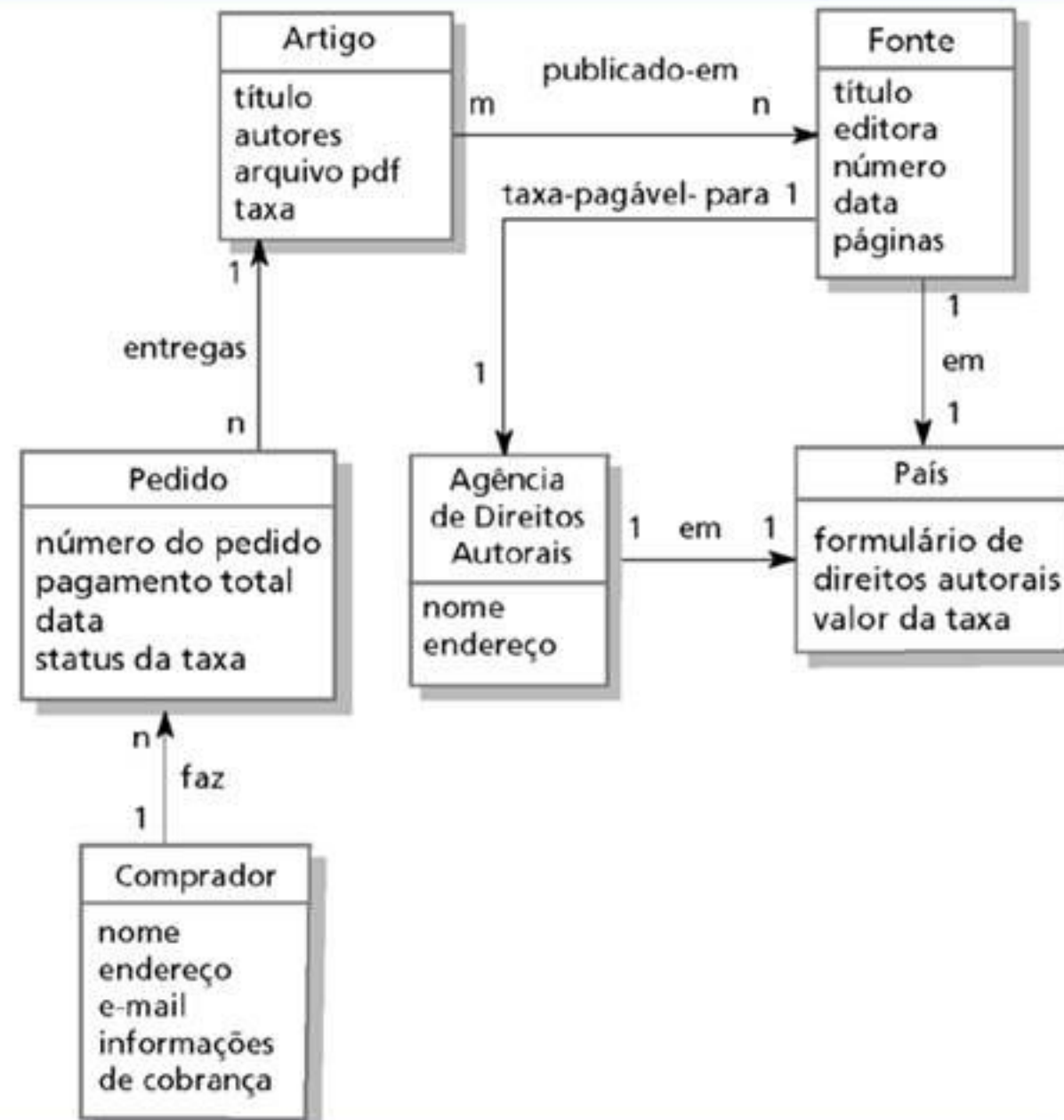
---

- São usados para descrever a estrutura lógica dos dados processados pelo sistema.
- Um modelo de Entidade-Relacionamento-Atributo (ERA – Entity-Relation-Attribute) define as entidades no sistema, as relações entre elas e os atributos de entidades.
- Amplamente usados no projeto de banco de dados. Podem ser prontamente implementados usando banco de dados relacionais.
- Nenhuma notação específica é fornecida na UML, contudo, objetos e associações podem ser usados.

# Modelo semântico de dados de biblioteca

**Figura 8.7**

Modelo semântico de dados para o sistema LIBSYS.



# Dicionários de dados

---

- Dicionários de dados são listas de todos os nomes usados nos modelos de sistema. Descrições das entidades, dos relacionamentos e dos atributos também são incluídos.
- Vantagens
  - Apóia o gerenciamento de nomes e evita duplicação;
  - Repositório de informações da organização ligando análise, projeto, implementação e evolução;
- Muitos workbenches CASE apóiam os dicionários de dados.

# Entradas de dicionário de dados

**Tabela 8.2** Exemplos de entradas do dicionário de dados

Nome	Descrição	Tipo	Data
Artigo	Detalhes do artigo publicado que pode ser pedido pelas pessoas que usam o LIBSYS.	Entidade	30.12.2002
Autores	Nomes dos autores do artigo que podem receber uma parte da taxa.	Atributo	30.12.2002
Comprador	A pessoa ou organização que pede uma cópia do artigo.	Entidade	30.12.2002
Taxa-pagável-para	Relacionamento 1:1 entre Artigo e Agência de Direitos Autorais à qual deve ser paga a taxa de direitos autorais.	Relação	29.12.2002
Endereço (comprador)	Endereço do comprador. É usado para quaisquer informações de cobrança necessárias.	Atributo	31.12.2002

# Modelos de objetos

---

- O modelos de objetos descrevem o sistema em termos de classes de objeto e suas associações.
- Uma classe de objeto é uma abstração de um conjunto de objetos com atributos comuns e os serviços (operações) fornecidos por cada objeto.
- Vários modelos de objetos podem ser produzidos
  - Modelos de herança;
  - Modelos de agregação;
  - Modelos de interação.



# Modelos de objetos

---

- Maneiras naturais de refletir as entidades do mundo real manipulados pelo sistema.
- As entidades mais abstratas são mais difíceis de modelar usando esta abordagem.
- A identificação da classe de objeto é reconhecida como um processo difícil que requer um entendimento profundo do domínio de aplicação.
- Classes de objetos que refletem entidades de domínio são reusáveis entre sistemas.

# Modelos de herança

---

- Organizam as classes de objeto de domínio em uma hierarquia.
- As classes do topo da hierarquia refletem as características comuns a todas as classes.
- As classes de objeto herdam seus atributos e serviços a partir de um ou mais superclasses. Estas podem ser, então, especializadas quando necessário.
- O projeto de hierarquia de classes pode ser um processo difícil se as duplicações nos diferentes ramos não forem evitadas.

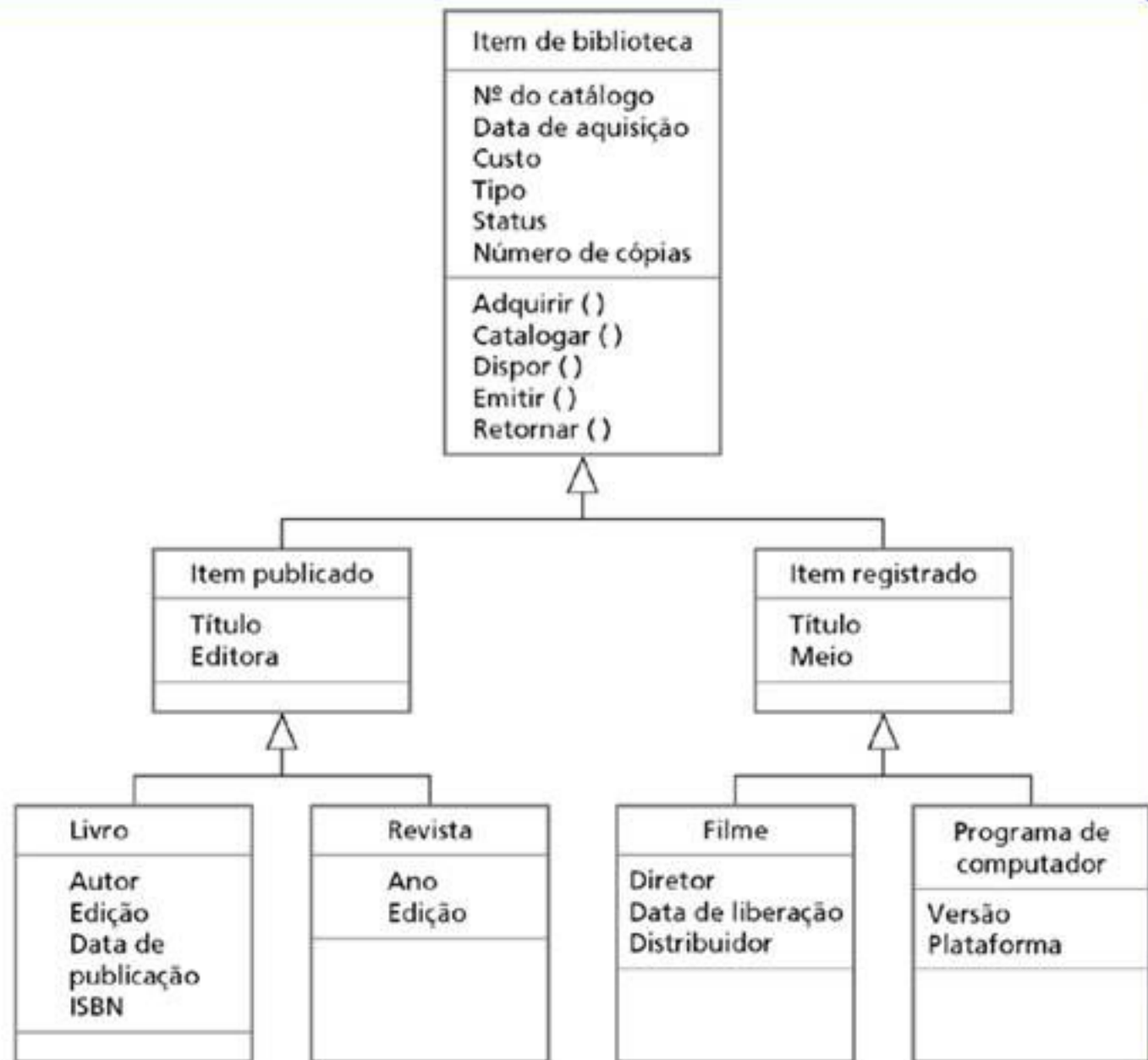
# Modelos de objetos e a UML

- A UML é uma representação padrão criada pelos desenvolvedores de métodos de análise e projeto orientados a objetos amplamente usados.
- Ela tornou-se um padrão efetivo para a modelagem orientada a objetos.
- Notação
  - As classes de objetos são retângulos com o nome na parte superior, atributos na seção intermediária e operações na seção inferior;
  - OS relacionamentos entre classes de objetos (conhecidos como associações) são mostrados como linhas que ligam os objetos;
  - O termo herança é referido como generalização e mostrada no 'sentido ascendente' ao invés de no 'sentido descendente' na hierarquia.

# Hierarquia de classes de biblioteca

**Figura 8.8**

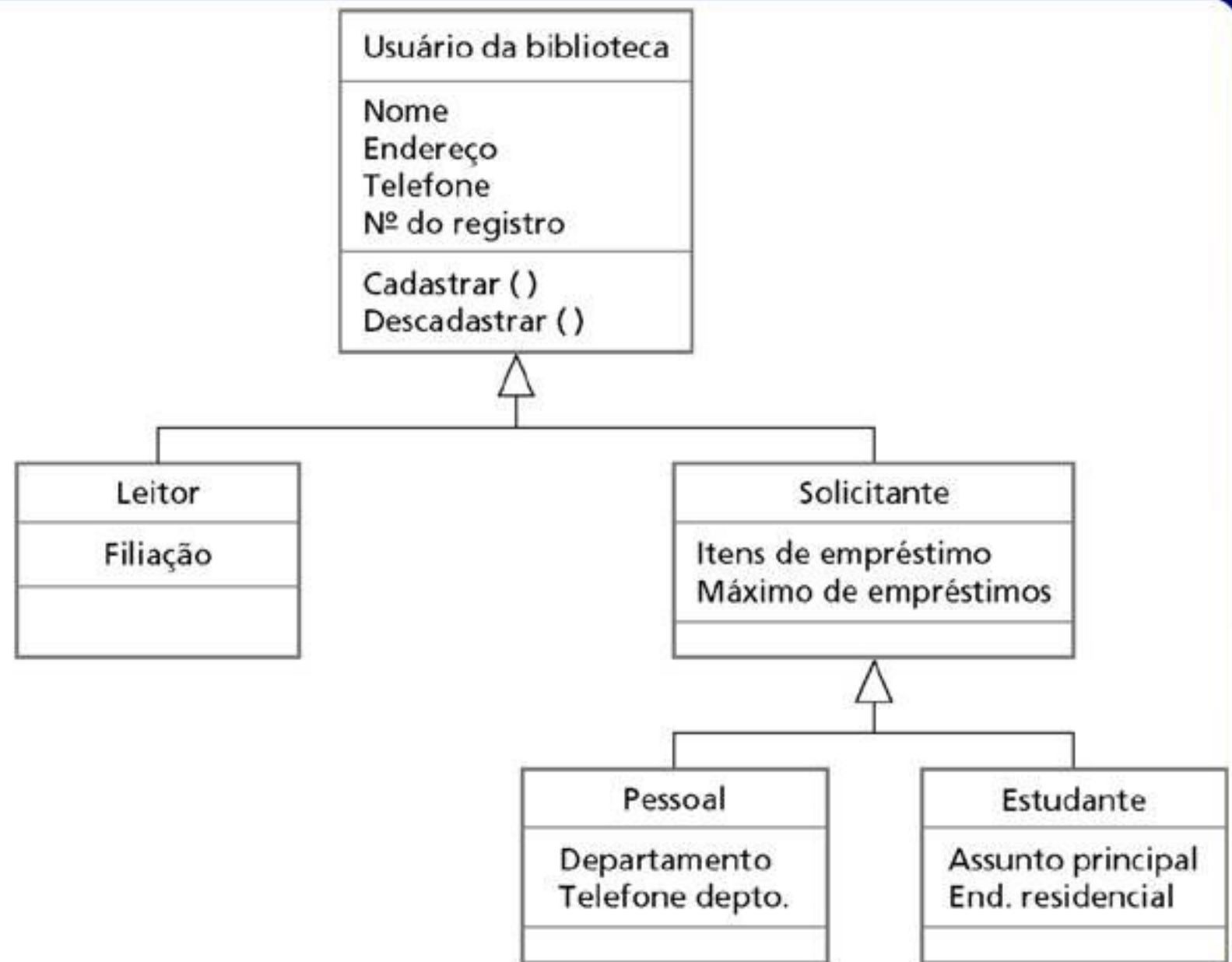
Parte da hierarquia de classes de uma biblioteca.



# Hierarquia de classes de usuário

**Figura 8.9**

Hierarquia de classes de usuário.



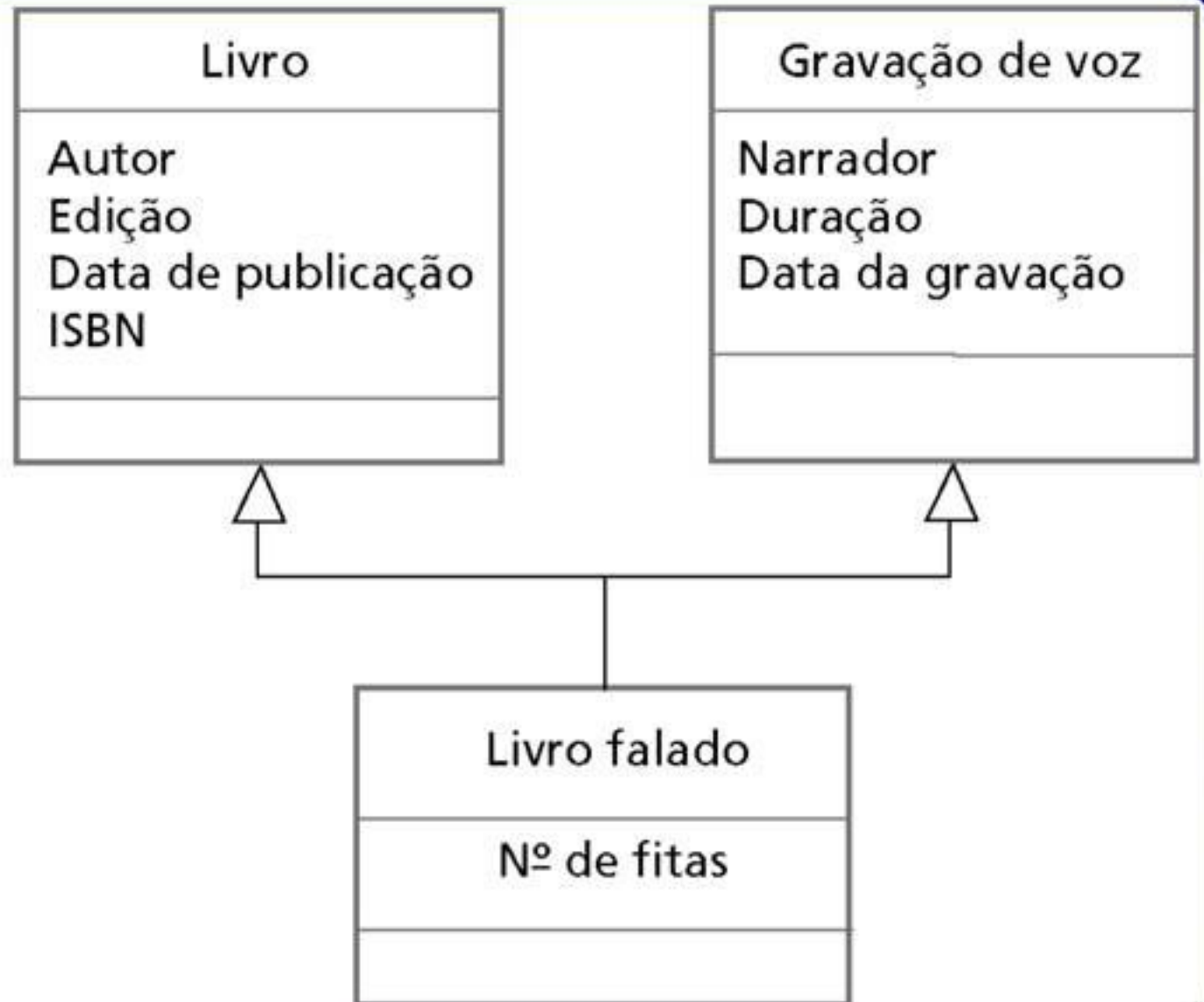
# Herança múltipla

- Ao invés de herdar os atributos e serviços a partir de uma única classe-pai, um sistema que apóia a herança múltipla permite que classes de objeto herdem a partir de superclasses.
- Isso pode levar a conflitos onde atributos/serviços com o mesmo nome em superclasses diferentes têm semânticas diferentes.
- A herança múltipla torna a reorganização de hierarquia de classes mais complexa.

# Herança múltipla

**Figura 8.10**

Herança múltipla.



# Aggregação de objetos

---

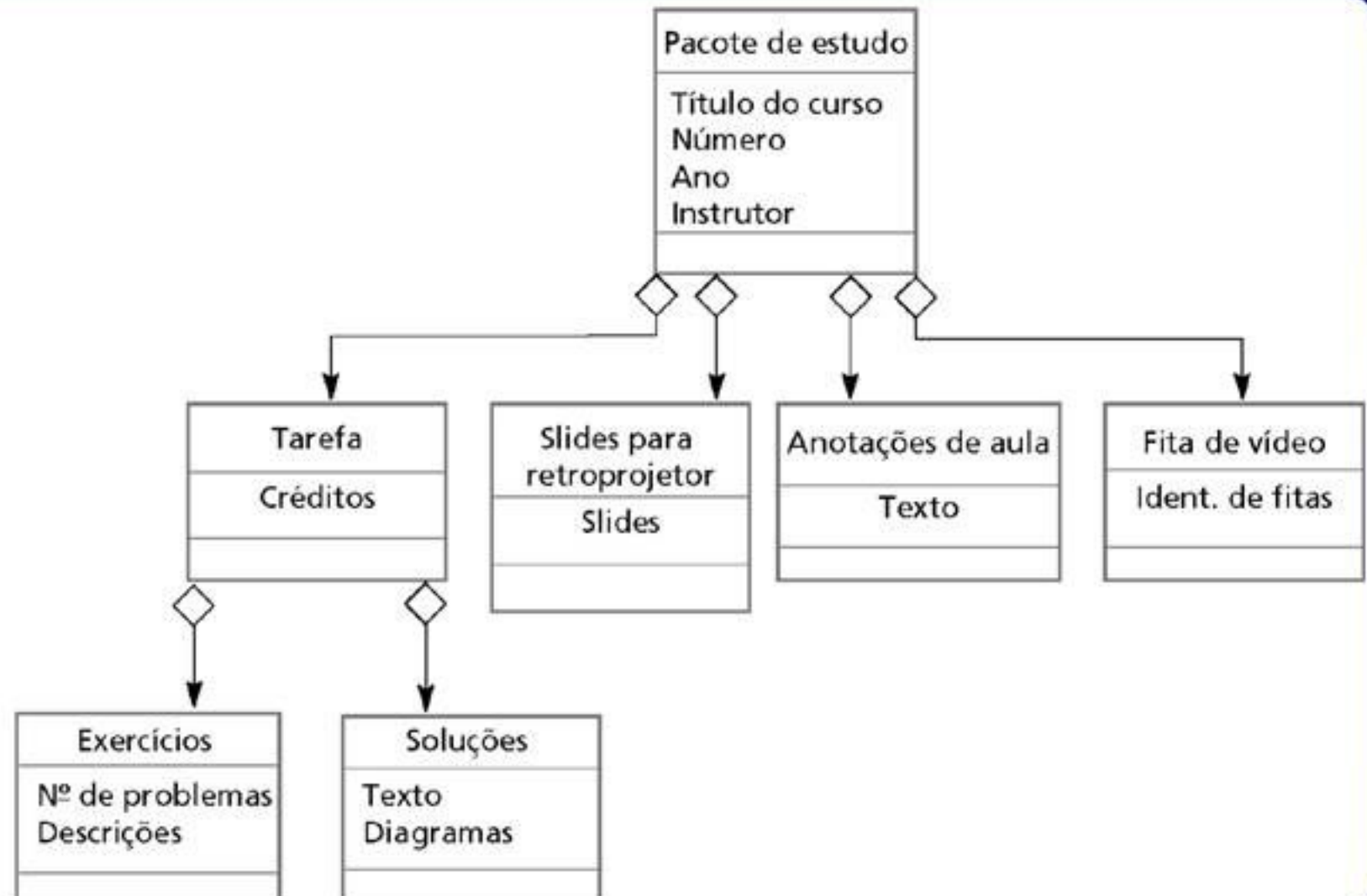
- Um modelo de agregação mostra como as classes que são coleções são compostas de outras classes.
- Modelos de agregação são similares ao relacionamento 'parte de' nos modelos semânticos de dados.



# Agregação de objetos

**Figura 8.11**

Objeto agregado que representa um curso.



# Modelagem de comportamento de objetos

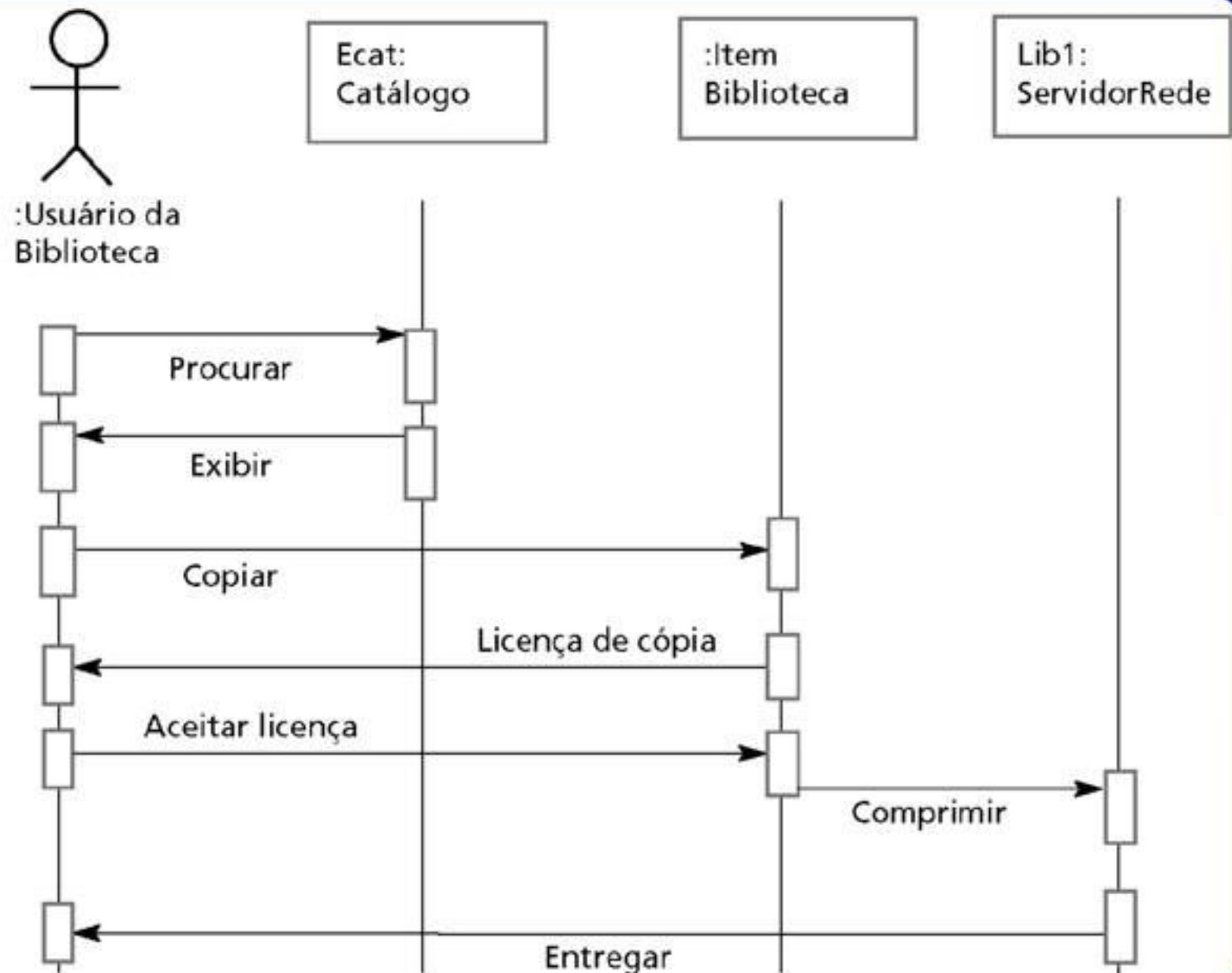
---

- Um modelo de comportamento mostra as interações entre objetos que produzem algum comportamento de sistema particular que é especificado como um caso de uso.
- Diagramas de seqüência (ou diagramas de colaboração) são usados na UML para modelar a interação entre objetos.

# Cópia de itens eletrônicos

**Figura 8.12**

Cópia de itens eletrônicos.



# Métodos estruturados

---

- Os métodos estruturados incorporam a modelagem de sistema como uma parte inerente do método.
- Os métodos definem um conjunto de modelos, um processo para derivá-los, e regras e diretrizes que se aplicam a esses modelos.
- As ferramentas CASE apóiam a modelagem do sistema como parte de um método estruturado.

# Pontos fracos de métodos

---

- Não modelam requisitos não funcionais de sistema.
- Geralmente não incluem informações sobre se um método é adequado para um determinado problema.
- Podem produzir um excesso de documentação.
- Os modelos de sistema são, às vezes, muito detalhados e difíceis para os usuários compreenderem.

# Workbenches CASE

---

- É um conjunto de ferramentas coerentes, projetado para apoiar as atividades relacionadas de processo de software, tais como, análise, projeto ou teste.
- *Workbenches* de análise e de projeto apóiam a modelagem de sistema durante ambos, engenharia de requisitos e projeto de sistema.
- Esses workbenches podem apoiar um método de projeto específico, ou podem fornecer apoio para a criação de vários tipos de modelos de sistema diferentes.

# Um *workbench* de análise e projeto

**Figura 8.13**

Componentes de uma ferramenta CASE para apoio de método estruturado



# Componentes de *workbench* de análise

---

- Editores de diagramas
- Ferramentas de análise e verificação de projeto
- Repositório e linguagem de consulta associada
- Dicionário de dados
- Ferramentas de definição e geração de relatórios
- Ferramentas de definição de formulários
- Recursos de importação/exportação
- Ferramentas de geração de códigos



# Pontos-chave

---

- Um modelo é uma visão abstrata do sistema. Tipos complementares de modelo fornecem diversas informações sobre o sistema.
- Os modelos de contexto mostram a posição de um sistema no seu ambiente com outros sistemas e processos.
- Os modelos de fluxo de dados podem ser usados para modelar o processamento de dados em um sistema.
- Os modelos de máquinas de estados modelam o comportamento do sistema em resposta a eventos internos ou externos.

# Pontos-chave

---

- Os modelos semânticos de dados descrevem a estrutura lógica que é importada ou exportada pelos sistemas.
- Os modelos de objetos descrevem as entidades lógicas do sistema, sua classificação e agregação.
- Os modelos de seqüência mostram as interações entre os agentes e os objetos de sistema usados.
- Os métodos estruturados fornecem um *framework* para desenvolvimento de modelos de sistema.