

SC-301 – Programação Orientada a Objetos II

Introdução à

UML
Parte 01

Prof. Dr. Fábio Fagundes Silveira

fsilveira@unifesp.br

<http://fabiosilveira.net>

UNIFESP – Universidade Federal de São Paulo

março/2010



Créditos

- Grade parte dos slides seguintes foram preparados/elaborados pelos Professores:
 - Prof. Dr. Jaelson Freire Brelaz de Castro; e
 - Prof. Dr. Ulrich Schiel



Bibliografia

- *The Unified Modelling Language - User Guide* (G. Booch, J. Rumbaugh, I. Jacobson) - Addison Wesley
- *The Unified Modelling Language - Reference Manual* (J. Rumbaugh, I. jacobson, G. Booch) - Addison Wesley
- *The Unified Software Development Process* (I. Jacobson, G. Booch, R. Rumbaugh) - Addison Wesley
- *UML Distilled* (Martin Fowler) - Addison Wesley

UML

UML



Conteúdo

- Introdução à UML
- Conceitos Gerais
- Apresentação dos diagramas da UML
- Novos diagramas da UML 2.0



Criadores da UML

- James Rumbaugh - Object Modeling Technique (OMT)
- Grady Booch - Booch Method
- Ivar Jacobson - Objectory (OOSE) Process

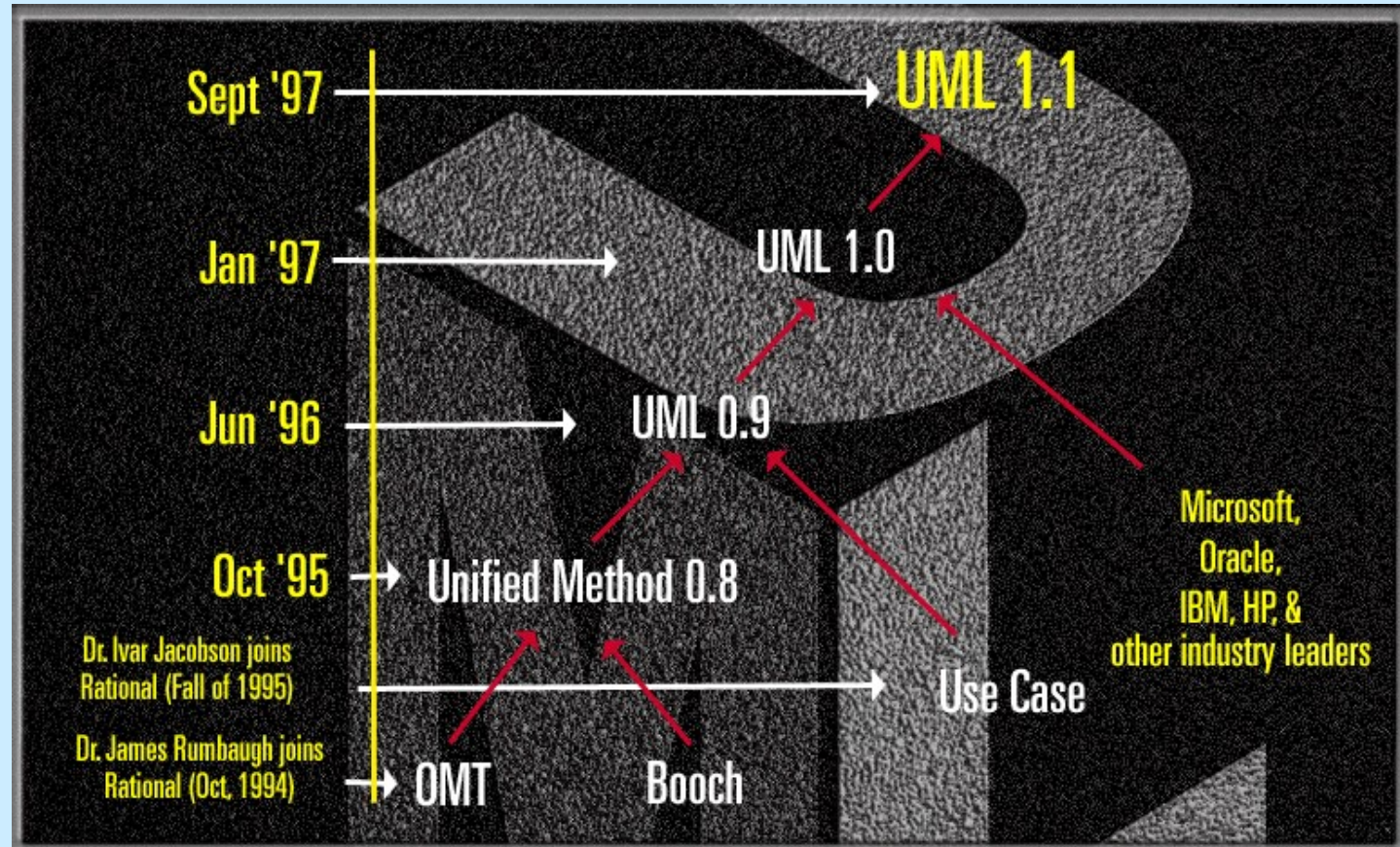


Por que os 3 autores resolveram criar a UML?

- Cada autor adotava idéias dos métodos dos outros, então, evoluindo juntos produziram melhorias
- A unificação dos 3 métodos trariam estabilidade para o mercado



História e Padronização





Tradicional e Moderna





Usos de UML

A UML é uma linguagem de modelagem para:

- Visualização
- Especificação
- Construção
- Documentação
- Comunicação



Elementos Essenciais

**Elementos
Estruturais**

**Elementos
Comportamentais**

**Elementos de
Agrupamento**

**Elementos de
Anotação**



Elementos Estruturais

São as partes estáticas de um modelo, representando elementos que são ou conceituais ou físicos.

Exemplos:

- **Classe**
- **Interface**
- **Use Cases**
- **Componente**
- **Nó**



Elementos Comportamentais

São as partes dinâmicas dos modelos da UML.

Exemplos:

- **Interação** - especifica um conjunto de mensagens trocadas entre objetos
- **Máquina de Estado** - especifica seqüências de estados de um objeto



Elementos de Agrupamento

São partes organizacionais dos modelos da UML.

Exemplo:

- **Pacotes** - mecanismo para organização de elementos dentro de grupos



Elementos de Anotação

São partes explicativas dos modelos da UML. São comentários que você aplica para descrever, iluminar e remarcar elementos no modelo.

Exemplo:

- **Nota** - símbolo contendo restrições ou comentários que são melhor expressadas em textos



Diagramas

São representações gráficas de um conjunto de elementos. São desenhados para visualizar um sistema de diferentes perspectivas.

A UML 2.0 possui 13 diagramas:

- Use Case
- Classe
- Objetos
- Seqüência
- Comunicação (Colaboração)
- Máquina de Estados (Diagrama de Estados)
- Atividade
- Componentes
- Implantação
- Pacotes
- *Estrutura Composta*
- *Interação Geral*
- *Diagrama de Tempo*



Diagrama Use Cases

- São especialmente importantes na organização e modelagem das principais funcionalidades de um sistema
- Use Case é a especificação de sequências de ações atender a uma funcionalidade do sistema, interagindo com seus agentes



Diagrama de Use cases

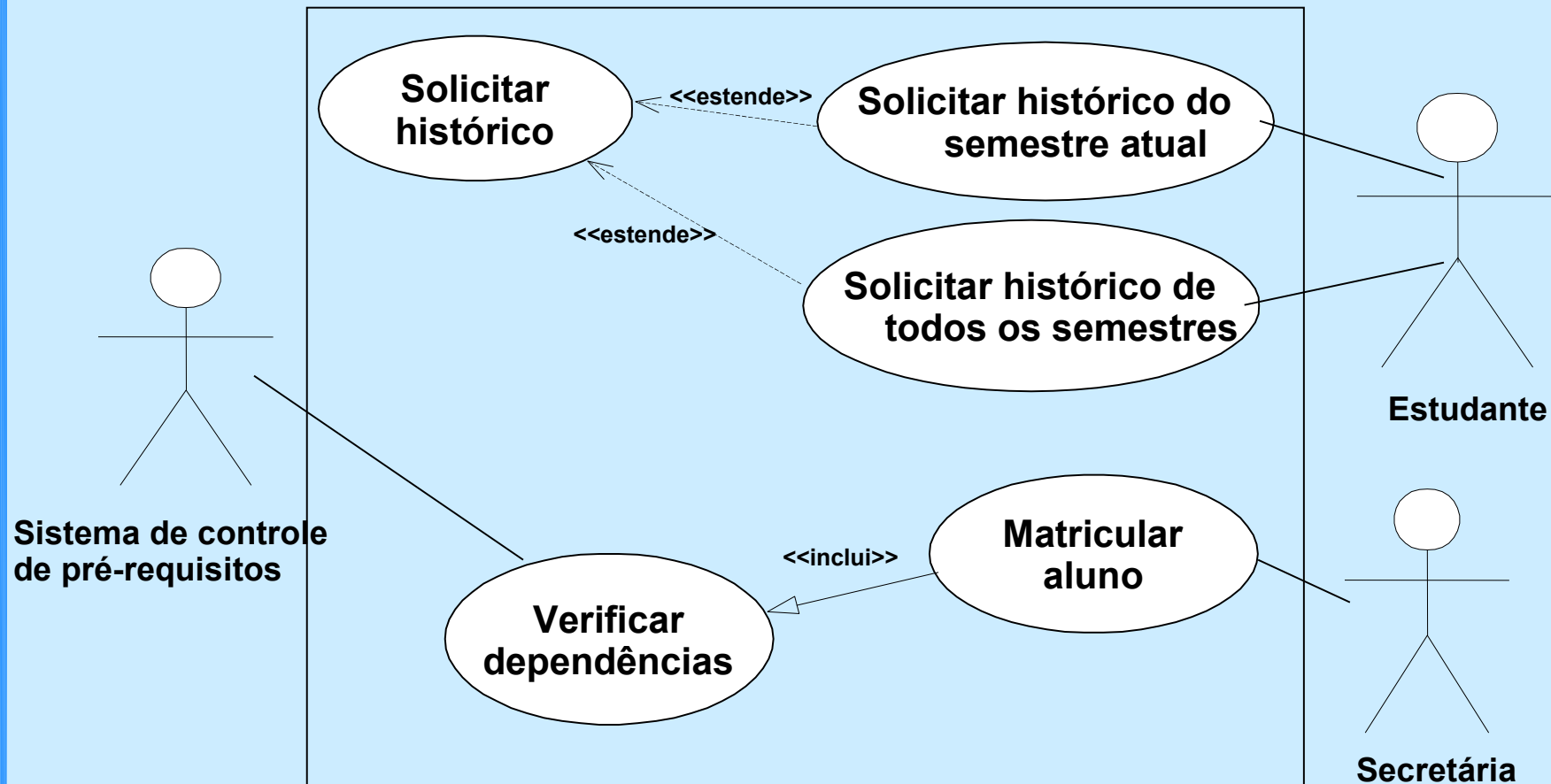




Diagrama de Classe

- Os diagramas de classes são os principais diagramas estruturais da UML
- Diagramas de classe mostram classes, interfaces e seus relacionamentos
- As classes especificam a estrutura e o comportamento dos objetos, que são instâncias de classes



Diagrama de Classe

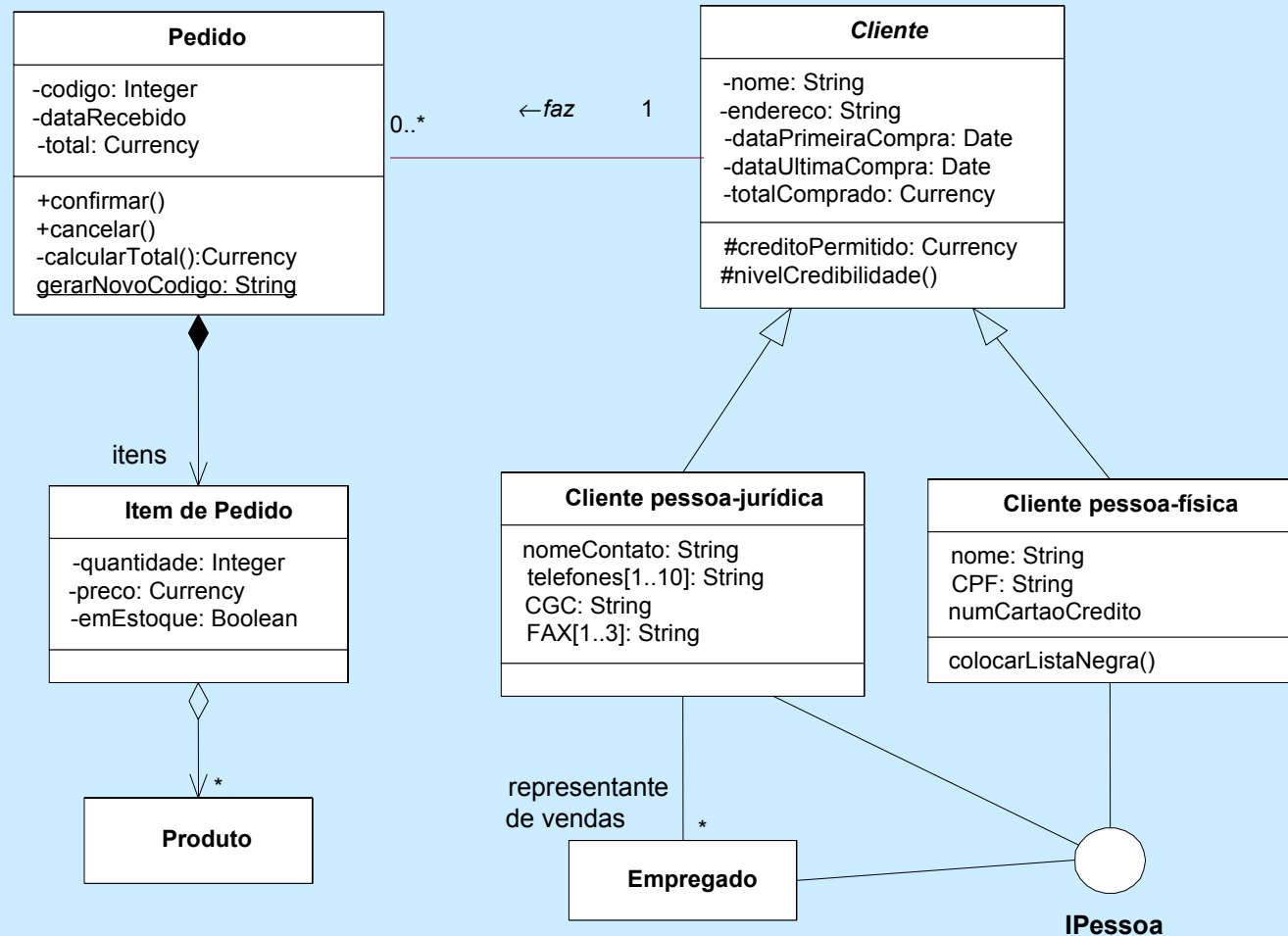




Diagrama de Objetos

- Mostram objetos e seus relacionamentos
- Representam instâncias estáticas de elementos dos diagramas de classes
- Os diagramas de objetos são úteis para a modelagem de estruturas de dados complexas



Diagrama de Objetos

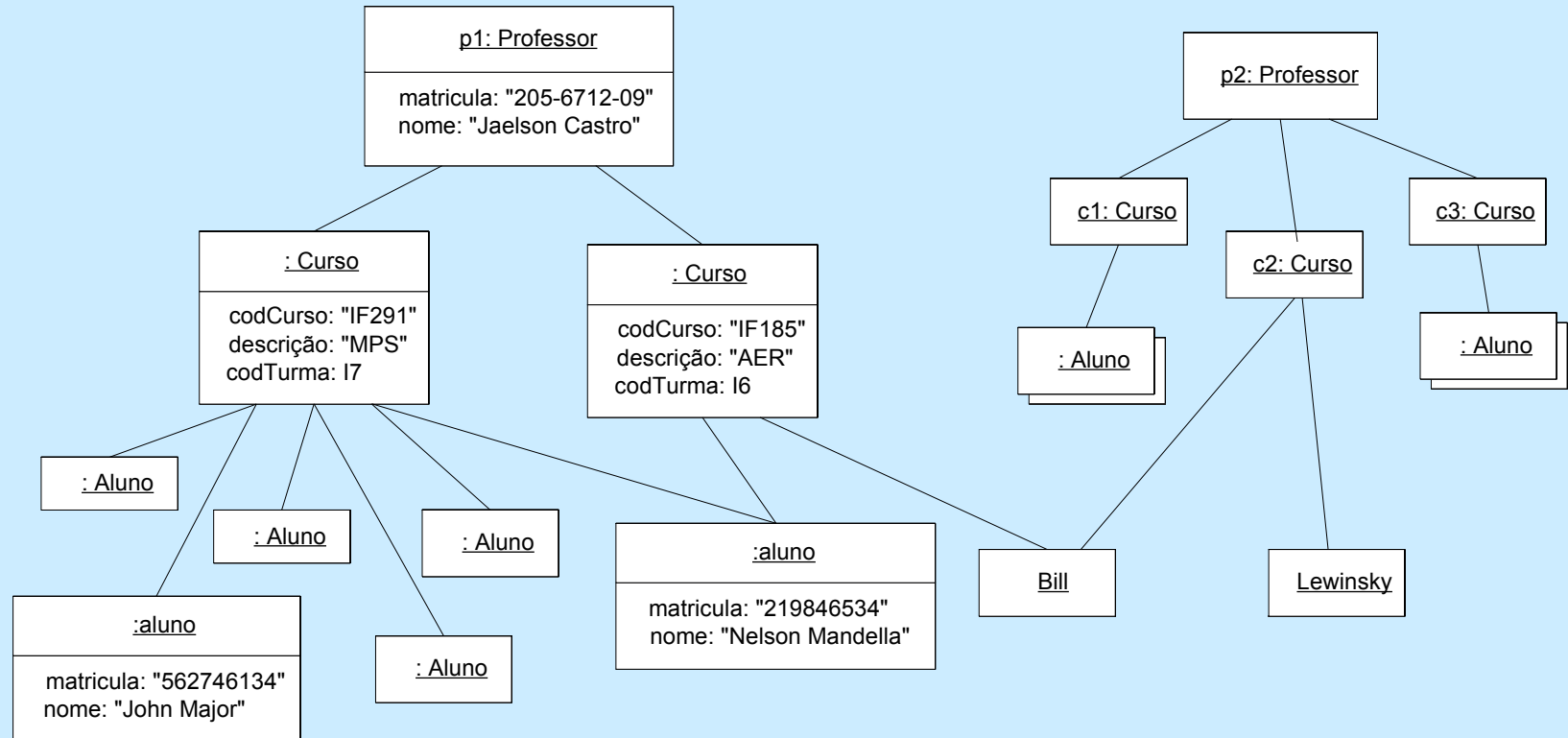
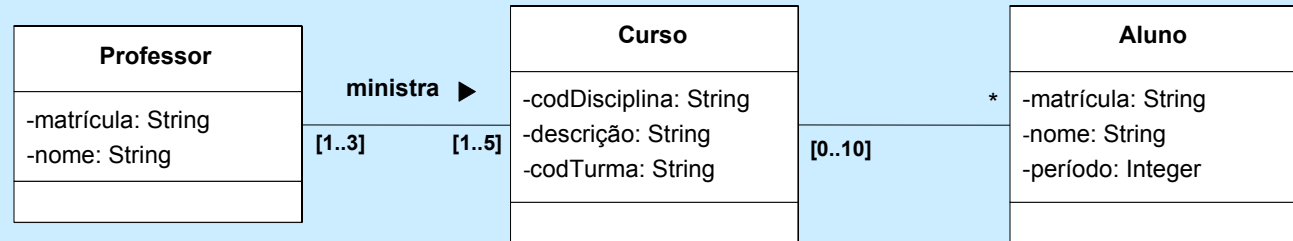




Diagrama de Seqüência

- Mostra um conjunto de objetos, seus relacionamentos e as mensagens que podem ser enviadas entre eles



Diagrama de Seqüência

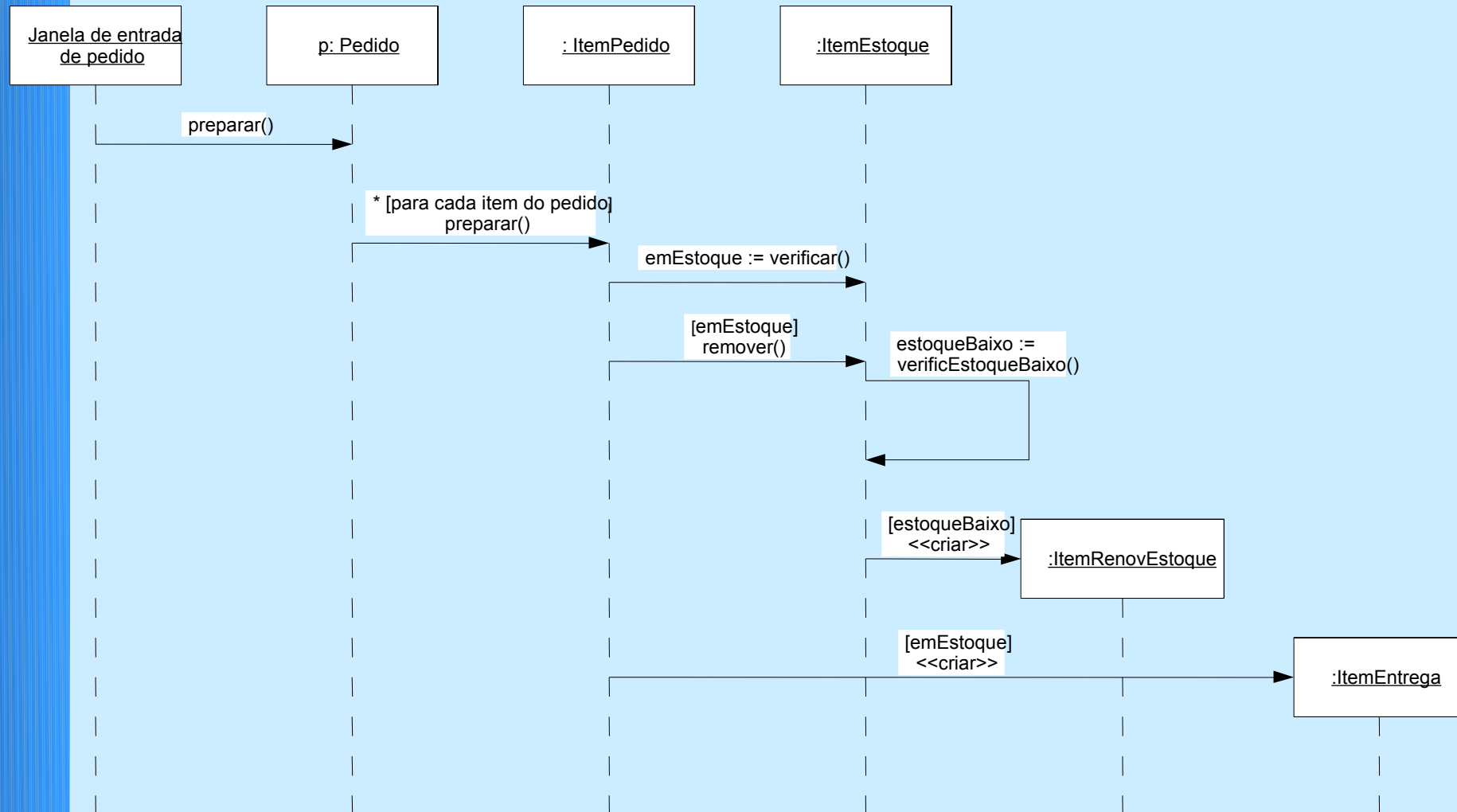




Diagrama de Comunicação

- Mostra um conjunto de objetos, seus relacionamentos e as mensagens que enfatizam a organização dos objetos que trocam mensagens



Diagrama de Comunicação

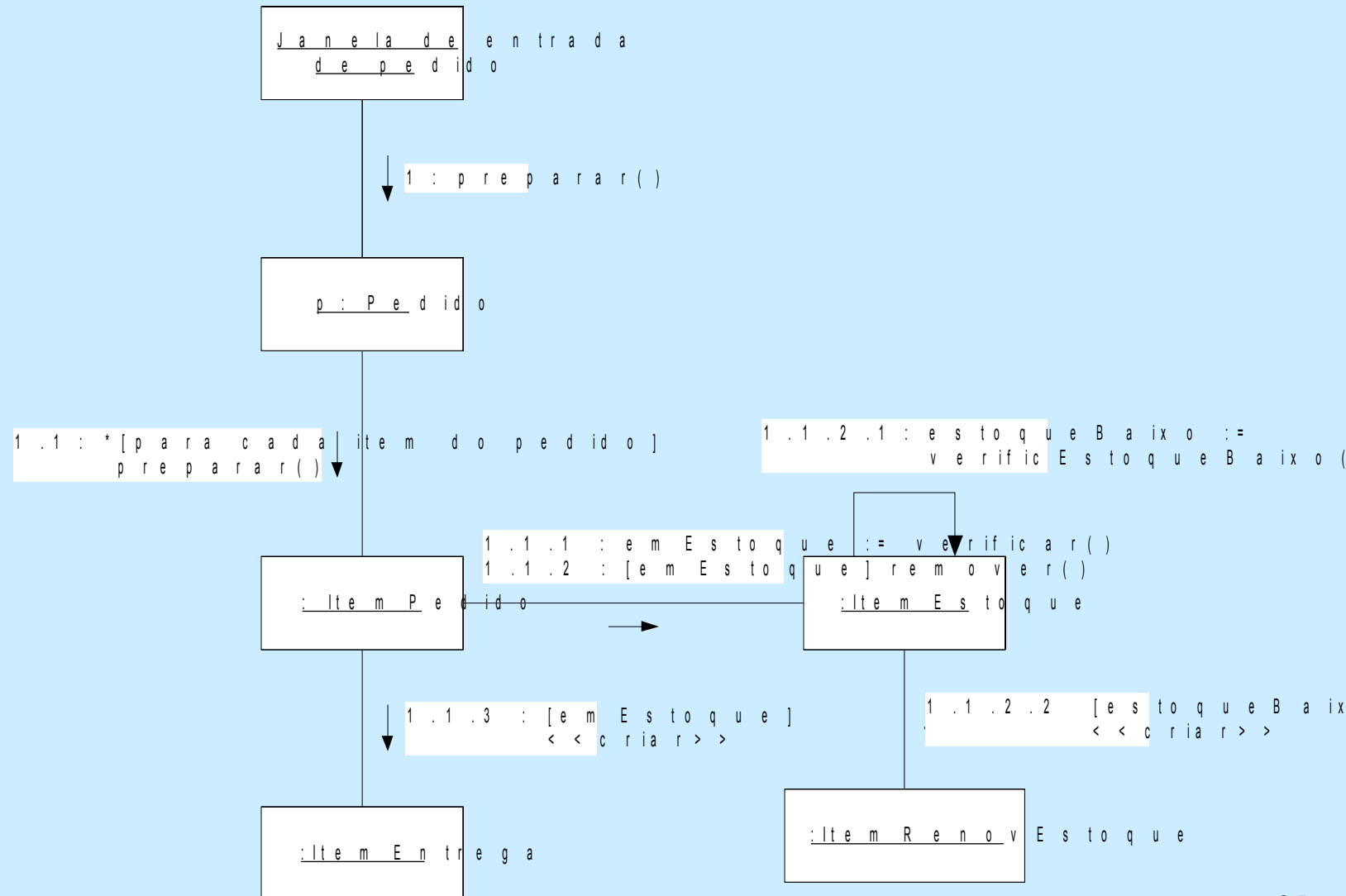




Diagrama de Máquina de Estados

- Mostra uma máquina contendo estados, transições, eventos e atividades
- Estes diagramas são usados para modelar o comportamento de objetos (com comportamento complexo)
- Nestes diagramas são modelados os estados em que um objeto pode estar e os eventos que fazem o objeto passar de um estado para outro



Diagrama de Máquina de Estados

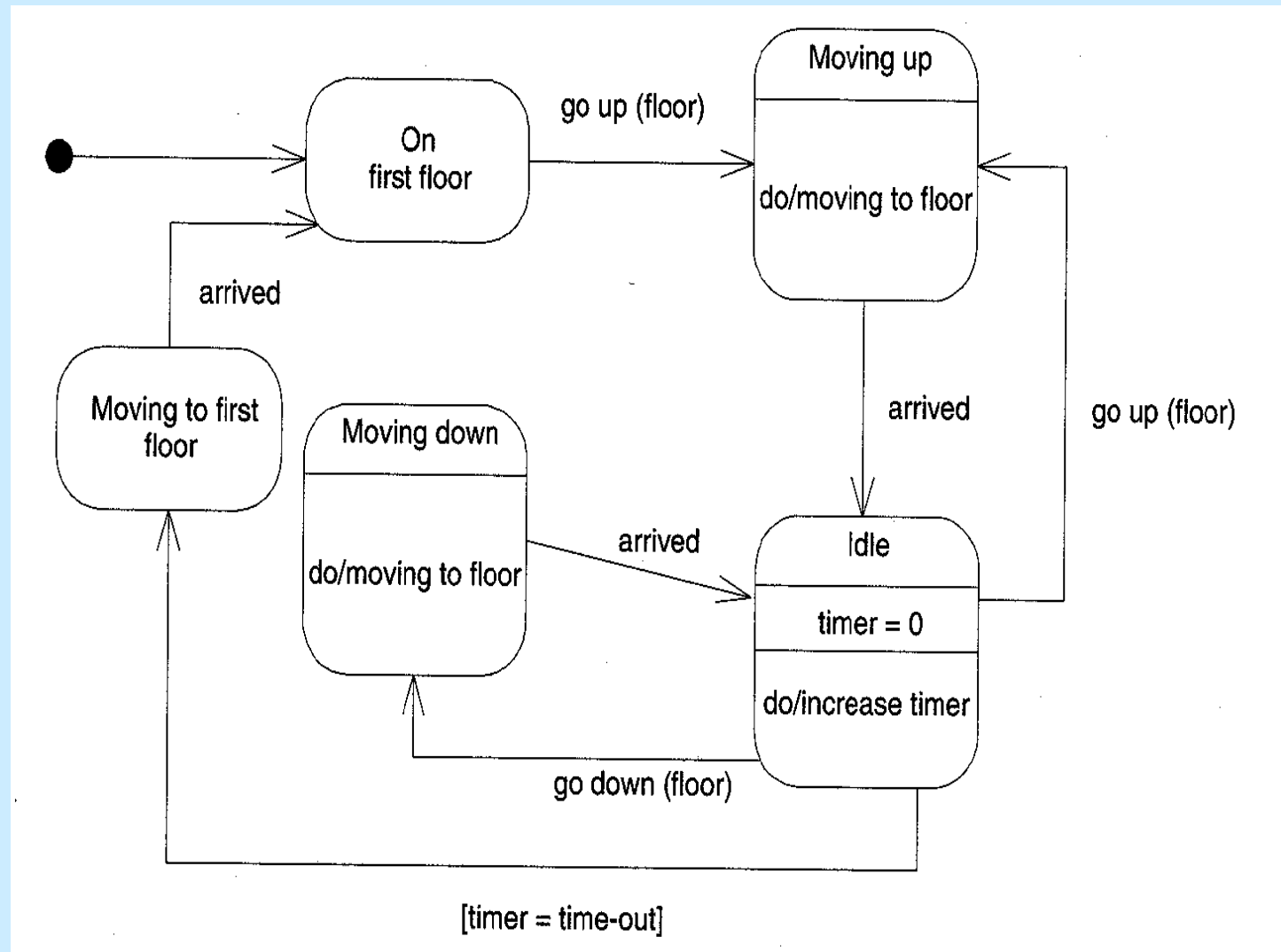




Diagrama de Atividade

- Destaca a lógica de realização de uma tarefa
- Mostra o fluxo entre atividades (ações não-atômicas)
- É semelhante aos antigos fluxogramas
- É usado também para modelar alternativas de execução e atividades concorrentes



Diagrama de Atividade

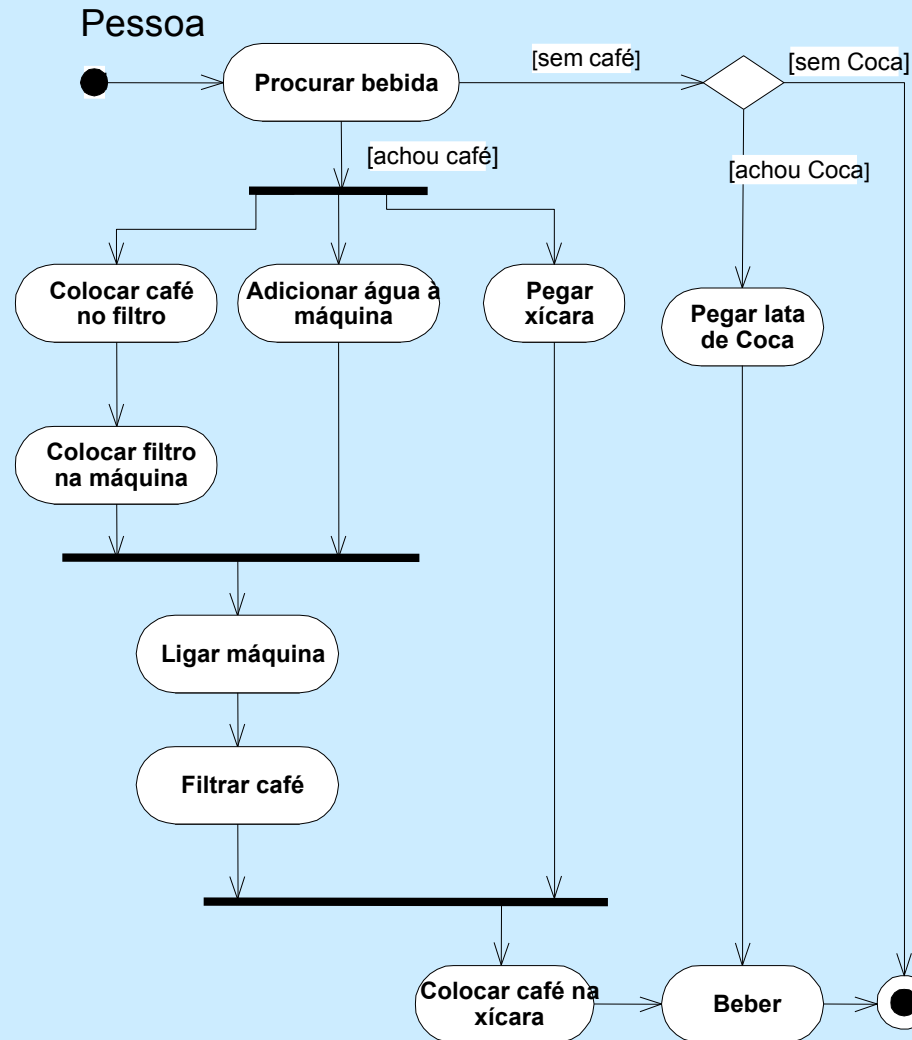




Diagrama de Componentes

- Mostra os componentes de hardware e software de uma aplicação e os relacionamentos entre eles
- É usado para modelar o aspecto físico de um sistema
- Exemplos de componentes são documentos, executáveis e tabelas de bancos de dados



Diagrama de Componentes

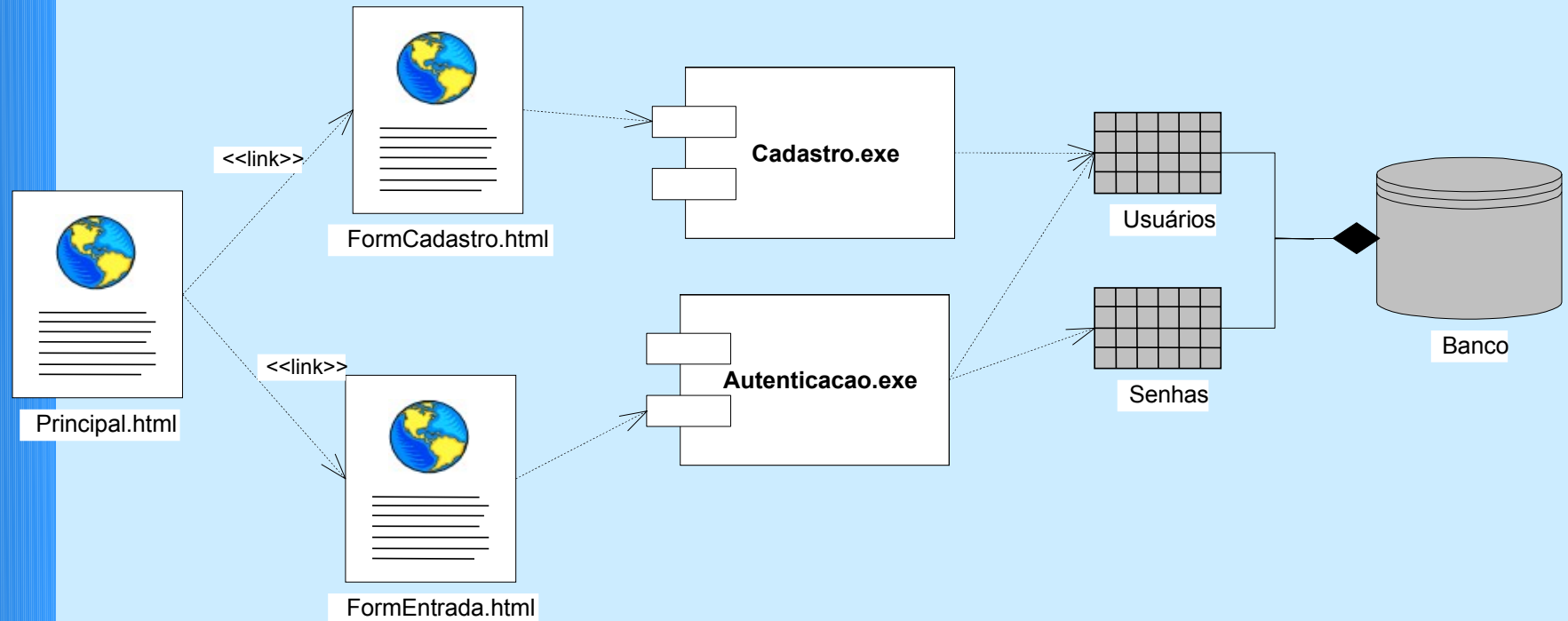


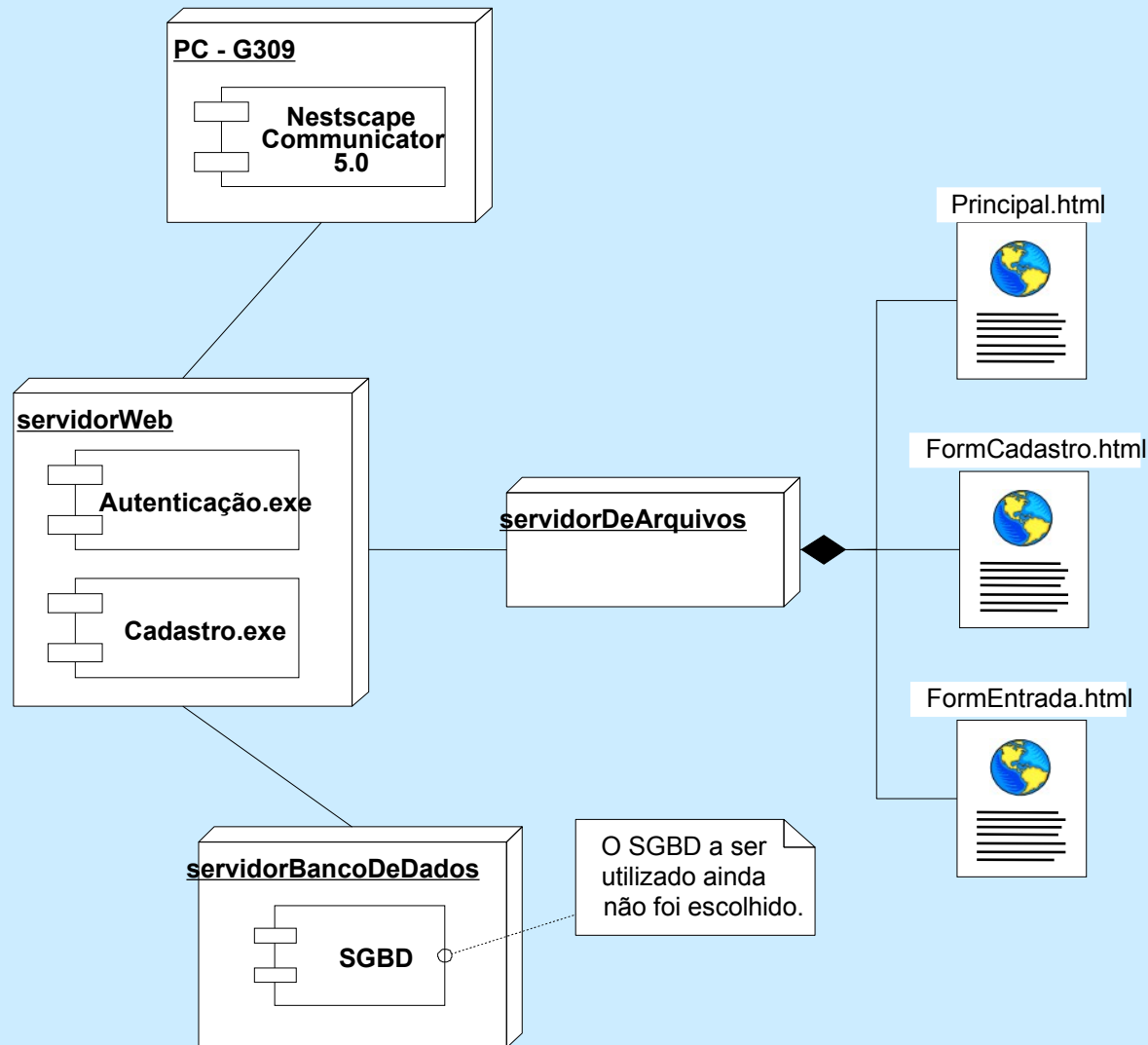


Diagrama de Implantação

- É usado para modelar a arquitetura de distribuição em que o sistema será executado
- É composto por nós e relacionamentos de comunicação
- Um nó pode ser um computador, uma rede, um disco rígido, um sensor, etc.



Diagrama de Implantação





Diagramas Use Cases



Diagramas de Use Cases

- Servem para facilitar o entendimento de um sistema mostrando a sua “visão externa”
- São usados para modelar o contexto de um sistema, subsistema ou classe
- São uma das maneiras mais comuns de documentar os requisitos do sistema
 - Delimitam o Sistema
 - Definem a funcionalidade do sistema



Use Case

- Um use case é uma unidade funcional que descreve o comportamento de um elemento da aplicação
- contém sequências de ações, interagindo com os atores que usam a aplicação
- inclui variantes, rotinas de erro, etc. que o sistema executa para produzir um resultado observável para um ator



Use Case: Representação Gráfica

- A coleção dos use cases deverá especificar todas as formas existentes de uso do sistema

Matricular aluno

**Solicitar
histórico**

**Verificar
pré-requisitos**

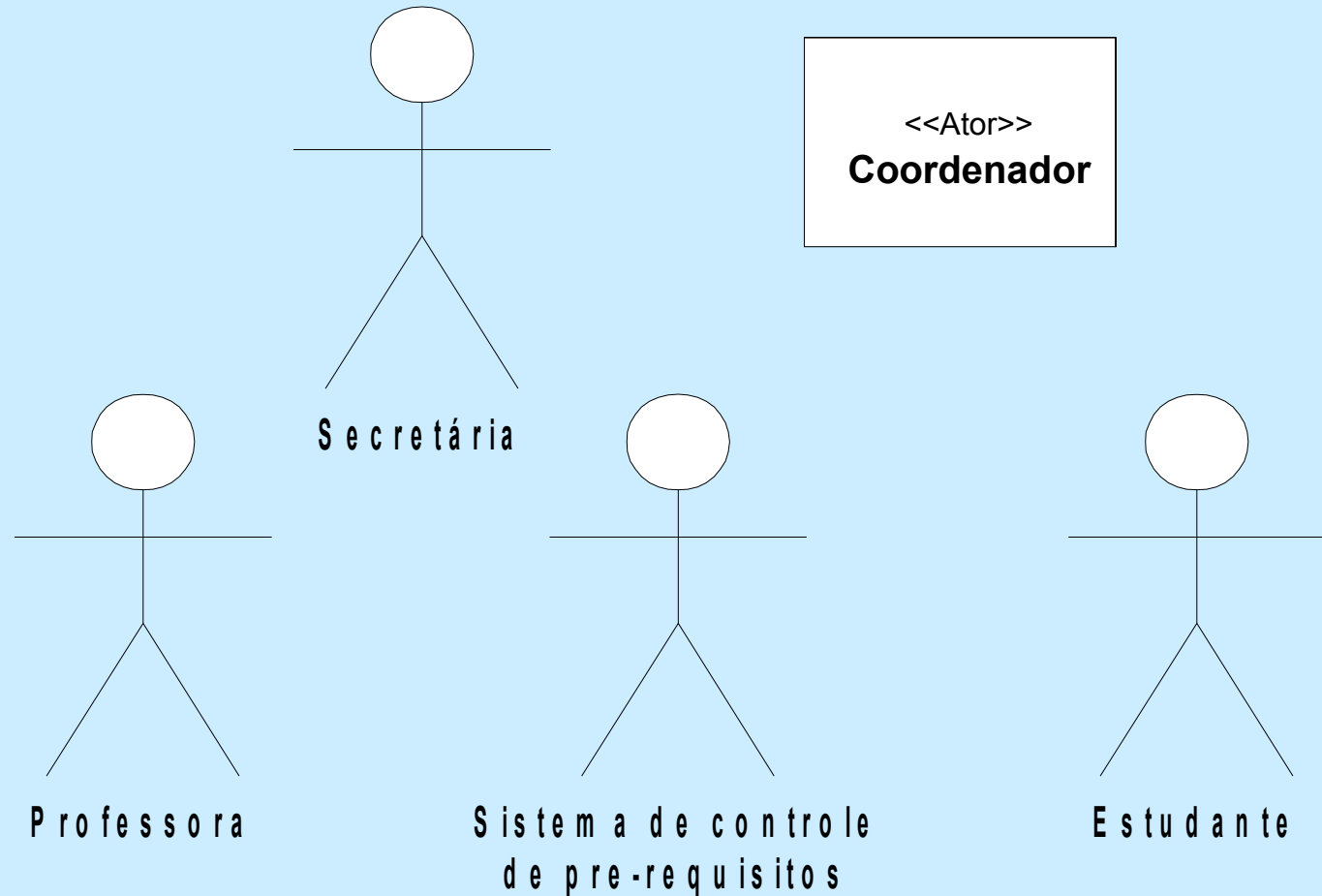


Atores

- O sistema será descrito através de vários *use cases* que são executados por um número de *atores*
- Atores constituem as entidades do ambiente do sistema
- São pessoas ou outros subsistemas que interagem com o sistema em desenvolvimento



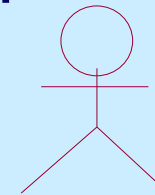
Atores - Notação



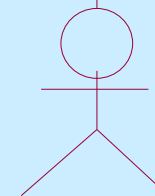


Atores: Especialização

- É possível definir tipos gerais de atores e especializá-los usando o relacionamento de especialização



Cliente



ClienteEspecial



Diagramas de Use Cases

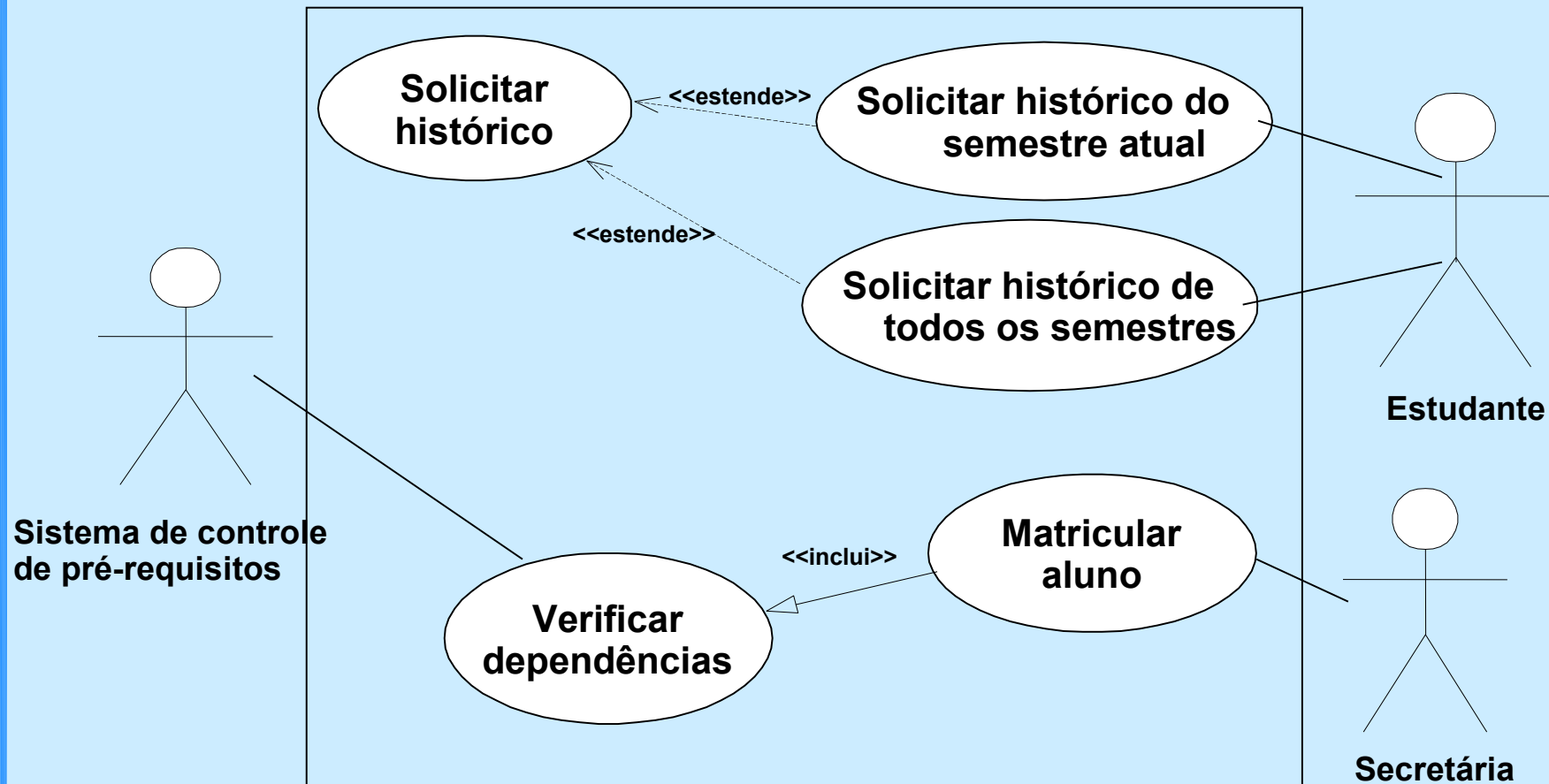
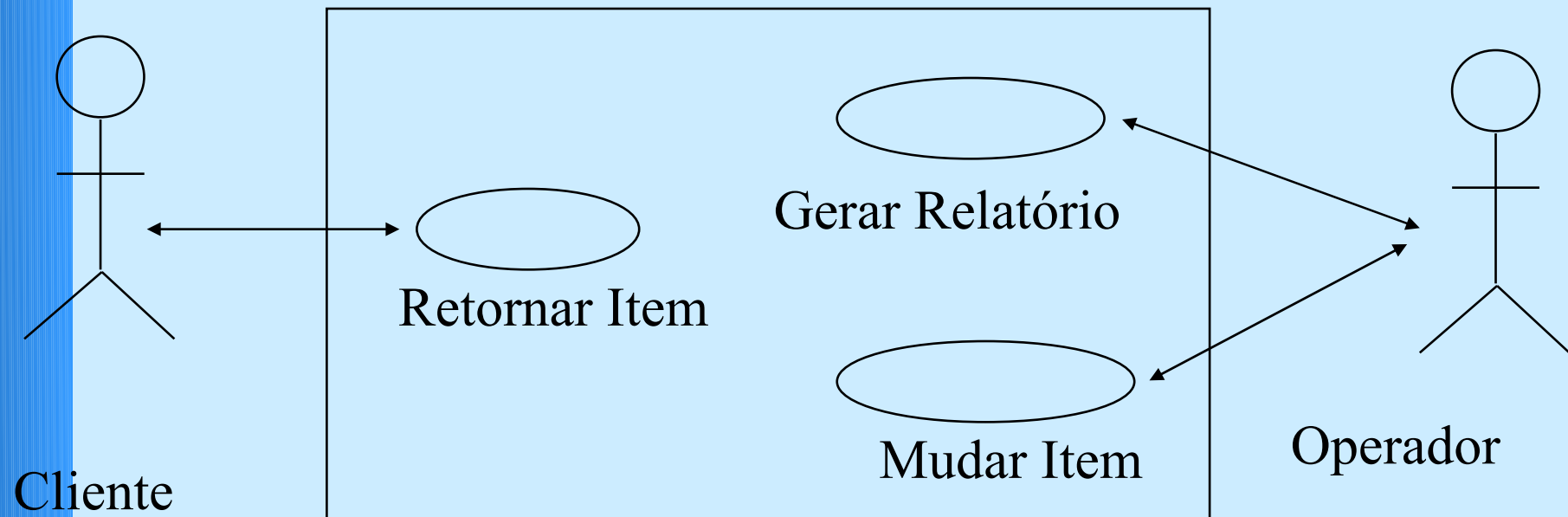




Diagrama de Use Case





Expressão de variantes de use case

- Nem sempre é óbvio decidir se uma funcionalidade corresponde a um novo use cases. As vezes trata-se de uma variação de um mesmo use case
 - Se as diferenças forem pequenas elas podem ser descritas através de variantes de um mesmo use case
 - Se as diferenças são grandes elas devem ser descritas como use cases separados



Expressão de variantes de use case

- Fluxo principal de eventos
 - Descreva a seqüência normal de eventos de um use case
- Fluxo excepcional de eventos
 - Descreva as variações dos cursos básicos de eventos (ex: Erros)



Expressão de Variantes

- Use Case Retornar item

Fluxo principal de eventos:

- Quando o *cliente* depositar os seus itens, ele/ela irá pressionar o *botão recibo* para obter o recibo. O recibo impresso irá listar os itens depositados, seus totais e o valor a ser pago ao *cliente*



Expressão de Variantes

- Use Case Retornar item

Fluxo excepcional de eventos:

- Quando o cliente retorna um item ele é medido pelo sistema. A medição é usada para determinar que tipo de lata ou garrafa foi depositado. Se aceito o total do cliente será incrementado. Se não for aceito, apresentar mensagem 'NÃO É VALIDA'

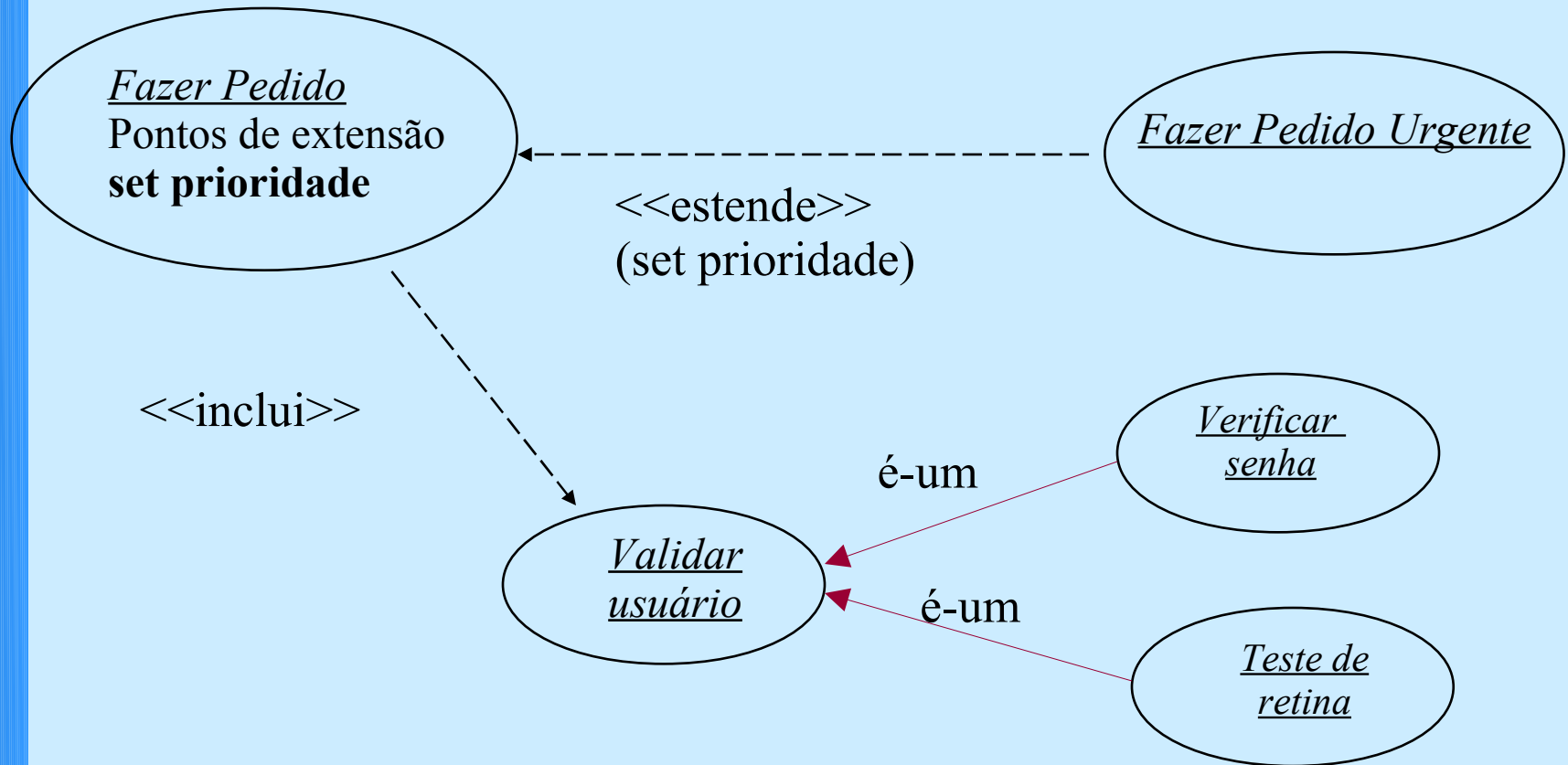


Organizando Use Cases

- Generalização
- Inclusão
- Extensão



Estruturação Use Case



Use Case Fazer Pedido

Fluxo principal de eventos: *inclui* (Validar usuário). Receber do usuário os itens do pedido. (set prioridade). Submeter o pedido para processamento