Análise de Sistemas – AULA 05

BCC Noturno - EMA908915A

Prof. Rafael Oliveira
rpaes@ic.unesp.br
Slides cedidos pelo Prof. Dr. Frank Affonso

Universidade Estadual Paulista
“Júlio de Mesquita Filho”
UNESP

Rio Claro 2014 (Sem 2)

Resumo

- Breve revisão OO
- A linguagem UML
  - Principais diagramas UML
    - Classe
    - Sequência
    - Caso de uso
    - Estado
- Exercícios

Resumo OO

Objeto	Estado	Comportamento
Pessoa	Nome, idade, RG	Falar, andar, cumprimentar
Cachorro	Nome, raça	Latir, correr
Conta bancária	Saldo, agência, número	Creditar, debitar
Carro	Cor, marca, modelo	Acelerar, frear, abastecer

Estado e **comportamento**, respectivamente, são transformados em **dados e procedimentos** quando programamos de forma estruturada e **atributos e métodos** quando utilizamos orientação a objetos.

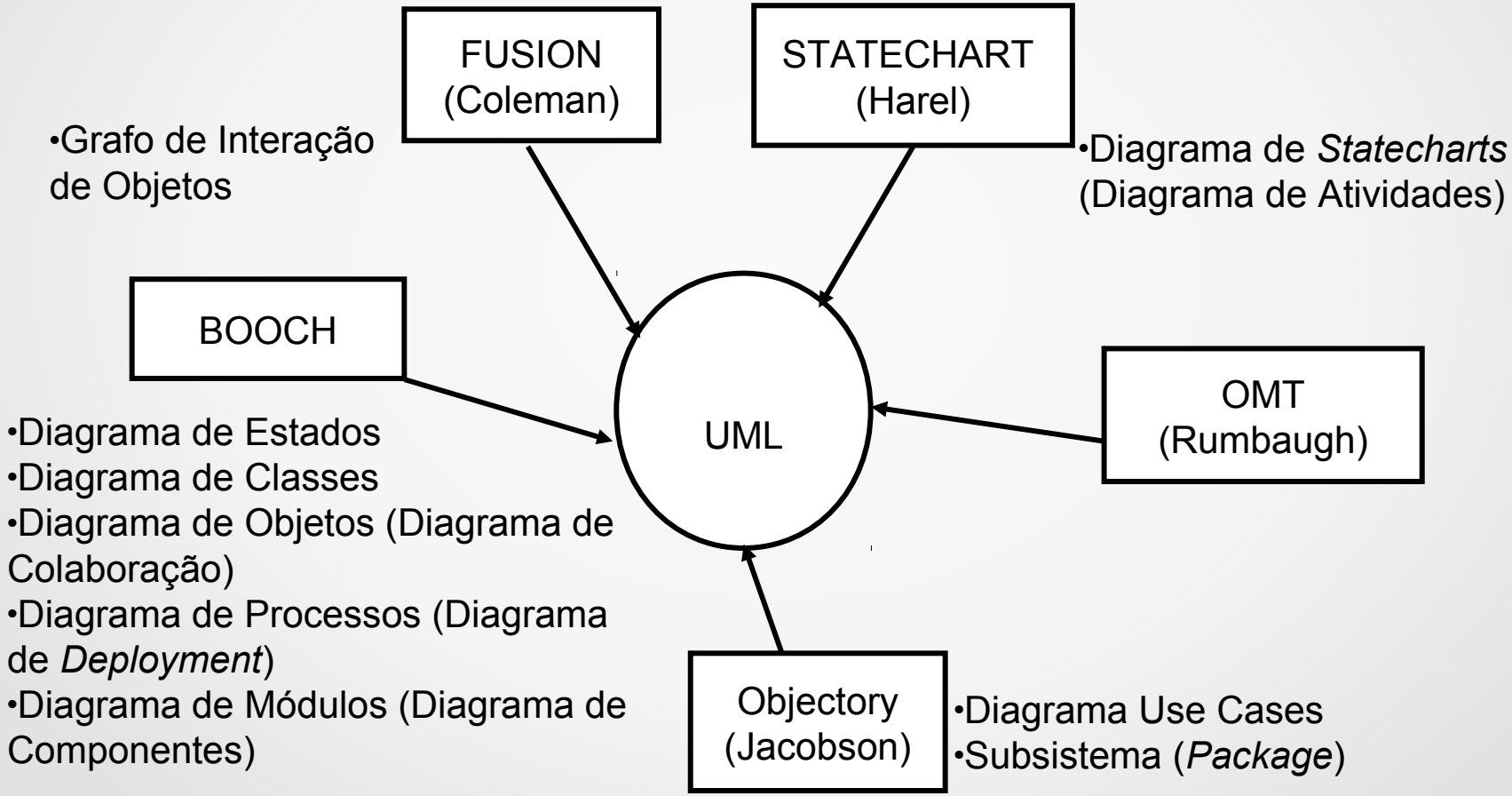
Principais Métodos OO

- Booch;
- OMT (Object Modeling Technique);
- OOSE(Object-Oriented Software Enginnering);
- Shalaer/Mellor;
- Coad/Yourdon Orientado a Objetos;
- Martin/Odell;
- Martin/Odell;
- Wirfs-Brock;
- Embley/Kurtz
- **UML – Unified Modeling Language;**
- Catalysis.

A Linguagem UML

- A UML é um **conjunto de técnicas** que apoiam a elaboração de projetos de sistemas orientados a objetos. Essas técnicas podem ser aplicadas na visualização, na especificação, na construção e na documentação de sistemas. Para isso, recebe contribuição de vários métodos OO.

A Linguagem UML (cont.)



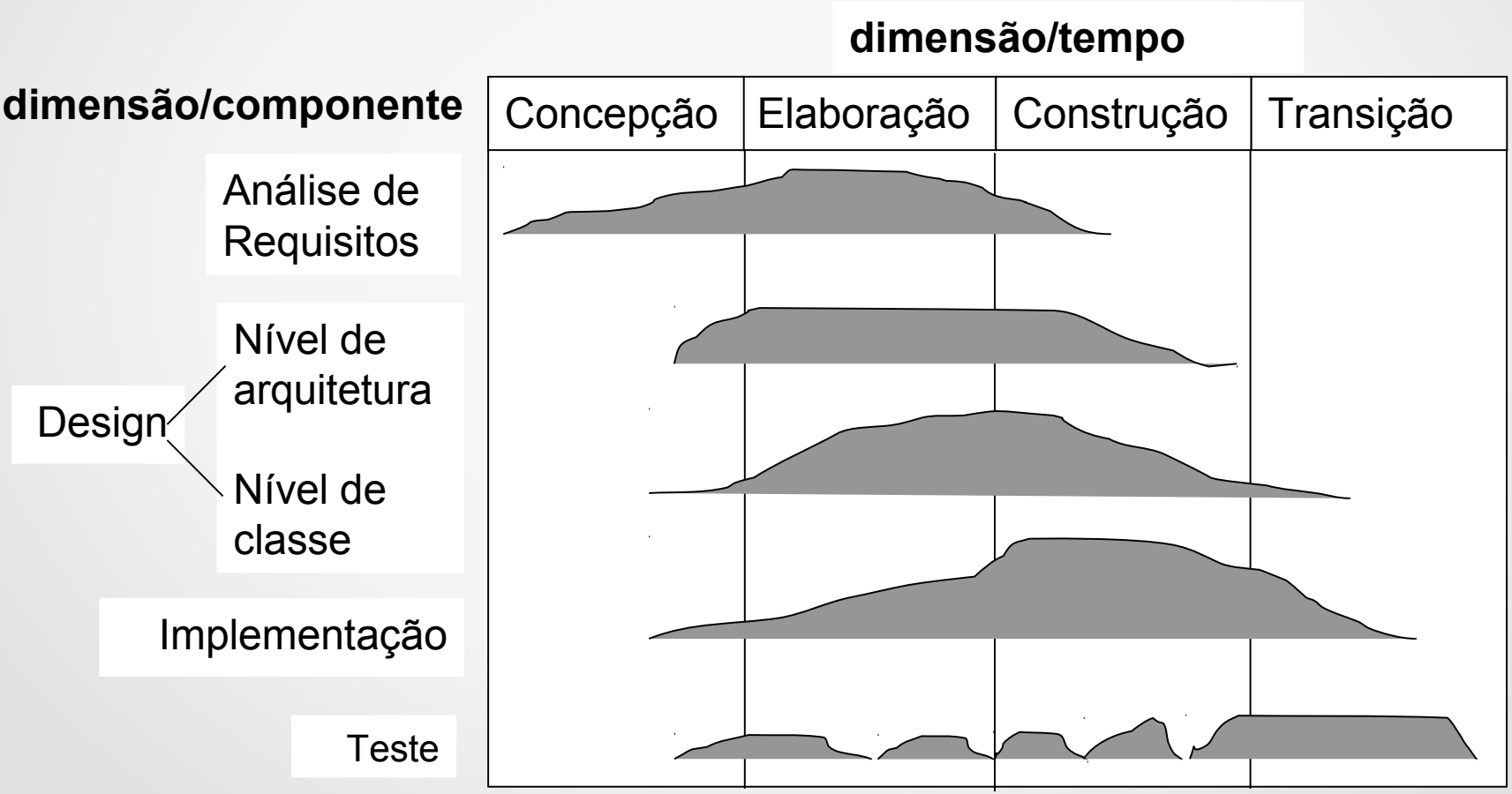
A Linguagem UML (cont.)

- **As técnicas da UML estão organizadas em diagramas** que atuam desde a representação dos requisitos até o projeto do sistema. Atualmente, existem algumas ferramentas que implementam os recursos da UML, tais como: Eclipse, **Netbeans**, Rational Rose, ArgoUML, Together, entre outras.

A Linguagem UML (cont.)

- O Processo de Desenvolvimento de Software (PDS) com a UML está estruturado em quatro fases:
  - **Concepção** quando se especifica da visão do sistema.
  - **Elaboração** quando se faz o planejamento das atividades necessárias e dos recursos requeridos e a especificação do sistema e design da sua arquitetura.
  - **Construção** desenvolvimento do produto como uma série de interações incrementais.
  - **Transição** fornecimento do produto para o usuário (fabricação, distribuição e treinamento).

A Linguagem UML (cont.)



A Linguagem UML (cont.)

- Ferramentas como, por exemplo, **Netbeans-UML**, implementam a UML em duas visões;
  - Use Case View;
  - Logical View;
- A linguagem UML prevê quatro visões para implementação das técnicas:
  - Use Case View;
  - Logical View;
  - Component View;
  - Deployment View;

A Linguagem UML (cont.)

- Visões:
  - **Use Case View** - descreve o sistema como um conjunto de transações do ponto de vista dos atores externos. Esta visão é inicialmente criada na fase de concepção do ciclo de vida e direciona o resto do processo.
  - **Logical View** - contém a coleção de packages, classes e relacionamentos. Esta visão é inicialmente criada na fase de elaboração e refinada na fase de construção.

A Linguagem UML (cont.)

- Visões:
  - **Component View** - contém módulos e subsistemas. Esta visão é inicialmente criada na fase de elaboração e refinada na fase de construção.
  - **Deployment View** - contém a parte física do sistema e a conexão entre estas partes. Esta visão é criada na fase de elaboração do processo.

A Linguagem UML (cont.)

- Baseado nessas visões, a UML oferece como representação do sistema as seguintes técnicas (diagramas):
- **Use Case View:**
  - Diagrama de Caso de Uso.
  - Diagrama de Sequência;
  - Diagrama de Colaboração.

A Linguagem UML (cont.)

- Baseado nessas visões, a UML oferece como representação do sistema as seguintes técnicas (diagramas):
- **Logical View:**
  - Diagrama de Classes;
  - Diagrama de Estado;
  - Diagrama de Atividade.

A Linguagem UML (cont.)

- Baseado nessas visões, a UML oferece como representação do sistema as seguintes técnicas (diagramas):
- **Component View:**
  - Diagrama de Componentes.
- **Deployment View:**
  - Diagrama de Deployment.

A Linguagem UML (cont.)

- As técnicas da UML para a modelagem de um sistema:
  - Diagrama de Classe;
  - Diagrama de Sequência;
  - Diagrama de Caso de Uso;
  - Diagrama de Estado;
  - Diagrama de Atividade;
  - Outros (**Outras Ferramentas**).

A Linguagem UML (cont.)

- Para mostrar a modelagem com os diagramas citados, será considerado um **sistema para locação de carros por um clientes**

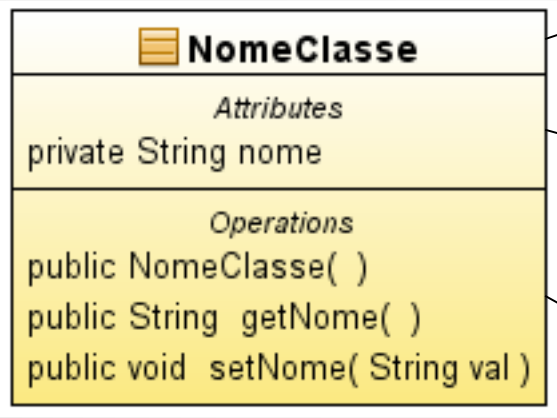
- **Considerações:**
  - O número de atributos serão reduzidos para não “poluir” o diagrama;
  - Os requisitos desse sistema estão em anexo.

A Linguagem UML (cont.)

- **Diagrama de Classe** é uma representação gráfica para descrições genéricas do sistema. Pode conter packages, tipos, relacionamentos e instâncias de classes. Utilizado, principalmente, nas fases de análise e projeto de um sistema:
  - **Análise**, mostrando regras e responsabilidades comuns de entidades que fornecem o comportamento do sistema.
  - **Projeto**, capturando a estrutura das classes que formam a arquitetura do sistema.

A Linguagem UML (cont.)

- Diagrama de Classe:



Nome da Classe

Atributos da Classe

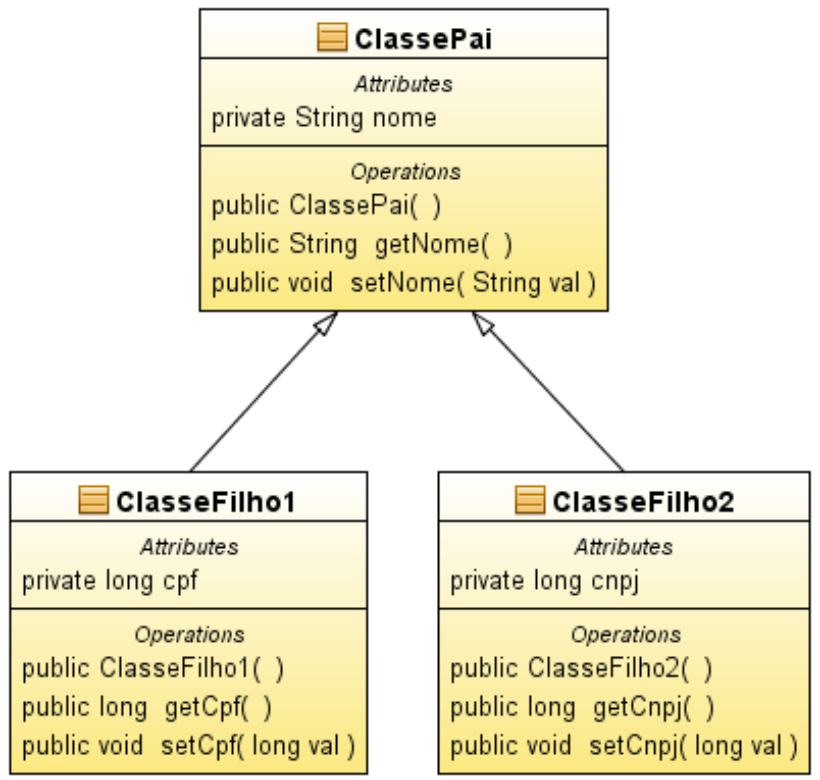
Métodos da Classe

A Linguagem UML (cont.)

- Os tipos de visibilidade dos atributos e métodos de uma classe são:
  - **public:** é visto por todos os membros internos da classe e qualquer membro externo à classe em que está declarado;
  - **private:** é visto apenas pelos elementos da classe, subclasses (métodos de acesso);
  - **protected:** os elementos são acessíveis somente pela própria classe ou pelas subclasses;

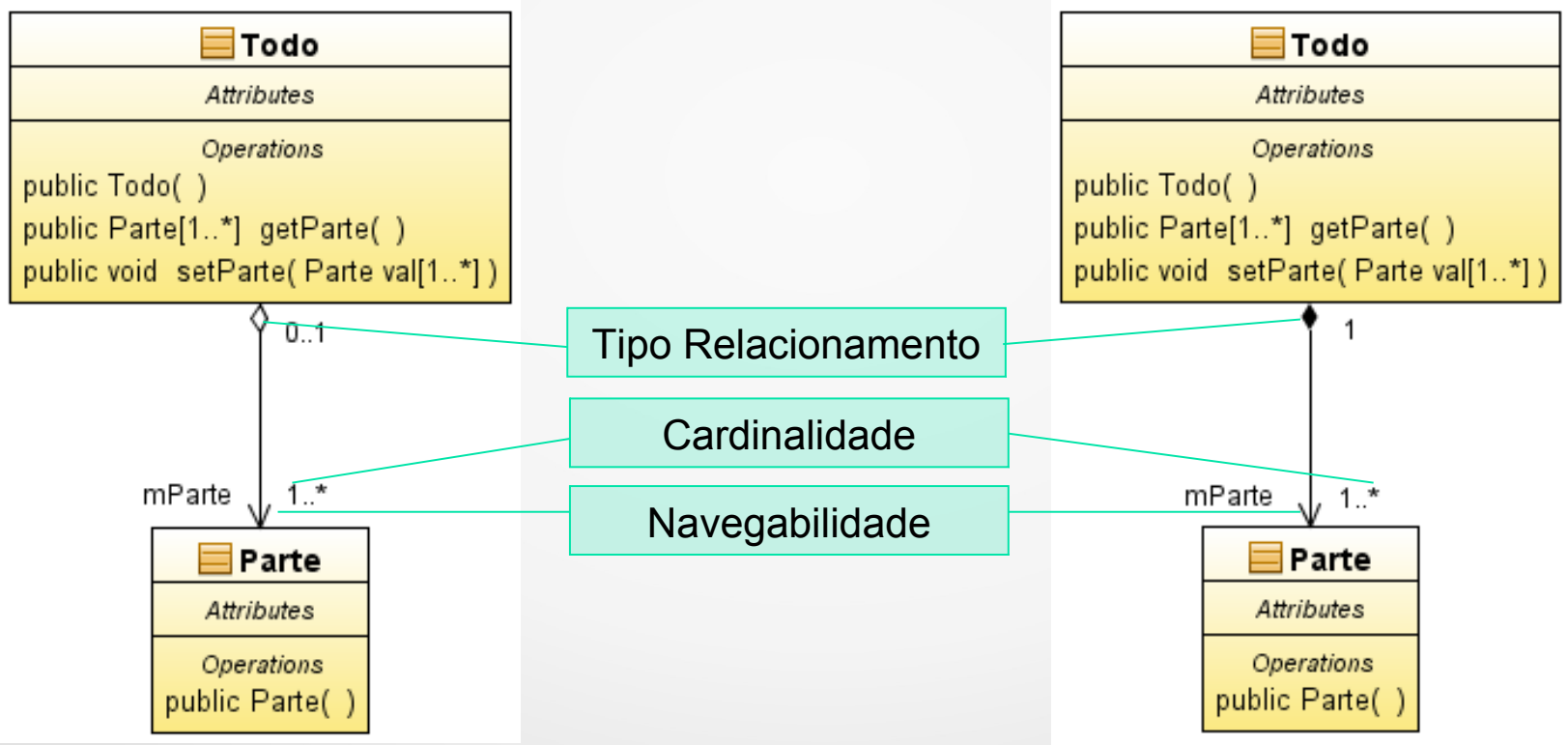
A Linguagem UML (cont.)

- Algumas notações (Herança):



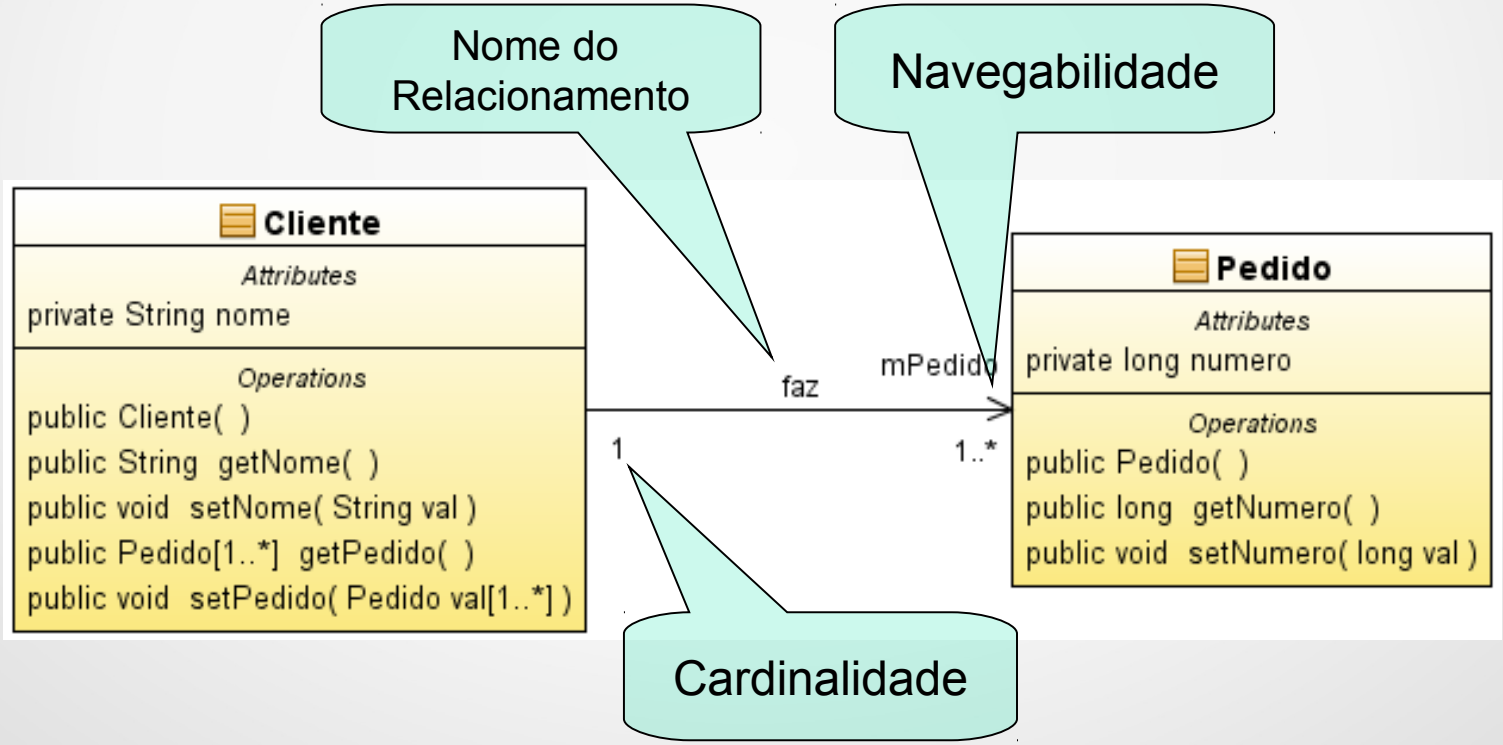
A Linguagem UML (cont.)

- Algumas notações (Agregação / Composição):



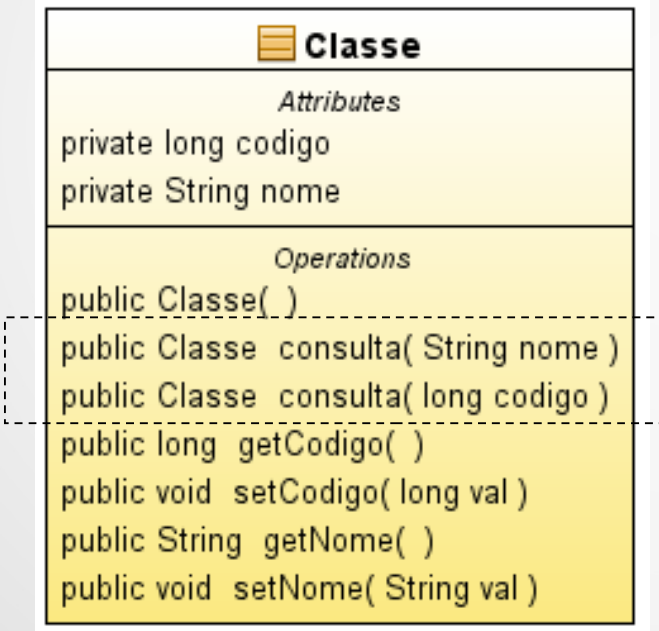
A Linguagem UML (cont.)

- Algumas notações (Associação):



A Linguagem UML (cont.)

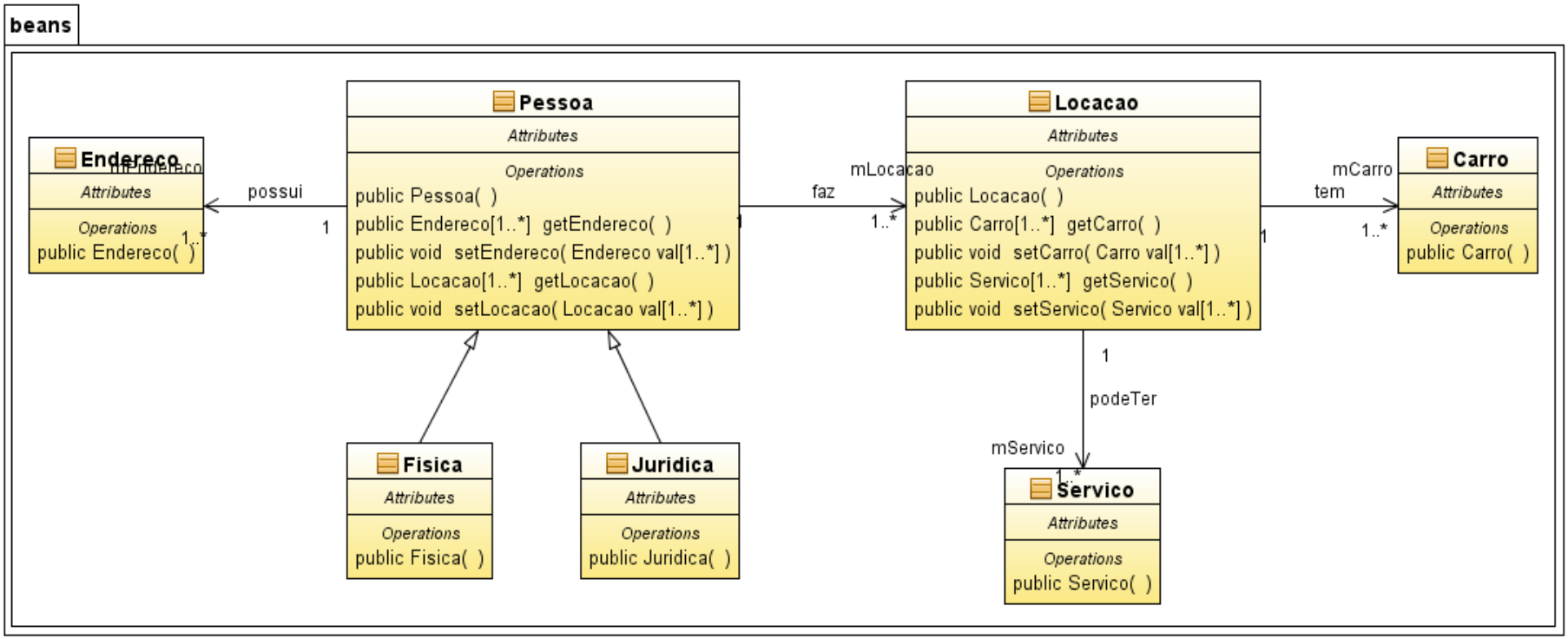
- Algumas notações (Sobrecarga):



Operações com **mesmo retorno**, **mesmo nome** e **diferentes parâmetros** de consulta

A Linguagem UML (cont.)

- Diagrama de Classe (Resumido):



A Linguagem UML (cont.)

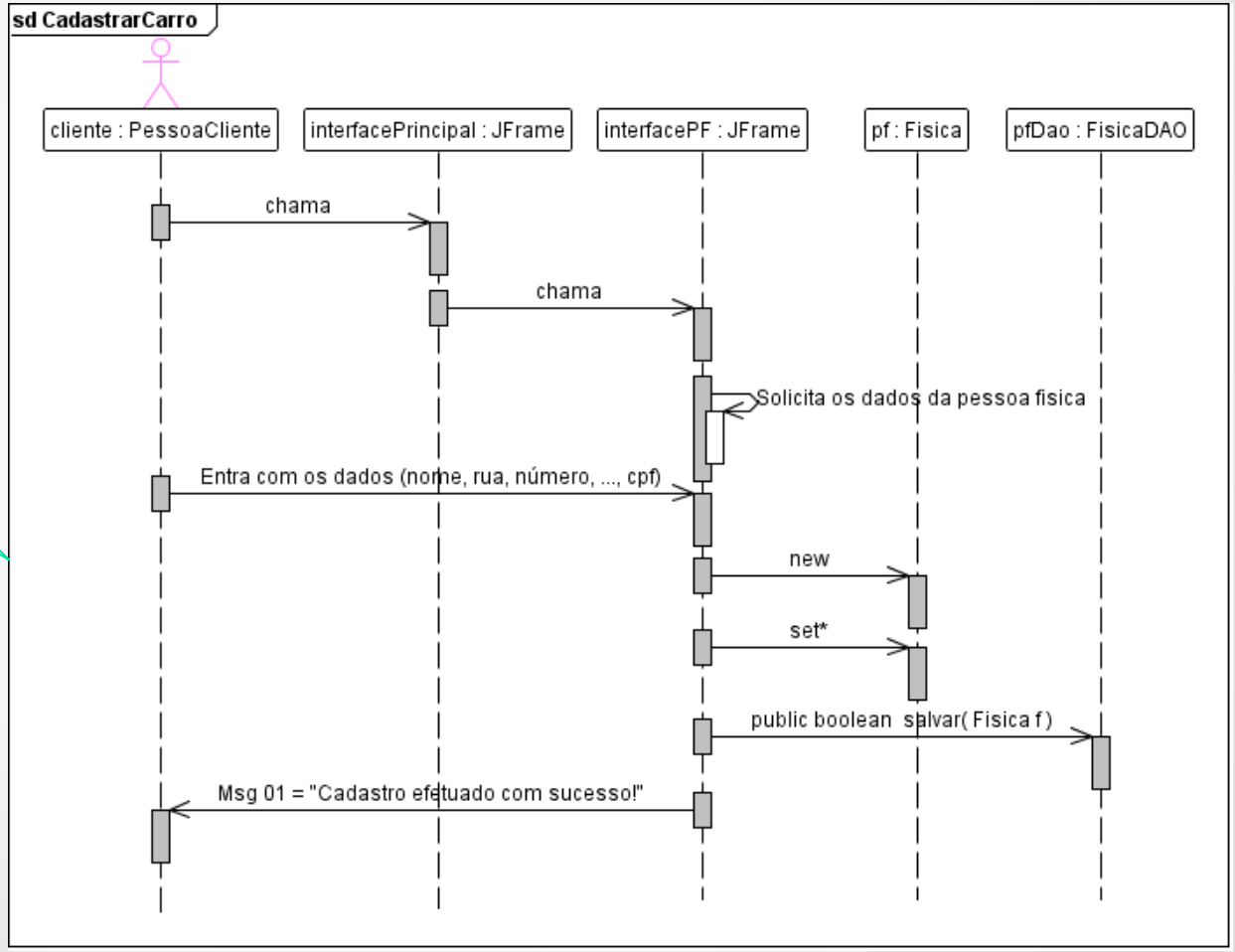
- Acesse o arquivo diagrama_classes.pdf no moodle!

A Linguagem UML (cont.)

- **Os Diagramas de Sequência** mostram uma interação organizada em uma **sequência de tempo** entre os objetos participantes de uma operação e as trocas de mensagens entre eles.
- Esses diagramas são bastante utilizados para especificar sistemas de tempo real e sistemas complexos. Um diagrama de seqüência possui duas dimensões:
  - **vertical** que representa o tempo e;
  - **horizontal** que representa diferentes objetos (se for necessário as dimensões podem ser invertidas).

A Linguagem UML (cont.)

Diagrama de Sequência para cadastro de uma pessoa física



A Linguagem UML (cont.)

- Acesse o arquivo diagrama_sequência.pdf no moodle!

A Linguagem UML (cont.)

- **Diagrama de Caso de Uso**, representa uma coleção de casos de uso e atores, os quais são tipicamente usados para especificar ou caracterizar as funcionalidades e o comportamento de um sistema.

A Linguagem UML (cont.)

- **Os agentes externos** e/ou qualquer outro sistema que possa interagir com o sistema que está sendo modelado são chamados de atores, pois fornecem uma visão clara do que será realizado, delimitando o problema;
- Os diagramas de caso de uso são desenvolvidos de acordo com os eventos que ocorrem entre os agentes externos e o sistema.

A Linguagem UML (cont.)

Diagrama de Caso de Uso para cadastro de uma pessoa física

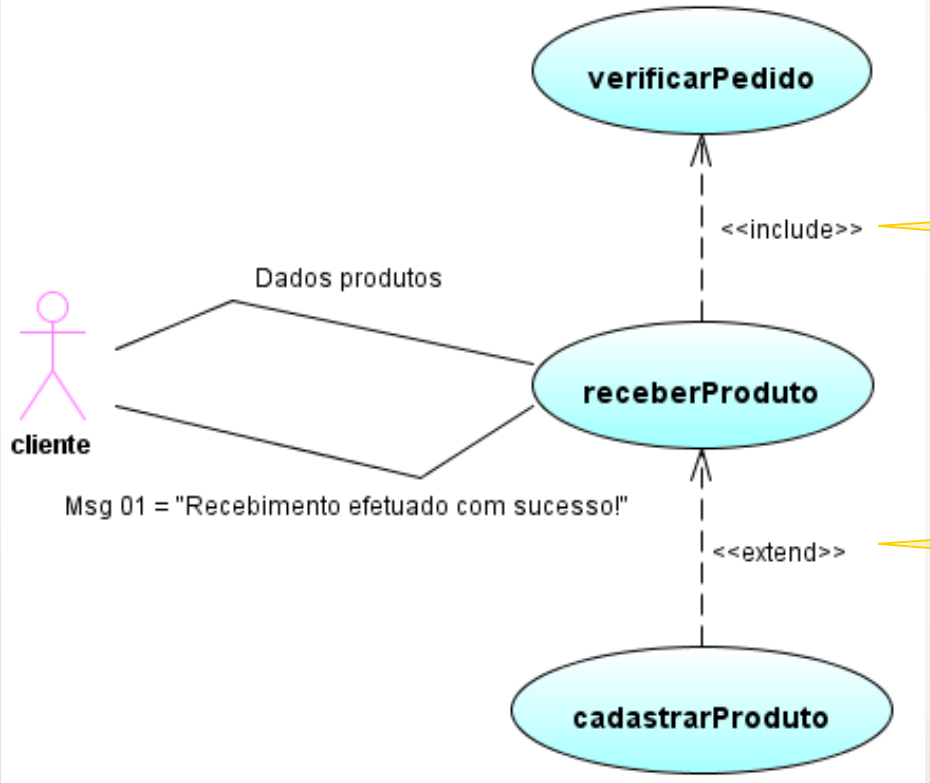
Curso Normal:
1. Pessoa solicita o cadastro.
2. O sistema solicita os dados da pessoa.
3. O sistema faz a validação do CPF.
4. O sistema exibe os dados da pessoa.
5. O sistema solicita a confirmação do cadastro.
6. A pessoa confirma a operação.
7. O sistema exibe a mensagem ("Cadastro efetuado com sucesso!").

Curso alternativo:
3. O sistema faz a validação do CPF.
3.1. O CPF da pessoa não é válido.
3.2. O sistema exibe uma mensagem informando o cliente.
3.3. O sistema volta para o passo 2.

5. O sistema solicita a confirmação do cadastro.
5.1. O usuário aborta a operação.
5.2. O sistema volta para o passo 2.

A Linguagem UML (cont.)

- Notações:



Uso Obrigatório

Uso Não Obrigatório

A Linguagem UML (cont.)

- Acesse o arquivo diagrama_casos_uso.pdf no moodle!

A Linguagem UML (cont.)

- **Diagrama de Estado** é usado para mostrar os estados dos objetos de uma classe. Os eventos do diagrama de estados causam uma transição de um estado para outro e as ações resultam na mudança de estado. Cada diagrama de estados está associado a uma classe ou a um diagrama de estados de um nível mais alto.

A Linguagem UML (cont.)

- Também é conhecido como um grafo direcionado de estados conectados por transições que mostra um estado inicial, um ou mais estados, um ou mais estados finais e as transições de estados entre eles. Cada classe que possui eventos significativos, pode conter um diagrama de estados para descrever este comportamento.
- Utilizados para modelar aspectos dinâmicos de um sistema;

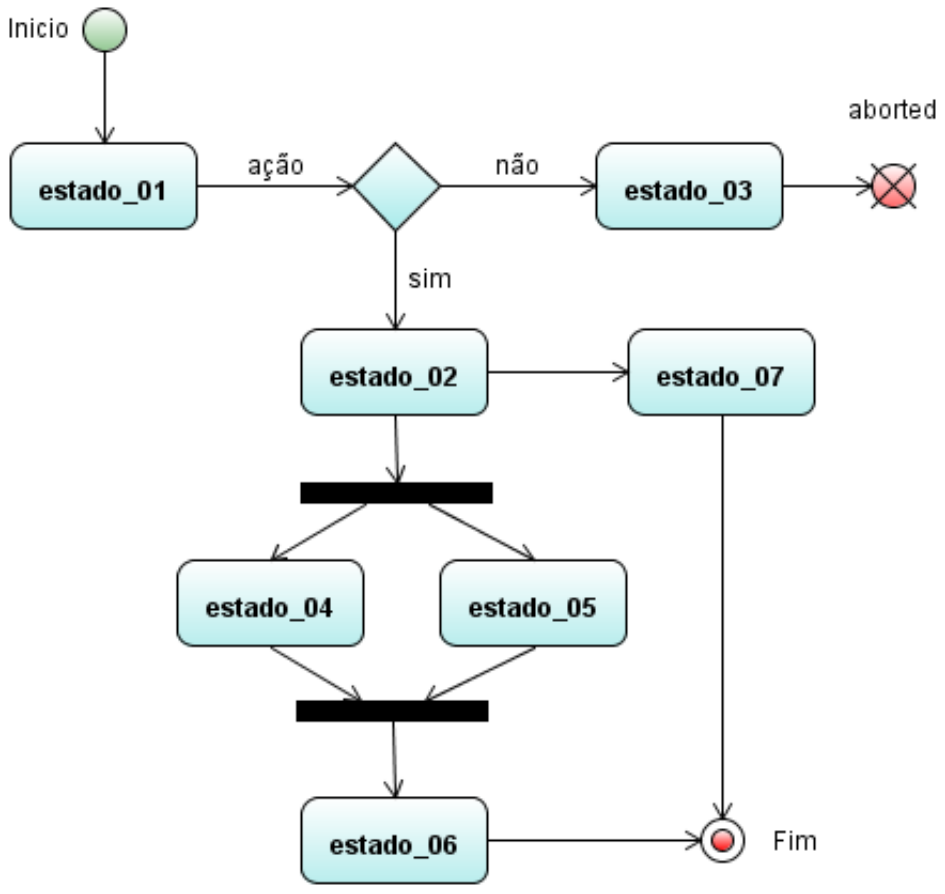
A Linguagem UML (cont.)

- Quando elaborar este tipo de diagrama?
  - Utilizado em classes cujos objetos têm atributo de estado com duas propriedades: o atributo pode assumir um pequeno número de valores possíveis e as transições permitidas entre esses valores são restritas. Um classe pode ter um ou mais atributos que podem ser proveitosamente modelados com diagramas de estado.

A Linguagem UML (cont.)

- Exemplo Genérico:

Apenas para demonstrar o uso dos elementos do diagrama



A Linguagem UML (cont.)

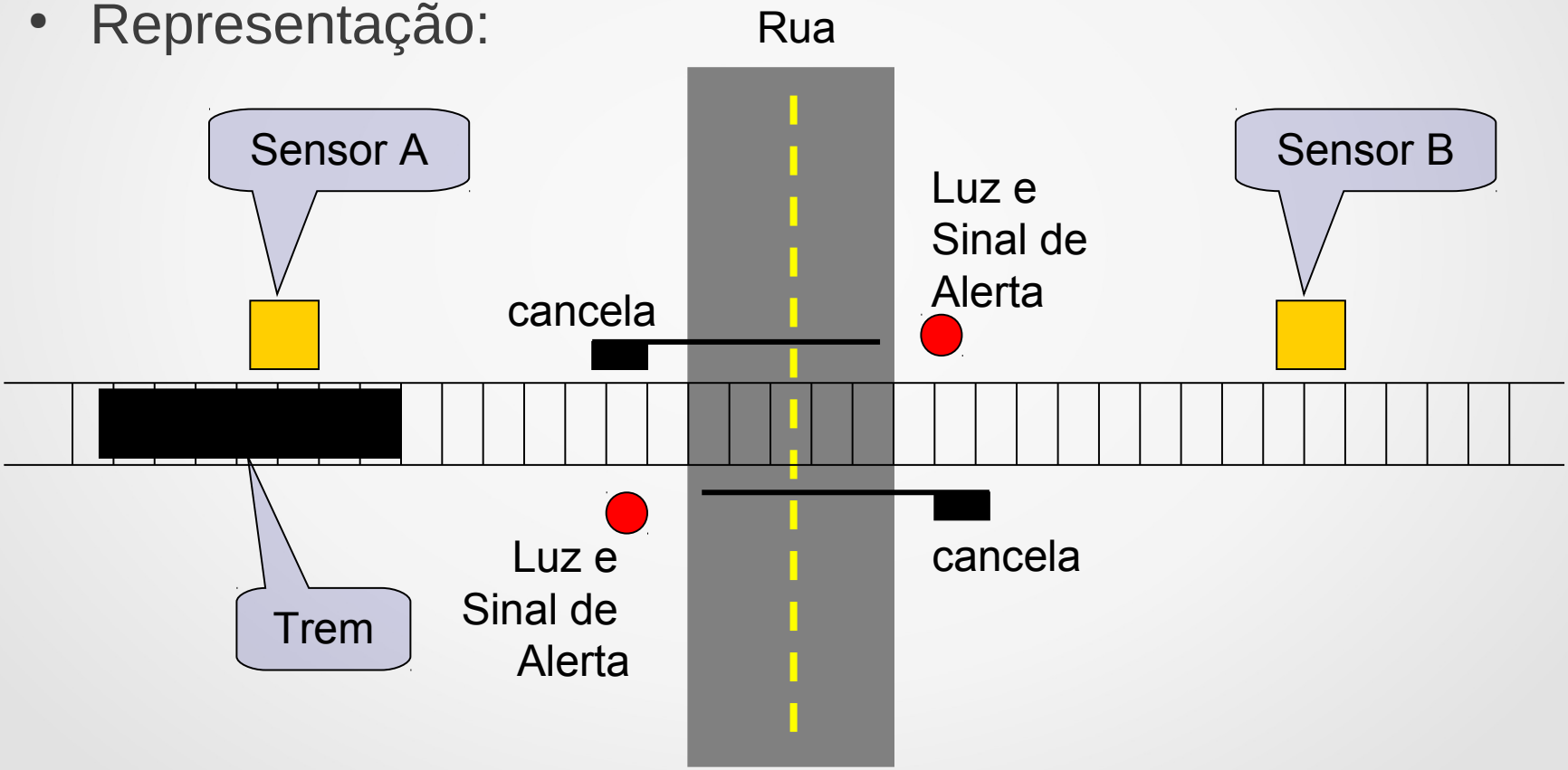
- Exemplo: vamos modelar o estado de um sensor que controla o estado de passagem de uma rua:

http://www.fahrplancenter.com/CancelaFormiga.jpg

Formalizando o problema

A Linguagem UML (cont.)

- Representação:

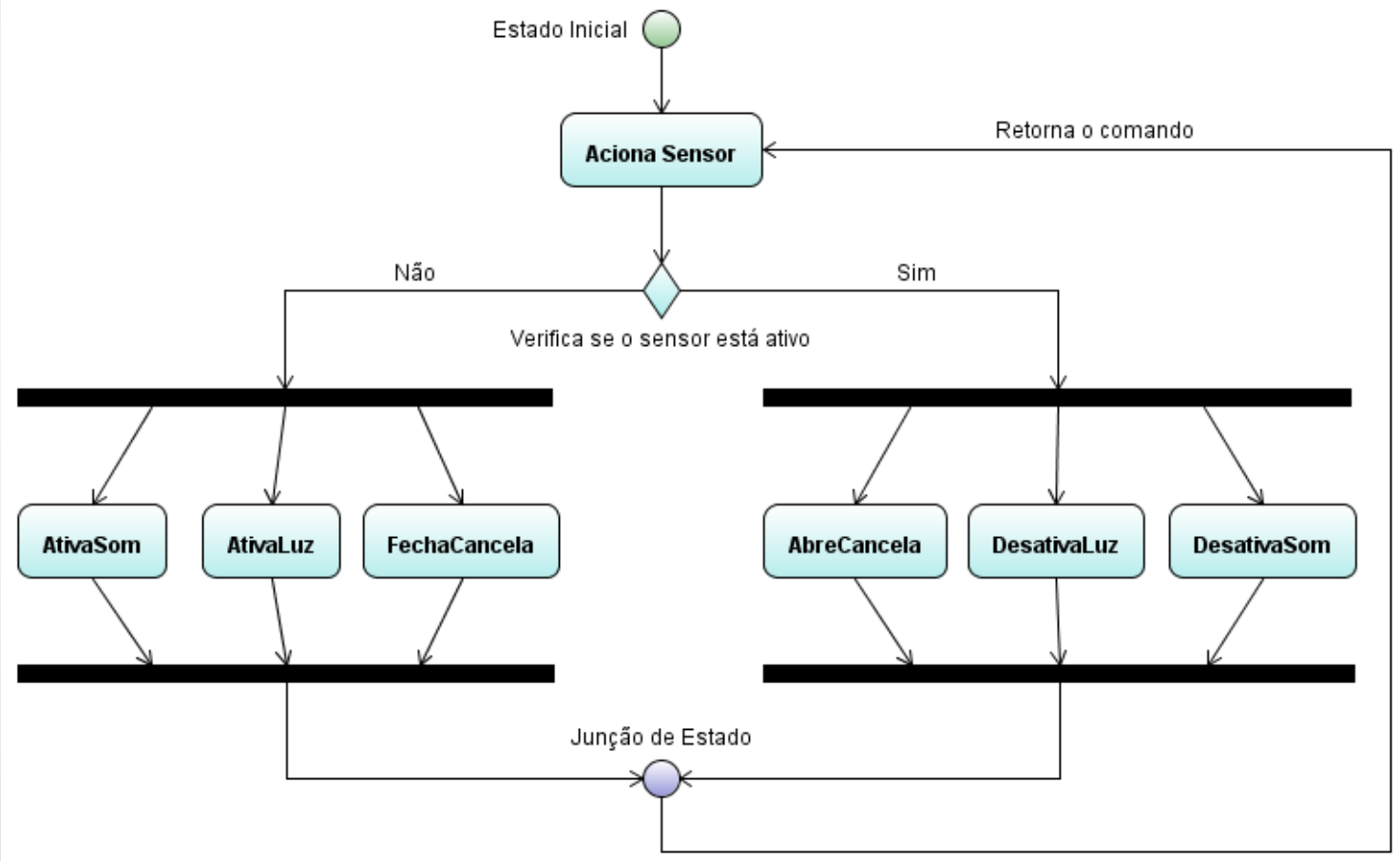


A Linguagem UML (cont.)

- Descrição:
  - Quando o trem passar pelo sensor A, a luz e o sinal sonoro serão acionados e a cancela será fechada lentamente após 30 segundos.
  - Após a passagem do trem pelo sensor B, a luz e o sinal sonoro são desativados e a cancela se abre lentamente.
  - Observação: o tamanho do trem e a distância entre os sensores são ideais.

A Linguagem UML (cont.)

- Modelo:

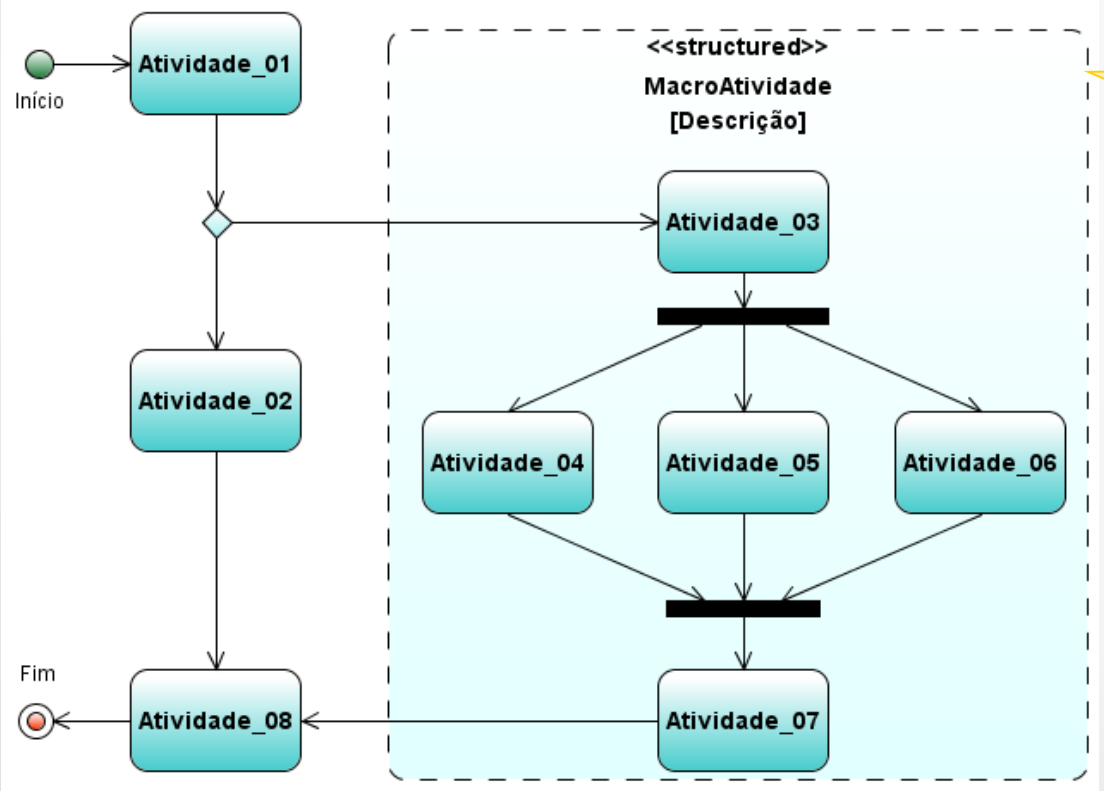


A Linguagem UML (cont.)

- **Diagrama de Atividade** é um diagrama de estado especial, onde a maioria das transições é ativada por conclusão das ações nos estado anteriores. Seu propósito é estudar o processamento interno e o fluxo das operações.
- Pode-se dizer que representa o fluxo de controle de uma atividade para outra, mostrando a concorrência e ramificações de controle.

A Linguagem UML (cont.)

- Exemplo Genérico:



Apenas para demonstrar o uso dos elementos do diagrama

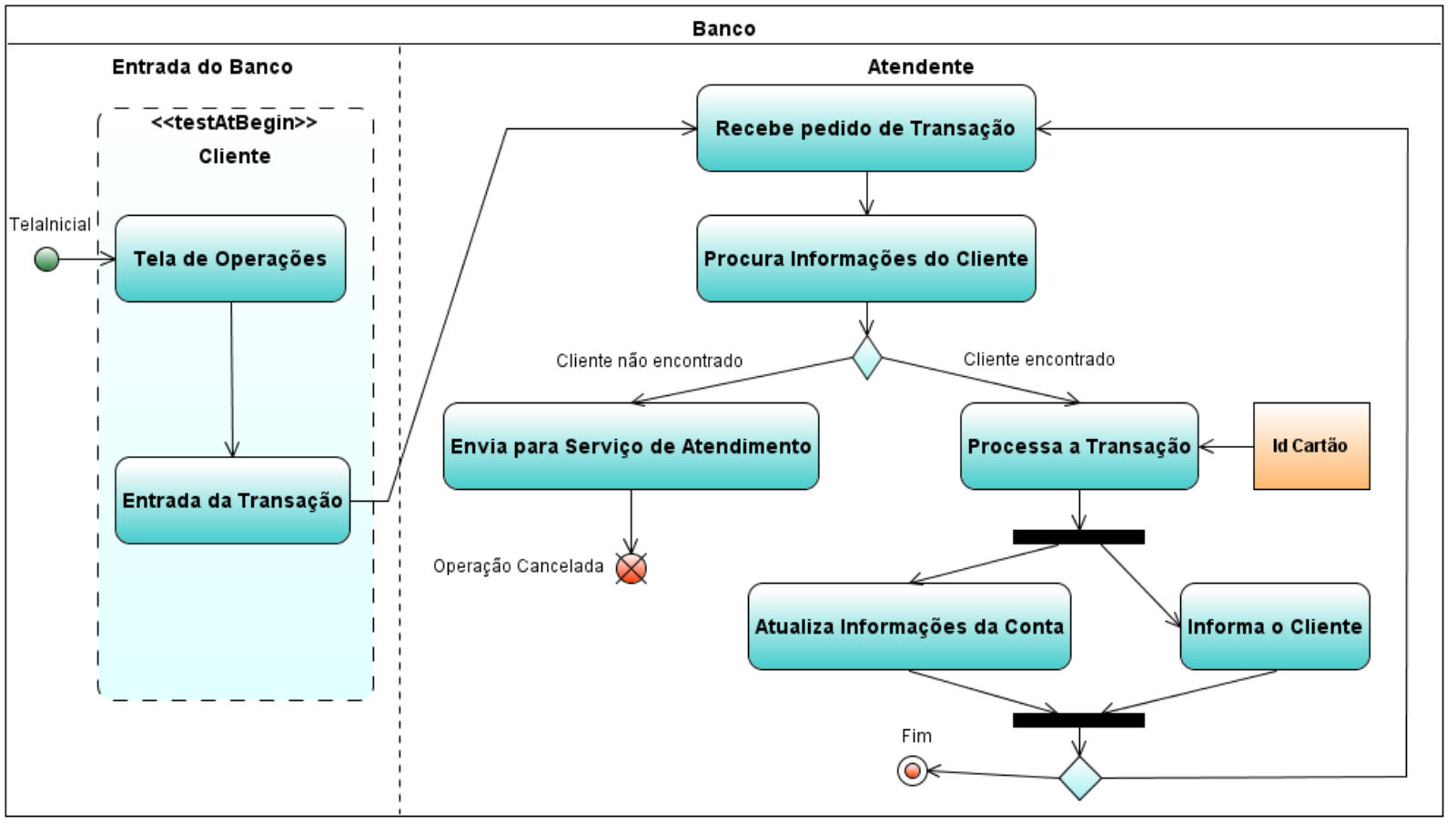
Exemplo:
Movimentação bancária

A Linguagem UML (cont.)

- Exemplo:

Movimentação bancária via ATM

A Linguagem UML (cont.)



A Linguagem UML (cont.)

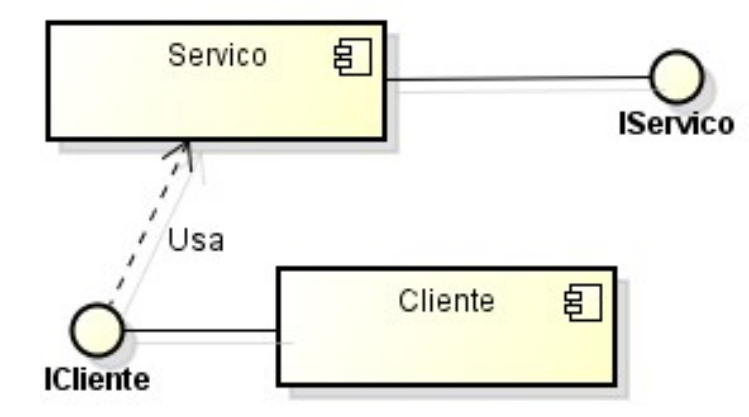
- O **Diagrama de Componentes** mostra as dependências entre componentes de software, incluindo componentes de código fonte, componentes de código binário e componentes executáveis. Um módulo de software é representado como um tipo de componente.

A Linguagem UML (cont.)

- Um diagrama de componentes contém elementos que representam:
  - Packages de componentes;
  - Componentes ou Módulos;
  - Packages;
  - Programa principal;
  - Subprogramas;
  - Tarefas;
  - Dependências.

A Linguagem UML (cont.)

- Componentes são conectados com outros componentes através de relacionamentos de dependências. Isto indica que um componente usa os serviços de outro.

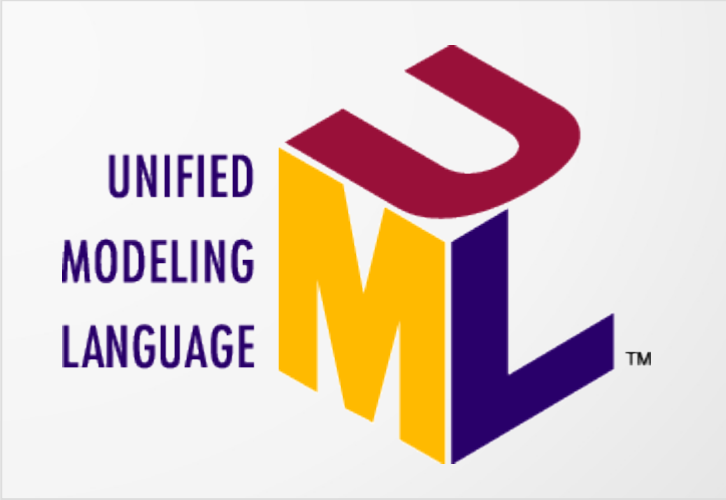


A Linguagem UML (cont.)

- O **Diagrama de Deployment** mostra as conexões físicas entre os processadores, dispositivos e a alocação dos processos aos processadores. Este diagrama mostra a organização do hardware e a ligação do software com os dispositivos físicos. O tipo do dispositivo de hardware é dado pelo seu stereotype, tais como, processador, vídeo, dispositivo, memória, disco e outros.

Estudo de Caso

- Modelagem de Sistema utilizando ferramenta Astah Community:



Referências Bibliográficas

- MARTIN, J.; ODELL J. J. Análise e Projeto Orientados a Objetos, Makron Books, 1995;
- PRESSMAN, R. S., Engenharia de Software, 5ª edição, Mc Graw Hill.
- FURLAN, J. D. Modelagem de objetos através da UML. Análise e desenho orientado a objetos, Makron Books, 1998.
- MEDEIROS, E. Desenvolvendo Software com UML, Ed. Person, 2004;
- Notas de Aula do Prof. Dr. Antonio Francisco do Prado - Departamento de Computação - UFsCar;
- Notas de Aula da Profa. Dra. Rosângela Aparecida Dellosso Penteado - Departamento de Computação - UFsCar;
- Notas de Aula da Profa. Dra. Maria Cecília V. S. Carneiro - DEMAC/UNESP;
- Notas de Aula do Prof. Dr. Frank J. Affonso - DEMAC/UNESP.

Exercícios resolvido

Sistema de Controle Acadêmico (SiCAd)

- Requisitos funcionais:

- RF01 – o sistema deve permitir à secretaria cadastrar cursos contendo código, descrição e coordenador.
- RF02 – o sistema deve permitir à secretaria cadastrar disciplinas de cursos, contendo código, descrição, carga horária, ementa, bibliografia e pré-requisitos.
- RF03 – o sistema deve permitir à secretaria cadastrar alunos, contendo matrícula, nome, endereço, telefone e curso para o qual foi aprovado.
- RF04 – o sistema deve permitir ao departamento de recursos humanos (RH) cadastrar professores, contendo nome, endereço, telefone e titulação máxima (graduação, especialização, mestrado, doutorado) e cursos que esteja vinculado.

Exercícios resolvido

Sistema de Controle Acadêmico (SiCAd)

- Requisitos funcionais:

- RF05 – o sistema deve permitir à secretaria abrir turmas de disciplinas de cursos, informando ano e semestre, dias da semana e horários de realização.
- RF06 – o sistema deve permitir aos coordenadores de curso alocar professores a determinadas turmas.
- RF07 – o sistema deve permitir à secretaria matricular alunos em turmas.
- RF08 – o sistema deve permitir aos professores lançar avaliações (duas notas parciais, nota da prova final e frequência) dos alunos das turmas que estejam sob sua responsabilidade.
- RF09 – o sistema deve permitir aos alunos consultar suas avaliações.

Exercícios resolvido

Sistema de Controle Acadêmico (SiCAd)

- Requisitos funcionais:

- RF10 – o sistema deve permitir à secretaria emitir diários de classe das turmas.
- RF11 – o sistema deve permitir à secretaria emitir históricos escolares dos alunos.
- RF12 – o sistema deve efetuar o cálculo da aprovação de alunos em turmas, sendo que, para ser aprovado, deve-se ter frequência mínima de 75%. Além disso, para aprovação sem prova final, a média das notas parciais deve ser maior ou igual a 70. para reprovação direta, esta média deve ser menor que 30. médias entre 30 (inclusive) e 70 (exclusive) colocam o aluno em prova final. Se a média da prova final com a média anterior for menor que 50, o aluno está reprovado, caso contrário, aprovado.
- RF13 – o sistema deve controlar a situação de um aluno, podendo estar matriculado, trancado, formado ou evadido.

Exercícios resolvido

Sistema de Controle Acadêmico (SiCAd)

- Cenário 01 – Descrição de um caso de uso: Matricular aluno

- Descrição: este caso de uso é iniciado pela secretaria quando requisita ao sistema matricular um aluno em uma determinada turma.
- Objetivo: possibilitar que ocorra a matrícula de alunos em turmas.
- Ator envolvido: Secretaria.

Exercícios resolvido

Sistema de Controle Acadêmico (SiCAd)

- Cenário 01 – Descrição de um caso de uso: Matricular aluno

- Interação entre sistema e ator:

Secretaria	Sistema
Solicita a matrícula de alunos em turmas.	
	Exibe uma interface com uma lista de turmas cadastradas, contendo descrição do curso, descrição da disciplina, ano, semestre e descrição da turma.
	Exibe uma lista de nomes de alunos cadastrados.
Seleciona uma turma e o aluno a ser matriculado, respeitando **RN01** e **RN02**.	
	Armazena a matrícula (**EX01**).
	Retorna o resultado da operação.
	Fecha a interface.

Exercícios resolvido

Sistema de Controle Acadêmico (SiCAd)

- Cenário 01 – Descrição de um caso de uso: Matricular aluno

- Previsão de exceções:

  - EX01 – caso o aluno já se encontre matricula naquela turma, a mensagem “Este aluno já possui matrícula na turma” é apresentada.

- Regras de Negócio:

  - RN01 – o aluno somente pode ser matriculado na turma de uma disciplina que não tenha pré-requisito(s) ou que já cursara o(s) pré-requisito(s), obtendo aprovação.

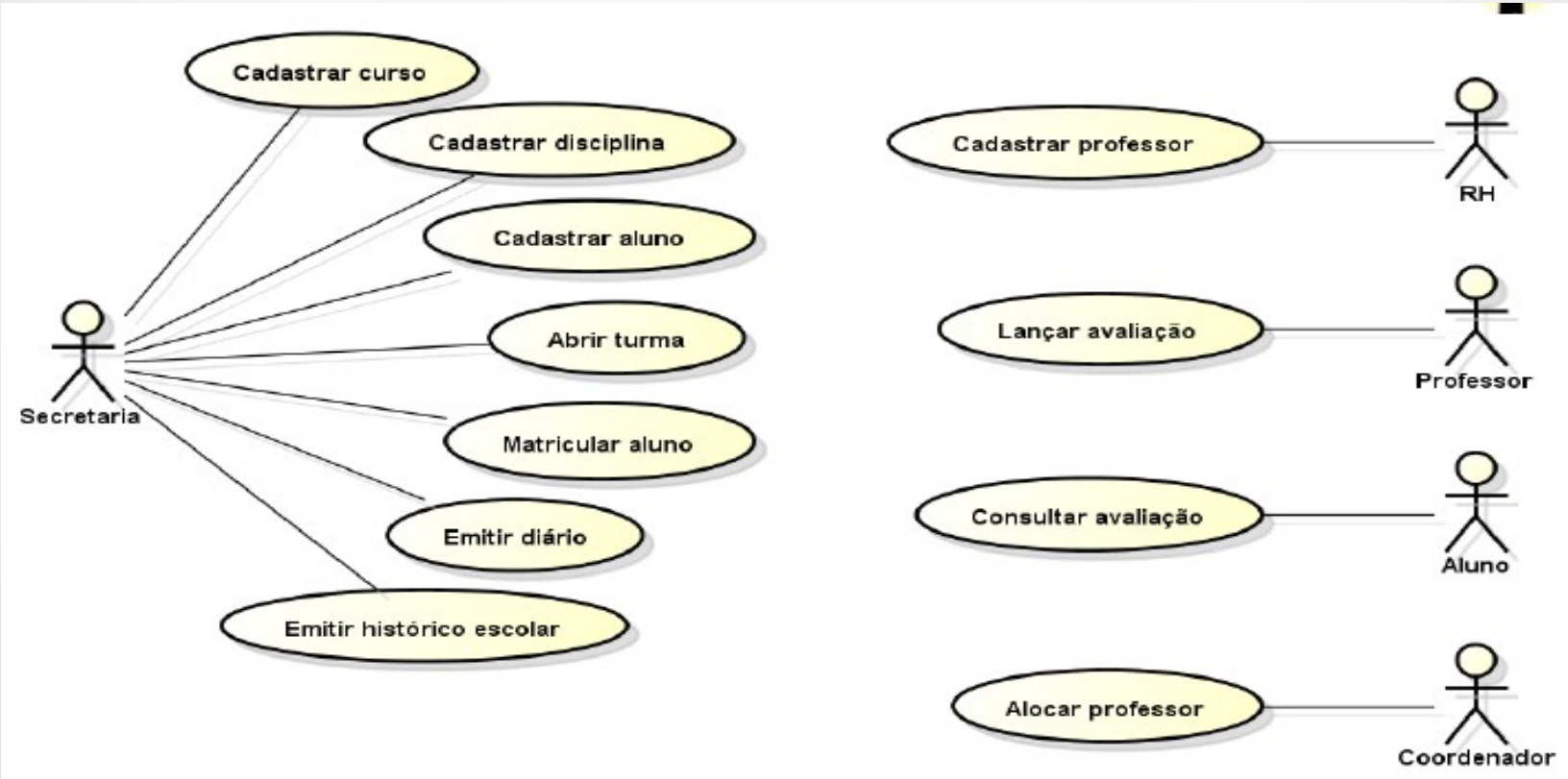
  - RN02 – o aluno somente pode ser matriculado em turmas de disciplinas do curso em que esteja matriculado.

Exercícios resolvido

Sistema de Controle Acadêmico (SiCAd)

- Cenário 01 – Descrição de um caso de uso: Matricular aluno

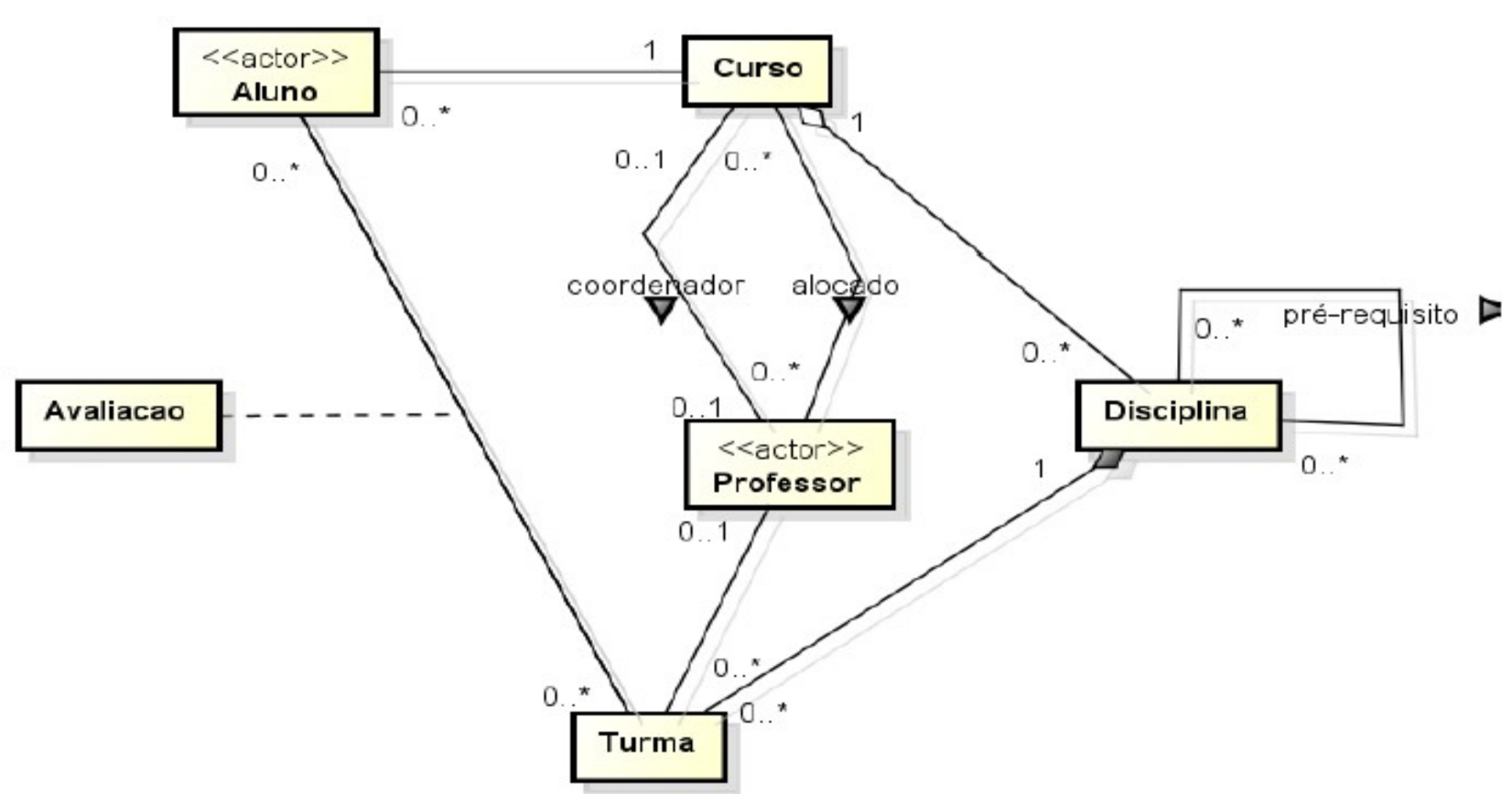
= Possível diagrama de caso de uso:



Exercícios resolvido

Sistema de Controle Acadêmico (SiCAd)

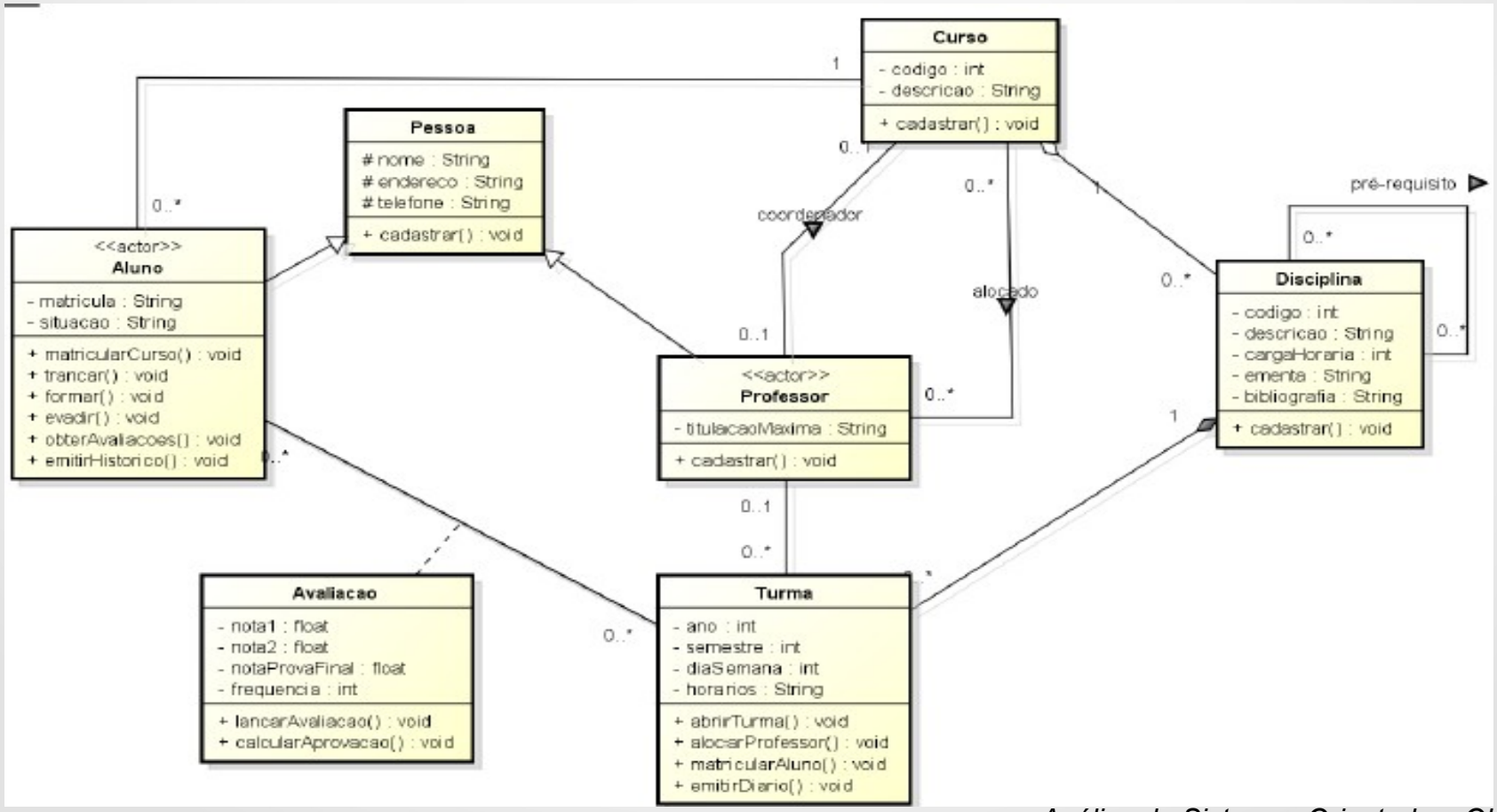
- Possível diagrama de Classes (nível de domínio)



Exercícios resolvido

Sistema de Controle Acadêmico (SiCAd)

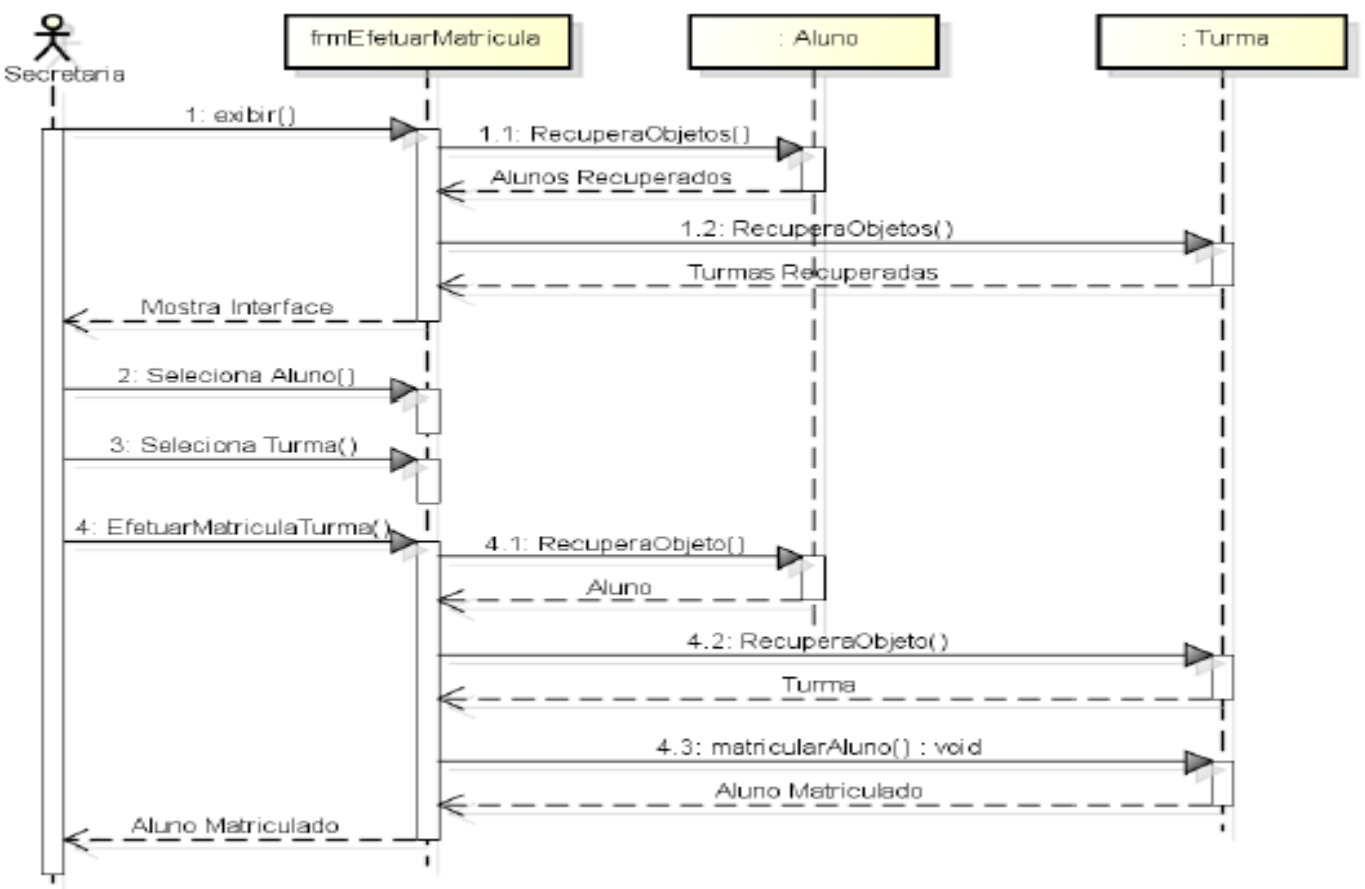
- Possível diagrama de Classes



Exercícios resolvido

Sistema de Controle Acadêmico (SiCAd)

- Possível diagrama de Sequência (Matricular aluno)



Exercícios resolvido

Dicas genéricas para a elaboração de um bom projeto usando UML

- Procurem por modelagem de sistemas parecidos.
- Procurem por modelos de documentos de outros sistemas.
- Leiam livros, revistas e sites sobre o assunto.
- Vejam exemplos em livros, artigos, revistas ou na Internet.