

## Unidade 5: Modelagem de classes de análise

De posse da visão dos casos de uso, os desenvolvedores necessitam prosseguir o desenvolvimento do sistema.

### 1 Colaboração entre objetos

Conceito: A funcionalidade externa de um sistema Orientado a Objetos (OO) é fornecida por meio da **colaboração entre objetos**. Externamente ao sistema, os atores visualizam resultados de cálculos, relatórios, confirmações, entre outras coisas. Internamente, os objetos do sistema colaboram uns com os outros para produzir os resultados visíveis de fora. Esta colaboração pode ser vistas sob os aspectos estático e dinâmico.

Conceito: O aspecto **estático** ou **estrutural** de uma colaboração permite compreender como o sistema é estruturado internamente para que as funcionalidades externamente visíveis sejam realizadas. Este aspecto é dito estático porque não apresenta informação sobre como os objetos do sistema interagem no decorrer do tempo. Ele é chamado estrutural porque representa a estrutura das classes de objeto componentes do sistema e as relações entre elas.

Conceito: O aspecto **dinâmico** de uma colaboração entre objetos descreve a troca de mensagens entre os mesmos e a sua reação a eventos que ocorrem no sistema. O aspecto dinâmico de uma colaboração é representado pelo Modelo de Interações, a ser estudado mais à frente.

Os aspectos estrutura e dinâmico de um sistema OO não são independente. Na verdade, a construção de um serve para adicionar detalhes ao outro.

Conceito: Assim como o Modelo de Casos de Uso representa o aspecto funcional de um sistema OO pela visão externa, o aspecto estático estrutural é representado pelo **Modelo de classes**. O Modelo de Classes é representado pelo **Diagrama de classes** e pela **descrição textual** associada ao mesmo.

### 2 Estágios do Modelo de classes

O Modelo de classes é utilizado durante a maior parte do desenvolvimento iterativo de um Sistema de Software Orientado a Objetos. À medida que o sistema é desenvolvido, o Modelo de classes é incrementado com novos detalhes. Há três estágios de abstração sucessivos pelos quais o Modelo de classes passa: Modelo de classes de análise, Modelo de classes de especificação ou de projeto e Modelo de classes de implementação.

1. O **Modelo de classes de análise** (MCA) representa as classes que se tornam evidentes na medida em que focamos a atenção sobre “o que” o sistema deve fazer. Por definição, esta fase do Modelo de classes não leva em consideração restrições inerentes à tecnologia a ser utilizada na solução do problema. O Modelo de classes de Análise é composto dos elementos identificados na análise do domínio e na análise da aplicação. Seu objetivo consiste em descrever o problema apresentado pelo sistema a ser desenvolvido num nível alto de abstração. O MCA e o Modelo de classes de análise são os dois principais modelos criados na atividade de Análise;
2. O **Modelo de classes de especificação** ou **Modelo de classes de projeto**, é um detalhamento do modelo de classes de análise. Este estágio normalmente exige a criação de novas classes pois o foco passa a ser o “como” o sistema realiza as funções previstas. Também ocorre a adição de detalhes às classes já definidas. Este modelo é construído na atividade de Projeto. Como analogia ilustrativa, podemos pensar nas fases de modelagem de uma casa. No modelo de análise pensamos em objetos tais como quartos, salas, banheiros. No modelo de projeto abordamos aspectos como encanamento, parte elétrica,

ou seja, detalhes mais físicos do que funcionais;

3. O **Modelo de classes de implementação** é um detalhamento do Modelo de classes de projeto e corresponde à implementação das classes em alguma linguagem de programação, normalmente uma linguagem OO (como java ou C++). Este modelo é construído na atividade de Implementação.

Nomenclatura proposta para os elementos do Modelo de classes:

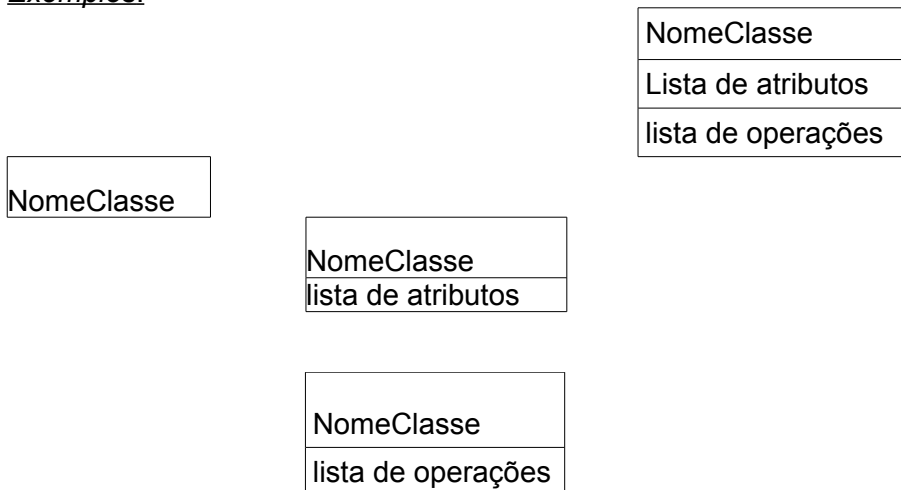
1. ao rotular identificadores, devem ser eliminados quaisquer espaços em branco e preposições;
2. Para nomes de classe e de relacionamento, todas as palavras componentes devem ser escritas começando por letra maiúscula. Exs.: ItemPedido, OrdemServiço,...;
3. Para nomes de atributos e de operações, deve-se escrever o começo da primeira palavra do nome do atributo em minúscula e as palavras subseqüentes começando com maiúscula. Exs.: precoUnitário, nome, dataNascimento, obterTotal,... Siglas ficam inalteradas. Exs.: CPF, CEP,...

### 3 Diagrama de classes

#### 3.1 Classes

Uma classe é representada por uma “caixa” com no máximo três compartimentos: o nome da classe – no singular e as palavras constituintes começando por maiúscula, os atributos e as operações – ações que um objeto sabe realizar.

Exemplos:



Conceito: Os **atributos** correspondem à descrição dos dados armazenados pelos objetos de uma classe. A cada atributo é associado um conjunto de valores que o atributo pode receber.

Conceito: As **operações** correspondem à descrição das ações que os objetos sabem realizar. Ao contrário dos atributos, para os quais cada objeto tem o seu próprio valor, objetos de uma classe compartilham as operações definidas para a mesma. O nome de uma operação normalmente contém um verbo e um complemento e é fechada por parênteses. A utilidade dos parênteses é determinada pelo nível de especificação.

Exemplo de representação de uma classe em diferentes níveis de abstração:

ContaBancária	ContaBancária número saldo dataAbertura	ContaBancária criar() bloquear() desbloquear() creditar()	ContaBancária número saldo dataAbertura ----- criar() bloquear() desbloquear() creditar() debitar()
ContaBancária -número : String -saldo : Quantia -dataAbertura : Date ----- +criar() +bloquear() +desbloquear() +creditar(in valor : Quantia) +debitar(in valor : Quantia)			

### 3.2 Associações

Conceito: Cada ocorrência de uma classe é chamada de **objeto** ou **instância**.

Conceito: Os objetos podem se relacionar uns com os outros. O **relacionamento entre objetos** torna possível a troca de mensagens. Portanto, os relacionamentos entre objetos permitem que os objetos colaborem entre si. A fim de produzir as funcionalidades requeridas do sistema.

Conceito: A **associação** entre objetos é um outro elemento do Diagrama de classes que representa os relacionamentos entre objetos durante a execução do sistema.

Notação: Uma associação é normalmente representada por um segmento que une as duas classes às quais os objetos pertencem. É importante notar que, embora representadas em relação às classes, as associações representam ligações entre objetos (ou instâncias) das classes envolvidas na associação.

Exemplos:

- 1) No domínio de vendas, um cliente compra produtos.

Cliente ————— Pedido

- 2) No domínio bancário, uma conta-corrente possui um histórico de transações.

ContaCorrente ————— HistóricoTransações

- 3) Num hotel, há vários hóspedes, assim como vários quartos. Os hóspedes ocupam os quartos.

Hóspede ————— Quarto

Propriedades: As associações possuem diversas propriedades: multiplicidade, nome, direção de leitura, papéis, tipo de participação e conectividade.

### 3.2.1 Multiplicidades

Conceito: As associações permitem representar a informação dos limites inferior e superior da quantidade de objetos aos quais outro elemento pode estar associado. Estes limites são chamados de **multiplicidade** da associação na terminologia UML.

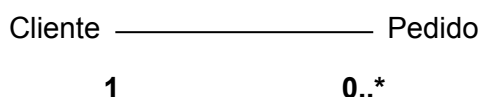
Este conceito é bastante semelhante ao conceito de cardinalidade associado a outras notações para modelos conceituais de bancos de dados.

Cada associação em um Diagrama de classes possui duas multiplicidades: uma em cada extremo do segmento que une as classes participantes da associação.

Simbologia para a representação de multiplicidades:

Nome	Símbolo
Apenas um	1 ou 1..1
Zero ou mais	0..*
Um ou mais	1...*
Zero ou um	0..1
Intervalo especificado	1 <sub>1</sub> ..1 <sub>s</sub>

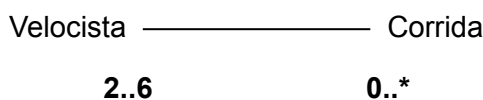
Exemplo1:



Interpretação: O “\*” à direita indica que um cliente pode estar associado a vários pedidos. Já o “0” indica que um cliente pode não estar associado a pedido algum.

O “\*” denota que não há limite superior predefinido para a quantidade máxima de objetos com os quais outro objeto pode se relacionar. No exemplo recém visto, um cliente pode fazer um número indefinido de pedidos. Isto, logicamente, é limitado de maneira física pela quantidade de memória em tempo de execução.

Exemplo2: Suponha a representação de informações sobre velocistas e as corridas das quais eles participam. Suponha, também, que em uma corrida deve haver no mínimo 2 e no máximo 6 participantes.



Simbologia adicional para a multiplicidade: É possível, também, indicar uma série de intervalos como multiplicidade.

**Conceito:** Embora haja infinitas possibilidades de associação entre objetos (na realidade, todas as combinações possíveis dos números naturais), há apenas três representações pertinentes nas quais todos os casos são agrupados: “**um para um**”, “**um para muitos**” e “**muitos para muitos**”. Denomina-se **conectividade** o tipo de associação entre duas classes.

Um para um: Empregado — Departamento  
1 0..1

**Um para muitos:** Empregado — Departamento  
0..\* 1

**Muitos para muitos:**

Empregado ————— Projeto

**0..\***                      **1..\***

**Conceitos:** Uma característica importante de uma associação está relacionada com a obrigatoriedade ou não da existência da associação entre todos os objetos das duas classes envolvidas nela. Este conceito é chamado **participação**. A participação de uma classe numa associação pode ser obrigatória ou opcional.

É preciso ter cuidado na definição de participações obrigatória,s pois elas “engessam” o modelo, o que dificulta futuras extensões do mesmo. Se houver dúvidas quanto à participação ser obrigatória, a decisão deve ser deixada para depois. Em particular, a informação obtida com a modelagem dinâmica do sistema serve para definir e/ou validar multiplicidades de associações.

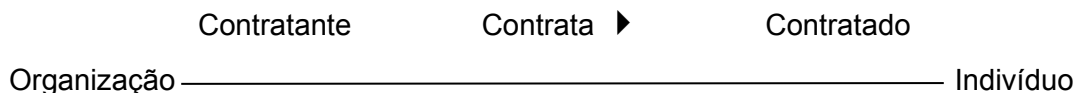
Análise e Projeto de Sistemas – Profa. Laura Sánchez García – 2o semestre de 2010

Conceito e notação: O **nome da associação** deve ser significativo. Ele deve ser posicionado em cima do segmento que determina a associação, no meio do caminho entre as duas classes envolvidas.

Conceito e notação: A **direção de leitura** serve para indicar como a associação deve ser lida, ou seja, qual das classes atuará como sujeito e qual como objeto.

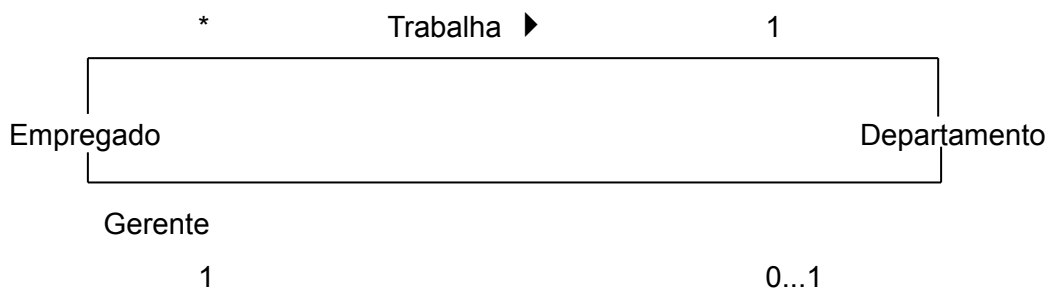
Conceito e notação: Quando um objeto participa de uma associação, ele tem um **papel** nela. Os papéis devem ser indicados próximos à classe e devem ser omitidos se eles forem passíveis de determinação pelos nomes das classes envolvidas e da própria associação.

Exemplo:



Pode haver associações entre pares de classes. Considere, por exemplo, duas classes: Empregado e Departamento. Considere, também, que um departamento precisa saber quais são seus empregados e quem é o seu gerente. Este caso determina as associações a seguir:

Exemplo de duas associações entre duas classes:

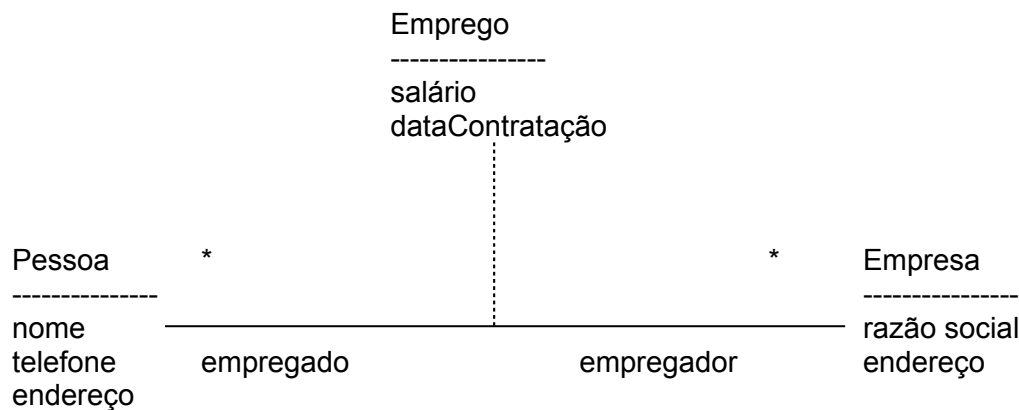


### 3.2.4 Classes associativas

Conceito: **Classes associativas** são classes que estão ligadas a associações, em vez de estarem ligadas a outras classes. Este tipo de classe normalmente aparece quando duas classes estão associadas e é necessário manter informações sobre a associação. Embora seja mais comum encontrar classes associativas com conectividade muitos para muitos, pode-se achar este tipo de classe com qualquer conectividade. De forma geral, este tipo de classe deve ser usada sempre que precisarmos representar a existência de uma informação que somente faz sentido se passível de aplicação a todos os objetos de duas outras classes.

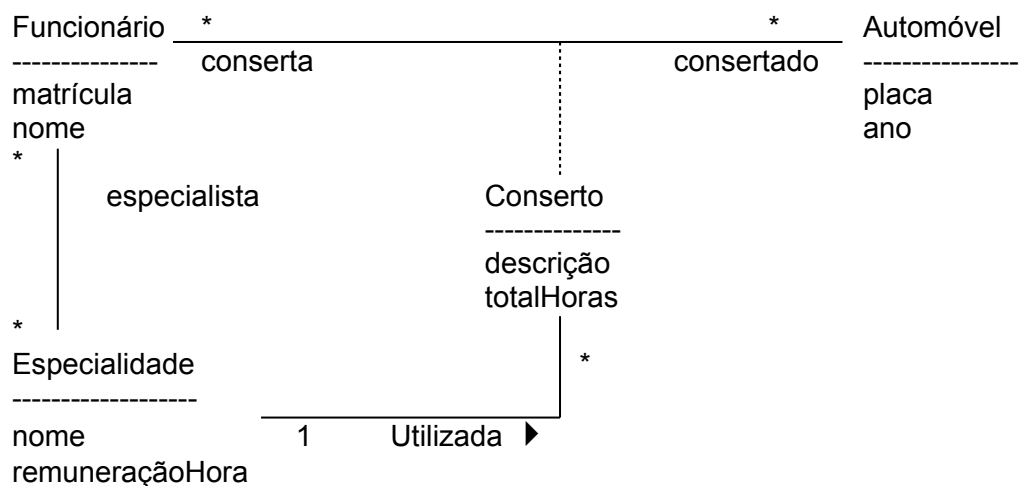
Notação: Classes associativas são indicadas da mesma forma que qualquer outra classe, mas são ligadas por linha tracejada a uma associação.

Exemplo: Considere as classes Pessoa, Empresa e Emprego. A classe associativa Emprego permite saber, para cada par de objetos (empregado, empregador), qual o salário e a data de admissão do empregado.



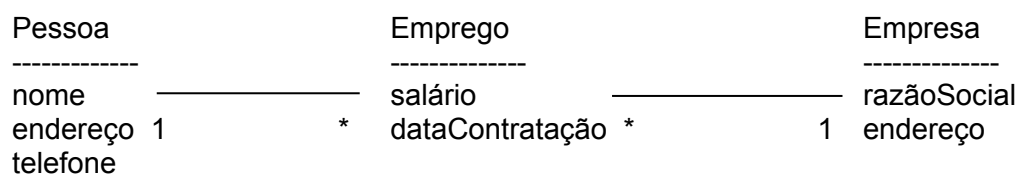
Uma classe associativa pode participar de outras associações ou relacionamentos.

Exemplo de classe associativa relacionada com outras classes que não as da própria associação:



Representação alternativa de classes associativas: Classes associativas podem ser eliminadas da representação incluindo a classe associativa como classe intermediária entre as duas classes que determinavam a associação.

Exemplo de representação alternativa:

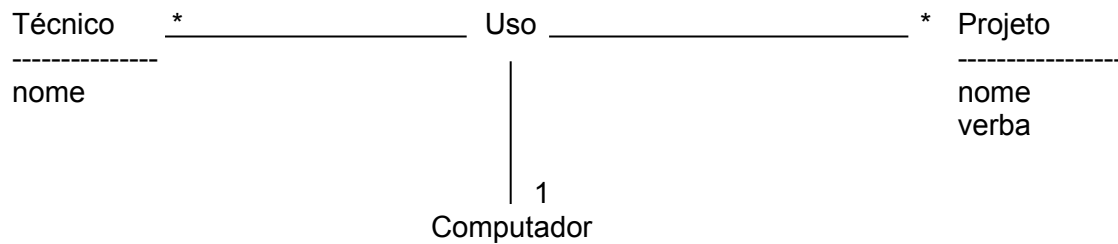


### 3.2.5 Associações ternárias

Conceito: Define-se como **grau de uma associação** como a quantidade de classes envolvidas na mesma. Na maioria dos casos de modelagem, as associações são binárias. Entretanto, o grau pode ser maior em alguns casos.

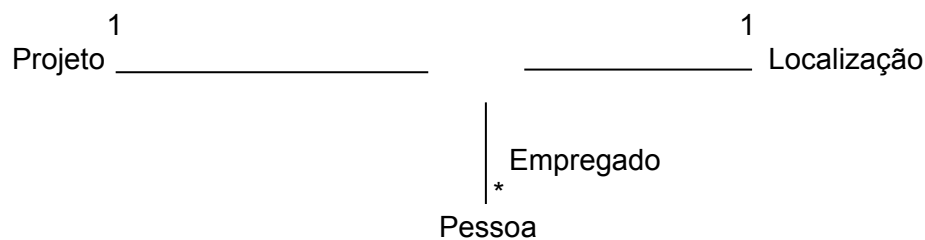
Dada uma associação ternária entre A, B e C, a multiplicidade do extremo C indica quantos objetos da classe C podem estar associados a um par de objetos (a,b), onde a é um objeto (ou instância) da classe A e b é um objeto (ou instância) da classe B.

Exemplo 1 de associação ternária:



O Diagrama do exemplo significa que para cada par instanciado (técnico, projeto) é utilizado um único computador. Em outras palavras, um técnico utiliza um único computador para cada projeto.

Exemplo 2 de associação ternária:



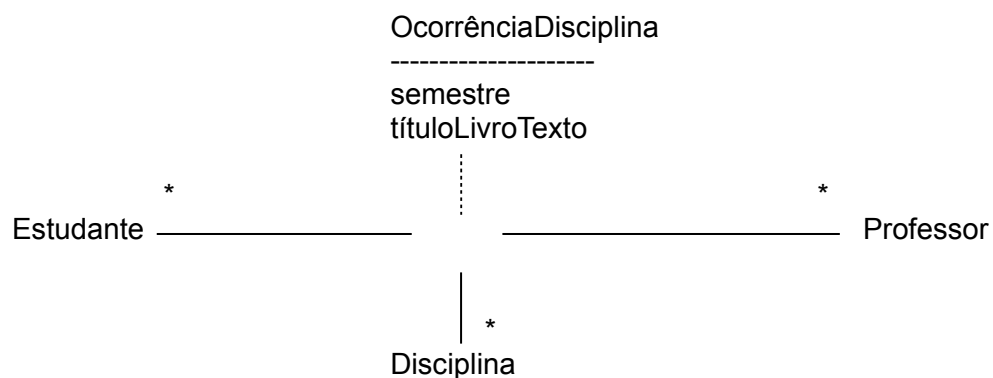
Este diagrama significa que só há um projeto ao qual cada par (pessoa, localização) está associado. Em outras palavras, cada pessoa localizada num determinado lugar só atua lá um único projeto.

Por outro lado, é possível dizer, também, que cada pessoa que atua num projeto, trabalha nele numa única localização.

Finalmente, os conceitos de classe associativa e de associação ternária podem ser integrados.

Exemplo de integração de classe associativa e associação ternária:

Neste exemplo, há três classes ligadas por uma associação ternária (Estudante, Professor e Curso). Além disso, existe a classe associativa OcorrênciaDisciplina, que permite armazenar informações sobre cada instância da associação anterior.



Este diagrama significa que várias disciplinas podem ter um mesmo par (estudante, professor). Em outras palavras, um professor pode dar aula a um mesmo estudante em várias disciplinas.



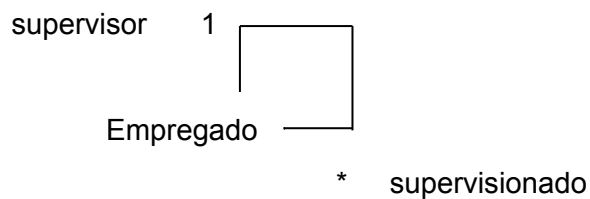
Vários professores podem associados a um par (estudante, disciplina). Ou seja, mais de um professor pode dar aula para um único estudante numa dada disciplina.

Vários estudantes podem estar associados ao par (professor, disciplina). Em outras palavras, uma turma de uma disciplina, alocada a um professor, pode ter vários alunos.

### 3.2.6 Associações reflexivas

Conceito: Uma **associação reflexiva** (também denominada “**auto-associação**”) associa objetos da mesma classe. Diferentes objetos da classe podem ter um papel distinto nessa associação.

Exemplo:



Observação: É preciso ter cuidado na interpretação do conceito de associação reflexiva. Ela não indica que um objeto se relaciona com ele mesmo, mas, sim, que um objeto de uma classe se associa com outro(s) objeto(s) da mesma.