

# **Manual de Modelagem Orientada a Objetos**

Fábio Nogueira de Lucena  
fabio@inf.ufg.br  
Instituto de Informática (UFG)  
Copyright © 2007

# Introdução

Este manual oferece uma abrangente visão de modelagem de domínio. Isto inclui a definição, quando tal modelagem é empregada ao longo do processo de desenvolvimento de software, como desenvolvê-la e, em particular, um extenso conjunto de exemplos. O objetivo deste manual é apresentar um conjunto significativo de oportunidades de aprendizado de modelagem de orientação a objetos por meio de cenários típicos.

## Modelagem de domínio

Modelo de domínio é o artefato no qual são descritas classes conceituais relevantes a um domínio de interesse bem como as associações entre estas classes. Não são classes de software, embora sejam uma fonte de inspiração para classes a serem implementadas em uma linguagem de programação como Java ou VB .NET, por exemplo. Esta inspiração, contudo, deve ocorrer durante a fase de projeto. Alguns tratam o modelo de domínio como um dicionário visual de abstrações de um dado domínio.

Modelo de domínio é o principal artefato produzido pelo que geralmente é referenciado na literatura como análise orientada a objetos. Geralmente é produzido concomitantemente com a eliciação dos requisitos. A investigação dos requisitos é imprescindível para a obtenção do modelo de domínio correspondente. Este artefato, juntamente com os requisitos de software, são os principais insumos para projetistas de software.

Modelos de domínios podem ser criados através do emprego de algumas técnicas básicas, que são apresentadas neste texto, juntamente com diretrizes para a obtenção de um modelo adequado e um processo simples a ser seguido. Primeiro trataremos da identificação de classes e, posteriormente, da identificação das associações relevantes entre elas.

É comum a confusão de classes conceituais com entidades de bases de dados. Não tenha dúvidas, são elementos distintos, com propósitos distintos. As entidades de bases de dados têm o propósito de armazenar informações em meio persistente. A preocupação é com o armazenamento, a organização, a recuperação e outras questões da natureza de dados, como formato e eficiência, por exemplo. Quando se constrói um modelo de domínio estas questões não são relevantes. Tratar uma classe conceitual como uma tabela é um erro. Tratar associações entre classes como relacionamentos entre entidades de dados também é um erro. Classe conceitual representa um conceito relevante para um domínio. Nada mais.

Descobrir classes não é assunto novo. Uma das abordagens mais conhecidas é denominada de CRC (*Classes, Responsibilities, Collaborations*). Existem livros e tutoriais sobre o assunto. Cartões CRC foram propostos por Kent Beck e Ward Cunningham no artigo [\*A Laboratory for Teaching Object-Oriented Thinking\*](#), publicado nos anais da OOPSLA '89. Tanto este artigo quanto tutoriais podem ser encontrados facilmente na Internet.

Neste texto é aplicada uma proposta diferente de CRC. Recomendo a leitura do livro *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process*, Craig Larman, Prentice-Hall, 2<sup>nd</sup>. Edition, 2002. Trata-se de texto onde o autor apresenta o assunto com rara legibilidade. Em *Developing Applications with Visual Basic and UML*, Paul R. Reed Jr., Addison-Wesley, 2000, também são fornecidas várias orientações acerca da modelagem de domínio.

## Identificando classes

### ***Onde procurar por classes?***

Relatórios, documentos, textos, outros sistemas e qualquer outro artefato que faça parte do domínio do problema. O analista investiga estes artefatos e identifica o vocabulário relevante. Cada léxico deste vocabulário dá origem a uma classe conceitual no modelo de domínio.



*Analista investigando artefatos  
à procura de classes conceituais.*

### ***Como criar um modelo de domínio?***





A experiência em um domínio é uma forte aliada, mas há outras abordagens que também contribuem:





- Empregue uma lista de categorias de classes
- Identifique substantivos a partir da especificação dos requisitos
- Faça uso de padrões de análise





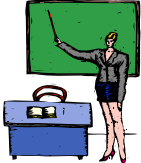

### **Lista de categorias de classes**



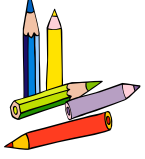



Não interprete as categorias abaixo como fronteiras bem definidas. De fato, uma mesma classe conceitual pode ser tratada como exemplo de mais de uma categoria. Esta classificação tem o único propósito de orientar o analista na elaboração de modelos de domínio.

<i>Objetos físicos ou tangíveis</i>	<i>Especificações, projetos ou descrições de coisas</i>
 <div data-bbox="406 1742 572 1825"> Medalha </div>	 <div data-bbox="984 1733 1289 1816"> DescricaoProjeto </div>

<i>Objetos físicos ou tangíveis</i>	<i>Especificações, projetos ou descrições de coisas</i>
 <div>Crucifixo</div>	 <div>Escritura</div>
 <div>Despertador</div>	 <div>Imagem dimensões formato</div>

<i>Lugares</i>	<i>Transações</i>
 <div>Campo</div>	 <div>Namoro</div>
 <div>Cozinha</div>	 <div>Venda</div>

<i>Papéis de pessoas</i>	<i>Contêineres de coisas</i>
 <div>Câmera</div>	 <div>Barril</div>
 <div>Policia</div>	 <div>Saco</div>
 <div>Professora</div>	 <div>Caminhão</div>

<i>Elemento de um conjunto</i>	<i>Conceitos abstratos</i>
 <div>Jogador</div>	 <div>Amor</div>
 <div>Lápis</div>	 <div>Medo</div>
 <div>Carta</div>	 <div>Carisma</div>

Muitas outras categorias podem ser utilizadas além daquelas ilustradas acima. Organizações: DepartamentoVenda, Escola. Eventos: Reunião, Aniversário, Decolagem, Eleição. Processos: AvaliaçãoProposta, TurbinandoCarro, ValidarDiploma. Regras e políticas: CancelamentoDisciplina, RequisiçãoDiplomaSegundaVia, CadastramentoRegra. Catálogos: CatálogoCurso, CatálogoLivro, CatálogoProduto. Registros de finanças, trabalho, contratos, avaliações: Recibo, Pedido, NotaFiscal, ContratoTrabalhista, Prova, OrdemServiço. Serviços financeiros: Empréstimo, Extrato, LinhaCrédito, Poupança. Manuais, documentos, referências: Artigo, Lei, Ajuda, ListaPreçosDiários, EspecificaçãoProduto, Livro.

Convém ressaltar que as categorias acima não formam uma lista exaustiva. Também não há interesse em, dado um modelo, classificar as classes conceituais conforme as categorias ilustradas acima. Tais categorias têm como objetivo apenas facilitar a identificação de classes conceituais.

### Identifique substantivos a partir dos requisitos

Uma análise lingüística também pode ser útil à identificação de classes conceituais além de ser atrativa dada a simplicidade. O analista deve ter em mente que um mapeamento de substantivos para classes conceituais é uma abordagem útil mas que apresenta limitações. Observe o fluxo básico do caso de uso abaixo para um futuro sistema bancário onde estão sublinhados os substantivos. Em tempo, *caso de uso* é uma abordagem que podemos empregar para identificar e registrar requisitos funcionais e, portanto, um dos artefatos dos quais podemos extrair classes conceituais. Um exemplo é fornecido abaixo.

#### Caso de Uso: *Autorizar liberação de crédito*

1. Gerente tem detalhes do pedido de crédito do cliente para o qual deseja liberar o crédito.
2. Gerente requisita liberação de certo valor de crédito para o cliente em questão.
3. Sistema pede confirmação do gerente e do valor do crédito.
4. Gerente autoriza o crédito.
5. Sistema atualiza crédito do cliente e emite extrato.

Nem todos os substantivos sublinhados acima identificam classes conceituais. Por exemplo, gerente não será uma classe conceitual. Em outros cenários poderia até identificar um papel desempenhado por um ser humano. Neste caso, gerente é um ator, elemento externo ao sistema. Detalhes de casos de uso podem ser obtidos em outras fontes como *Applying Use Cases: A Practical Guide*, 2<sup>nd</sup> edition, Addison-Wesley, 2001.

O substantivo detalhes também não dá origem a uma classe conceitual. Neste caso refere-se a elementos do pedido de crédito, por exemplo, valor, data da requisição, finalidade do empréstimo e outros. Observe que valor é outro substantivo que, conforme vimos, trata-se de um detalhe de um pedido. Da perspectiva orientada a objeto, neste caso, valor é melhor tratado como atributo de Crédito, em vez de uma classe conceitual. Tratar uma informação como atributo ou classe conceitual é a discussão da seção seguinte. Confirmação também pode ser eliminado por representar muito mais a interação entre o gerente e o sistema do que elemento conceitual relevante para o domínio. O mesmo ocorre com o substantivo liberação. Se é relevante ressaltar alguma confirmação, então provavelmente é melhor identificar a classe Autorização. Um dos atributos relevantes desta classe é a data da autorização, por exemplo.

Retirando as classes conceituais induzidas erroneamente por alguns dos substantivos sublinhados, tem-se o modelo abaixo.

Crédito	Autorização	Pedido de crédito
valor	data	valor

## Padrões de análise

Quando o assunto é padrões de análise, *Analysis Patterns: Reusable Object Models*, Martin Fowler, Addison-Wesley, 1996 é uma fonte a ser consultada. Lá você encontrará padrões de análise adequados em cenários recorrentes. Mesmo que não seja, os padrões lá apresentados serão úteis como exercício intelectual relevante.

## Classe ou atributo, como decidir?

Se houver impasse quanto a tratar um elemento X como classe conceitual ou atributo, então trate-a inicialmente como classe. Para evitar a criação de classes indesejáveis, que deveriam ser tratadas como atributos, pode-se empregar filtros, conforme a seção seguinte.

## Use filtros para eliminar classes espúrias

Em tempo, conforme o Houaiss, espúrio é não genuíno, suposto, hipotético. O fato é que nem todos os substantivos que podem ser identificados em um caso de uso e mesmo “classes” identificadas conforme algumas categorias fornecidas acima não necessariamente merecem o tratamento de classes. Para separar o joio do trigo podemos empregar filtros. Estes filtros não são regras de ouro, mas oferecem contextos para os quais alguns substantivos não são apropriadamente tratados como classes.

*Redundância.* Dois ou mais substantivos podem representar a mesma coisa e, neste caso, identifique

o mais apropriado e elimine os demais. Por exemplo, dentre aluno, estudante e discente este filtro pode eliminar estudante e discente, ficando apenas a classe aluno.

*Atributo.* Conforme sugere Craig Larman, *se nós não imaginamos uma classe conceitual X como um número ou um texto, então X é provavelmente uma classe conceitual e não um atributo*. Este especialista ainda sugere: *na dúvida, crie uma classe conceitual*.

Por exemplo, a descrição de um logradouro pode ser tratada como um atributo. Trata-se de um simples texto. Um endereço, por outro lado, é composto de logradouro, CEP, número e outros elementos que não são adequadamente tratados como um texto, o que sugere que endereço merece o tratamento de classe. É observando a orientação fornecida acima que somos inclinados a tratar preço e altura, por exemplo, como atributos, enquanto livro e avião são adequadamente tratados como classes. Em tempo, livro é publicado por uma editora, possui um autor, tem um título, tem uma edição, tem ano de publicação e outras. Provavelmente encontraremos um conjunto bem maior de elementos se considerarmos um avião.

A diretriz fornecida, contudo, nem sempre é suficiente para elucidar alguns casos. Cartão de crédito, ou melhor, número de cartão de crédito é um atributo ou uma classe?

Depende. Se o número é visto como apenas uma seqüência de caracteres, provavelmente é adequada a visão dele como um atributo. Contudo, podemos estar interessados na formatação deste número conforme vemos impresso em um cartão, podemos estar interessados na validação deste número e, portanto, só para ficar nestas duas operações, já identificamos funcionalidade que justifica a existência de uma classe.

*Irrelevante.* Substantivos podem não estar relacionados ao escopo de interesse. Quando se deseja informatizar uma farmácia, por exemplo, vassoura não é um substantivo que dê origem a uma classe. Provavelmente outros substantivos como medicamento, prescrição e outros dêem origem a classes.

*Operação.* Um substantivo pode ser uma responsabilidade de uma classe. Por exemplo, o cálculo do total de uma nota fiscal pode ser uma responsabilidade da classe que representa o substantivo nota fiscal e não uma classe propriamente dita. O total de uma nota é adequadamente modelado como uma responsabilidade da classe nota fiscal e, portanto, em fase posterior do desenvolvimento esta responsabilidade será atribuída a esta classe através da definição de uma operação (método). Operações geralmente não são definidas em um modelo de domínio.

*Papel.* O substantivo “empregado” talvez não dê origem a uma classe, desde que o substantivo “funcionário”, uma classe legítima, seja definida. Neste caso, empregado é apenas a identificação do papel que uma instância da classe Funcionário pode desempenhar quando ligada a uma instância da classe Empresa, por exemplo. Caso “empregado” seja considerado uma classe, teremos uma redundância com a existência da classe Funcionário e, neste caso, uma delas teria que ser eliminada. Gerente (da classe Funcionário), professor (da classe Docente), entre outros, são papéis legítimos desempenhados por instâncias das classes fornecidas entre parênteses.

*Evento.* Ocorrências específicas em um determinado instante de tempo são geralmente conhecidas por eventos. Por exemplo: ligar alarme e queda de energia são duas ocorrências “instantâneas”, são eventos. Um evento pode ser tratado, em alguns casos, como uma classe. Neste caso, tanto LigarAlarme quanto QuedaEnergia representariam estas ocorrências ou eventos.

## Processo de criação de modelos de domínio

A criação de um modelo de domínio pode ser orientado pela execução das seguintes atividades:

1. Liste classes conceituais candidatas. Use a lista de categorias e a identificação de substantivos para tal finalidade.
2. Desenhe o modelo de domínio correspondente. O diagrama de classes (UML) neste primeiro instante contém apenas classes.
3. Acrescente associações entre as classes.
4. Acrescente atributos para que as classes identificadas forneçam as informações desejadas.

### ***Diretrizes***

1. Use nome no singular para uma classe conceitual.
2. Use um nome que identifique um único objeto em vez de uma coleção deles.
3. Um modelo de domínio é um modelo da interpretação que possuímos de um domínio. Isto sugere que é imprescindível a identificação tanto do domínio quanto da interpretação que se deseja. Por exemplo, o mesmo domínio de “feiras de automóveis” pode ser observado de várias perspectivas, como a econômica, o impacto no público e, em particular, aquela que se tem interesse em investigar ao contrário de uma tentativa impossível de modelar estas feiras como de fato e realmente são.

## Identificando associações

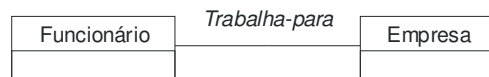
Em uma perspectiva orientada a objetos de determinado domínio vê-se uma coleção de objetos que trocam mensagens entre eles. A troca de mensagens pressupõe a existência de relacionamentos. O relacionamento entre objetos é descrito através de uma associação entre as classes correspondentes. Em um modelo de domínio, após a identificação dos conceitos relevantes, modelados como classes conceituais, voltamos nossa atenção para a identificação de relações relevantes entre as classes.

Segundo o dicionário Houaiss, relevante é o que tem importância; que se sobressai; de grande valor ou interesse. Por mais impreciso que seja, já fizemos uso deste termo para identificar as classes conceituais, afinal, classes conceituais que não são relevantes não devem fazer parte de modelos de domínios. Uma estratégia similar é seguida aqui para a identificação de associações entre os conceitos identificados.

Afinal, o que é uma associação? Associação registra um relacionamento semântico entre classes. A forma mais comum de associação é entre duas classes. Há também associações de uma classe para ela própria, assim como associações onde mais de duas classes estão envolvidas.

Uma associação estabelece uma possível conexão entre instâncias das classes envolvidas. Por exemplo, em uma associação entre a classe *Funcionário* e a classe *Empresa*, uma instância de *Funcionário*, por exemplo, João da Silva, trabalha para a empresa SuperProdutos, uma instância de *Empresa*. Neste caso, esta associação pode ser rotulada por *Trabalha-para* de *Funcionário* para *Empresa*, para esclarecer qual o relacionamento que se deseja registrar entre instâncias destas classes. Veja ilustração abaixo.





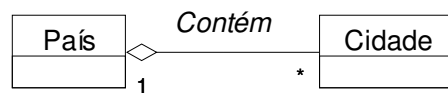
Entre duas classes podem existir vários relacionamentos entre as respectivas instâncias. Cada relacionamento pode ser registrado através de uma associação. Por exemplo, entre *Funcionário* e *Empresa* pode existir o relacionamento *Gerente*. Neste caso, a instância de *Funcionário* ligada à instância de *Empresa* por esta associação representaria o gerente da empresa em questão. Naturalmente, neste caso, convém observar, nem toda instância de *Funcionário* está ligada à alguma instância de *Empresa*.

Associações são estabelecidas apenas quando o significado correspondente deve ser preservado. Por exemplo, se for útil identificar os funcionários de uma empresa que não usam o serviço de almoço oferecido ou, noutras palavras, quais as empresas para as quais um determinado funcionário faz ou fez uso do serviço de refeições oferecido pela empresa, então será necessário acrescentar mais uma associação entre *Funcionário* e *Empresa*, provavelmente rotulada por *Usa-serviço-refeição*.

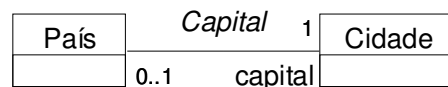
## Prática 1

Para cada descrição abaixo forneça a modelagem correspondente empregando a UML.

1. *Em um país há várias cidades.* Dado uma instância de País temos zero ou mais instâncias de Cidade correspondentes. Dada uma instância de Cidade há uma instância de País correspondente (exatamente uma). Pode-se modelar esta relação com uma agregação, que representa um relacionamento todo/parte, como normalmente as pessoas imaginam existir entre um país e as cidades deste país. Esta agregação é identificada por Contém, conforme o diagrama abaixo. Convém ressaltar a interpretação precisa da cardinalidade oferecida do lado da classe Cidade. O asterisco indica zero ou mais, noutras palavras, o modelo abaixo admite a existência de instância de País sem que este esteja associado a uma Cidade. Em vez do asterisco poderia ser empregada a cardinalidade 1..\*, o que significaria pelo menos uma instância de Cidade. Ou seja, não teríamos um país sem a existência de uma cidade.

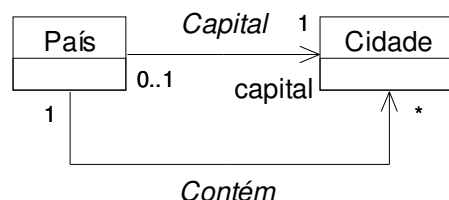


2. *Uma das cidades de um país é a capital.* Dada uma instância de País necessariamente teremos uma instância da classe Cidade correspondente conforme a associação identificada por Capital. Ou seja, todo país necessariamente possui uma cidade que é a capital, exatamente uma instância. A instância da classe Cidade desempenha o papel de capital, conforme ilustrado. No sentido inverso, dada uma instância da classe Cidade pode existir ou não uma instância da classe País correspondente,



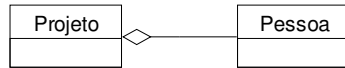
conforme a cardinalidade 0..1. Observe que se a cardinalidade fosse 1, então teríamos um modelo no qual toda instância de Cidade é capital de algum país, o que não condiz com a realidade.

3. *Em um país há várias cidades e uma delas é a capital.* Uma agregação reforça a relação todo/parte, afinal, cidades pertencem a países. Pode-se especular, contudo, que a associação abaixo também representa este tipo de relacionamento sem nenhuma perda semântica. As associações estão orientadas. Quando não há orientação elas são ditas bidirecionais. A semântica é simples, *é fácil identificar as cidades existentes em um país*, assim como *é fácil identificar a capital de um país*. Tal facilidade não existe quando desejamos saber se uma dada cidade é capital ou mesmo em qual país está localizada esta cidade. A cardinalidade ou multiplicidade no lado da classe País é diferente para as associações Capital e Contém. Para a associação Capital a multiplicidade é 0..1, indicando que uma cidade pode ou não ser capital de algum país. Para a associação Contém a multiplicidade é 1, ou seja, não existe cidade que não seja parte de algum país, pelo menos conforme o modelo.

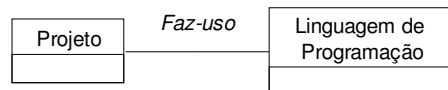


4. *Um projeto envolve várias pessoas.* A agregação indica que pessoas são partes de projetos. Neste exemplo a multiplicidade não é fornecida. Há pelo menos dois casos para serem analisados. Em um deles a multiplicidade não é fornecida por não ser relevante no contexto. Em muitos casos é natural

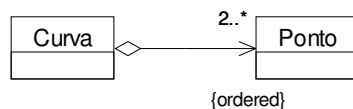
identificar as classes e associações entre estas, o que é relevante, e deixar para momento posterior a identificação precisa de quantas instâncias de uma classe se relacionam com uma instância de outra classe. No outro caso a ser analisado, a multiplicidade não foi fornecida porque está sendo adotada a multiplicidade padrão da UML. O padrão \*, ou seja, zero ou mais instâncias. Neste caso a interpretação seria que um projeto pode estar associado a zero ou mais pessoas assim como uma pessoa pode estar associada a zero ou mais projetos.



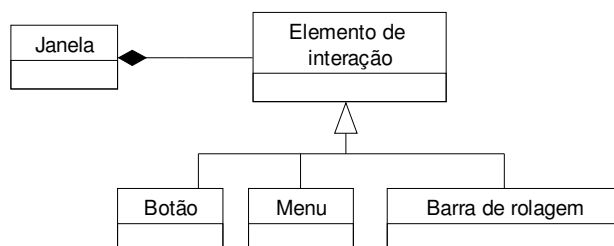
5. *Um projeto de software pode empregar várias linguagens de programação.*



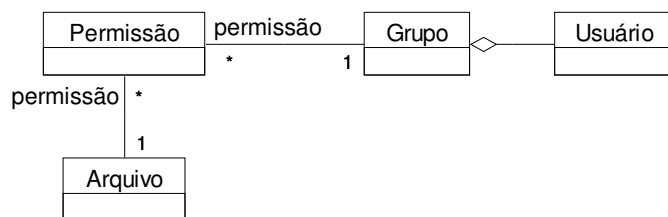
6. *Uma curva pode ser definida como uma coleção de pontos ordenados.* O modelo informa que toda curva está associada a pelo menos dois pontos ordenados. Podem existir bem mais de 2 pontos, mas todos eles estão ordenados, o que assegura a possibilidade de reconstrução da curva em questão. O modelo indica que as instâncias da classe Ponto associadas a uma instância da classe Curva estão ordenados pela restrição {ordered} associada ao extremo direito da associação apresentada.



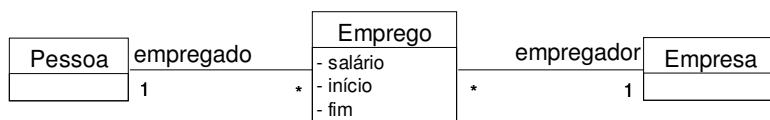
7. *Uma janela gráfica (interface com o usuário) compreende vários elementos de interação. Por exemplo, botões, menus e barras de rolagem.* Embora uma instância de Janela possa ser interpretada como uma agregação de instâncias de Elemento de interação, convém ressaltar um relacionamento “mais forte” entre instâncias desta classe. Quando uma instância de Janela é criada, as instâncias correspondentes da classe Elemento de interação também são criadas. Quando uma instância de Janela é destruída, naturalmente as instâncias da classe Elemento de interação associadas também são destruídas. Quando objetos apresentam este relacionamento todo/parte com semântica mais rigorosa que aquela da agregação, empregamos uma composição, conforme ilustrado no diagrama. Em tempo, os elementos de interação de uma janela considerados no modelo abaixo incluem botões, menus e barras de rolagem, conforme as classes exibidas.



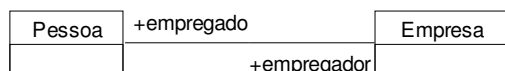
8. *Um arquivo possui permissões de acesso. Cada permissão está associada a um grupo (de usuários).* Conforme o diagrama, toda Permissão está associada necessariamente a um Grupo e a um Arquivo. Naturalmente, dada uma instância de Arquivo, podem existir várias instâncias de Permissão, cada uma delas conforme a instância de Grupo associada. Observe que pode não existir nenhuma permissão correspondente para um dado arquivo. No outro sentido a interpretação é a mesma, ou seja, um Grupo possui uma instância de Permissão para cada Arquivo. Por último, um Grupo define um conjunto de instâncias de Usuário, cada uma delas representa um usuário que é membro do grupo em questão, daí o emprego da agregação.



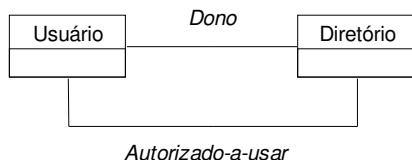
9. *Pessoas trabalham para empresas por um determinado salário e intervalo de tempo. Conforme o modelo, uma empresa oferece vários empregos, cada um deles ocupado exclusivamente por um empregado. No sentido inverso, uma pessoa pode ocupar vários empregos, cada um deles oferecido por um empregador. Cada instância de Emprego possui informações pertinentes à vaga em questão.*



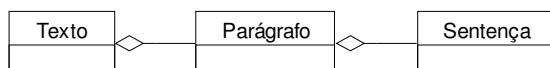
10. *Uma pessoa (empregado) trabalha para uma empresa (empregador). No modelo abaixo, ao lado do papel empregado e empregador há o símbolo +. Isto significa que os empregados de uma determinada empresa podem ser obtidos além das fronteiras da classe Empresa, ou seja, não se trata de informação privada de uma instância desta classe. Analogamente, dada uma instância de Pessoa, podemos saber quais as empresas pelas quais o ser humano em questão ofereceu os seus serviços além das fronteiras da classe. Em geral este modificador de acesso associado aos papéis não é fornecido. São elementos específicos de implementação e, portanto, em geral, podem ser decididos pelos responsáveis pela construção propriamente dita do software em questão.*



11. *Um usuário é o “dono” de um diretório. Cada diretório pode ser consultado por usuários autorizados. Dada uma instância de Usuário temos instâncias de Diretório que são propriedades do usuário em questão. Dado um Diretório, aqueles autorizados (instâncias de Usuário) devem estar ligados pela associação Autorizado-a-usar.*



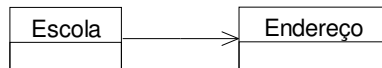
12. *Um texto é uma combinação de parágrafos que, por sua vez, são combinações de sentenças. Talvez você imagine que uma composição seja mais apropriada, por ressaltar que a destruição do texto significa a destruição dos parágrafos e respectivas sentenças e que, em outro sentido, quando se criam as sentenças e os parágrafos o texto está sendo criado. Por outro lado, apesar de não usual, o modelo abaixo permite compartilhar parágrafos e sentenças, possibilidade que desaparece caso seja empregada a composição.*



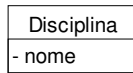
13. *Toda escola possui um endereço. Embora alguns especulem a definição de endereço como atributo da classe Escola, a proposta abaixo ressalta a distinção entre os conceitos envolvidos. Enquanto entidade, uma instância de Escola possui atributos como nome, capacidade de alunos e outros. Endereços nos dias atuais inclui CEP, logradouro e outros. Tratar estas informações como atributos*

espalhados por várias classes é inconveniente óbvio de não tratar endereço como uma classe.

14. *Toda disciplina possui um nome.* Observe que nome não é considerado um conceito que mereça o



tratamento de classe. Um atributo fornece um detalhe intimamente ligado à classe na qual é definido. Infelizmente não existe um conjunto de diretrizes que, uma vez seguidas, identificam com clareza, se uma determinada informação deve ser tratada como atributo ou como classe. Felizmente, a prática tem mostrado que esta questão torna-se cada vez menor com o aumento da experiência do responsável pela modelagem.



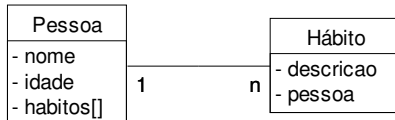
## Prática 2

Para cada descrição abaixo forneça a modelagem correspondente empregando a UML.

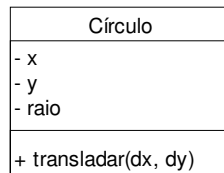
1. Uma pessoa possui hábitos. Talvez nenhum hábito. Todo hábito está associado a uma pessoa, ou seja, conforme modelado abaixo, hábitos não são compartilhados, cada um possui os seus.



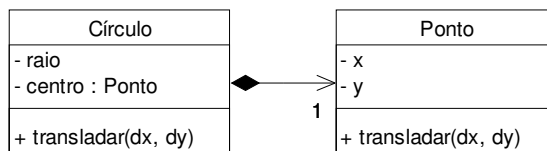
2. Uma pessoa possui um nome, idade e hábitos. Um hábito possui uma descrição.



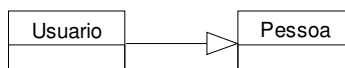
3. Um círculo é descrito por uma posição (x,y), correspondente ao centro e um valor para o raio e pode ser transladado de um deslocamento em x e outro em y.



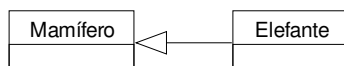
4. Um círculo é descrito por um ponto, correspondente ao centro, e um valor para o raio. O círculo é transladado de um deslocamento em x e outro em y.



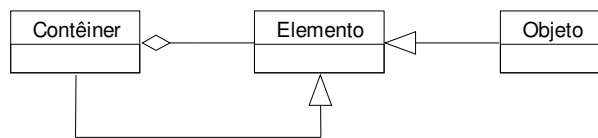
5. Um usuário de um sistema computacional é uma pessoa. Embora o modelo abaixo permita que uma instância de Usuário seja tratada como uma instância de Pessoa, o que decorre da herança, convém ressaltar que um modelo alternativo, talvez melhor seja uma simples associação entre Usuário e Pessoa. Nesta associação, a instância de Usuário ressaltaria uma atividade, um papel que a instância de Pessoa correspondente desempenharia.



6. Um elefante é um mamífero.

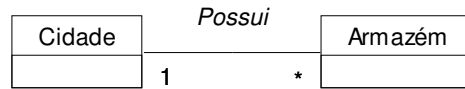


7. Um contêiner contém contêineres e objetos.

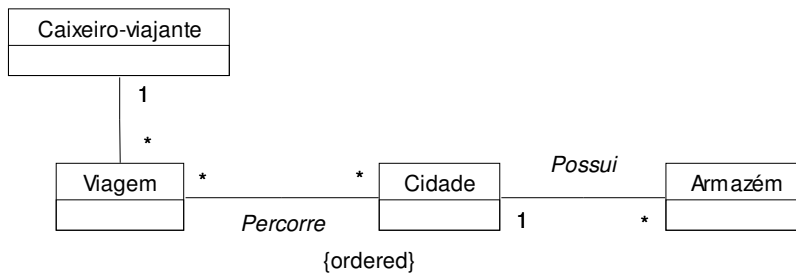


8. Um caixeiro-viajante faz uso de uma lista de cidades pelas quais terá que percorrer, na ordem fornecida e, para cada uma delas, colher pedidos de armazéns lá localizados.

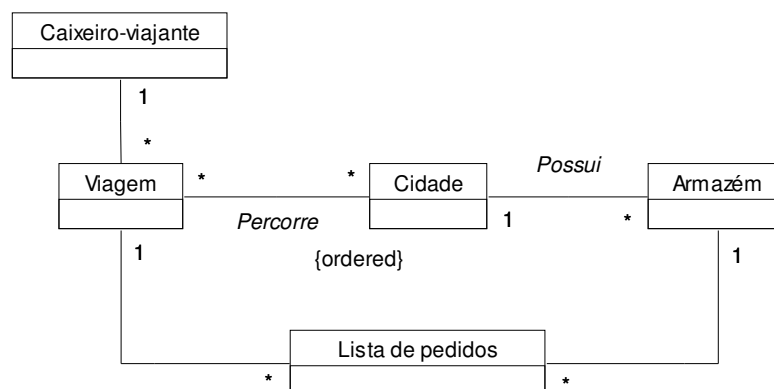
Naturalmente cada cidade possui vários armazéns, o que pode ser descrito pelo modelo parcial abaixo.



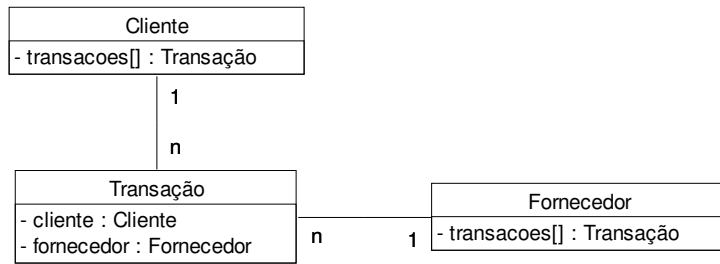
A lista de cidades corresponde a uma viagem do caixeiro-viajante. Espera-se uma lista de cidades para cada viagem. Ou seja, um caixeiro-viajante está associado a várias viagens, cada uma delas é descrita por uma lista de cidades, conforme ilustrado abaixo. Observe que as cidades estão ordenadas (*ordered*) pois, caso contrário, não será possível recuperar a ordem em que foram percorridas, caso esta informação seja relevante.



Falta ao modelo acima, contudo, a informação pertinente aos pedidos colhidos em cada armazém. Naturalmente será preciso associar a lista de pedido ao armazém do qual esta foi gerada. Esta informação, contudo, não é suficiente, pois também será preciso identificar a viagem na qual esta foi definida. Uma alternativa é estabelecer uma associação de cada lista de pedido para o armazém do qual esta originou-se e outra com a viagem em questão, conforme modelado abaixo.



9. Cliente e fornecedor envolvem-se em transações econômicas.

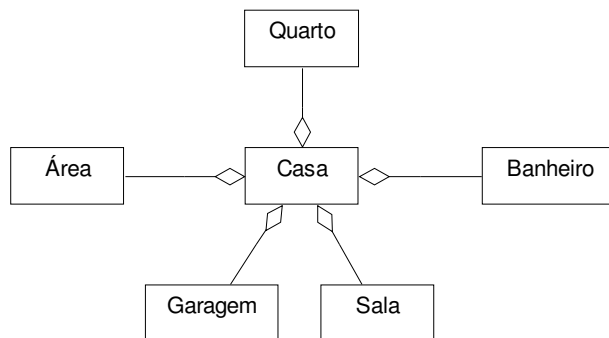




## Prática 3

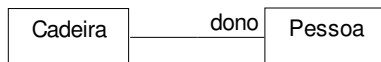
Modele cada um dos itens abaixo conforme são apresentados, independente de serem ou não adequadamente relevantes e/ou semanticamente corretos.

1. *Uma casa compreende quartos, banheiros, salas, áreas e garagens.*



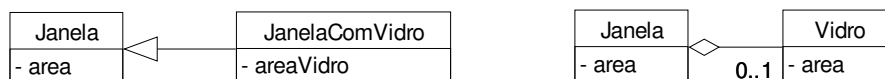
O diagrama acima não apresenta as cardinalidades nos extremos das agregações, mas também seria adequado o emprego de \* para representar zero ou mais instâncias de salas. Também é possível a interpretação na qual toda casa possui pelo menos um banheiro, pelo menos uma sala e assim por diante e, neste caso, a cardinalidade deveria ser 1..\*. Pode-se argumentar que sala, garagem e os demais elementos são partes da casa, como é a interpretação comum. Convém ressaltar que o próprio verbo “compreender” resalta esta relação do tipo todo/parte. Neste caso, temos duas opções: agregação ou composição. Se imaginarmos que Banheiro, por exemplo, pode estar representando o banheiro de um rodoviária ou outro local público, então a composição não seria adequada. Observe que a sentença da qual este modelo foi produzido não é clara a este respeito.

2. *Toda cadeira possui um dono, uma pessoa, que pode ser uma mulher ou homem.*



O modelo abaixo não é explícito quanto ao sexo da pessoa. Dado o fato de que os atributos foram omitidos, não há porque imaginar que este não será incluído. Também não foi incluída a cardinalidade, para os mais rigorosos, poderíamos indicar que a associação no extremo de pessoa possui como cardinalidade o valor 1, enquanto do outro extremo a sentença modelada não fornece nenhuma pista.

3. *Há janelas com vidros, outras sem. Toda janela com vidro encontra-se dividida em duas áreas: aquela ocupada pelo vidro e a restante. Naturalmente, a área total da janela é a soma destas duas áreas.*

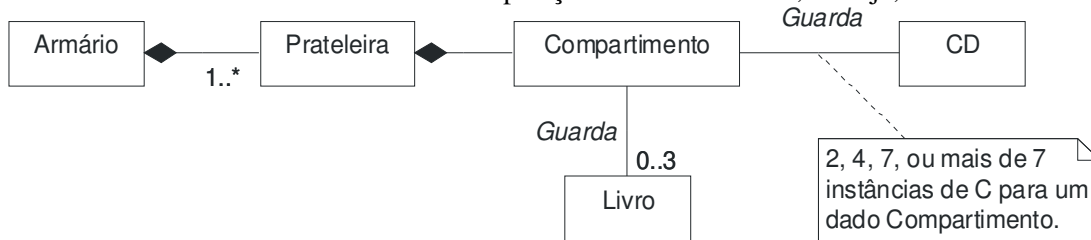


Uma janela com vidro pode ser interpretada como uma especialização de uma janela, digamos, comum, sem vidro, ou vice-versa. Aquela com vidro possui como atributo areaVidro, que informa a área da janela ocupada pelo vidro, enquanto a janela sem vidro possui como atributo area. Uma janela com vidro, portanto, possui pelo menos dois atributos suficientes para determinarmos duas informações relevantes: a área do vidro e aquela área da janela que não é ocupada por vidro. Uma alternativa é exibida no lado esquerdo da figura abaixo.

4. *Todo armário possui várias prateleiras. Cada uma delas divididas em compartimentos. Em cada*

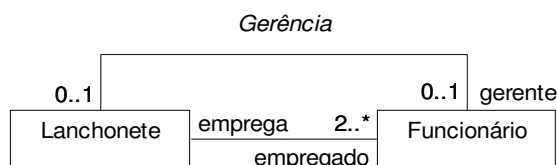
compartimento são armazenados objetos de dois tipos: livros e CDs. Cada compartimento pode guardar no máximo 3 livros, enquanto deve guardar 2, 4, 7, ou mais de 7 CDs.

No modelo acima Armário é uma composição de Prateleira, ou seja, não existe Prateleira sem



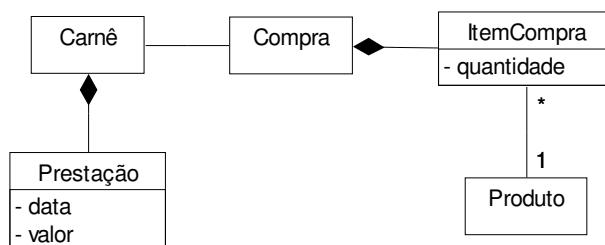
que seja parte de um armário. Um raciocínio similar é válido entre Prateleira e Compartimento. Este último guarda vários objetos em seu interior. Até três livros e uma combinação exótica mas bem definida de CDs. Afinal, ou teremos 2, ou teremos 4, ou teremos 7, ou teremos um número superior a 7 CDs.

5. Toda lanchonete possui pelo menos 2 funcionários, é possível que um deles seja gerente.

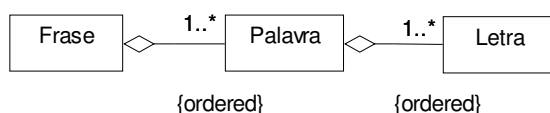


Um funcionário pode desempenhar o papel de gerente de uma lanchonete. Cada funcionário, além da possibilidade de estar ligado à lanchonete via Gerência, é empregado da lanchonete.

6. Todo carnê de prestações refere-se a uma determinada compra, que pode incluir vários produtos, em quantidades distintas para cada um deles. Cada prestação possui um valor correspondente e uma data limite para a quitação correspondente.



7. Uma frase é uma seqüência de palavras. Cada palavra é uma seqüência de caracteres.

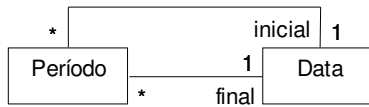


Não queremos confusão com os lingüistas e, dessa forma, podemos estar assumindo que existe frase formada por apenas uma única palavra, conforme o modelo abaixo registra. Tirando este cenário no mínimo *sui generis*, todas as palavras estão em uma seqüência, assim como as letras correspondentes a cada uma delas.

8. Preço é uma combinação de um valor e uma moeda. Por exemplo, valor 10 e moeda dólar.

Preço
- valor
- moeda

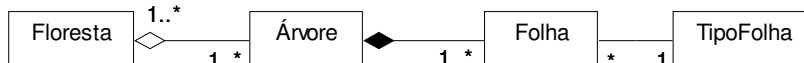
9. *Período é formado por uma data inicial e uma data final.*



Abaixo este cenário foi modelado com o emprego de associações. Também poderíamos ter empregado atributos na classe **Período**. A decisão entre estas opções é quase sempre uma questão do contexto para o qual o modelo está sendo construído. Se uma data é um elemento relevante do modelo, então trate-o como uma classe, doutra forma, um atributo é suficiente.

10. *Florestas são formadas por árvores que, por sua vez, são formadas por folhas. Cada folha possui sua forma, dentre todo um conjunto de formas possíveis. Existem árvores que participam de mais de uma floresta.*

Se uma árvore pode participar de mais de uma floresta, então **Floresta** não é uma composição de

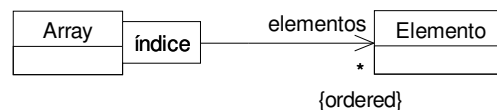


Árvore, mas uma agregação. Por outro lado, não é razoável imaginar que uma folha possa participar de mais de uma árvore. Para cada **Folha** há um **TipoFolha** correspondente.

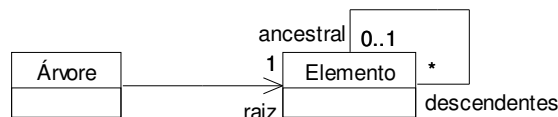
## Prática 4

Modelagem conceitual de algumas estruturas de dados. Não se tem a pretensão de apresentar, portanto, modelos de projeto, contendo classes de software, mas apenas aquelas que conceitualmente representam a correspondente estrutura de dados.

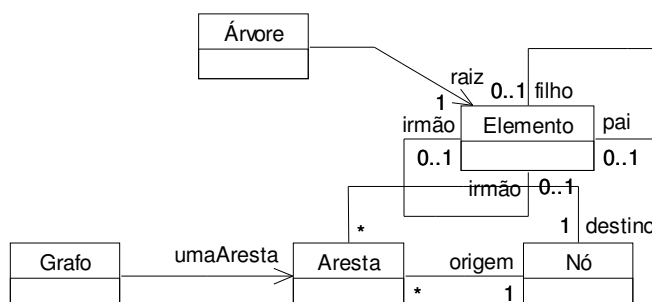
1. *Array*. Sequência de elementos homogêneos. Observe que o índice do *array* é suficiente para identificar um único elemento desta estrutura de dados. Em decorrência foi empregada uma associação qualificada pelo *índice*, conforme ilustrado abaixo. Os elementos de um *array* formam uma sequência, o que é indicado no modelo pela restrição *{ordered}*. Por último, é comum o emprego da navegabilidade para dizer, neste caso, que a classe *Array* possui um atributo, denominado de *elementos*, através do qual pode-se obter cada um dos elementos, instância de *Elemento*, do *array* em questão.



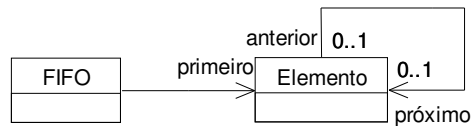
2. *Árvore*. Toda árvore possui um elemento denominado raiz, conforme o modelo abaixo. Em consequência, este modelo não contempla árvores nulas, sem nenhum elemento. Cada elemento pode ou não possuir descendentes. Se não possuir, então o elemento considerado é uma folha. Observe que nem todo elemento da árvore possui ancestral. Este caso particular ocorre quando o elemento em questão é a raiz da árvore.



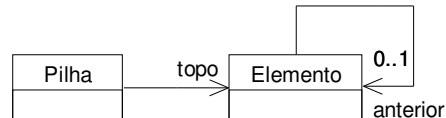
3. *Árvore*. Nesta proposta observe que cada elemento pode ou não possuir um ancestral (*pai*), assim como também pode ou não possuir um irmão (*irmão*). Se um elemento é filho único, então não possui irmãos. Se um elemento é a raiz, então não possui ancestral. Se tivermos um elemento da árvore que possui vários descendentes, então cada um deles irá indicar este elemento como pai e, além disso, todos os irmãos formarão uma lista circular. Cada irmão indica o seguinte formando esta lista.



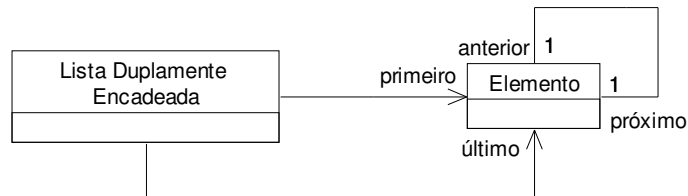
4. *Grafo*. Um grafo dirigido pode ser modelado como abaixo. Para cada aresta tem-se um nó que funciona como origem e outro como destino. Se fosse desejável adicionar pesos a cada uma das arestas, seria suficiente definir um atributo para a classe *Aresta*.
5. *FIFO*. A fila FIFO (*first-in first-out*) pode ser modelada conforme abaixo. Observe a presença de um primeiro elemento e, deste, todos os demais na fila, em ordem bem definida pela associação que indica o anterior e o próximo.



6. *Pilha*. A estrutura de pilha permite o acesso somente ao elemento do topo da pilha. Dado o elemento do topo, o próximo elemento que se tem acesso é o que desempenha o papel de **anterior** na associação de Elemento para Elemento.



7. *Lista duplamente encadeada*. Uma lista duplamente encadeada apenas permite, dado um determinado elemento, a definição imediata do sucessor e do anterior, caso estes existam. Um modelo correspondente é apresentado abaixo. Uma peculiaridade desta estrutura pode ser observada na cardinalidade da associação de **anterior** e **próximo**. Ambas são 1, indicando que, dado um elemento desta lista, necessariamente há um anterior e um elemento próximo, mesmo que seja o próprio elemento. Esta é uma característica comum de listas ditas duplamente encadeadas.

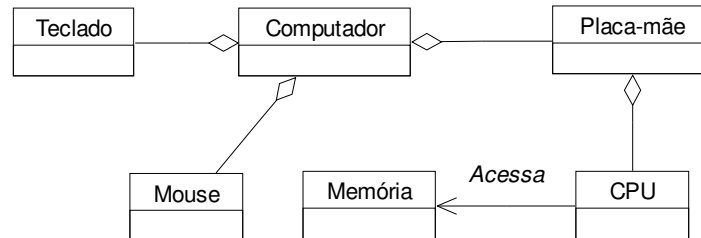


## Prática 5

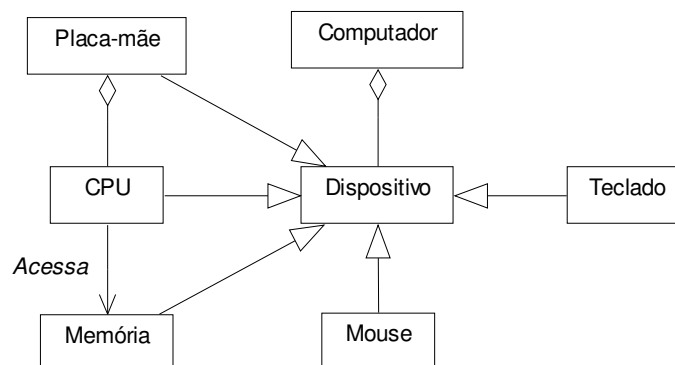
Para cada descrição abaixo forneça a modelagem correspondente empregando a UML.

1. Um computador compreende mouse, teclado, monitor e placa-mãe. A placa-mãe compreende memória e CPU. A CPU faz acesso à memória.

Um modelo que pode ser diretamente obtido da descrição acima é fornecido abaixo.

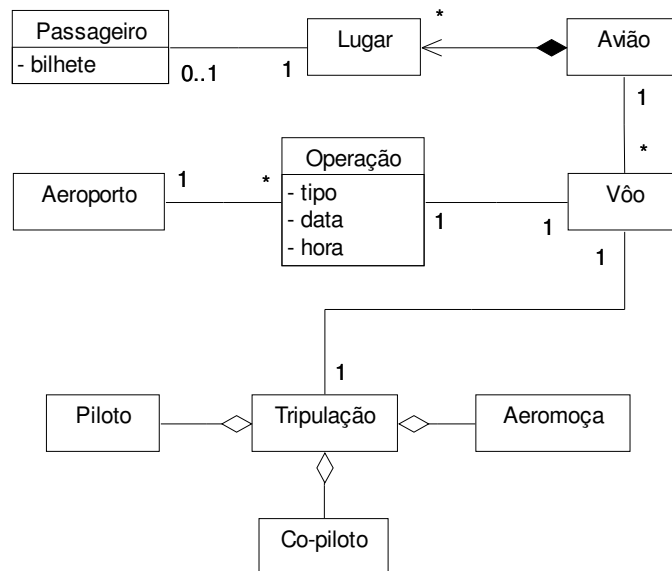


O modelo acima pode ser “trabalhado”, resultando naquele abaixo onde podemos acrescentar outros dispositivos à medida que se fizer necessário. No modelo abaixo optamos por não representar outros compartimentos de uma classe, ou seja, aquele dos atributos e das operações (métodos). São formas alternativas, que devem ser empregadas conforme a necessidade. Para este caso, por exemplo, nem atributos nem operações fazem falta.

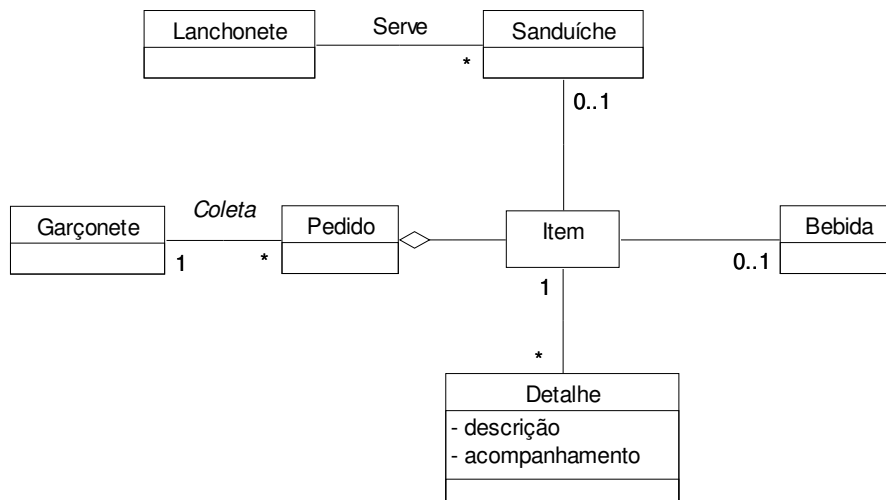


2. Em um aeroporto tem-se a ocorrência de vôos. Alguns decolam outros aterrissam. Para cada vôo há um avião, contendo vários lugares, cada um deles possivelmente ocupado por um passageiro. Para cada vôo também está associada toda a tripulação que inclui, necessariamente, um piloto, um co-piloto e uma ou mais aeromoças.

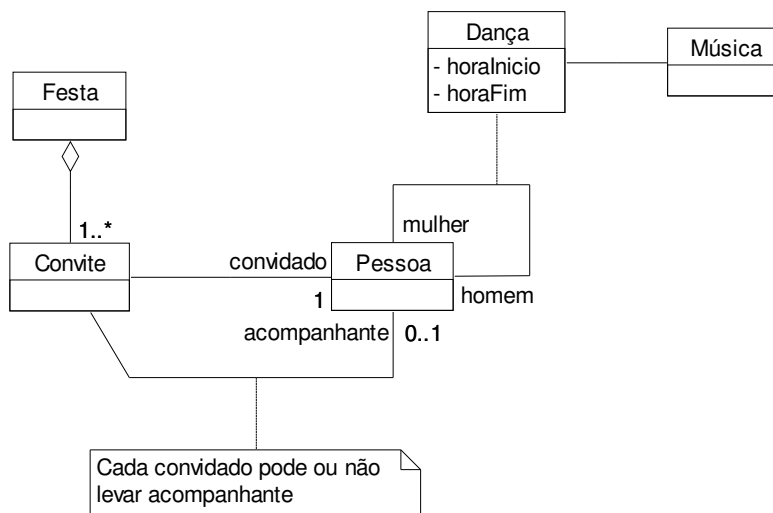
Para o caso acima é fácil observar alguns elementos principais a serem considerados. A simples identificação dos substantivos empregados nos revela os conceitos vôo, aeroporto, avião, lugar, passageiro, tripulação, piloto, co-piloto e aeromoça. Este conjunto candidato de classes é refinado à medida que modelamos e acrescentamos outros elementos. Neste caso estão faltando os relacionamentos entre estas classes. Por exemplo, deve ser destacado o fato de que um aeroporto decolam e aterrissam vôos. Avião contém lugares e assim por diante. Um modelo possível é fornecido abaixo.



3. Em lanchonetes são servidos vários tipos de sanduíche, alguns com vários tipos de acompanhamento como, por exemplo, ovo, salada, queijo e outros. O acompanhamento é opcional, não faz parte do sanduíche. Cada pedido recebido pelas garçonetes também inclui, em geral, bebidas.



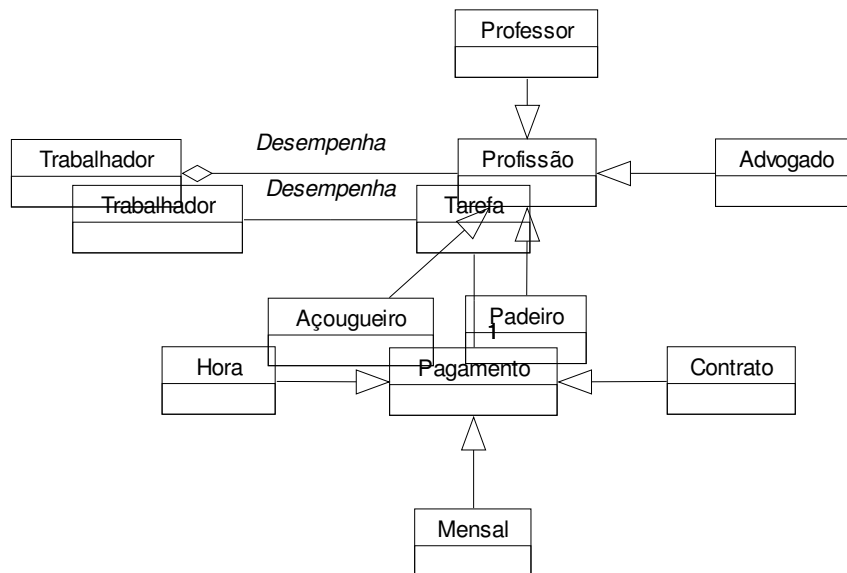
4. Em uma festa convencional homens dançam com mulheres. Cada dança está associada a



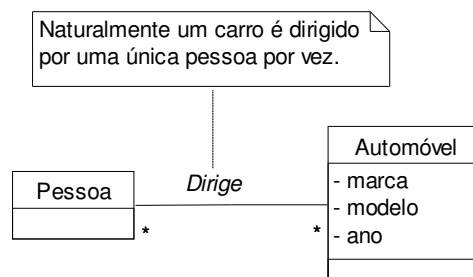
uma música. Cada convidado da festa pode ou não ir acompanhado.

5. Um trabalhador pode ser um açougueiro, um padeiro, um professor e um advogado.

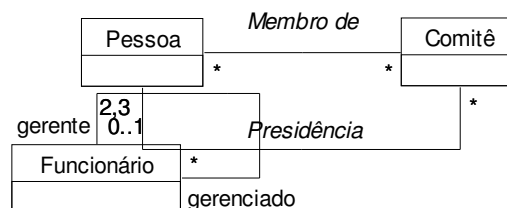




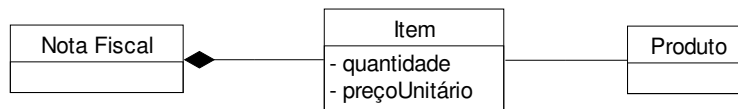
6. Uma determinada tarefa desempenhada por um trabalhador pode ser paga por hora, por um salário mensal ou por contrato. Vários pagamentos de formas distintas podem ser efetuados para uma mesma tarefa.
7. Pessoas dirigem automóveis, cada um deles de uma determinada marca, modelo e ano.



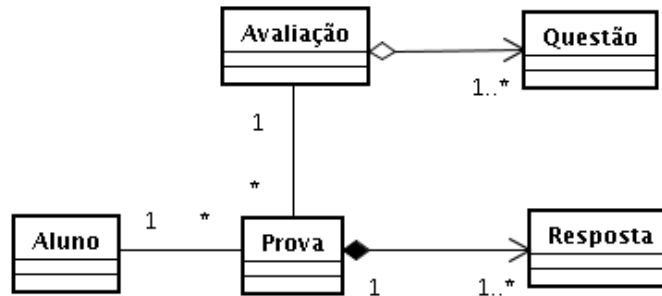
8. Uma empresa possui empregados organizados hierarquicamente onde gerentes gerenciam funcionários que, por sua vez, podem gerenciar outros funcionários. Todo empregado possui um gerente, exceto aquele do topo da hierarquia. Este último não é gerenciado por ninguém.
9. Pessoas podem ser membros de comitês. Cada comitê necessariamente possui dois ou três presidentes.



10. Uma nota fiscal contém vários itens, cada um descreve um produto, a quantidade correspondente e o preço unitário.



11. Em um sistema acadêmico avaliação é o nome que se dá a um conjunto de questões, elaborada com determinada finalidade e aplicada em uma determinada data. Neste sistema prova é o nome que se dá às respostas fornecidas por um aluno. Ou seja, alunos são submetidos a avaliações e, para cada uma delas, cada prova correspondente tem o o aluno

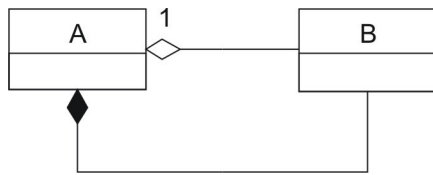


tem a correspondente prova.

## Prática 6

---

(a) Comente o diagrama abaixo.



O diagrama informa que para uma instância da classe B há uma instância de A, via a associação que é uma agregação, da qual B é membro. A composição diz que, para uma instância de B tem-se que esta é parte de uma instância de A e, portanto, sugere que uma instância de B pode ser parte de uma instância de A e membro de outra, ao mesmo tempo. Para ressaltar que uma instância de B pode estar associada a uma instância de A por uma associação ou, exclusivamente por outra, a cardinalidade destas associações do lado da classe A deveria ser  $0..1$ .

(b) Marque os itens abaixo que sugerem a quebra do princípio de substituição de Liskov. O princípio de substituição de Liskov é atendido quando, onde é esperada uma instância de um determinado tipo, qualquer instância de um subtipo daquele esperado pode ser fornecido. Cada item é uma relação (herança) entre dois nomes que devem ser interpretados da seguinte forma: o primeiro refere-se à superclasse e o segundo à subclasse.

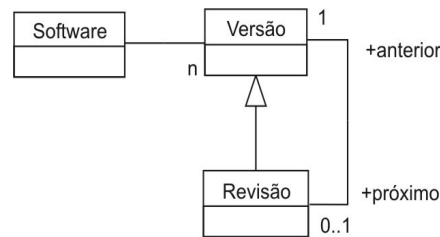
- (a) Biblioteca/Livro
- (b) Telefone/Comunicação
- (c) Animal/Macaco
- (d) Música/CD-ROM
- (e) Janela/Vidro
- (f) Moradia/Casa
- (g) Retângulo/Quadrado

Quebrar o princípio de substituição de Liskov é fazer uso de uma herança na qual uma instância da subclasse não pode se comportar adequadamente como uma instância da superclasse. Dos itens acima, claramente: (a) um livro não se comporta como uma biblioteca pois, por exemplo, biblioteca tem horário de funcionamento, local, telefone, funcionários e várias outras considerações que simplesmente não existem em um livro; (b) comunicação não se comporta como telefone pois, por exemplo, telefone tem marca, cor e peso, entre outras, que não estão presentes em uma comunicação; (d) CD-ROM não se comporta como música, por exemplo, porque música possui compositor e intérpretes, entre outros, não presentes em um CD-ROM; (e) Uma janela possui um estado que pode ser aberta ou fechado, o que não existe em vidro; (g) em um retângulo podemos atribuir um valor para dois dos lados e outro valor para os outros dois lados, o que não é possível em um quadrado.

(c) Analise cada um dos itens abaixo. Entenda cada item como um cenário. Para cada um deles, verifique se o modelo ao lado o contempla, ou seja, se o cenário pode ocorrer dado o modelo fornecido. Também entenda que a palavra “revisão” deve ser interpretada como uma instância criada a partir da classe *Revisão* e a palavra “versão” como uma instância criada a partir da classe *Versão*. Por último, “software” é uma instância criada a partir da classe *Software*.

- (a) Há softwares e nenhum deles possui uma revisão.

- (b) Há revisão sem software.
- (c) Há versão sem revisão.
- (d) Há revisões sem versão.
- (e) Pode haver mais versões que revisões.
- (f) Pode haver mais revisões que versões.
- (g) Para cada revisão há uma versão correspondente.
- (h) Para cada versão há uma revisão correspondente.
- (i) Uma revisão não pode participar de uma ligação com outra revisão da qual a primeira desempenha o papel de próximo.



Itens verdadeiros: (a), (b), (c), (d), (e), (f) e (i)

Itens falsos: (g) e (h)

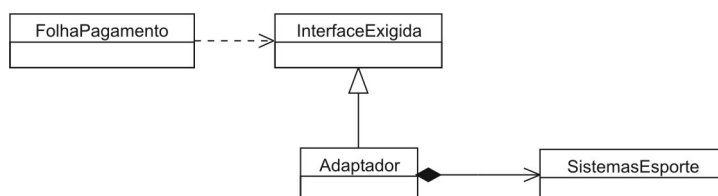
Itens comentados: (d) observe que uma instância de **Revisão** está associada, necessariamente, a uma instância que pode se comportar como uma instância de **Versão** através da associação na qual esta instância desempenha o papel de **anterior**. Observe que não é necessário a criação de uma instância de **Versão**, afinal, toda instância de **Revisão** pode se comportar como uma **Versão**. Se tivermos apenas instâncias de **Revisão** então elas formarão um ciclo onde uma sucede e precede qualquer outra. (g) pelo comentado no item (d) observa-se que podemos criar várias instâncias de **Revisão** sem explicitamente criarmos instâncias de **Versão**.

- (d) Uma empresa de grande tradição no ramo de materiais esportivos, com milhares de funcionários, faz uso de um sistema de folha de pagamento fornecido por uma empresa goiana. O sistema da folha de pagamento faz uso de inúmeras informações fornecidas pelos vários sistemas em operação da empresa de materiais esportivos. A interação entre o sistema da folha de pagamento e os sistemas da empresa de materiais esportivos sempre é um problema. Motivo: grande volatilidade dos sistemas da empresa de materiais esportivos. Embora as informações requisitadas pela folha de pagamento nunca tenham sido alteradas, as mudanças nos sistemas da empresa de materiais esportivos sempre forçaram mudanças no código do sistema da folha de pagamento com o propósito de obter, como já sabemos, as mesmas informações. Questão: modele uma solução para este problema.

Um dos padrões de projeto orientado a objetos é conhecido por Adapter. Quando duas partes precisam de interagir e uma delas sofre freqüentes mudanças, pode-se empregar o padrão Adapter para encapsutar a parte volátil e oferecer uma interface estável para uso da outra parte.

- (e) Tratando **FolhaPagamento** como uma destas partes, a parte estável, **SistemasEsporte** como a parte volátil, abstratamente representando os sistemas existentes na empresa de materiais esportivos, **InterfaceExigida** como o conjunto de requisições da folha de pagamento com o propósito de obter as informações necessárias (bem estável conforme o enunciado) e, por último, **Adaptador** como uma abstração para os possíveis e vários adaptadores a serem produzidos para cada mudança nos sistemas que lidam com os materiais esportivos, teremos o seguinte modelo como resultado:

Nesta proposta, observe que **FolhaPagamento** está imune às mudanças que eventualmente ocorrerem nos

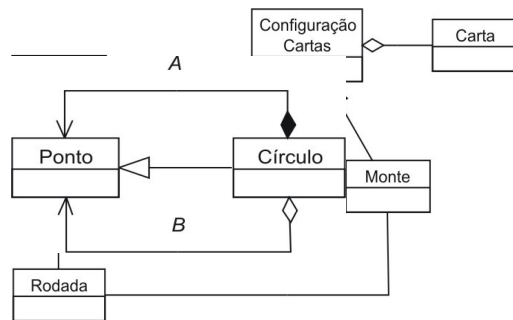


sistemas da empresa de materiais esportivos.

## Prática 7

Modele cada um dos cenários abaixo, quando o exercício não exigir algo diferente.

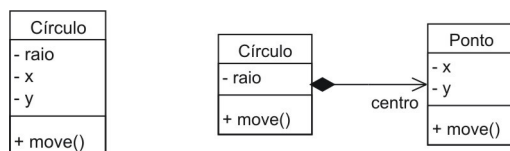
- (a) Em um jogo de cartas há dois tipos de conjuntos delas: um conhecido por “monte” e outro para cada jogador. O conjunto de cartas com cada jogador pode variar a cada jogada. O conjunto de cartas do monte também pode variar a cada jogada. Uma jogada é executada por um jogador e consiste em retirar uma carta daquelas disponíveis para serem “compradas” e, se assim preferir, esta carta pode ser substituída por alguma carta do conjunto de cartas do jogador. Neste caso, a carta substituída é depositada no monte. Um jogo é definido por uma sequência de jogadas.



- (b) Quais das associações abaixo é mais adequada para representar que um círculo, além de outras propriedades, não exibidas, possui um ponto como centro? Justifique.

Observe que herança não é uma associação. A herança e as associações acima, contudo, são todas exemplos de relacionamentos. Temos portanto, uma composição ou uma agregação. A composição, podem muitos argumentar, seria mais apropriada, pois o ponto é parte indissociável do círculo do qual designa o centro.

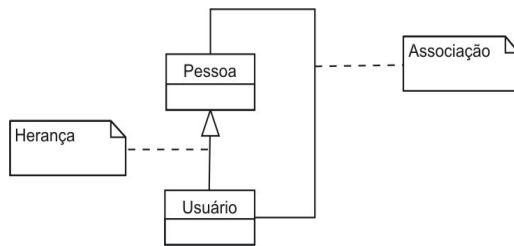
- (c) Quais dos modelos abaixo é mais adequado para representar o fato de que um círculo possui um ponto como centro? Justifique.



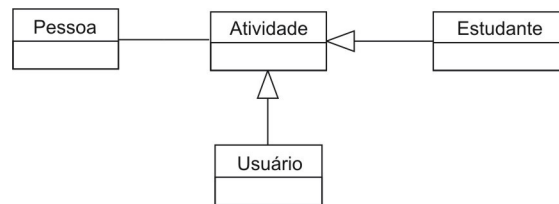
A segunda versão oferece maior independência entre entidades distintas. Observe que na primeira versão é necessário conhecer como se move um ponto para que o círculo seja deslocado. Na segunda versão, sabe-se que para mover um círculo basta mover o centro deste círculo, cujo conhecimento correspondente para tal é melhor depositado na classe Ponto, em vez da classe Círculo, que entende de círculos.

Observe ainda que esta idéia não escala. Para um triângulo teríamos três pares ordenados, ou seja, seis propriedades para representar os três pontos. Já pensou em um polígono?

- (d) Quais dos relacionamentos entre as classes abaixo é o mais adequado? Justifique.

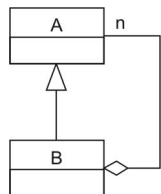


Gostaria de dizer que a herança está definitivamente errada e a associação está definitivamente correta. Contudo, a resposta depende do cenário em questão. Cenário omitido nos deixa apenas com especulações. A que prefiro, neste contexto, é a seguinte. Um usuário representa um papel dentre todo um conjunto deles que um ser humano (pessoa) pode desempenhar. Por exemplo, usuário e aluno podem ser papéis desempenhados por uma pessoa ao longo de sua vida. Se empregarmos a herança, então “confundimos” o papel com a pessoa. De fato, todo usuário é uma pessoa, contudo, tratar João da Silva Sauro como usuário e termos dificuldade de dissociar a pessoa do papel de usuário que este desempenha não parece elegante. O diagrama abaixo apresenta um modelo compatível com esta perspectiva.

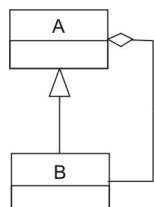


(e) Abaixo segue um diagrama contendo duas classes e dois relacionamentos. Também são fornecidos quatro itens que fornecem, cada um, um par de nomes. Qual destes itens os nomes, respectivamente, melhor representam A e B no diagrama abaixo?

- (a) moradia/casa;
- (b) ItemDiretório/Diretório;
- (c) Convidado/Festa
- (d) Jogador/Time.



(f) Prática de programação e Programação podem ser as classes referenciadas, respectivamente, por A e B no diagrama abaixo? (Assuma que programação refere-se a um esforço de desenvolvimento de código, ou seja, trata-se de uma prática de programação, que pode envolver várias atividades de desenvolvimento.) Justifique.



(g) Seja uma revisão um caso particular de versão de um produto de software. Todo software pode estar relacionado a várias versões (pelo menos uma). Para cada versão podem existir revisões subsequentes, uma após a outra, em uma seqüência bem definida. Este cenário é modelado adequadamente pelo

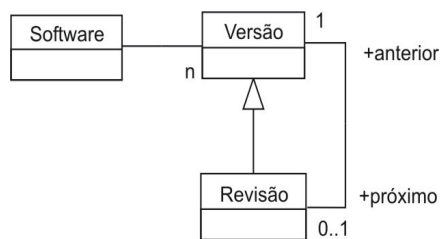


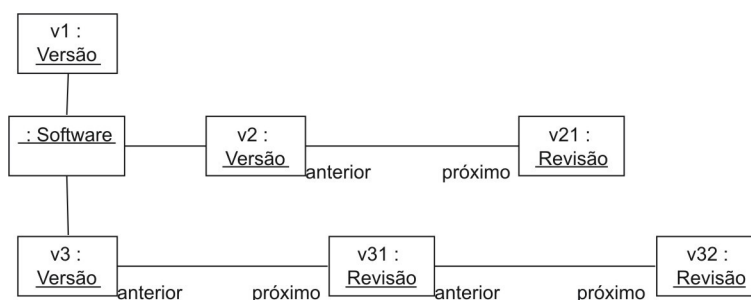
diagrama abaixo? Justifique.

Segundo este modelo, um software pode estar associado a várias versões. Contudo, pode existir software sem versão e, portanto, a cardinalidade deve ser ajustada.

Toda revisão é uma versão, conforme a herança, e não pode haver revisão sem uma instância de *Versão* que a preceda. Observe que para cada *Revisão* há uma *Versão* que desempenha o papel de *anterior*, conforme associação entre *Revisão* e *Versão*. Esta mesma associação também mostra que pode haver *Revisão* que irá suceder uma *Versão*, mas não se trata de uma obrigatoriedade, conforme a cardinalidade  $0..1$ . Se existir, então a versão precede a revisão e, naturalmente, a revisão sucede a versão, em uma ordem muito bem definida.

O modelo contempla a situação de uma versão ser sucedida por uma seqüência bem definida de revisões? Vimos que para uma versão pode haver uma revisão. Caso exista uma revisão, sabemos que uma revisão também é uma versão, conforme a herança. Em consequência, toda revisão também pode possuir um sucessor, um próximo, pois uma versão pode possuir um próximo e uma revisão é uma versão. Ou seja, uma mesma instância de *Revisão* pode participar de duas associações: (a) como uma revisão e, neste caso, desempenha o papel de próximo de alguma versão e (b) como uma versão e, neste caso, desempenha o papel de anterior em alguma instância da associação que liga esta revisão à revisão seguinte.

O diagrama de objetos abaixo pode ser útil na compreensão deste modelo. Observe que uma instância de *Software* está ligado a três versões identificadas por v1, v2 e v3. Estas ligações são instâncias da associação entre as classes *Software* e *Versão*, exibida na figura anterior.

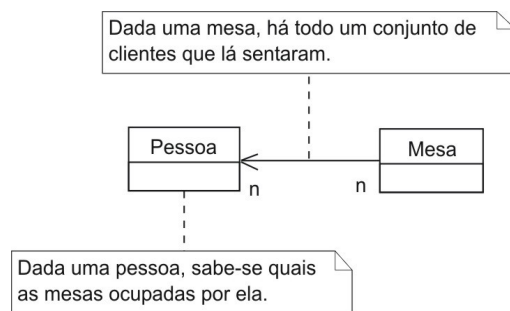


Se observarmos as instâncias v32 e v21 veremos que estas não participam de ligações onde desempenham o papel de *anterior*. Naturalmente, sempre teremos, em determinado instante de tempo, a última revisão para determinada versão. Ao observarmos o diagrama de classe veremos que uma revisão também é uma versão e, portanto, pode existir ou não uma instância de *Revisão* que sucede uma determinada versão. As instâncias citadas, v32 e v21, são exemplos de versões, pois são instâncias de *Revisão*, para as quais não há sucessores.

Enquanto v32 e v21 podem ou não possuir sucessores como versões, necessariamente possuem antecessores como revisões. Observe que o antecessor de uma revisão é uma instância de *Versão* que, portanto, pode ser outra revisão ou uma instância de *Versão*.

A instância v21 possui uma versão que a precede: v2. Esta ligação é “explicitamente” modelada pela associação entre *Versão* e *Revisão*. A instância v32 possui como anterior uma revisão: v31. Esta ligação não é “explicitamente” modelada pela associação, como dizem alguns, pois a associação é entre *Versão* e *Revisão*. Devemos estar atentos, contudo, ao fato de uma *Revisão* ser uma *Versão* e, portanto, uma instância de *Revisão* pode estar associada a outra instância de *Revisão*. Neste caso, a primeira delas desempenha o papel de *anterior*, enquanto a segunda o papel de *próximo*. Este é exatamente o cenário exibido no diagrama com os objetos acima.

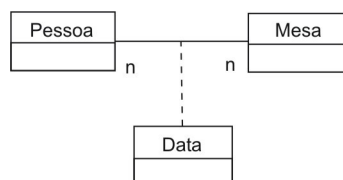
- (h) Clientes de um restaurante sentam-se em mesas. Em geral, um ou mais clientes compartilham uma mesma mesa. Dada uma mesa é desejado conhecer todos os clientes que já fizeram uso desta mesa. Comente o modelo abaixo para representar este cenário.



Se tratarmos pessoa como cliente e, conforme o enunciado, estivermos interessado em conhecer aqueles que tiveram o privilégio de se sentar à mesa de interesse, então o modelo está completo. Se observarmos a nota ligada à classe *Pessoa*, contudo, seremos obrigados a reconhecer que a associação deve ser bidirecional, pois também gostaríamos de obter todas as mesas ocupadas por determinada pessoa.

A realidade, contudo, é um pouco diferente do que o enunciado estabelece. Por exemplo, conforme modelado, para uma mesa, digamos M, saberemos todas as pessoas que ocuparam algum lugar nesta mesa. Não saberemos, contudo, em que momento ocorreu esta ocupação. Também não saberemos a ordem em que estas pessoas ocuparam M, se nos primeiros dias do restaurante ou apenas mais recentemente. Também não saberemos quais os grupos que se formaram nesta mesa, ou seja, quem estava acompanhado de quem. Também não saberemos quantas vezes uma determinada pessoa sentou-se nesta mesa, ou seja, se se trata de um usuário casual ou freqüentador assíduo deste restaurante suspeito. Muitas informações não são possíveis de serem registradas com este modelo.

- (i) Em restaurantes, mesas são ocupadas por pessoas ao longo do tempo. O modelo abaixo adequadamente reflete este fato? Justifique.

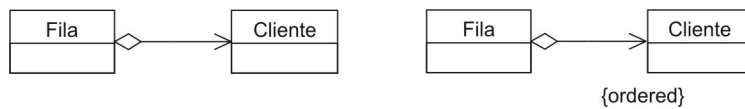


Veja que uma pessoa pode ocupar mesas e que uma mesa pode ser ocupada por pessoas. Esta ocupação ocorre em determinada data e, portanto, o modelo permite estabelecer um histórico, ao longo do tempo, de quem ocupou qual mesa. Com a identificação do tempo em que a ocupação ocorreu, pode-se estabelecer uma ordem das ocupações e, com algum esforço, até especular quais foram os grupos (pessoas) que se sentaram a mesa em determinada data.

- (j) Alguém estava preocupado, em uma grande agência bancária de Goiânia, com um modelo que refletisse

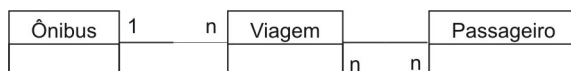


as filas que lá eram freqüentes. Neste momento surge um funcionário! Extrovertido, logo tratou de apresentar os dois modelos seguintes. Explicou ambos e concluiu que o segundo é superior. Qual o argumento que o funcionário provavelmente utilizou para convencer seus ouvintes?

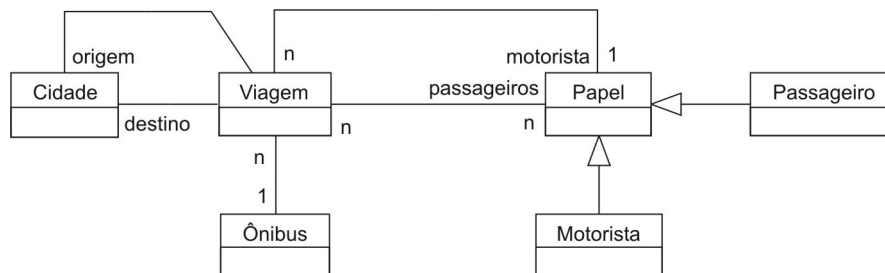


A versão à esquerda apenas agrupa clientes em uma fila, o que é pouco para organizar de forma justa o atendimento. Para tal é preciso estabelecer uma seqüência de atendimento e, neste caso, precisamos de ordenar nos clientes, conforme a versão fornecida à direita.

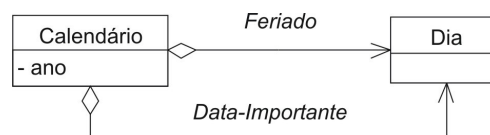
- (k) Ao longo de sua vida útil um ônibus transporta um grande número de passageiros. Represente os passageiros transportados por um ônibus para cada uma de suas viagens.



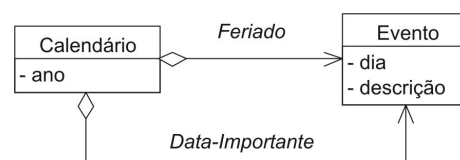
Para cada ônibus podem existir várias viagens. Para cada viagem temos exatamente um único ônibus. Cada viagem envolve vários passageiros. Se desejarmos saber qual a origem, o destino, o horário de partida e chegada e o motorista pode ser que o modelo resultante se pareça com aquele abaixo.



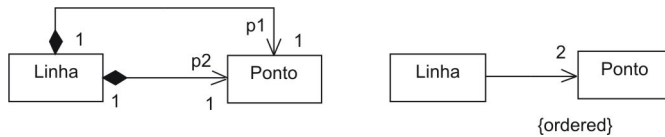
- (l) Ao longo de um ano há muitos dias que são feriados, enquanto outros referem-se a acontecimentos relevantes (seja uma data de aniversário ou outro).



O modelo acima apenas registra, para um dado calendário de um certo ano, quais os feriados e datas relevantes. Não é possível, contudo, registrar o acontecimento relevante ou de quem é o aniversário, por exemplo. O modelo abaixo, por outro lado, já permite um número maior de informações que podem ser úteis ao contexto considerado.

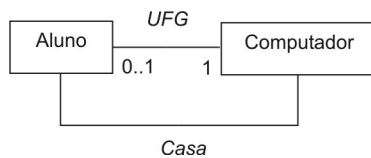


- (m) Uma classe **Linha** com duas composições para uma classe **Ponto**, cujos papéis são **p1** e **p2** é uma versão melhor que uma classe **Linha** associada a **Ponto** onde o extremo da associação com **Ponto** possui cardinalidade 2 e encontra-se ordenado? Justifique.
- As duas propostas são equivalentes. Contudo, caso seja interesse ressaltar que uma linha é formada por



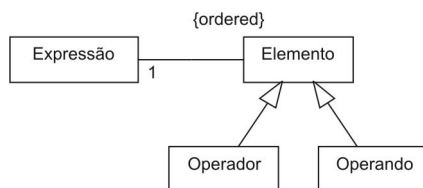
dois pontos (definida por dois pontos), então a primeira versão é mais explícita.

- (n) No âmbito da UFG, cada estudante possui um computador, que não é compartilhado com nenhum outro estudante. Fora da UFG, contudo, nem todos os alunos têm computador, enquanto alguns possuem vários computadores.



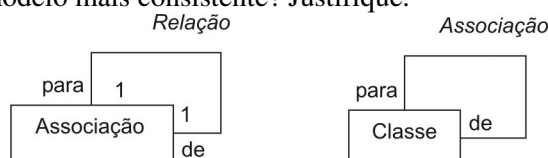
Se na UFG cada aluno possui um computador, então da esquerda para a direita a associação UFG reflete este cenário. Contudo, isto não é suficiente para deduzir que todo computador possui um “aluno-dono”, o que justifica a cardinalidade do lado esquerdo. Nos domicílios dos estudantes, contudo, o cenário é diferente. Um aluno pode estar associado a vários computadores e um computador pode estar associado a vários alunos. Convém lembrar que alguns alunos podem compartilhar um mesmo computador fora da UFG.

- (o) Modele expressões aritméticas como sendo seqüências ordenadas de elementos que são operadores ou operandos.



Não cabe assegurar, via estrutura, que as sentenças são válidas. Este tipo de verificação é melhor realizado por software, não por um modelo. Sentenças aritméticas, válidas ou não, podem ser registradas conforme o modelo acima.

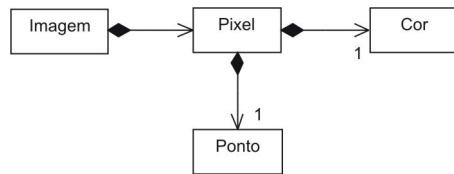
- (p) Uma classe **Associação** possui uma associação que parte dela para ela mesma. Em ambos os extremos a cardinalidade é 1. Em um deles o papel é **para**, enquanto no outro é **de**. O nome da associação é **Relação**. Outra classe, denominada de **Classe**, possui uma associação que também parte dela para ela mesma. Os papéis são, à semelhança do caso anterior, **para** e **de**. Neste último associação, contudo, o nome é **Associação** e nenhuma cardinalidade foi fornecida. Quais destas classes, com a respectiva associação representa um modelo mais consistente? Justifique.



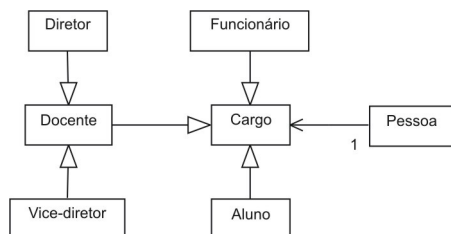
Sabemos que uma associação é um relacionamento entre classes. Desta forma, de imediato, a versão mais à direita do modelo acima está correto para associações unidirecionais. Associações bidirecionais não podem ser registradas por este modelo conforme está.

A versão mais à esquerda informa que uma associação está relacionada à outra de forma unidirecional. É fácil perceber que se trata de uma interpretação errada do conceito de associação e, portanto, a outra versão é mais apropriada, apesar de considerar apenas aquelas unidirecionais.

- (q) Uma imagem é um conjunto de pixels, cada um deles possui uma determinada posição e uma cor. Faça a modelagem de uma imagem. Em tempo, um pixel (*picture element*) é a menor unidade exibida em um monitor.



- (r) Estabeleça relações entre as classes Diretor, Vice-diretor, Docente, Funcionário e Aluno sabendo-se que, diretor e vice-diretor são docentes. Caso considere apropriado, acrescente outras classes. Um docente também pode ser funcionário e, possivelmente, também aluno.



No modelo acima, diretor, vice-diretor e demais elementos fornecidos foram tratados como casos particulares de cargo. Embora não seja natural tratar um aluno como ocupante de um cargo em uma instituição de ensino, este modelo permite relacionar um ser humano com vários “cargos”, o que é uma situação típica.

- (s) Um presidente nomeia ministros e estes seus respectivos assessores.

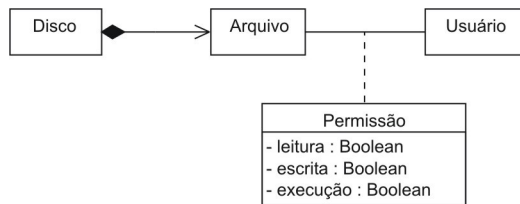


Em um cenário mais realístico pode ser necessário indicar, por exemplo, a data da nomeação, entre outras. Uma classe associativa pode registrar a informação data de nomeação, caso seja necessário.

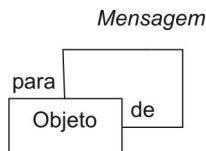
- (t) Todo usuário possui, para cada arquivo nos diretórios de um disco, acesso de escrita, leitura e/ou gravação.

Um disco pode ser interpretado como uma composição de arquivos. Entre um usuário e um arquivo há uma permissão, que indica o grau de poder do usuário. Isto é melhor modelado como uma classe associativa, conforme abaixo. A interpretação é simples: dado um arquivo e um usuário nós teremos, necessariamente, uma permissão. Os atributos de uma classe associativas são atributos da associação. Lembre-se de que classe associativa é uma construção que possui características tanto de uma associação como de uma classe.

Para que não haja dúvida, dado um aluno e um arquivo associados, teremos os atributos leitura, escrita e execução, conforme o diagrama abaixo.



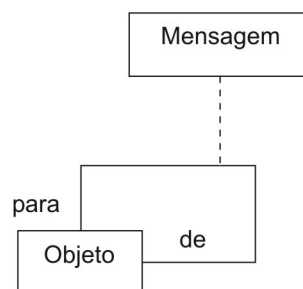
(u) Em um sistema orientado a objetos, objetos cooperam uns com os outros através da troca de mensagens.



Uma mensagem é, necessariamente, um relacionamento direcionado de um objeto para outro. Para que exista uma mensagem é necessário a existência de um objeto que envia a mensagem. Para que esta possa ser enviada, é preciso a existência do objeto destino. Observe que a origem e o destino podem ser distintos. Não necessariamente precisam ser o mesmo objeto, nem o modelo acima sugere que seja o mesmo objeto, apesar de muitos se enganarem.

O modelo acima é interpretado da seguinte forma: dada uma instância `obj` de **Objeto**, podem existir várias instâncias de **Objeto** relacionadas a `obj` através da associação **Mensagem**. Ou seja, de `obj` podem partir várias ligações para instâncias de **Objeto**. O destino destas mensagens podem ser o próprio objeto, pois um objeto pode enviar várias mensagens, inclusive para si próprio.

A associação, contudo, não permite descrever qual a mensagem, o que pode ser uma informação de interesse e não apenas quais são os objetos que recebem mensagem de determinado objeto. Em consequência, o modelo abaixo parece mais completo.



## Prática 8

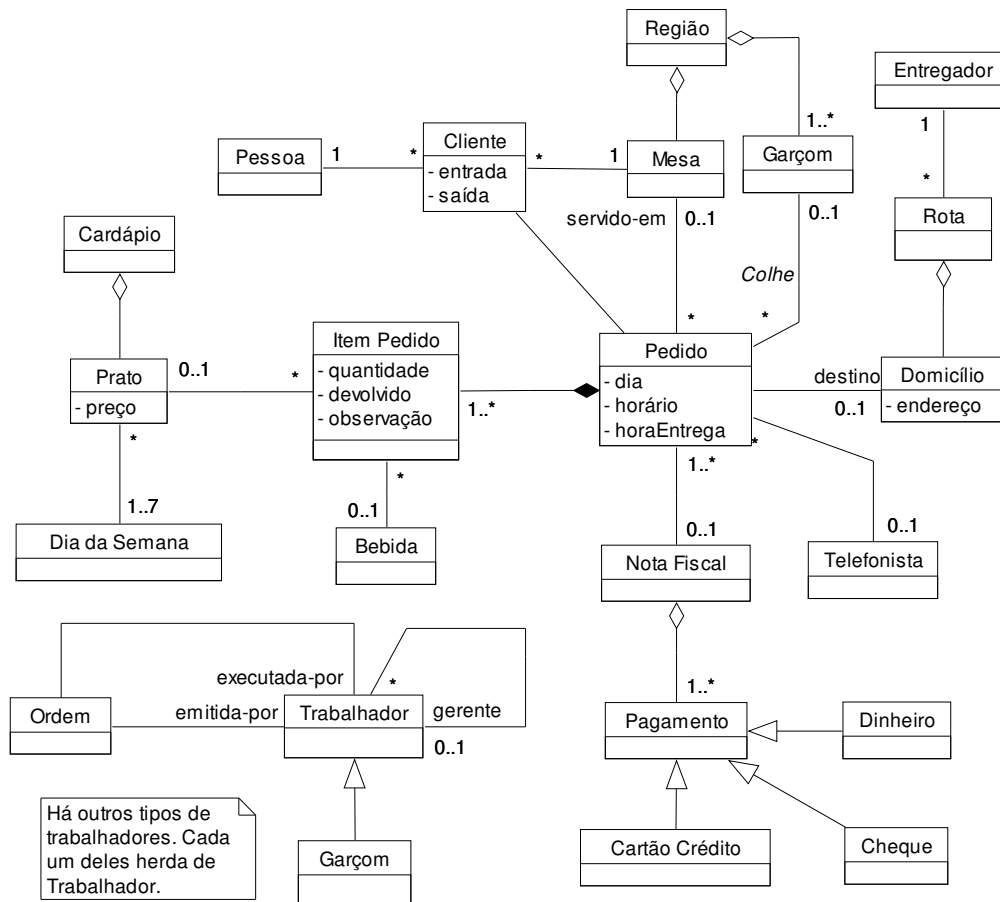
Modele de forma orientada a objetos o cenário caracterizado pelos vários itens abaixo.

---

1. Um restaurante serve pratos de um cardápio. O cardápio contém o preço de cada prato e o dia da semana em que este está disponível.
2. Clientes encontram-se distribuídos em mesas, servidas por garçons.
3. Garçons atendem os clientes conforme a região do restaurante. Cada região define um conjunto de mesas e um conjunto de garçons que as servem.
4. Pedidos podem ser coletados por um dos garçons ou pela telefonista.
5. O pedido pode ser atendido imediatamente, cenário mais comum, ou ser servido no dia e horário especificados, em uma mesa ou entregue em determinado domicílio.
6. Toda entrega à domicílio é realizada por um entregador, identificado conforme o conjunto de pedidos a serem entregues, o momento em que devem ser entregues e a disponibilidade deles, com o propósito de minimizar os custos através de uma rota menor. Cada entregador, no momento em que sai para um conjunto de entregas, recebe uma rota contendo a descrição de cada entrega e o endereço correspondente.
7. Pedidos são contabilizados para que uma nota seja posteriormente emitida para pagamento, via cartão de crédito, cheque ou dinheiro.
8. A nota emitida e entregue ao cliente contém o período de permanência do cliente(s) no restaurante, bem como o tempo médio de atendimento do pedido.
9. Cada pedido deve identificar o cliente que o requisita e o garçom (ou telefonista) que o colheu.
10. O pedido pode dar origem a um prato que é negado pelo cliente e, em consequência, devolvido. Isto também é válido para bebidas. Um suco de laranja com açúcar, um filé malpassado são alguns exemplos de possíveis “devoluções”.
11. Periodicamente podem ser emitidas duas listagens relevantes para a qualidade dos serviços prestados pelo restaurante: (a) tempo médio de atendimento de pedido por garçom (usado para premiar garçons eficientes) e (b) pratos que com mais frequência são devolvidos pelos clientes.
12. Em restaurantes há vários trabalhadores. Na cozinha há um chefe. No atendimento, um dos garçons é chefe dos demais. Há um gerente de todo o restaurante. Também há aqueles que estacionam os automóveis e fazem o serviço de segurança.
13. O gerente do restaurante é responsável pelo bom andamento das atividades e, em consequência, dele emanam ordens para todos os demais trabalhadores do restaurante.
14. Neste restaurante, quando uma nota é emitida, o gerente ordena que o responsável por guardar os automóveis coloque o automóvel do cliente disponível na portaria do restaurante.
15. Os pedidos de bebida, neste restaurante, conforme a prática da casa, são atendidos juntamente com os pedidos de pratos correspondentes, exceto quando o cliente desejar de forma diferente. As bebidas são servidas por garçons que só servem bebidas e não coletam

pedidos.

16. Acrescente um item e o contemple na modelagem correspondente.

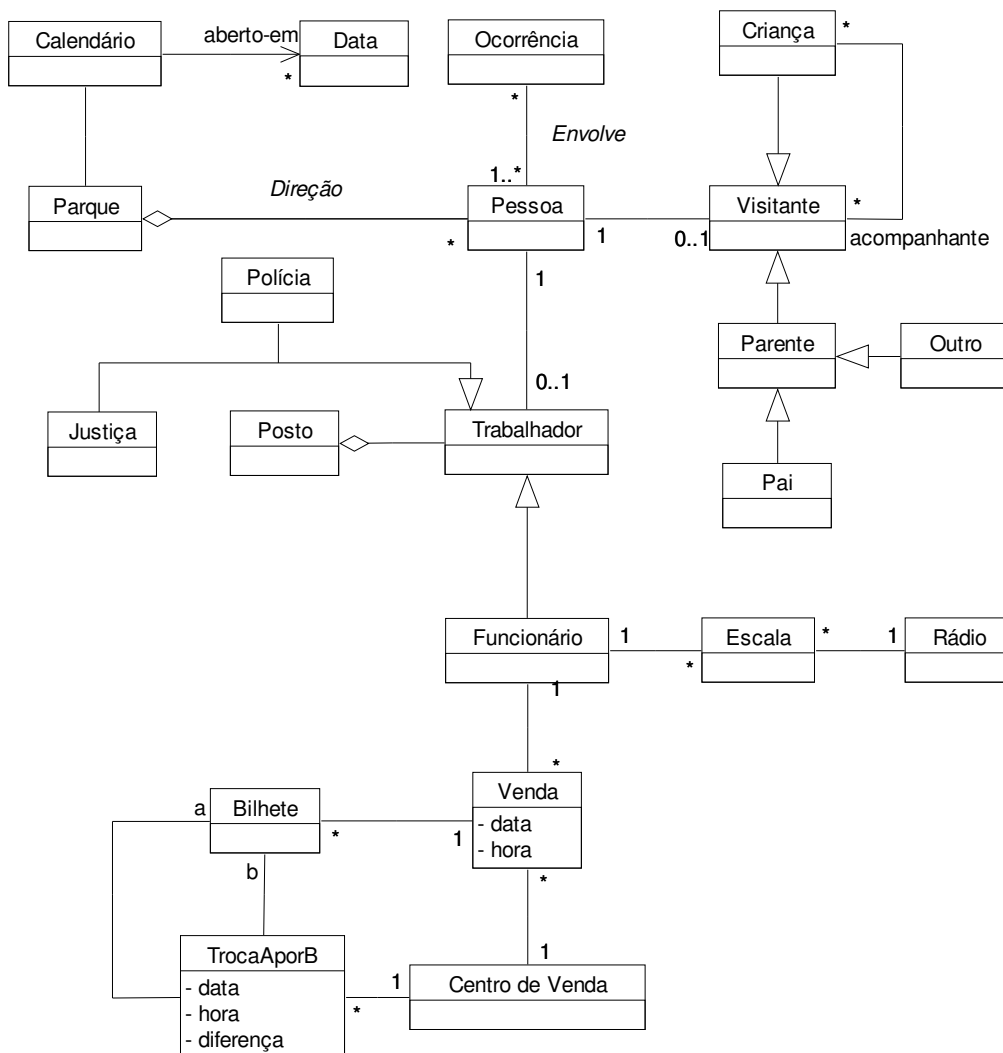


## Prática 9

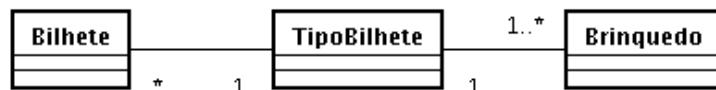
Modele de forma orientada a objetos o cenário caracterizado pelos vários itens abaixo.

---

1. Em um parque de diversões, nos dias em que se encontra aberto, conforme um calendário anual, geralmente se encontram várias pessoas.
2. No parque há crianças, algumas acompanhadas de seus pais, parentes ou amigos. Também há visitantes que não fazem uso dos brinquedos.
3. Alguns brinquedos são pagos, outros são usados gratuitamente.
4. Há várias centrais de venda de bilhetes.
5. Alguns bilhetes são “universais”, ou seja, são passaportes para qualquer brinquedo do parque. Outros são específicos e podem ser empregados apenas para um subconjunto dos brinquedos, ou até mesmo um único brinquedo.
6. Cada bilhete é adquirido em uma determinada data e uma determinada hora. A venda de cada bilhete é realizada por um funcionário do parque.
7. Bilhetes podem ser trocados por outros nas centrais de vendas. A data e horário da troca é relevante para a contabilidade do parque. A troca pode envolver devolução de diferença ou pagamento adicional.
8. Infelizmente, crianças geralmente se perdem dos seus pais. Nestes casos, é comum a existência de pessoas que as encaminham até os postos da polícia.
9. Nestes postos há representante da justiça e da polícia, que lavram ocorrências.
10. Durante o funcionamento do parque ocorrências (eufemismo para acidentes) acontecem. Uma ocorrência envolve, em geral, várias pessoas. Por exemplo, uma criança que se perdeu dos pais, que a acompanhavam. Cada ocorrência está associada a um local e instante de tempo, além de uma descrição contendo detalhes do evento.
11. O parque possui vários funcionários. Todos os funcionários e demais representantes da justiça e da polícia, que não são funcionários do parque, estão sob a coordenação do diretor do parque.
12. Cada funcionário faz uso de um rádio através do qual a comunicação com os demais é possível, em particular com o diretor em exercício. Cada rádio possui uma identificação única e a alocação de um rádio a um funcionário é estabelecida pela escala.



O modelo acima ainda pode ser enriquecido com o acréscimo do diagrama abaixo, onde o tipo de um bilhete define o conjunto de brinquedos correspondentes onde este pode ser consumido.





## Prática 10

Modele de forma orientada a objetos a descrição abaixo, produzida por Euclides da Cunha em seu famoso e secular livro *Os Sertões* – tido por muitos como a maior obra da literatura brasileira.

---

### *O sertão é um paraíso*

E o sertão é um paraíso...

Ressurge ao mesmo tempo a fauna resistente das caatingas: disparam pelas baixadas úmidas os caítilos esquivos; passam, em varas, pelas tigüeras, num estrídulo estrepitar de maxilas percutindo, os queixadas de canela ruiva; correm pelos tabuleiros altos, em bandos, esporeando-se com os ferrões de sob as asas, as emas velocíssimas; e as seriemas de vozes lamentosas, e as sericóias vibrantes, cantam nos baledos, à fimbria dos banhados onde vem beber o tapir estacando um momento no seu trote brutal, inflexivelmente retilíneo, pela caatinga, derribando árvores; e as próprias suçuaranas, aterrando os mocós espertos que se aninham aos pares nas luras dos fraguados, pulam, alegres, nas macegas altas, antes de quedarem nas tocaias traiçoeiras aos veados ariscos ou novilhos desgarrados...

Algumas palavras e os respectivos significados segundo o dicionário Houaiss:

*baledo*

terreno pantanoso repleto de plantas.

*estrídulo*

que ou o que se caracteriza pelo som agudo, ruidoso, penetrante.

*fimbria*

parte que delimita; beira, orla.

*fraguedo*

grupo de fragas (rochas escarpadas).

*lura*

buraco feito na terra; esconderijo; toca.

*macega*

erva daninha que nasce em terras cultivadas

*mocó*

roedor da família dos caviídeos do tamanho aproximado de um preá.

*sericóia*

saracura popular no Brasil.

*tapir*

designação dos mamíferos da família dos tapirídeos de corpo pesado e membros curtos; anta.

*tigüera*

roça depois de feita a colheita; milharal já colhido e extinto.

## Prática 11

Uma das obras mais famosas de Platão é o livro *A República*. Abaixo seguem alguns excertos desta obra. Modele-os da perspectiva orientada a objetos usando a UML.

---

- Na medida em que vão murchando para uns os prazeres físicos, nessa mesma aumentam o desejo e o prazer da conversa.
  - A justiça não é outra coisa senão a conveniência do mais forte.
  - Não há nada de grandioso que não tenha dificuldades.
  - Em qualquer empreendimento, o mais trabalhoso é o começo.
  - O corpo mais saudável e mais forte não é o que menos se altera pela ação da comida, da bebida e do esforço, bem como qualquer planta sujeita ao calor do sol, ao vento ou a qualquer acidente dessa espécie? E quanto à alma, não será a mais corajosa e mais sensata a que é menos abalada e alterada por qualquer acidente externo?
  - A mim não me parece ser o corpo, por perfeito que seja, que, pela sua excelência, torna a alma boa.
-

## Prática 12

Modelo de forma orientada a objetos os excertos da mais importante obra de Thomas Hobbes: *Leviatã*.

---

- Se apesar disso verificares que meu trabalho é atacado por todos, talvez te apraza desculpar-me, alegando que sou homem que ama suas próprias opiniões, o qual acredita em tudo o que diz.
- Quando acreditamos que qualquer espécie de afirmação é verdadeira, com base e argumentos que não são tirados da própria coisa nem dos princípios da razão natural, mas são tirados da autoridade e da opinião favorável que temos acerca de quem fez essa afirmação, neste caso o objeto de nossa fé é o orador ou a pessoa em quem acreditamos ou em quem confiamos e cuja palavra aceitamos; e a honra feita ao acreditar é feita apenas a essa pessoa. Conseqüentemente, quando acreditamos que as Escrituras são a palavra de Deus, sem ter recebido qualquer revelação imediata do próprio Deus, o objeto de nossa crença, fé e confiança é a igreja, cuja palavra aceitamos e à qual aquiescemos.
- Os secretos pensamentos de cada pessoa percorrem todas as coisas, sagradas ou profanas, limpas ou obscenas, sérias ou frívolas, sem vergonha ou censura. Coisa que o discurso verbal não pode fazer, limitado pela aprovação do juízo quanto ao momento, ao lugar e à pessoa.
- O valor de um homem, tal como o de todas as outras coisas, é seu preço. Tanto quanto seria dado pelo uso de seu poder. Portanto, não absoluto, mas algo que depende da necessidade e julgamento de outrem. Um hábil condutor de soldados é de alto preço em tempo de guerra presente ou iminente, mas não o é em tempo de paz. Um juiz douto e incorruptível é de grande valor em tempo de guerra. Tal como nas outras coisas, mas não o é tanto em tempo de guerra. Tal como nas outras coisas, também no homem não é o vendedor, mas o comprador quem determina o preço. Porque mesmo que um homem – como muitos fazem – atribua a si mesmo o mais alto valor possível, apesar disso seu verdadeiro valor não será superior ao que lhe for atribuído pelos outros.
- Aqueles que pouca ou nenhuma investigação fazem das causas naturais das coisas, todavia, devido ao medo que deriva da própria ignorância, daquilo que tem o poder de lhes ocasionar grande bem ou mal, tendem a supor e a imaginar por eles mesmos várias espécies de poderes invisíveis, e a se encher de admiração e respeito por suas próprias fantasias. Em épocas de desgraça tendem a invocá-las. Quando esperam um bom sucesso tendem a agradecer-lhes, transformando em seus deuses as criaturas de sua própria fantasia. Foi dessa maneira que aconteceu, devido à infinita variedade da fantasia, terem os homens criado no mundo inúmeras espécies de deuses. Esse medo das coisas invisíveis é a semente natural daquilo a que se chama religião. Esse mesmo medo, naqueles que veneram e temem esse poder de maneira diferente da sua, se chama superstição.  
Tendo esta semente da religião sido observada por muitos, alguns dos que a observaram tenderam a alimentá-la, revesti-la e conformá-la às leis, e a acrescentar-lhe, de sua própria invenção, qualquer opinião sobre as causas dos eventos futuros que melhor parecesse capaz de lhes permitir governar os outros, fazendo o máximo uso possível de seus poderes.

## Prática 13

Modele de forma orientada os domínios laconicamente apresentados abaixo. O modelo correspondente deverá envolver pelo menos dez conceitos.

---

- (a) Oficina mecânica
- (b) Farmácia
- (c) Livraria
- (d) Hospital
- (e) Esporte (aquele de sua preferência)
- (f) Ateliê de costura
- (g) Escola
- (h) Religião
- (i) Escolha um dos ícones abaixo e modele o que ele te evoca. **NOTA:** *cada pessoa pode ter uma reação distinta, ou interpretação diferente para cada uma das figuras abaixo. O sucesso do desenvolvimento de software depende, com certeza, da nossa habilidade em identificar o que se passa na mentes de clientes ao observar o cenário de negócio deles.*



## Considerações finais

O presente manual apresentou a conceituação de modelo de domínio e como produzir tais modelos empregando a orientação a objetos. O objetivo principal é fornecer um conjunto relativamente extenso de situações típicas de qualquer usuário de orientação a objeto. Adicionalmente, alguns cenários não foram comentados e servem como exercícios para exercitar a habilidade de modelagem empregando o paradigma orientado a objetos.