DIAGRAMA DE CLASSES
PERSPECTIVA CONCEITUAL
2ª PARTE

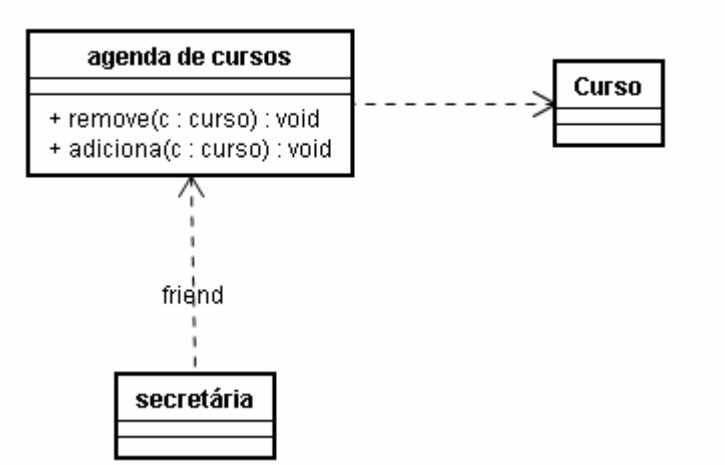
- DICAS
- DEPENDÊNCIAS AVANÇADO
- AGREGAÇÃO
- ATRIBUTOS E ASSOCIAÇÕES DERIVADAS
- ASSOCIAÇÃO TERNÁRIA
- GENERALIZAÇÃO
- ORGANIZAÇÃO DAS CLASSES EM PACOTES
- ELABORANDO O DIAGRAMA
- ERROS COMUNS

DICAS

- Foco: aspecto estático do sistema
- Não prejudicar a leitura com minimalismos
- Generalizações: evitar mais do que 5 níveis
- Nome para cada diagrama
- Evitar linhas cruzadas
- Elementos semânticos semelhantes próximos fisicamente
- Pode-se usar notações visuais que chamem a atenção
- É possível usar mais que um relacionamento, mas tentar evitar

DEPENDÊNCIAS AVANÇADO

- Tipos Definidos pela UML
  - Bind: origem instancia o destino
  - Derive: Origem computada através do destino (ex. Idade -> Data de Nascimento)
  - Friend: Origem recebe visibilidade especial no destino



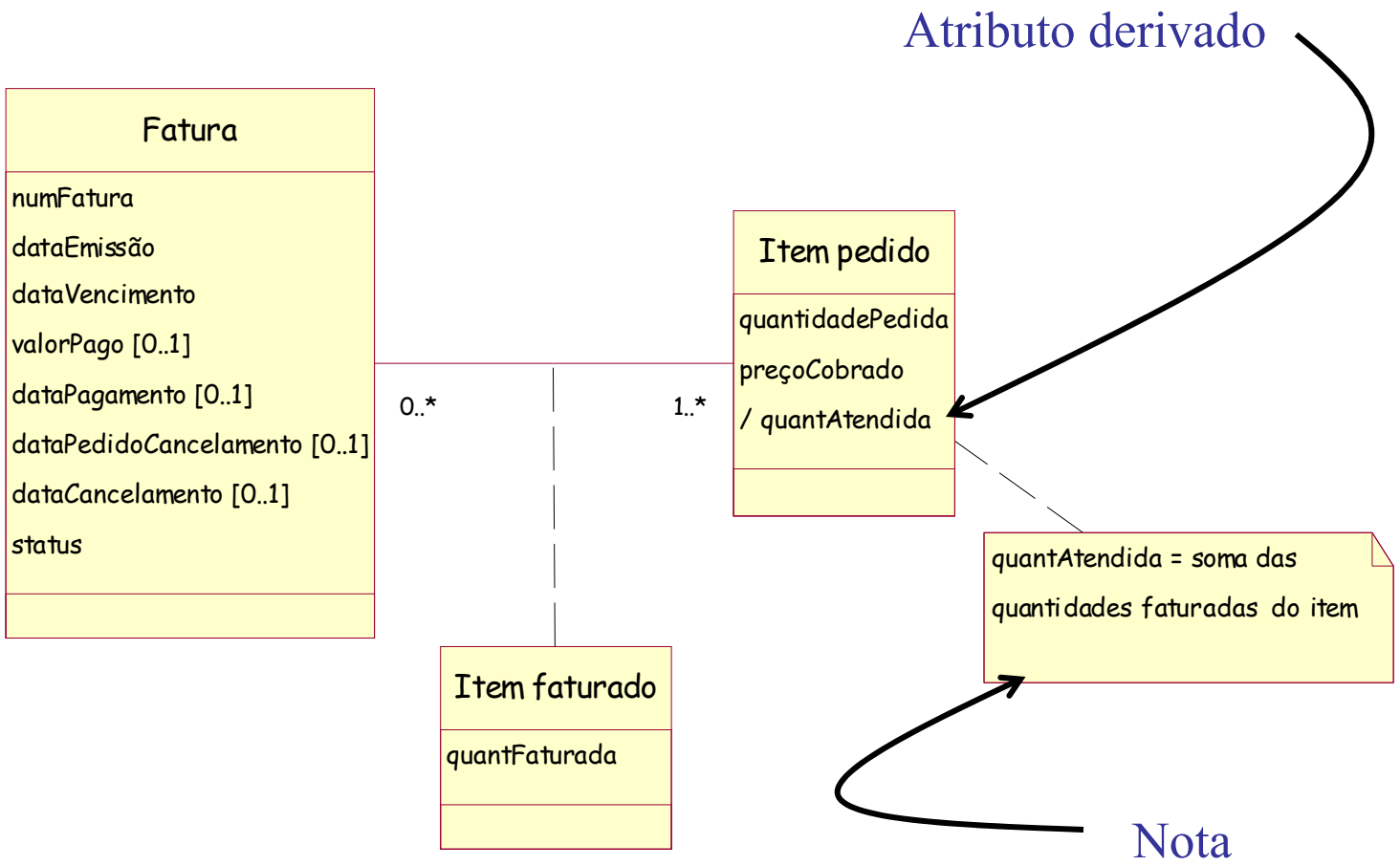
  - InstanceOf
  - Instantiate
  - Powertype
  - Refine
  - Use

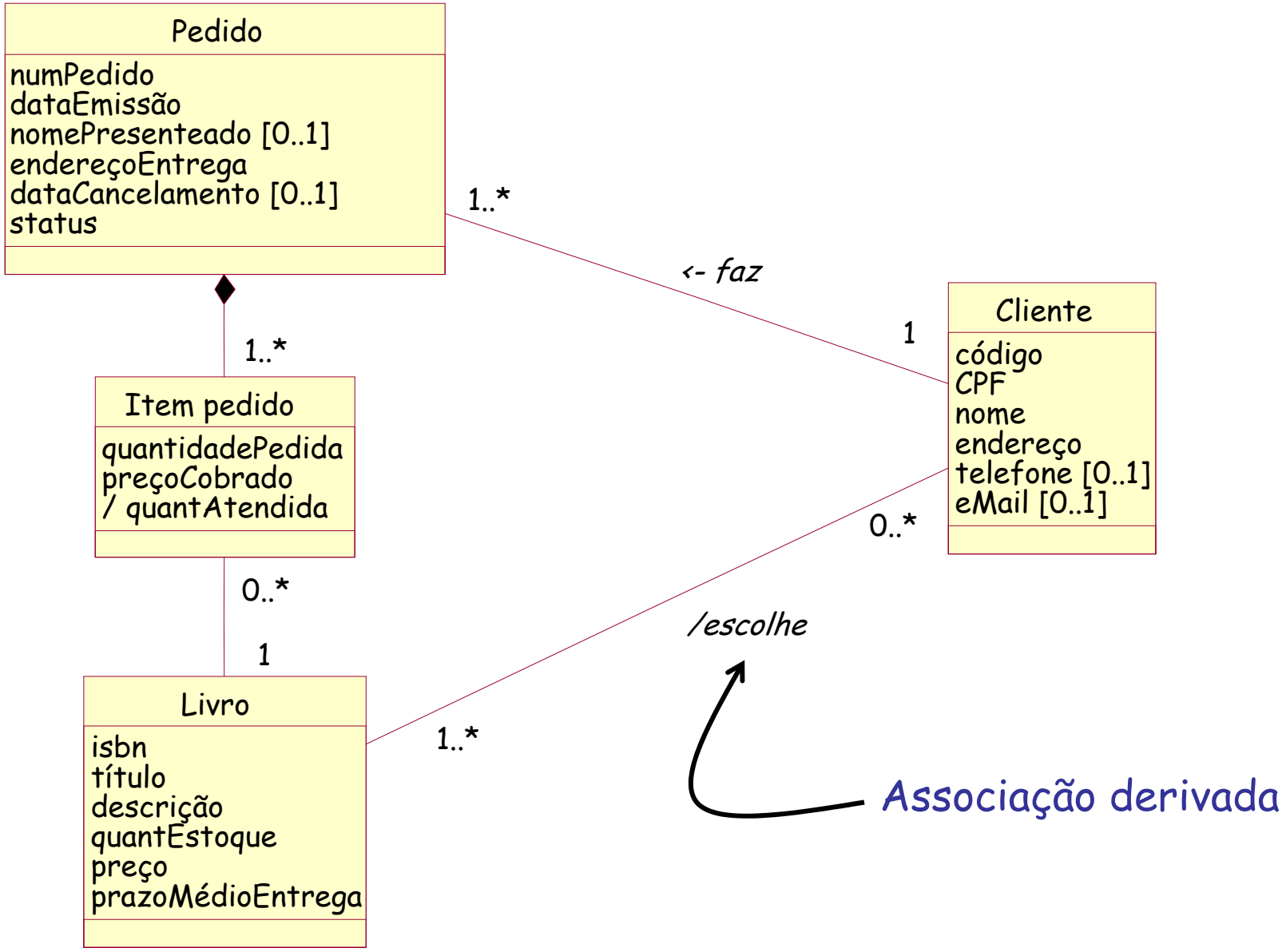
II. ATRIBUTOS E ASSOCIAÇÕES DERIVADAS

- Um elemento derivado é aquele que pode ser calculado a partir de um ou mais elementos mas é incluído no modelo:
  - com o objetivo de tornar algo mais claro ou
  - por algum motivo de projeto. Por exemplo, para tornar uma operação mais fácil de ser realizada.

- Atributos ou associações derivadas podem ser representadas através de uma barra (/) antes do nome da associação ou do atributo.

- Os detalhes sobre o cálculo do elemento derivado podem ser descritos junto a ele, através de uma nota (utilizada na UML para anexar informações a um modelo)


Atributo derivado
Fatura
numFatura
dataEmissão
dataVencimento
valorPago [0..1]
dataPagamento [0..1]
dataPedidoCancelamento [0..1]
dataCancelamento [0..1]
status
0..*
1..*
Item pedido
quantidadePedida
preçoCobrado
/ quantAtendida
quantAtendida = soma das
quantidades faturadas do item
Item faturado
quantFaturada
Nota


Pedido
numPedido
dataEmissão
nomePresenteado [0..1]
endereçoEntrega
dataCancelamento [0..1]
status
1..*
<- faz
1
Cliente
código
CPF
nome
endereço
telefone [0..1]
eMail [0..1]
1..*
Item pedido
quantidadePedida
preçoCobrado
/ quantAtendida
0..*
/escolhe
0..*
1
Livro
isbn
título
descrição
quantEstoque
preço
prazoMédioEntrega
1..*
Associação derivada

II. ASSOCIAÇÃO TERNÁRIA

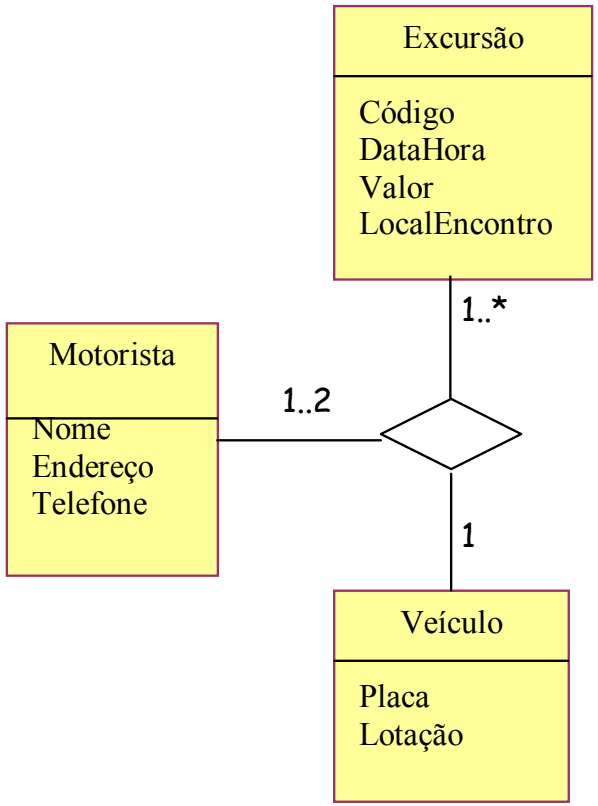
- Associações podem ser binárias, ternárias ou de outro grau.

- Uma associação ternária é uma associação entre três classes, onde uma classe pode ocorrer mais de uma vez.

Exemplo:

Num sistema de controle de reservas de excursões, ao ser criada uma excursão:

- são selecionados um ou mais veículos e
- para cada veículo, podem ser alocados até dois motoristas que o conduzirão nesta excursão.



- Neste caso não ocorre de serem selecionados veículos para a excursão e só depois serem alocados motoristas. Se fosse essa a situação não teríamos um ternário porque a classe motorista não estaria participando do relacionamento e num ternário as três classes têm que participar.

Para incluir a multiplicidade podemos pensar da seguinte maneira:

- Analisamos excursão-veículo com relação ao motorista. Para cada veículo em uma excursão é alocado um ou no máximo dois motoristas. Desta forma do lado do motorista incluímos o multiplicidade 1..2.
- Analisamos motorista-veículo com relação a excursão. Um motorista dirigindo um determinado veículo pode ter participado de várias excursões. Desta forma do lado da excursão teríamos a multiplicidade 1..*.
- Analisando motorista-excursão com relação ao veículo. Um motorista numa determinada excursão só pode ter dirigido um veículo. Desta forma ao lado do veículo teríamos a multiplicidade 1.

III. GENERALIZAÇÃO

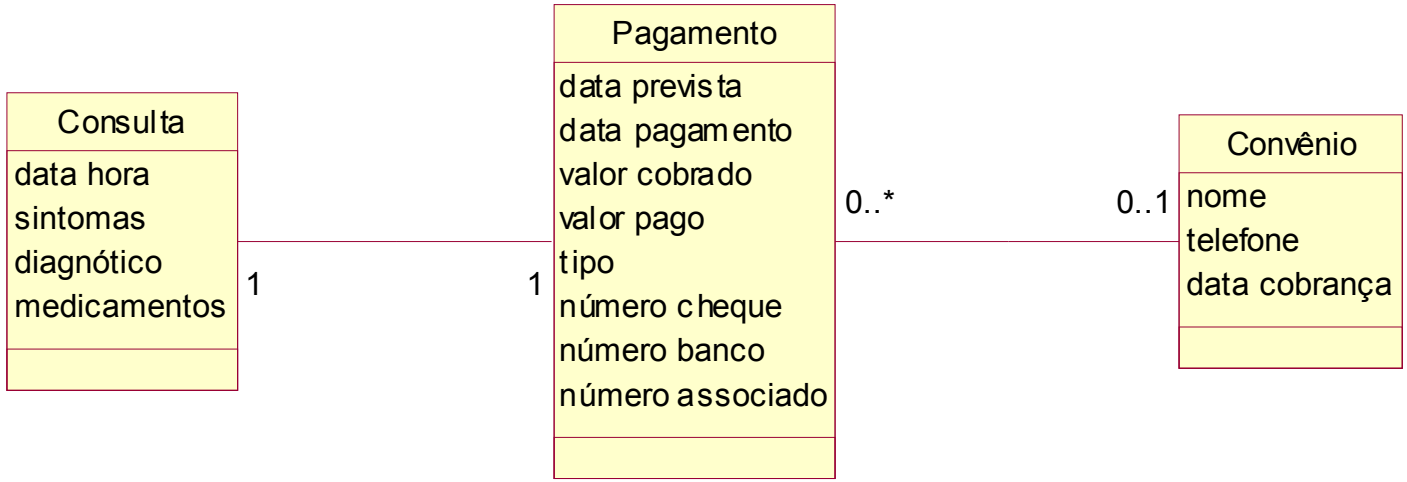
- A generalização é um relacionamento entre um elemento mais genérico (o pai) e um elemento mais específico (o filho) que é totalmente consistente com o primeiro elemento e acrescenta informações adicionais. É um relacionamento utilizado em classes mas também em casos de uso, atores, e outros elementos.

- No diagrama de classes, a generalização é apresentada como uma linha da classe filha para a classe pai com um triângulo ao fim da linha onde está o pai. Representa um relacionamento entre a superclasse e a subclasse. A subclasse herda as propriedades do pai, como atributos e operações e pode ter suas próprias propriedades.

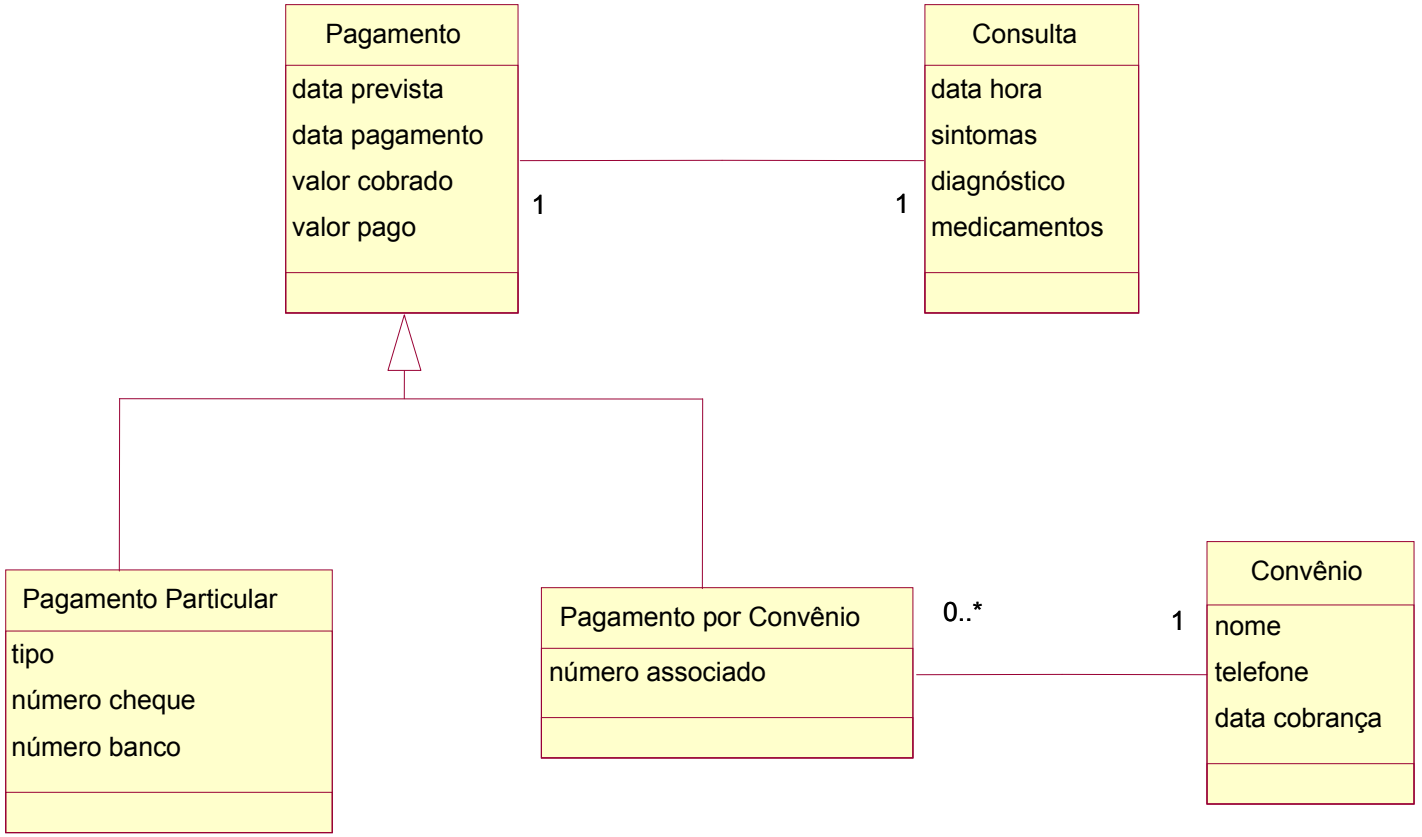
- Às vezes a decisão por modelar uma generalização começa quando temos uma classe com informações de vários tipos.

Por exemplo, num consultório médico o pagamento de uma consulta pode ser feito através de convênio ou particular. Neste último caso o cliente pode pagar com cheque ou dinheiro.

Na solução a seguir foram incluídos em Pagamento atributos relacionados a cada um desses tipos de pagamento.



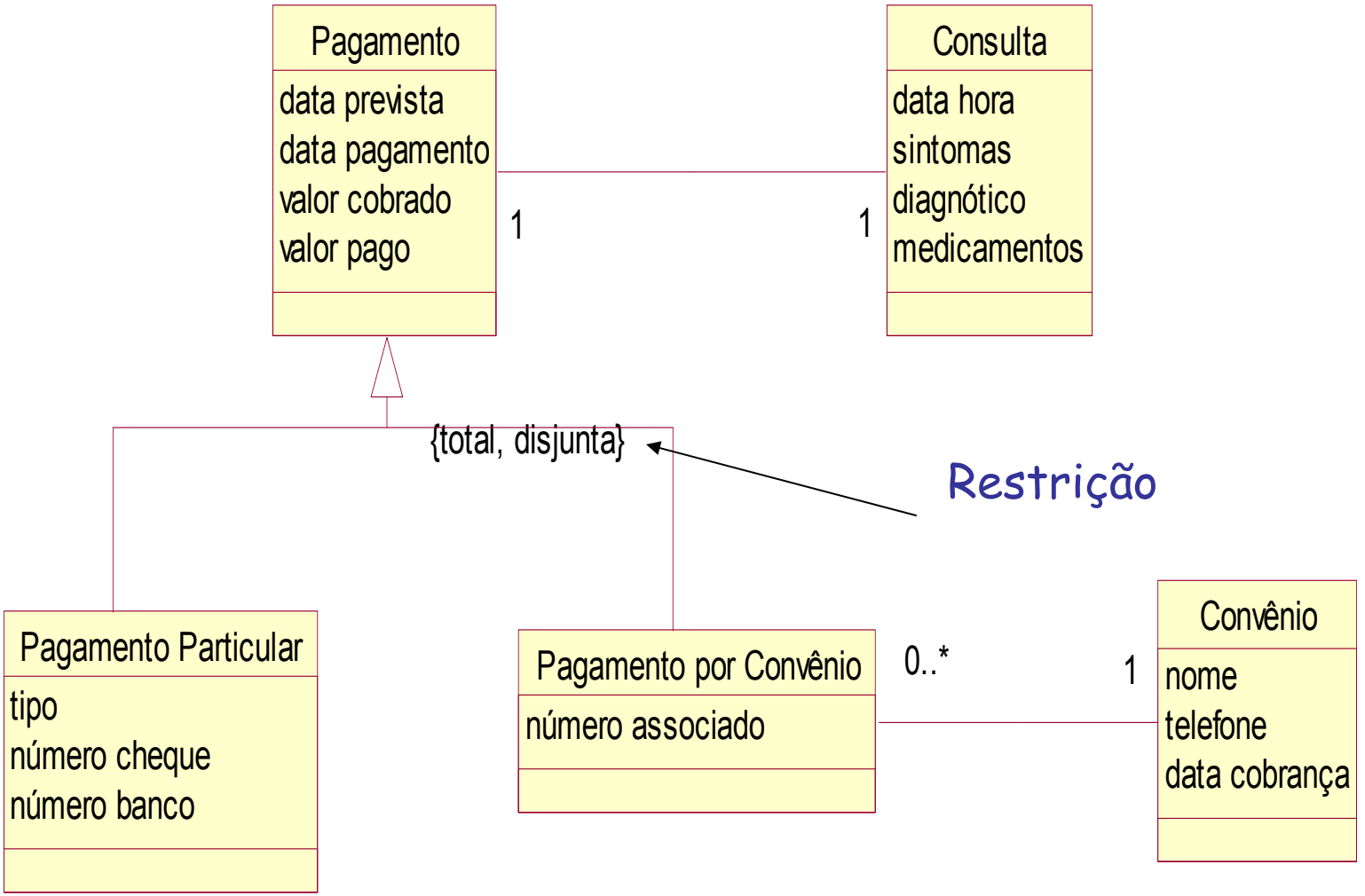
- Modelando com a *generalização* podemos ter maior clareza sobre pagamento, como podemos observar no diagrama a seguir.
  - Foram criadas duas subclasses, pagamento particular e pagamento por convênio.
  - Em pagamento estão os atributos comuns a pagamento particular e por convênio .
  - Em pagamento particular temos os atributos próprios só deste tipo e em pagamento por convênio os atributos e associações particulares ao pagamento por convênio.
  - Como consulta está relacionada à pagamento, todas as subclasses têm este relacionamento.


Pagamento
data prevista
data pagamento
valor cobrado
valor pago
1
1
Consulta
data hora
sintomas
diagnóstico
medicamentos
Pagamento Particular
tipo
número cheque
número banco
Pagamento por Convênio
número associado
0..*
1
Convênio
nome
telefone
data cobrança

- Uma generalização pode ser total ou parcial, disjunta ou sobreposta.

  - **Total:** Todas as subclasses são especificadas
  - **Parcial:** Nem todas as subclasses são especificadas
  - **Sobreposta:** As classes derivadas de uma superclasse podem ocorrer simultaneamente
  - **Disjunta:** As classes derivadas de uma superclasse podem ocorrer de maneira mutuamente exclusivas.

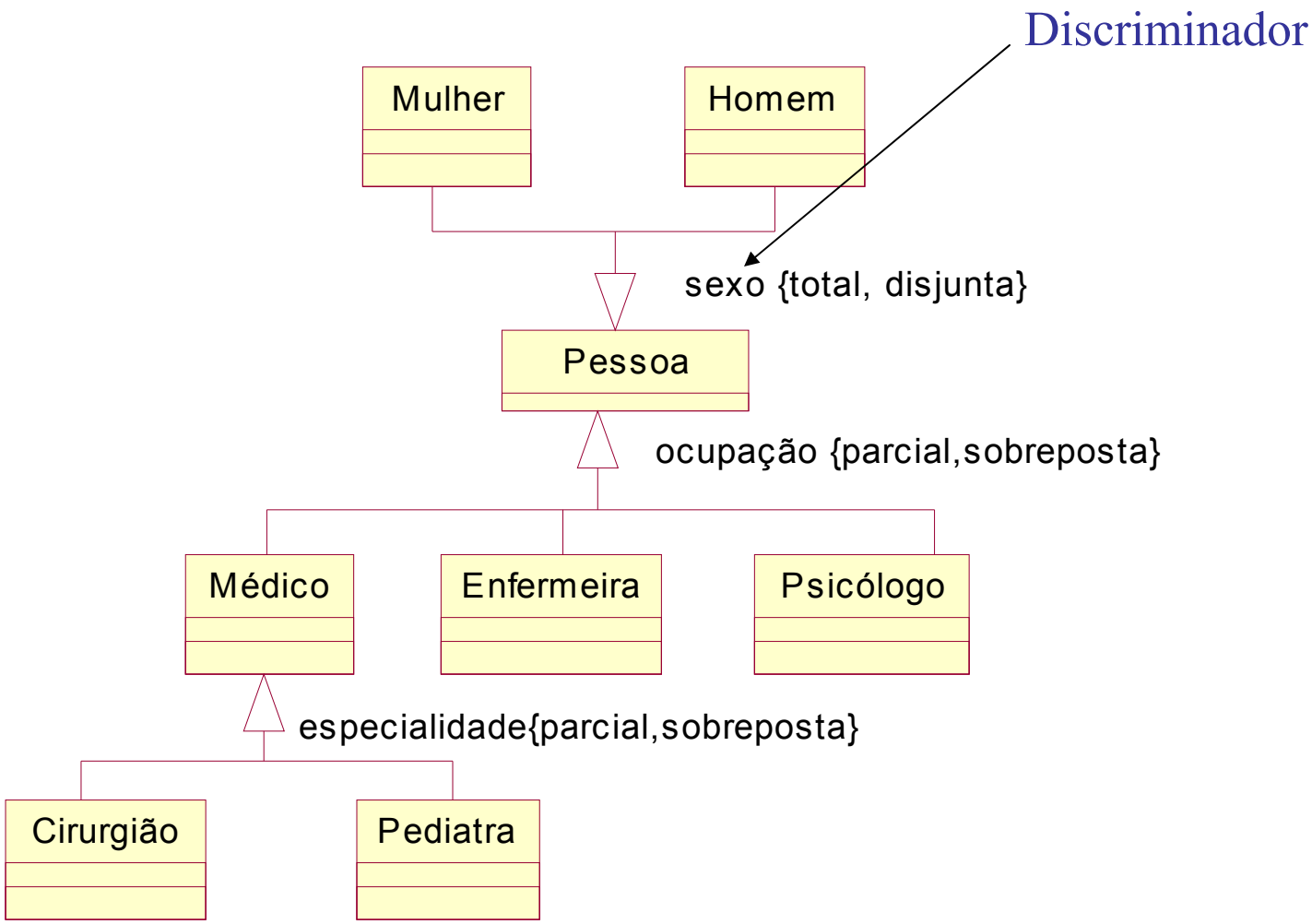
Restrições, já comentadas anteriormente, podem ser utilizadas para descrever o tipo da generalização.


Pagamento
data prevista
data pagamento
valor cobrado
valor pago
1
1
Consulta
data hora
sintomas
diagnótico
medicamentos
{total, disjunta}
Restrição
Pagamento Particular
tipo
número cheque
número banco
Pagamento por Convênio
número associado
0..*
1
Convênio
nome
telefone
data cobrança

- No caso anterior, as subclasses de pagamento foram criadas em função do tipo de pagamento. Pode ser necessário, no entanto, fazer várias classificações.

  - Classificação única: No exemplo a seguir médico é classificado somente quanto a sua especialidade.
  - Classificação múltipla: No exemplo a seguir temos pessoa classificada por sexo e por ocupação.

Quando temos uma classificação múltipla, discriminadores devem ser utilizados para diferenciar esses tipos. Ao lado do discriminador podemos escrever as restrições entre chaves.


Discriminador
Mulher
Homem
sexo {total, disjunta}
Pessoa
ocupação {parcial,sobreposta}
Médico
Enfermeira
Psicólogo
especialidade{parcial,sobreposta}
Cirurgião
Pediatra

- Quando uma subclasse tem apenas uma classe pai temos uma herança simples. Mas é possível ocorrer de uma classe ter várias classes pai e desta forma teremos uma herança múltipla.

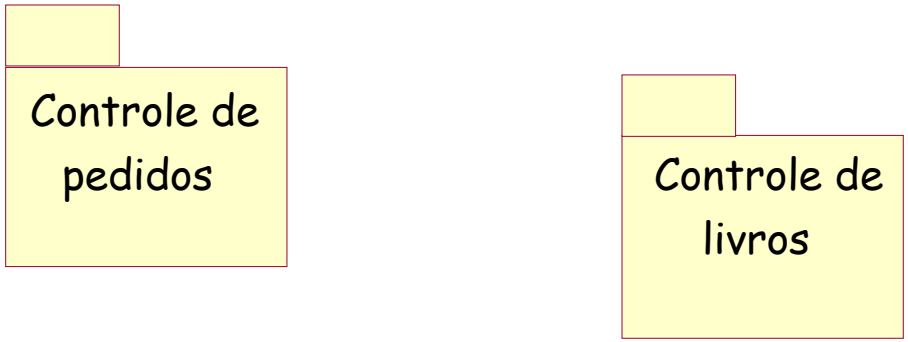
IV. ORGANIZAÇÃO DAS CLASSES EM PACOTES

- Na UML os modelos podem ser organizados em packages (ou pacotes) de forma que possamos compreendê-los mais facilmente.

- O package é formado por um grupo de elementos com um tema comum. Esses elementos podem ser classes, componentes, casos de uso e até mesmo outros pacotes.

Exemplo:

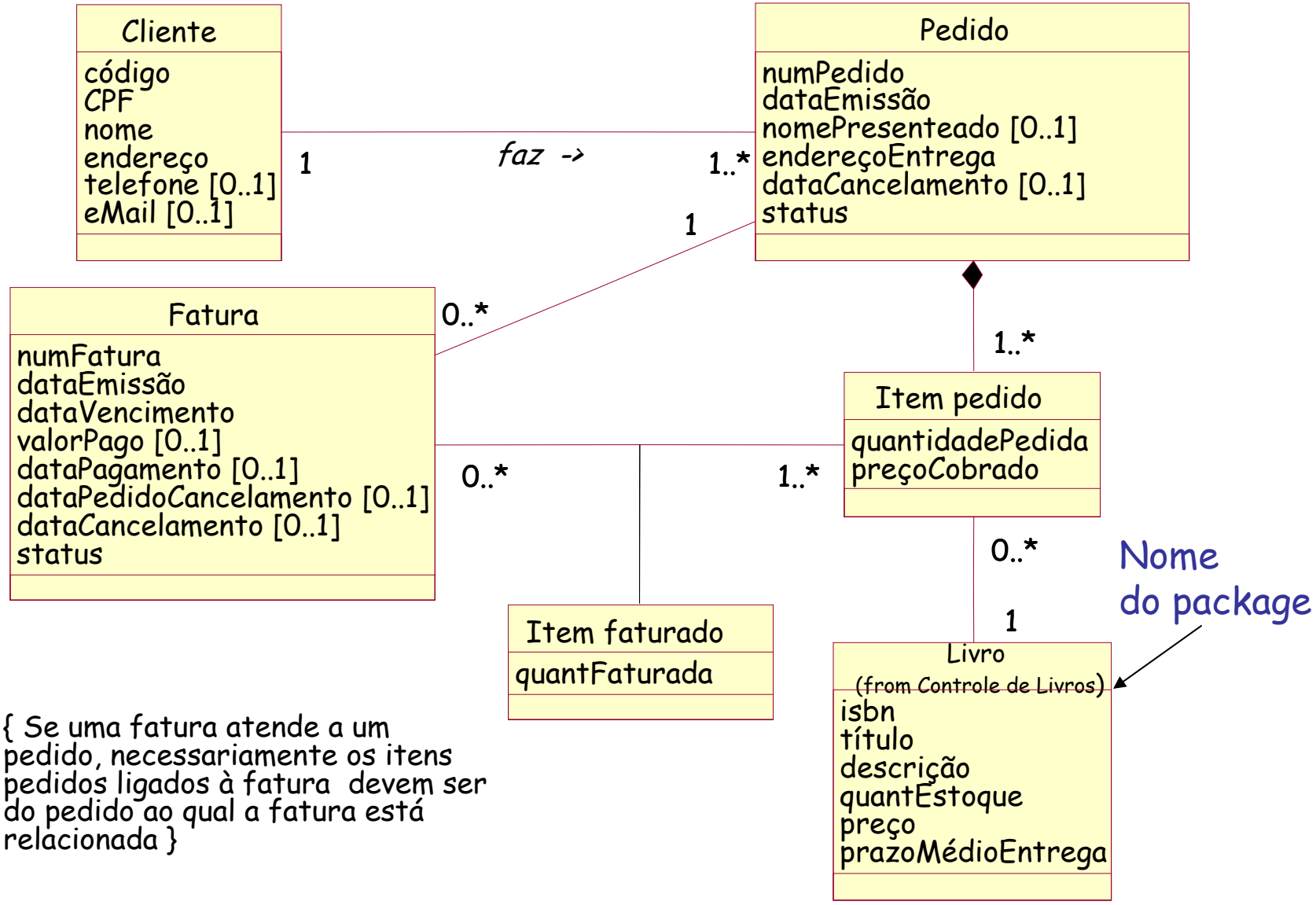
- Poderíamos no caso exemplo, ter dois packages: Controle de pedidos e Controle de Livros



- O diagrama de classes apresentado a seguir pertence ao package Controle de pedidos, que contém as classes próprias à administração de pedidos e faturamento

Classes de outros packages podem aparecer neste diagrama, caso seja importante, como é o caso de Livro. Ao lado do nome, no entanto, vem discriminado o nome do package ao qual pertence.

- O package Controle de Livros contém a classe livro que pertence ao sistema de Controle de Livros



{ Se uma fatura atende a um pedido, necessariamente os itens pedidos ligados à fatura devem ser do pedido ao qual a fatura está relacionada }

V. ELABORANDO O DIAGRAMA

1. A elaboração da primeira versão do diagrama de classe com uma perspectiva conceitual pode se realizada em paralelo à leitura dos casos de uso. Na descrição de cada caso de uso há uma série de informações, conceitos, nomes de atributos, etc, que serão úteis para descobrir as classes, associações e atributos

2. Ao elaborar o diagrama de classes é possível que se constate a falta de informações, a existência de inconsistências e haverá necessidade então de se consultar o cliente.

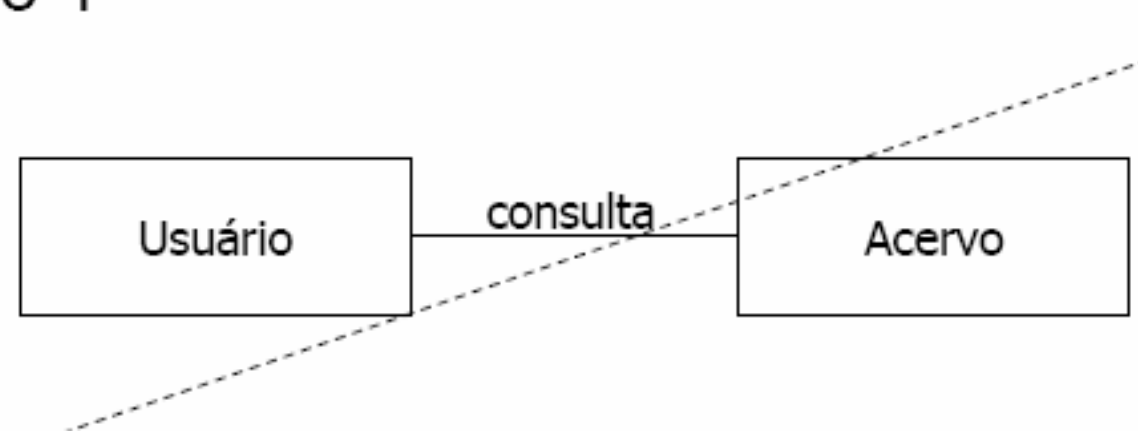
3. A elaboração de modelos é realizada através de várias repetições e o modelo deve ser construído em conjunto com os demais modelos, já que alterações em um modelo provocam alterações em outros.

4. Após a elaboração da primeira versão podemos tentar refinar o modelo verificando a possibilidade de incluir notações como composição, generalização que permitem que as informações fiquem descritas de forma mais clara.

5. Conforme é elaborado o diagrama pode ser necessário também organizar as classes em packages.

Erros comuns

- Usar classes ou associações para representar consultas ou operações do sistema que não devem ser registradas
  - Exemplo 1



Erros Comuns

- Usar classes ou associações para representar consultas ou operações do sistema que não devem ser registradas
  - Exemplo 2

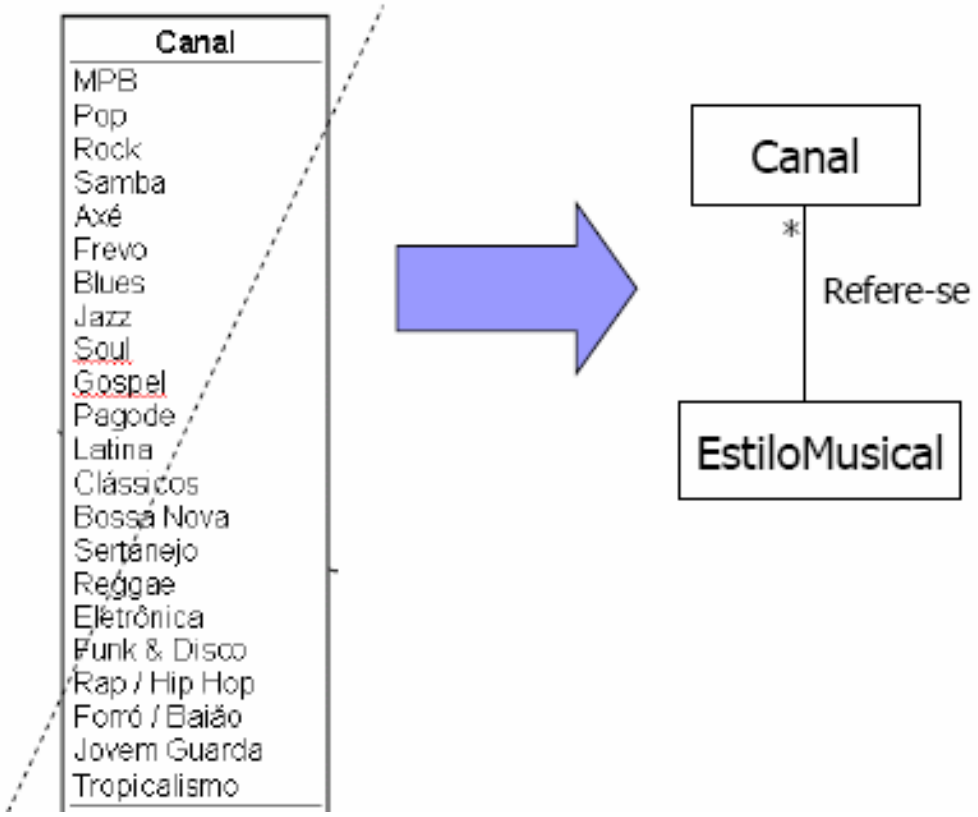
Erros Comuns

- Identificar métodos nas classes sem ter feito a modelagem temporal

O que é sintonizar?
-Quem usa?
-Quais os parâmetros?

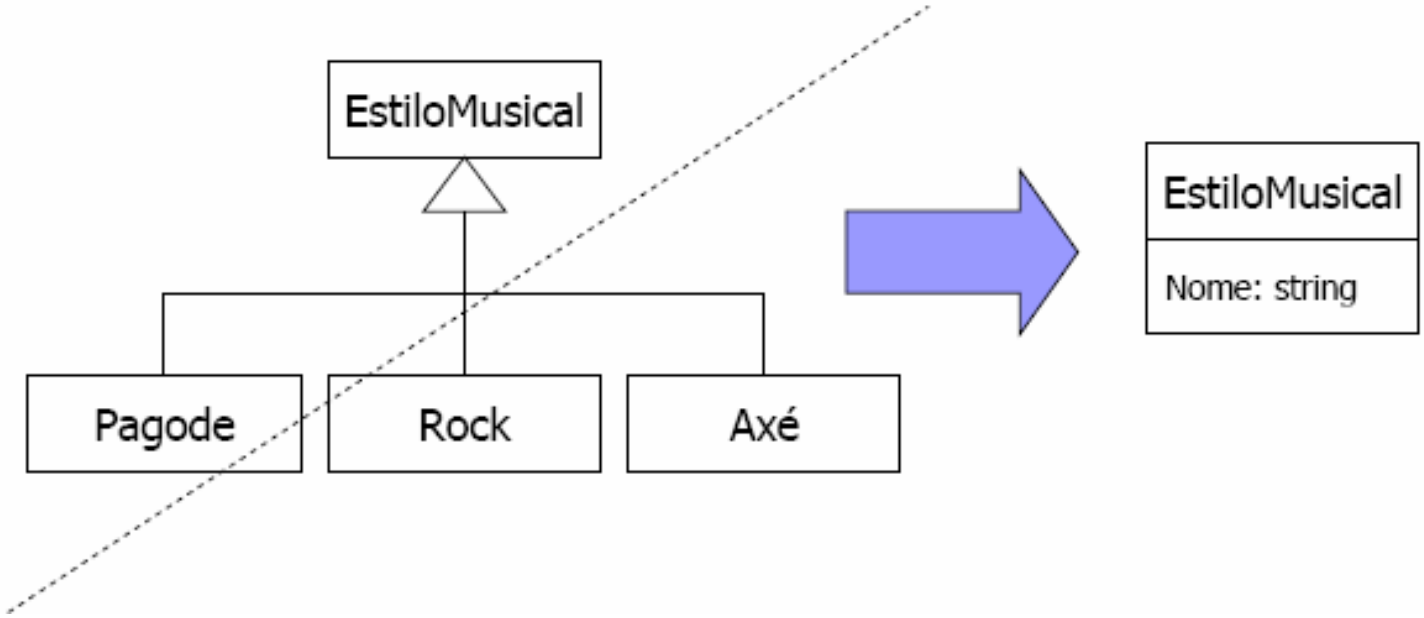
Erros Comuns

- Inserir atributos quando o ideal é criar uma classe



Erros Comuns

- Usar herança quando a quantidade de tipos é grande ou dinâmica



Erros Comuns

- Inserir chaves-estrangeiras no diagrama de classes
  - As associações são suficientes