

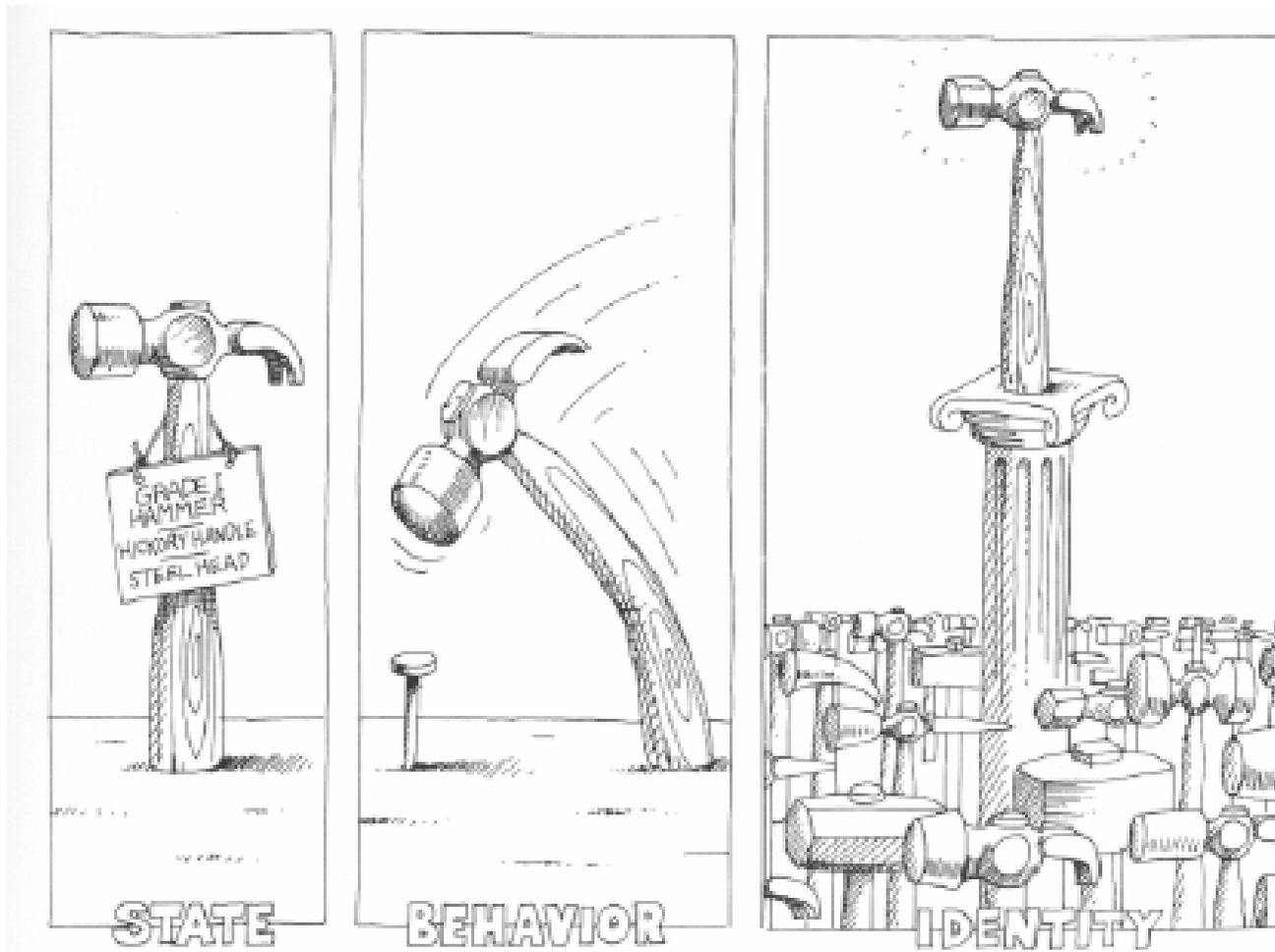
# DIAGRAMA DE CLASSES PERSPECTIVA CONCEITUAL 1ª PARTE

- DIAGRAMA
- CLASSE, ATRIBUTO E OPERAÇÃO
- ASSOCIAÇÃO
- CLASSE ASSOCIATIVA
- AGREGAÇÃO E COMPOSIÇÃO
- RESTRIÇÕES
- ELABORANDO O DIAGRAMA

## I. DIAGRAMA DE CLASSES

- Possibilita uma visão estática do sistema
- Ao elaborarmos um diagrama de classe devemos fazê-lo numa única perspectiva: conceitual ou de implementação
- Seguindo a perspectiva conceitual, os objetos do modelo devem ser conceitos do domínio da aplicação e não conceitos de implementação computacional.

- Estado, Comportamento e Identidade de Objetos



## □ Estado

- É determinado pelos valores armazenados em um objeto
- Sempre que houver alteração em um valor do objeto, diz-se que houve uma transição de estado
- Transição de estado: importante para identificar operações e necessidade de sincronização de eventos em um sistema

## □ **Comportamento**

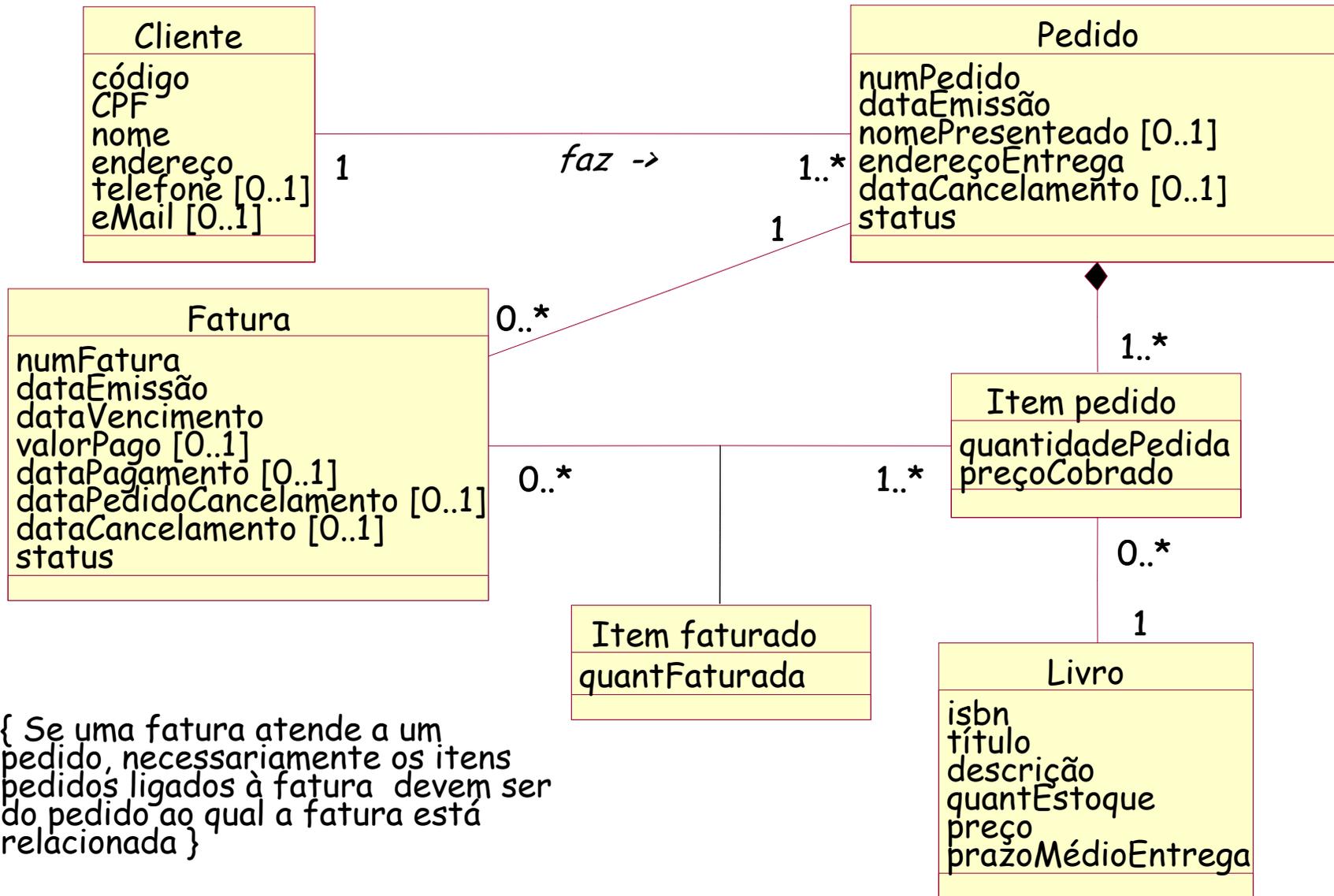
- Mensagens que um objeto/classe pode receber
- Todo objeto é passivo, esperando por requisições de outros objetos ou classes
  - A “passividade” de objetos é a justificativa para a necessidade de métodos especiais como o `main()` em Java

## □ Identidade de Objetos

- Todo objeto criado - seja em uma linguagem de programação ou sistema de BD OO - é associado a um identificador único
- Não existe chave primária (ou estrangeira) no modelo de objetos!
- Em algumas linguagens, a função a seguir retorna falso:

```
boolean testIdentidade() {  
    String s1, s2;  
    s1 = "Java"; s2 = "Java";  
    return (s1 == s2);  
}
```

## Diagrama de Classes (com perspectiva conceitual)

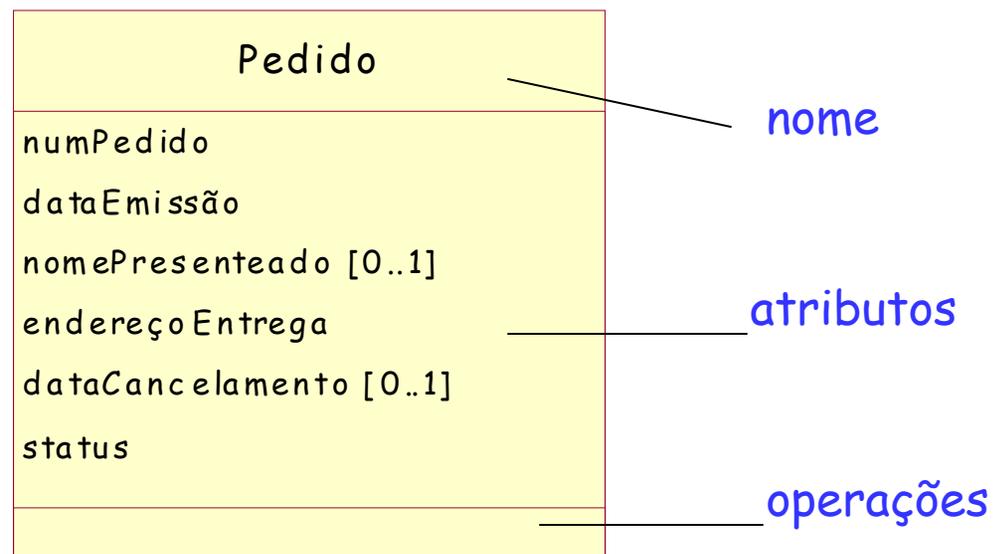


## II. CLASSE, ATRIBUTO E OPERAÇÃO

### CLASSE

- Na abordagem orientada a objetos os dados são subdivididos em **objetos**
- Cada objeto tem sua própria identidade. Assim, dois pedidos, no sistema de controle de pedidos, por mais semelhanças que contenham constituem cada um, um único objeto
- Objetos com a mesma estrutura de dados (atributos), com o mesmo comportamento (operações) e relacionamentos são agrupados numa **classe**.

- Classe é uma abstração que descreve propriedades importantes para uma aplicação e ignora as restantes.
- É representada através de um retângulo, com espaço para descrição do nome, dos atributos e das operações:



## Elicitação de Classes na Etapa de Análise

- Responsabilidades
- Empacotamento de Classes (Responsabilidades comuns ou semelhantes)
- Colaborações Mútuas
- Cuidado: Análise x Projeto (Conceito x Implementação)

## ATRIBUTO

- Atributo: é uma propriedade de uma classe.

O atributo *dataEmissão* da classe Pedido significa que um objeto pedido possui data de emissão. Este atributo também indica que um objeto pedido pode dizer, a quem solicitar, sua data de emissão e tem alguma forma de modificá-lo através de uma operação.

- Sintaxe completa de um atributo:

[visibilidade] nome [ [multiplicidade] ] [:tipo]  
[= valor inicial] [{propriedade}]

- Se o diagrama está sendo elaborado com uma perspectiva conceitual basta representar o nome do atributo e a sua multiplicidade, que indica a quantidade mínima e máxima de ocorrências daquele atributo em um objeto.

- Opções de multiplicidade:

1 - exatamente uma ocorrência

0..1 - zero ou uma

1..\* - uma ou mais

0..\* - zero ou mais

\* - zero ou mais

É possível também determinar o número exato de ocorrências. Por exemplo, 2..6

- Pedido tem seis atributos:
  - numPedido, dataEmissão, endereçoEntrega e status têm uma única ocorrência. Neste caso não é necessário indicar a multiplicidade.
  - nomePresenteado: multiplicidade mínima = zero, já que nem sempre há presenteado
  - dataCancelamento: multiplicidade mínima = zero, já que nem sempre o pedido é cancelado

| Pedido                  |
|-------------------------|
| numPedido               |
| dataEmissão             |
| nomePresenteado [0..1]  |
| endereçoEntrega         |
| dataCancelamento [0..1] |
| status                  |

## OPERAÇÃO

- Operação: um serviço que pode ser solicitado por algum objeto da classe. Por exemplo, um objeto pedido pode solicitar o seu cancelamento.
- Uma classe pode ter várias operações ou nenhuma
- O motivo de não serem descritas operações na classe Pedido não é o fato de não haver operações. Optamos por incluir operações somente na etapa de Projeto quando o diagrama de classes for elaborado com uma perspectiva de implementação.

## Exemplo

### Faz pedido

*Cenário principal: Venda realizada com sucesso por um novo cliente*

1. Cliente informa dados pessoais (cpf, nome, endereço, telefone e e-mail) e endereço de entrega
2. Cliente informa a lista dos livros desejados e respectivas quantidades
3. Sistema armazena além dos dados fornecidos pelo cliente, a data de emissão do pedido e o preço cobrado por cada livro, já que pode ser dado algum desconto e o valor cobrado não será o de tabela
4. Sistema envia ao cliente a confirmação da venda com o número de seu pedido, seu código, a lista dos itens pedidos com quantidade e preço de cada item e o preço total do pedido.

| Cliente         |
|-----------------|
| código          |
| CPF             |
| nome            |
| endereço        |
| Telefone [0..1] |
| eMail [0..1]    |

| Pedido          |
|-----------------|
| numPedido       |
| dataEmissão     |
| endereçoEntrega |

| Item pedido      |
|------------------|
| quantidadePedida |
| preçoCobrado     |

| Livro             |
|-------------------|
| isbn              |
| título            |
| descrição         |
| quantEstoque      |
| preço             |
| prazoMédioEntrega |

Obs: A classe livro faz parte de outro subsistema.

## Faz pedido para presentear

1. Cliente informa dados pessoais (cpf, nome, endereço, telefone e e-mail)

Cliente informa dados do presenteado:  
nome e endereço de entrega

Continua a partir do passo 2.

## Pedido

numPedido

dataEmissão

nomePresenteado [0..1]

endereçoEntrega

## Fatura pedido

*Cenário principal: faturamento de pelo menos um item do pedido*

1. Funcionário seleciona um pedido que não tenha sido integralmente atendido (faturado)
2. Sistema verifica a quantidade pendente ( $\text{quantidadePedida} - \text{quantidadeAtendidaTotal}$ ) de cada item (*Extend - Comunica atraso*)
3. Sistema cria uma fatura com o número da fatura, a data de emissão, a data limite de pagamento e a quantidade de cada item.
4. Sistema emite a fatura que deverá ser encaminhada ao cliente juntamente com os livros. A fatura deverá conter:
  - Número da fatura
  - Número do pedido
  - Data de emissão
  - Data de vencimento
  - Para cada item: a quantidade e o preço unitário cobrado
  - Preço total

*Obs:*

- *A quantidade faturada de cada item (livro) está limitada ao que há em estoque. Caso não possa ser feito um atendimento completo neste momento, mais adiante, logo que haja o item em estoque, será criada uma nova fatura.*
- *Uma fatura faz referência a um e apenas um pedido. No entanto ela pode estar atendendo apenas parcialmente àquele pedido.*

| Cliente  |
|----------|
| código   |
| CPF      |
| nome     |
| endereço |
| telefone |
| eMail    |

| Pedido                 |
|------------------------|
| numPedido              |
| dataEmissão            |
| nomePresenteado [0..1] |
| endereçoEntrega        |
| status                 |

| Fatura         |
|----------------|
| numFatura      |
| dataEmissão    |
| dataVencimento |
| status         |

| Item pedido      |
|------------------|
| quantidadePedida |
| preçoCobrado     |

| Item faturado |
|---------------|
| quantFaturada |

| Livro             |
|-------------------|
| isbn              |
| título            |
| descrição         |
| quantEstoque      |
| preço             |
| prazoMédioEntrega |

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

## Comunica atraso

- Sistema verifica que um ou mais itens pedidos não poderão ser entregues e que não há previsão de entrega.
- Sistema comunica ao cliente o atraso descrevendo o número do pedido e os itens para os quais não há previsão de entrega.

## Diminui quantidade de um item do pedido

### *Cenário principal: Quantidade diminuída*

1. Cliente informa seu código
2. Sistema valida código
3. Cliente informa número do pedido
4. Sistema valida número do pedido (include - Valida pedido)
5. Cliente informa o item a ser reduzido
6. Sistema apresenta ao cliente a quantidade pendente (quantidade pedida - quantidade faturada)
7. Cliente informa a nova quantidade (no máximo a quantidade pendente)
8. Sistema armazena a nova quantidade
9. Sistema envia ao cliente a confirmação da alteração realizada

## Diminui quantidade de um item do pedido

*Cenário alternativo: Pedido de redução de item é recusado por já ter sido totalmente faturado*

### 6. Sistema comunica ao cliente:

- o item não pode ser reduzido por ter sido completamente faturado
- caso deseje realmente esta redução o cliente deverá solicitar o cancelamento parcial ou total da fatura. Esse pedido será avaliado pelo gerente.

Os passos seguintes não são realizados

## Solicita cancelamento de pedido

*Cenário principal: Pedido cancelado por não haver fatura emitida*

1. Cliente informa seu código
2. Sistema valida código
3. Cliente informa número do pedido
4. Sistema valida número do pedido (*include - Valida pedido*)
5. Sistema cancela o pedido (não há nenhuma fatura emitida para ele) e armazena a data de cancelamento
6. Sistema envia ao cliente a confirmação do cancelamento solicitado.

## Pedido

numPedido

dataEmissão

nomePresenteado [0..1]

endereçoEntrega

dataCancelamento [0..1]

status

## Solicita cancelamento de pedido

*Cenário alternativo: Pedido não pode ser cancelado por já ter sido emitida pelo menos uma fatura*

### 5. Sistema comunica ao cliente:

- o pedido não pode ser cancelado por já ter sido emitida pelo menos uma fatura.
- o cliente deverá solicitar o cancelamento das faturas já emitidas caso deseje cancelar o pedido.

Os passos seguintes não são realizados

## Valida pedido

1. Sistema verifica a existência do número do pedido
2. Sistema envia ao cliente os dados relativos a seu pedido:
  - a lista dos itens pedidos com quantidade e preço unitário de cada item e o preço total do pedido
  - a lista das faturas emitidas contendo para cada fatura:
    - Número da fatura
    - Número do pedido
    - Data de emissão
    - Data de vencimento
    - Para cada item: a quantidade e o preço unitário
    - Preço total

## Solicita cancelamento de fatura

*Cenário principal: Solicitação de cancelamento integral da fatura realizada com sucesso*

1. Cliente informa seu código
2. Sistema valida código
3. Cliente informa número da fatura
4. Sistema verifica a existência deste número
5. Sistema envia ao cliente os dados da fatura, contendo: a data de emissão, o preço total e para cada item: a quantidade e o preço unitário
6. Cliente solicita o cancelamento integral da fatura
7. Sistema armazena a solicitação de cancelamento da fatura e a data da solicitação
8. Sistema envia ao cliente a confirmação do cadastramento de sua solicitação e a informação de que o seu pedido só será analisado quando a Empresa receber os livros relativos à fatura.

## Fatura

numFatura  
dataEmissão  
dataVencimento

dataPedidoCancelamento [0,1]

status

## Paga fatura

*Cenário principal: Pagamento da última fatura de um pedido*

1. Cliente paga a fatura
2. Sistema armazena o número da fatura, o valor pago e a data de pagamento.
3. Sistema fecha o pedido relacionado à fatura (foi paga a última fatura de um pedido)
4. Sistema envia ao cliente a confirmação de pagamento da fatura

## Fatura

numFatura

dataEmissão

dataVencimento

valorPago [0..1]

dataPagamento [0..1]

dataPedidoCancelamento [0..1]

status

## Avalia cancelamento de fatura

*Cenário principal: Gerente cancela a fatura e automaticamente é cancelado o pedido*

1. Gerente analisa o pedido de solicitação de cancelamento de uma fatura
2. Gerente cancela a fatura
3. Sistema armazena a data de cancelamento e atualiza o estoque, considerando a devolução dos livros
4. Sistema cancela o pedido (todas as faturas de um pedido foram canceladas)
5. Sistema envia ao cliente notificação do cancelamento da fatura e do pedido

## Fatura

numFatura

dataEmissão

dataVencimento

valorPago [0..1]

dataPagamento [0..1]

dataPedidoCancelamento [0..1]

dataCancelamento [0..1]

status

## Exercicio 1: Elabore as Classes para o problema abaixo:

Sua empresa foi solicitada para o desenvolvimento de um software para controle de um hospital. Este Hospital possui 5 centros: CTI, Emergência, Maternidade, Pediatria e Clinica Geral. Cada um destes centros possui uma série de características, tais como número de leitos, número de médicos, capacidade de espera, horários de atendimento, etc. Um paciente (que também deve possuir uma série de atributos), ao chegar, deve ser encaminhado para um destes centros, mas antes deve passar pela recepção do hospital para que sua ficha seja preenchida no sistema e para que o sistema, além de registrá-lo, realize o encaminhamento para o centro e respectivo médico responsável. Devem haver prioridades de atendimento ao paciente, sendo um dos atributos a gravidade do caso (esta gravidade deve variar de 1 a 5). Caso seu problema não se adeque, o hospital possui convênio com outros hospitais, aos quais encaminhará estes pacientes. Este encaminhamento será efetuado por uma das ambulâncias do primeiro hospital. Os médicos do hospital possuem diversas características, tais como nome, especialidade, CRM, turno. Enfermeiros são outros elementos distintos e também devem possuir uma serie de características.

- A operação é especificada através de sua **assinatura** (que contém o nome, tipo de todos os parâmetros e o tipo do valor a ser retornado) além de outras características, como visibilidade. Mas esses detalhes serão vistos na etapa de Projeto.
- Caso operações sejam incluídas no diagrama de classes elaborado com uma perspectiva conceitual, devem apenas indicar as responsabilidades principais da classe.

## RELACIONAMENTOS

## Relacionamentos - Dependências

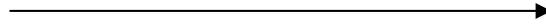
----->

- Dependência: Relacionamento de Utilização

Pode usar como assinatura, por exemplo

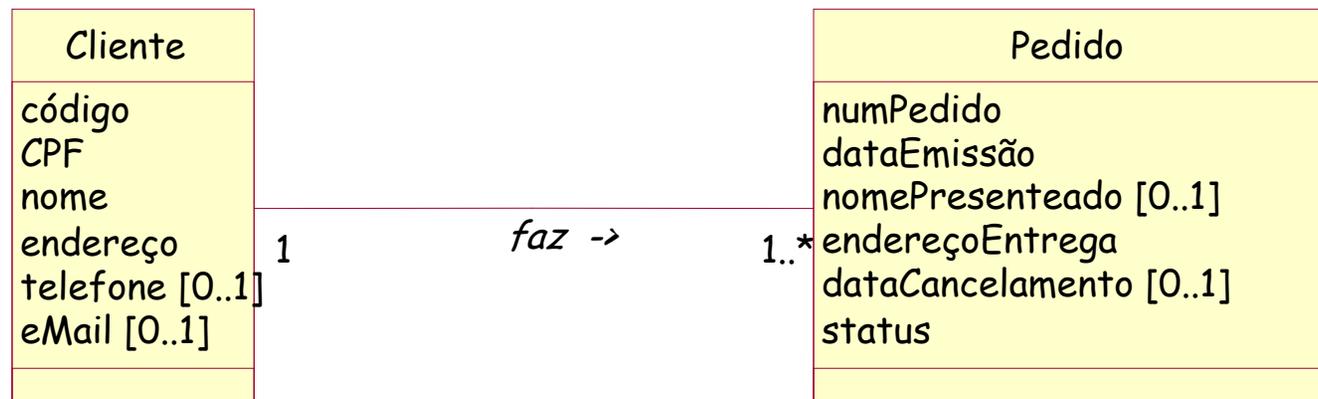
## Relacionamentos - Generalização

Todo - Parte



## Relacionamentos - Associação

- Associação: representa o relacionamento entre instâncias de classes. Especifica que um objeto de uma classe está ligado a um objeto de outra classe.
- Exemplo: faz é o nome da associação. Pode ser utilizada uma seta para indicar uma direção para este nome de forma a facilitar a leitura e entendimento. Neste caso leríamos *cliente faz pedido*. Quando não há dúvidas com relação ao nome da associação, ele pode ser omitido.



## Relacionamentos - Associação

- Nome Opcional
- Direção do Nome

- Associações possuem *multiplicidades* que indicam os limites mínimos e máximos para a participação dos objetos. No exemplo anterior, a associação entre cliente e pedido pode ser lida da seguinte forma: **cada cliente pode ter feito um ou mais pedidos e cada pedido necessariamente é de apenas um cliente.**

- Opções de multiplicidade:

1 - exatamente 1

0..1 - zero ou um

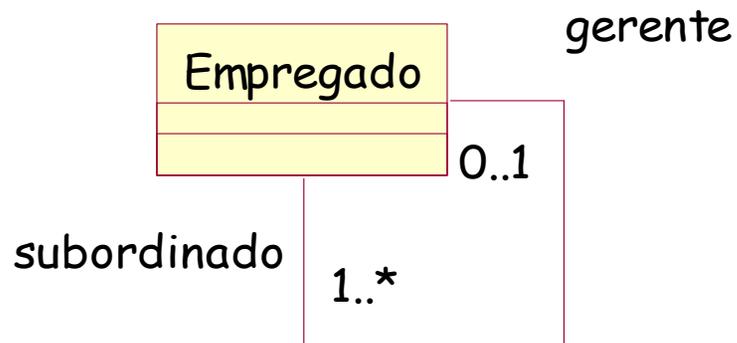
1..\* - um ou mais

0..\* - zero ou mais

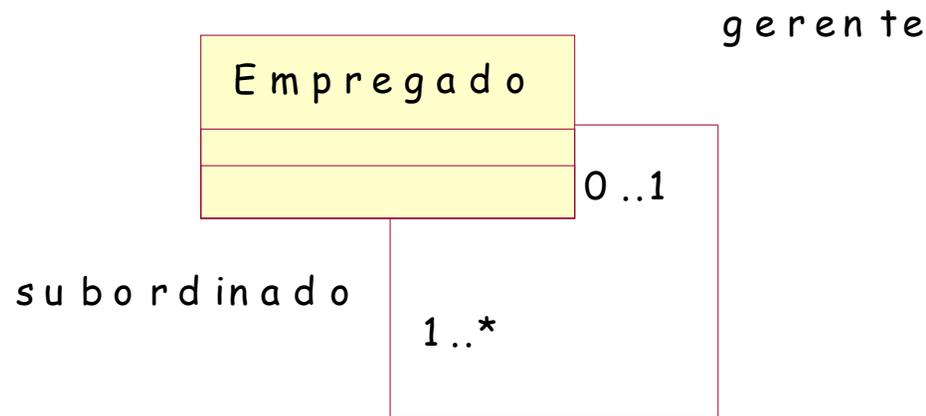
\* - zero ou mais

É possível também determinar o número exato. Por exemplo, 2..6

- Uma associação binária liga os objetos de exatamente duas classes, como a associação entre *Cliente* e *Pedido*.
- Uma associação binária, de acordo com a UML, pode também representar o relacionamento dos objetos de uma mesma classe como no exemplo a seguir no qual existe um relacionamento de gerência. Neste caso em particular um empregado pode ter nenhum ou no máximo um gerente e um gerente pode ter de um a vários subordinados.



## Papéis:

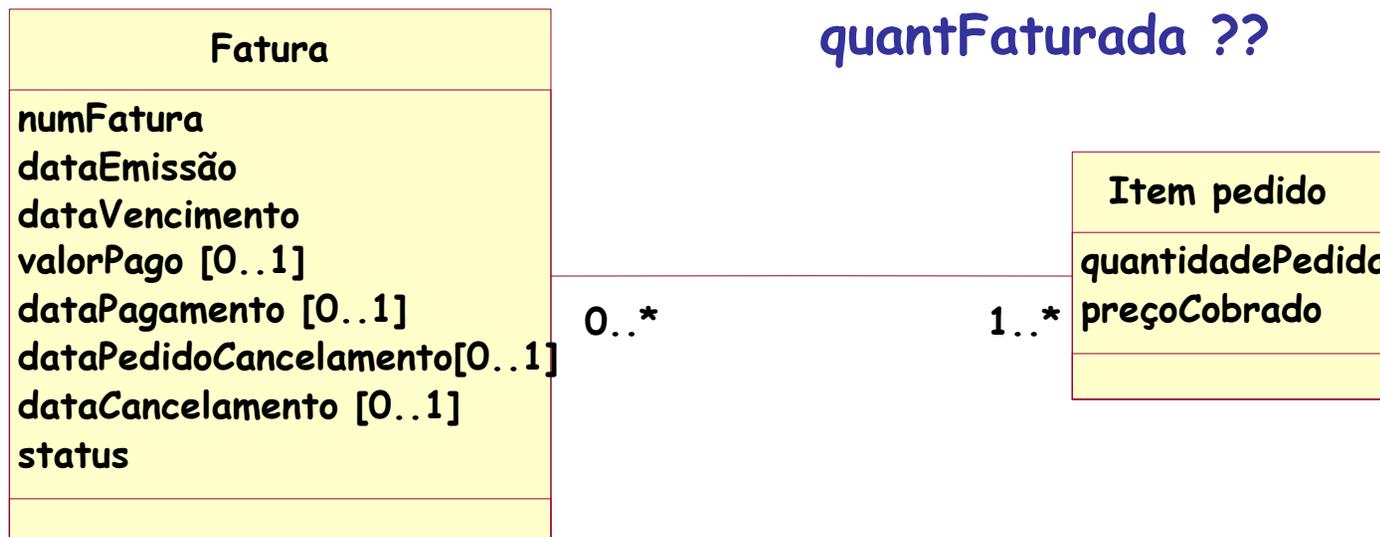


- A classe que participa de uma associação desempenha um papel neste relacionamento.
- Empregado pode desempenhar o papel de gerente ou subordinado.
- No caso em que o cliente faz pedido, não há dúvidas com relação a esses papéis, mas podem haver situações como esta acima em que seja importante destacá-los.

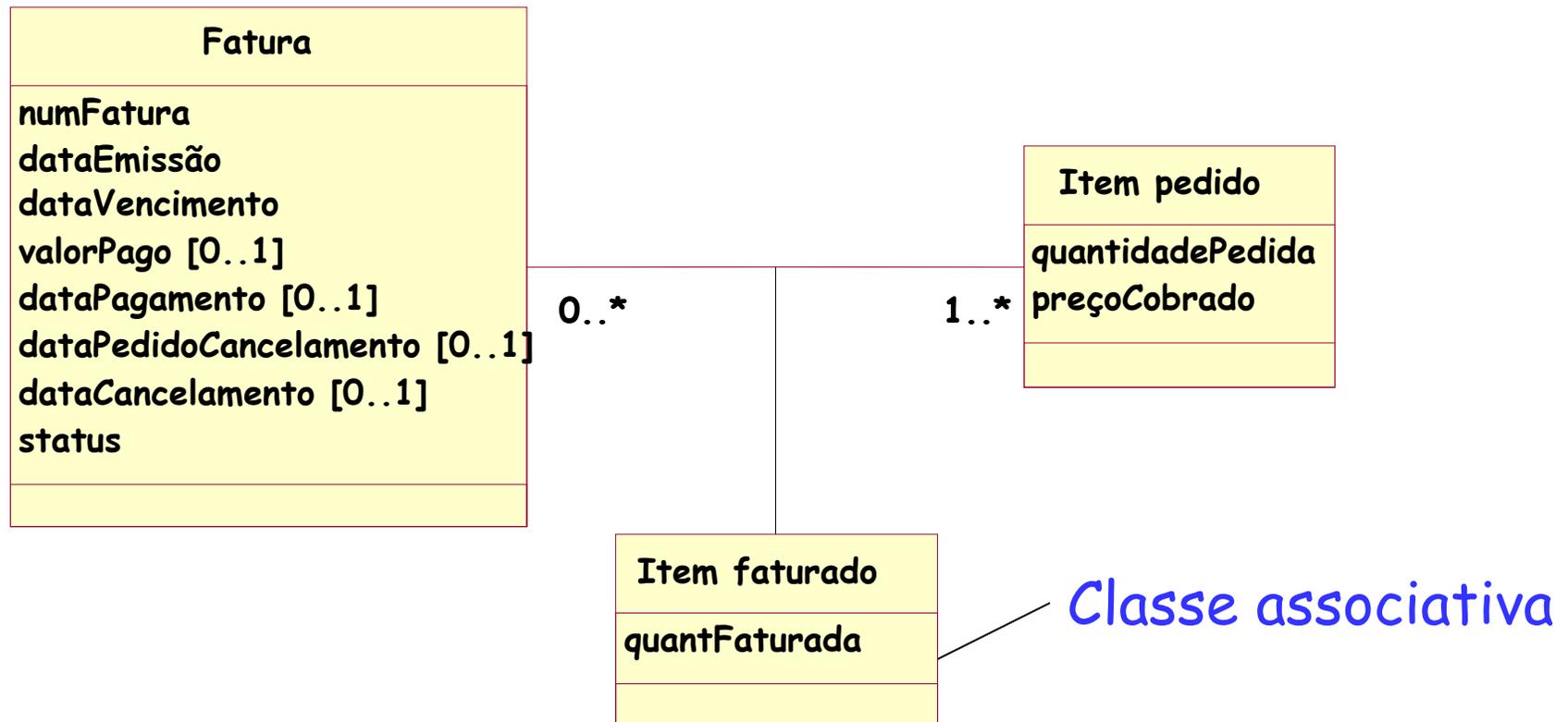
## IV. CLASSE ASSOCIATIVA

- Utilizada em situações em que uma associação tem propriedades, como por exemplo, atributos.

- Em nosso caso exemplo, quando é criada uma fatura deve ser descrita qual a quantidade faturada de cada item que será enviado ao cliente. A quantidade faturada de cada item não poderia ficar em fatura e nem em item pedido porque faz referência a uma determinada associação fatura - item pedido, ou seja é um atributo da associação.



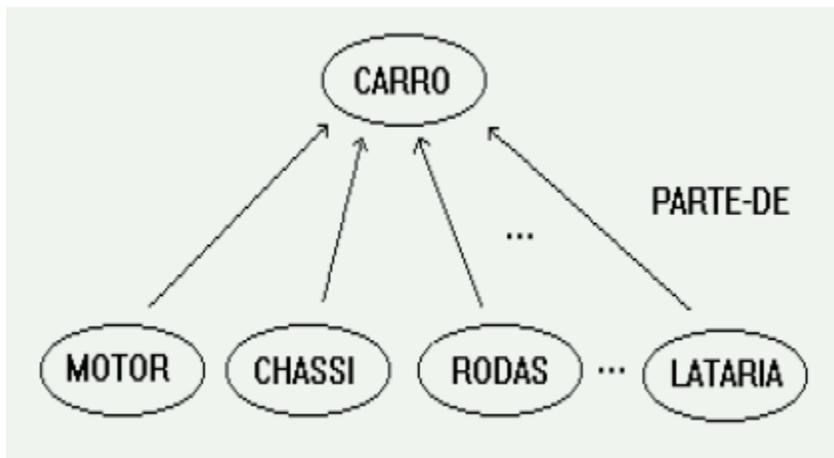
- A solução é colocar numa classe associativa o atributo `quantFaturada`, que descreve a quantidade que foi faturada.



- Uma classe associativa é uma associação que tem propriedades de classe. Assim, podem ser acrescentados a esta classe: atributos, operações e associações (com outras classes ou com ela própria)

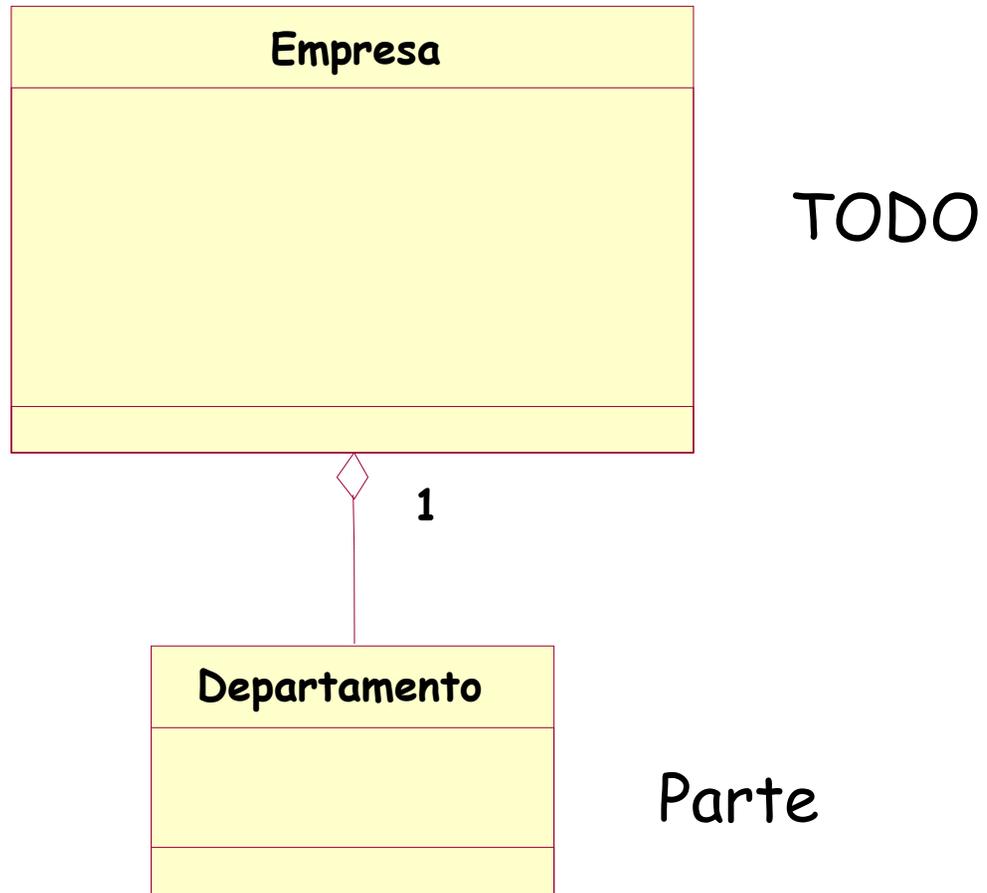
## V. AGREGAÇÃO E COMPOSIÇÃO

- Utilizamos a agregação quando temos um relacionamento entre duas classes do tipo todo/parte de tal forma que um objeto-parte pode pertencer a mais de um objeto-todo.
- A composição é uma variação da agregação onde há este relacionamento todo/parte mas cada objeto-parte pertence a um único objeto-todo e os objetos-parte desaparecem quando o objeto-todo ao qual pertencem é apagado. As partes podem ser criadas após o surgimento do todo assim como serem removidas antes que o todo seja apagado.

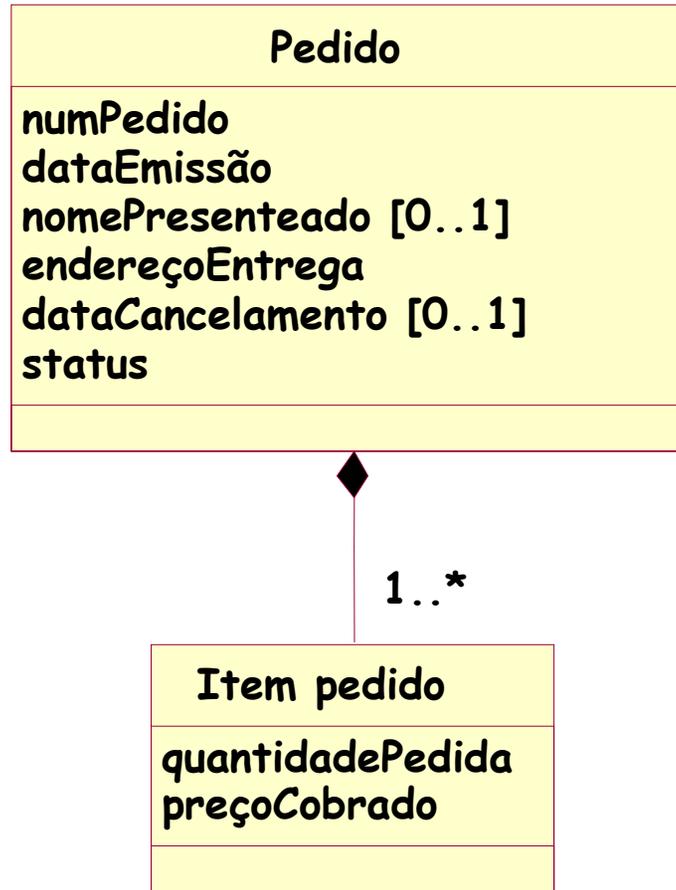


- A agregação na UML é representada por um losango e a composição por um losango cheio, ambos posicionados na extremidade referente ao todo. Na composição a multiplicidade na extremidade do todo é no máximo um enquanto na agregação pode ser maior que um.

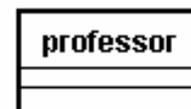
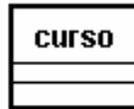
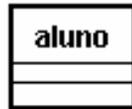
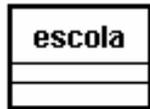
- Agregação:

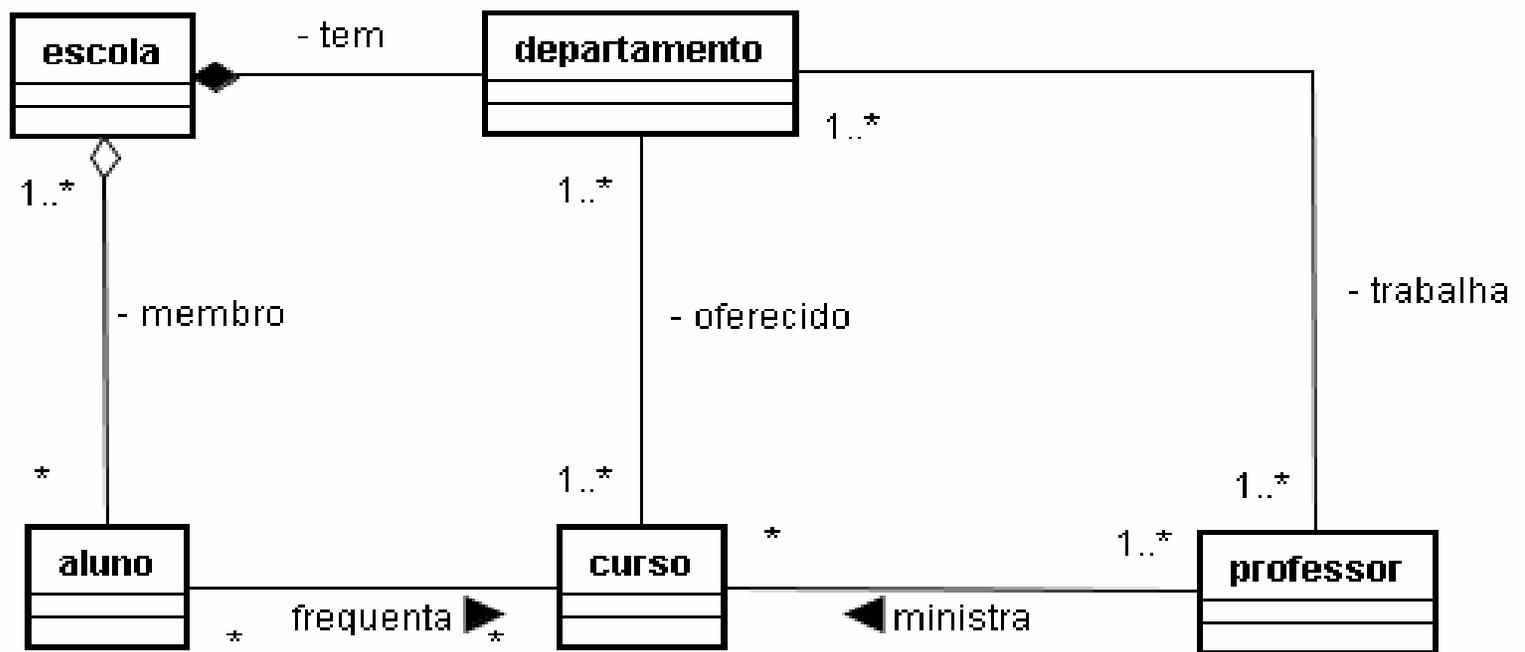


- Composição:



Podemos dizer que temos uma composição porque temos um relacionamento todo/parte e um determinado item pedido pertence a um único pedido.

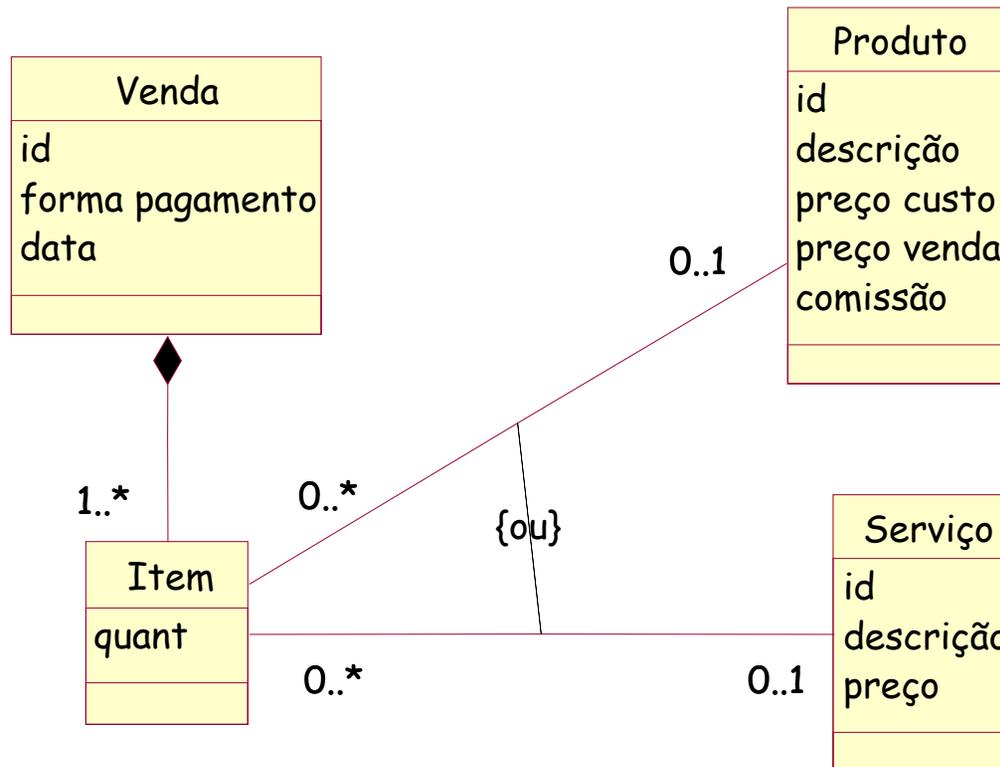




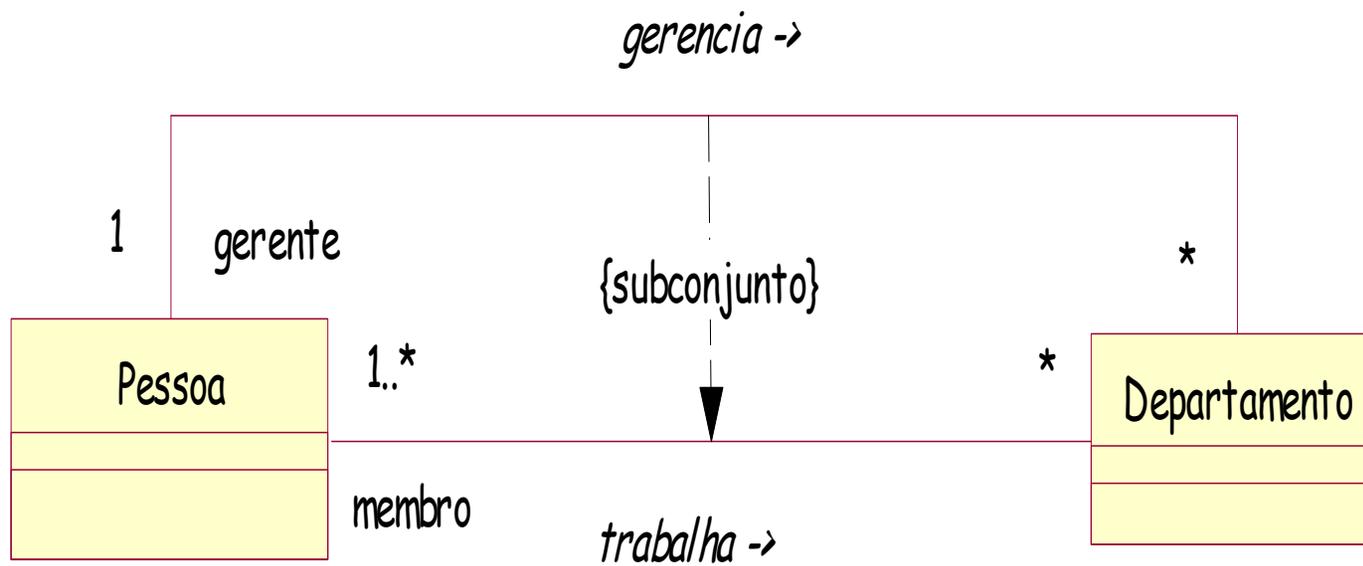
## VI. RESTRIÇÕES

- Restrição: é utilizada quando não podemos modelar uma informação importante com as notações já existentes.
- A restrição é representada através uma cadeia de caracteres entre chaves que é colocada ao lado do elemento ao qual faz referência. É portanto um mecanismo de extensibilidade da linguagem.

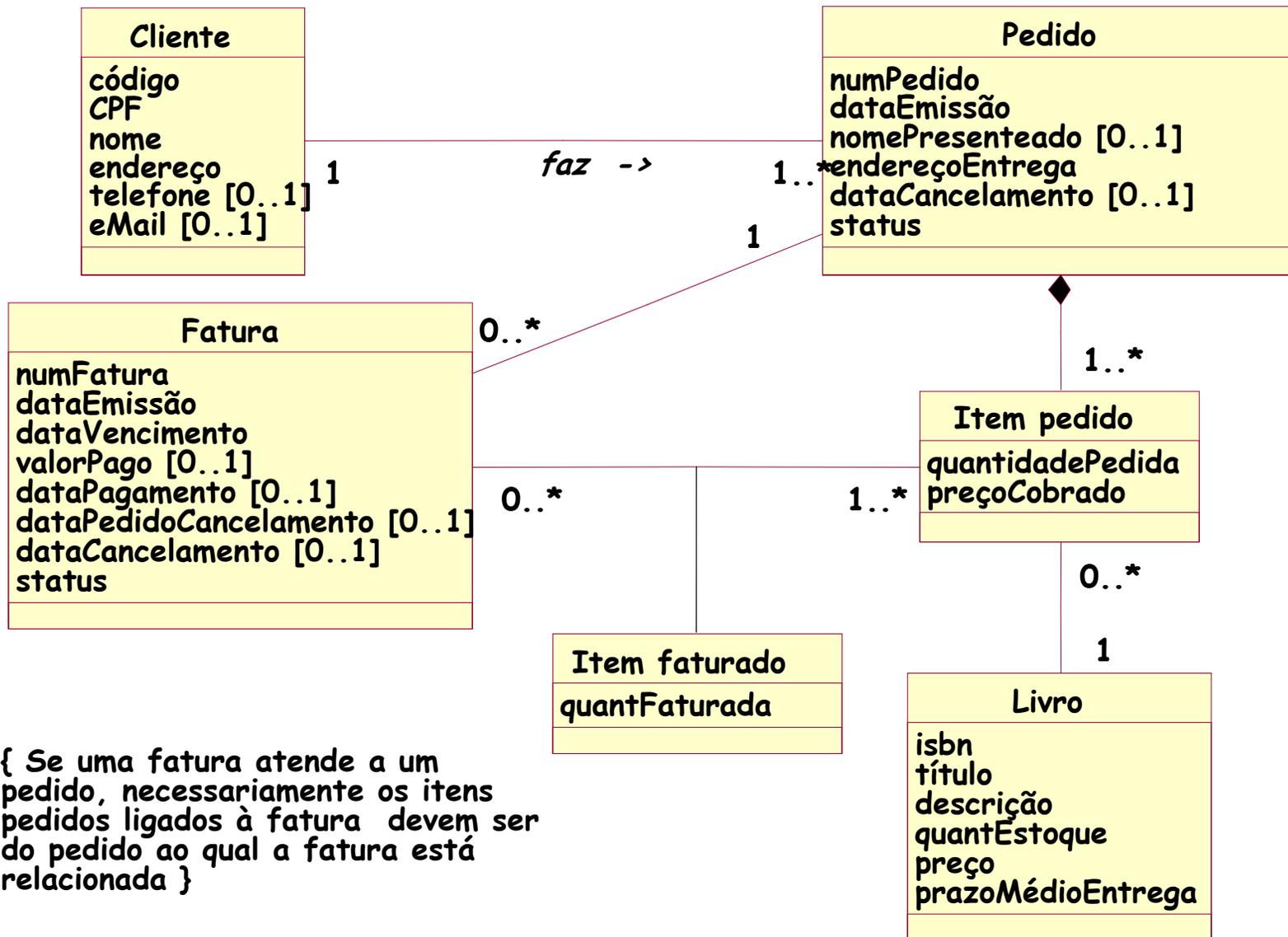
## Exemplo de restrição ou:



## Exemplo de restrição subconjunto:



## Exemplo de restrição relacionada a associações:



## VII. ELABORANDO O DIAGRAMA

1. A elaboração da primeira versão do diagrama de classe com uma perspectiva conceitual pode ser 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.

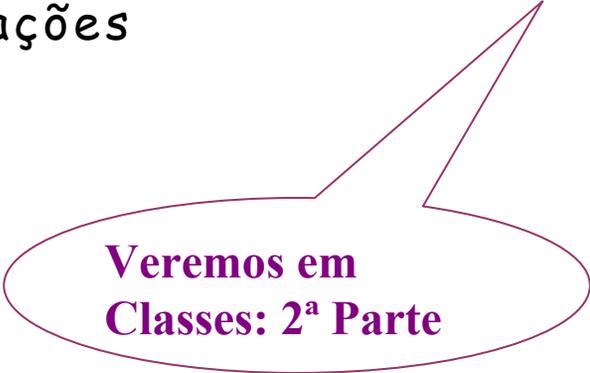
Uma estratégia é elaborar este diagrama incluindo somente as classes persistentes, ou seja, aquelas cujos objetos são preservados após serem criados ou modificados por uma aplicação. Estes objetos podem ser armazenados em um banco de dados de forma a serem recuperados mais adiante. Outras classes, não-persistentes, serão incluídas no diagrama de classes elaborado com a perspectiva de implementação.

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.



**Veremos em  
Classes: 2ª Parte**



**Veremos em  
Classes: 2ª Parte**

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

Exemplo:

Elaborando o diagrama de classes do Sistema de controle de pedidos, com uma perspectiva conceitual, a partir da descrição dos casos de uso.



## Faz pedido

*Cenário principal: Venda realizada com sucesso por um novo cliente*

1. Cliente informa dados pessoais (cpf, nome, endereço, telefone e e-mail) e endereço de entrega
2. Cliente informa a lista dos livros desejados e respectivas quantidades
3. Sistema armazena além dos dados fornecidos pelo cliente, a data de emissão do pedido e o preço cobrado por cada livro, já que pode ser dado algum desconto e o valor cobrado não será o de tabela
4. Sistema envia ao cliente a confirmação da venda com o número de seu pedido, seu código, a lista dos itens pedidos com quantidade e preço de cada item e o preço total do pedido.

## Faz pedido para presentear

1. Cliente informa dados pessoais (cpf, nome, endereço, telefone e e-mail)

Cliente informa dados do presenteado:  
nome e endereço de entrega

Continua a partir do passo 2.

## Fatura pedido

*Cenário principal: faturamento de pelo menos um item do pedido*

1. Funcionário seleciona um pedido que não tenha sido integralmente atendido (faturado)
2. Sistema verifica a quantidade pendente ( $\text{quantidadePedida} - \text{quantidadeAtendidaTotal}$ ) de cada item (*Extend - Comunica atraso*)
3. Sistema cria uma fatura com o número da fatura, a data de emissão, a data limite de pagamento e a quantidade de cada item.
4. Sistema emite a fatura que deverá ser encaminhada ao cliente juntamente com os livros. A fatura deverá conter:
  - Número da fatura
  - Número do pedido
  - Data de emissão
  - Data de vencimento
  - Para cada item: a quantidade e o preço unitário cobrado
  - Preço total

*Obs:*

- *A quantidade faturada de cada item (livro) está limitada ao que há em estoque. Caso não possa ser feito um atendimento completo neste momento, mais adiante, logo que haja o item em estoque, será criada uma nova fatura.*
- *Uma fatura faz referência a um e apenas um pedido. No entanto ela pode estar atendendo apenas parcialmente àquele pedido.*

## Comunica atraso

- Sistema verifica que um ou mais itens pedidos não poderão ser entregues e que não há previsão de entrega.
- Sistema comunica ao cliente o atraso descrevendo o número do pedido e os itens para os quais não há previsão de entrega.

## Diminui quantidade de um item do pedido

*Cenário principal: Quantidade diminuída*

1. Cliente informa seu código
2. Sistema valida código
3. Cliente informa número do pedido
4. Sistema valida número do pedido (include - Valida pedido)
5. Cliente informa o item a ser reduzido
6. Sistema apresenta ao cliente a quantidade pendente (quantidade pedida - quantidade faturada)
7. Cliente informa a nova quantidade (no máximo a quantidade pendente)
8. Sistema armazena a nova quantidade
9. Sistema envia ao cliente a confirmação da alteração realizada

## Diminui quantidade de um item do pedido

*Cenário alternativo: Pedido de redução de item é recusado por já ter sido totalmente faturado*

### 6. Sistema comunica ao cliente:

- o item não pode ser reduzido por ter sido completamente faturado
- caso deseje realmente esta redução o cliente deverá solicitar o cancelamento parcial ou total da fatura. Esse pedido será avaliado pelo gerente.

Os passos seguintes não são realizados

## Solicita cancelamento de pedido

*Cenário principal: Pedido cancelado por não haver fatura emitida*

1. Cliente informa seu código
2. Sistema valida código
3. Cliente informa número do pedido
4. Sistema valida número do pedido (*include - Valida pedido*)
5. Sistema cancela o pedido (não há nenhuma fatura emitida para ele) e armazena a data de cancelamento
6. Sistema envia ao cliente a confirmação do cancelamento solicitado.

## Solicita cancelamento de pedido

*Cenário alternativo: Pedido não pode ser cancelado por já ter sido emitida pelo menos uma fatura*

### 5. Sistema comunica ao cliente:

- o pedido não pode ser cancelado por já ter sido emitida pelo menos uma fatura.
- o cliente deverá solicitar o cancelamento das faturas já emitidas caso deseje cancelar o pedido.

Os passos seguintes não são realizados

## Valida pedido

1. Sistema verifica a existência do número do pedido
2. Sistema envia ao cliente os dados relativos a seu pedido:
  - a lista dos itens pedidos com quantidade e preço unitário de cada item e o preço total do pedido
  - a lista das faturas emitidas contendo para cada fatura:
    - Número da fatura
    - Número do pedido
    - Data de emissão
    - Data de vencimento
    - Para cada item: a quantidade e o preço unitário
    - Preço total

## Solicita cancelamento de fatura

*Cenário principal: Solicitação de cancelamento integral da fatura realizada com sucesso*

1. Cliente informa seu código
2. Sistema valida código
3. Cliente informa número da fatura
4. Sistema verifica a existência deste número
5. Sistema envia ao cliente os dados da fatura, contendo: a data de emissão, o preço total e para cada item: a quantidade e o preço unitário
6. Cliente solicita o cancelamento integral da fatura
7. Sistema armazena a solicitação de cancelamento da fatura e a data da solicitação
8. Sistema envia ao cliente a confirmação do cadastramento de sua solicitação e a informação de que o seu pedido só será analisado quando a Empresa receber os livros relativos à fatura.

## Paga fatura

*Cenário principal: Pagamento da última fatura de um pedido*

1. Cliente paga a fatura
2. Sistema armazena o número da fatura, o valor pago e a data de pagamento.
3. Sistema fecha o pedido relacionado à fatura (foi paga a última fatura de um pedido)
4. Sistema envia ao cliente a confirmação de pagamento da fatura

## Avalia cancelamento de fatura

*Cenário principal: Gerente cancela a fatura e automaticamente é cancelado o pedido*

1. Gerente analisa o pedido de solicitação de cancelamento de uma fatura
2. Gerente cancela a fatura
3. Sistema armazena a data de cancelamento e atualiza o estoque, considerando a devolução dos livros
4. Sistema cancela o pedido (todas as faturas de um pedido foram canceladas)
5. Sistema envia ao cliente notificação do cancelamento da fatura e do pedido

## Exercicio 2: Modele os Relacionamentos

Usar classes e associações para definir o glossário do sistema "Jogo de Futebol" descrito de seguida: "O jogo de futebol é realizado por duas equipes de jogadores. Cada equipe é composta por 11 jogadores, com diferentes funções: o goleiro, zagueiros, médios, atacantes, e pontas de lança. O ponta de lança é um atacante especial por ter especiais características de goleador... O jogo é realizado num campo com medidas regulamentares (em comprimento e largura), tem duas balizas, cada qual em extremos opostos do campo. Ganha o jogo a equipe que marcar mais gols (i.e., colocar a bola) na baliza do adversário. No jogo apenas existe uma única bola, que apresenta características (peso, diâmetro, ...) regulamentares... O jogo de futebol é mediado por uma equipe de 3 árbitros, em que um é o árbitro principal, e os outros dois são árbitros auxiliares..."

## Exercicio 3: Modele os Relacionamento

Sua empresa foi solicitada para o desenvolvimento de um software para controle de um hospital. Este Hospital possui 5 centros: CTI, Emergência, Maternidade, Pediatria e Clinica Geral. Cada um destes centros possui uma série de características, tais como número de leitos, número de médicos, capacidade de espera, horários de atendimento, etc. Um paciente (que também deve possuir uma série de atributos), ao chegar, deve ser encaminhado para um destes centros, mas antes deve passar pela recepção do hospital para que sua ficha seja preenchida no sistema e para que o sistema, além de registrá-lo, realize o encaminhamento para o centro e respectivo médico responsável. Devem haver prioridades de atendimento ao paciente, sendo um dos atributos a gravidade do caso (esta gravidade deve variar de 1 a 5). Caso seu problema não se adeque, o hospital possui convênio com outros hospitais, aos quais encaminhará estes pacientes. Este encaminhamento será efetuado por uma das ambulâncias do primeiro hospital. Os médicos do hospital possuem diversas características, tais como nome, especialidade, CRM, turno. Enfermeiros são outros elementos distintos e também devem possuir uma serie de características.