DCC056 – Modelagem de Sistemas

Prática de Modelagem de Sistemas pela UML

(Slides cedidos pelo Prof. Tarcísio de Souza Lima)

UNIVERSIDADE FEDERAL DE JUIZ DE FORA

Objetivos do Curso

- Introduzir os princípios e conceitos necessários para se iniciar a utilização efetiva da ***Unified Modeling Language*** (UML)[1], norma do *Object Management Group* (OMG)[2] para a construção de modelos de sistemas de software.
- Abordar linhas de orientação metodológica e ferramentas que potenciam a utilização efetiva da UML em projetos de software.
- Consolidar os conhecimentos transmitidos, pela prática através de exercícios e estudos de caso.

1. http://www.uml.org/
2. http://www.omg.org/

Metodologia

- Sessões de exposição de conhecimentos, alternadas, na medida do possível, com sessões laboratoriais para experimentação.
- Desenvolvimento ao longo do curso de um projeto em UML.

Resultados Esperados

- Compreender e saber distinguir as diferentes situações/necessidades de modelagem;
- Conhecer os principais diagramas da UML e sua notação;
- Entender a aplicação prática de cada um desses diagramas;
- Modelar um sistema fazendo uso da UML;
- Propiciar o uso efetivo de ferramentas de software de suporte à UML.

Bibliografia

Especificações:

•Unified Modeling Language: Superstructure, OMG (Objetct Management Group), 2007;

•Unified Modeling Language: Superstructure, OMG, 2007.

Livros (exemplos):

•Unified Modeling Language User Guide, The, Second Edition, Grady Booch, James Rumbaugh, Ivar Jacobson, Addison Wesley Professional, 2005 (496 pgs.) ;

•Unified Modeling Language Reference Manual, The, Second Edition, James Rumbaugh, Ivar Jacobson, Grady Booch, Addison Wesley Professional, 2004 (752 pgs.);

•Learning UML 2.0, Russell Miles, Kim Hamilton, O'Reilly, 2006 (286 pgs.);

•The Elements of UML 2.0 Style, Scott W. Ambler, Cambridge University Press, 2005 (200 pgs);

Visão geral da linguagem UML e de seus diagramas

O que é a UML?

- UML = *Unified Modeling Language* (Linguagem de modelagem unificada)
- UML é uma **linguagem** (notação com semântica associada) para

  especificar / descrever /representar

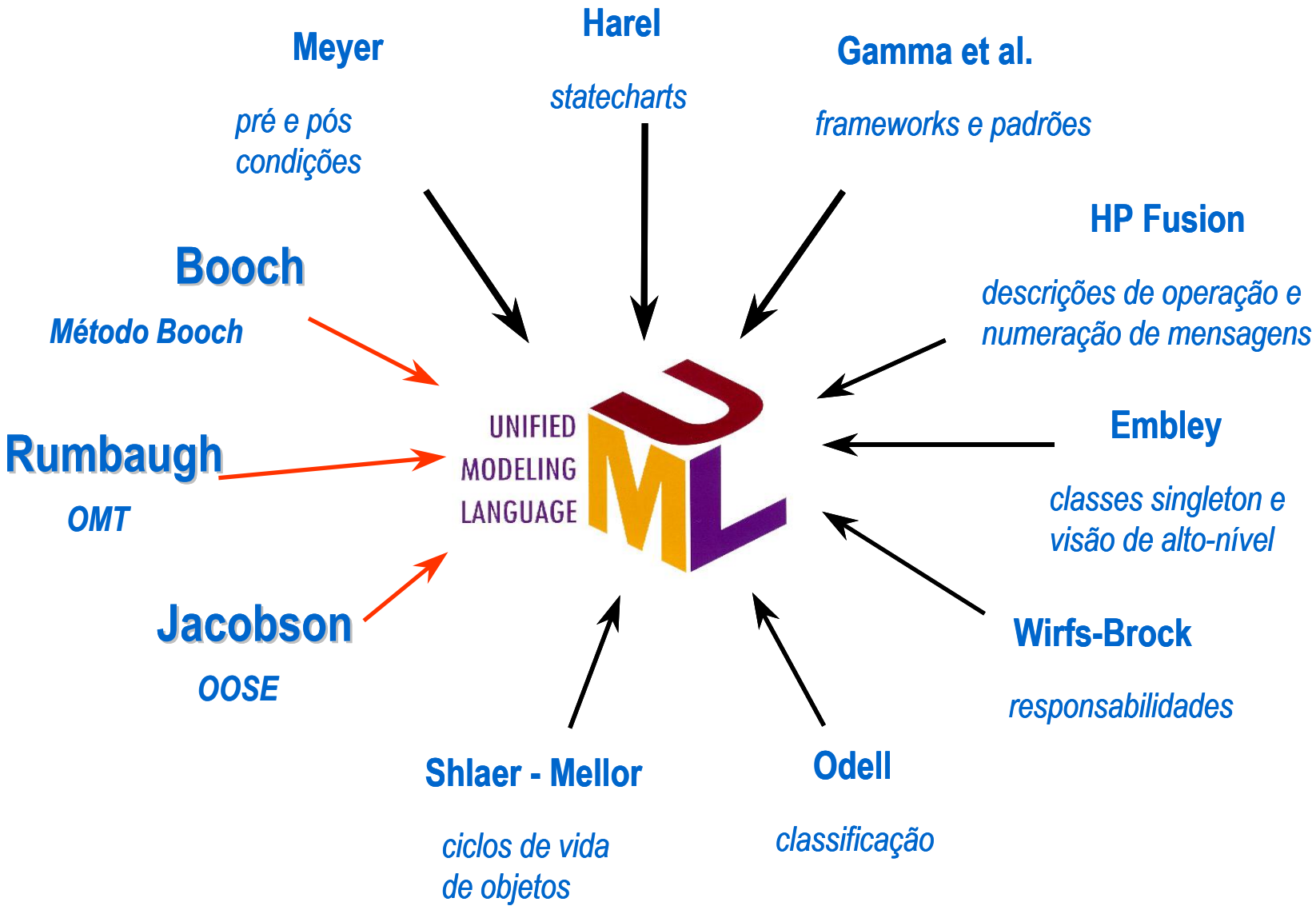
  os artefatos de um sistema com uma componente intensiva de software (*software intensive system*)
- UML não é uma metodologia
  - não diz quem deve fazer o quê, quando e como
  - UML pode ser usado segundo diferentes metodologias, tais como RUP (*Rational Unified Process*), FDD (*Feature-Driven Development*), etc.
- UML não é uma linguagem de programação

Valor da UML

- É um padrão aberto
  - Versões sucessivas vêm sendo aprovadas pelo OMG (*Object Management Group)*[1] desde Novembro de 1997
  - Versão oficial corrente: **UML 2.3** (2010)
- Suporta todo o ciclo de vida do software
  - modelagem do negócio (processos e objetos de negócio)
  - modelagem de requisitos de software
  - modelagem da solução de software
- Suporta diversas áreas de aplicação
  - Por exemplo: negócio, engenharia, medicina, ...
- É baseado na experiência e necessidades da comunidade de usuários (indústria, universidade, governo, especialistas, etc.)
- É suportado por muitas ferramentas computacionais

1 http://www.omg.org/

Origens da UML



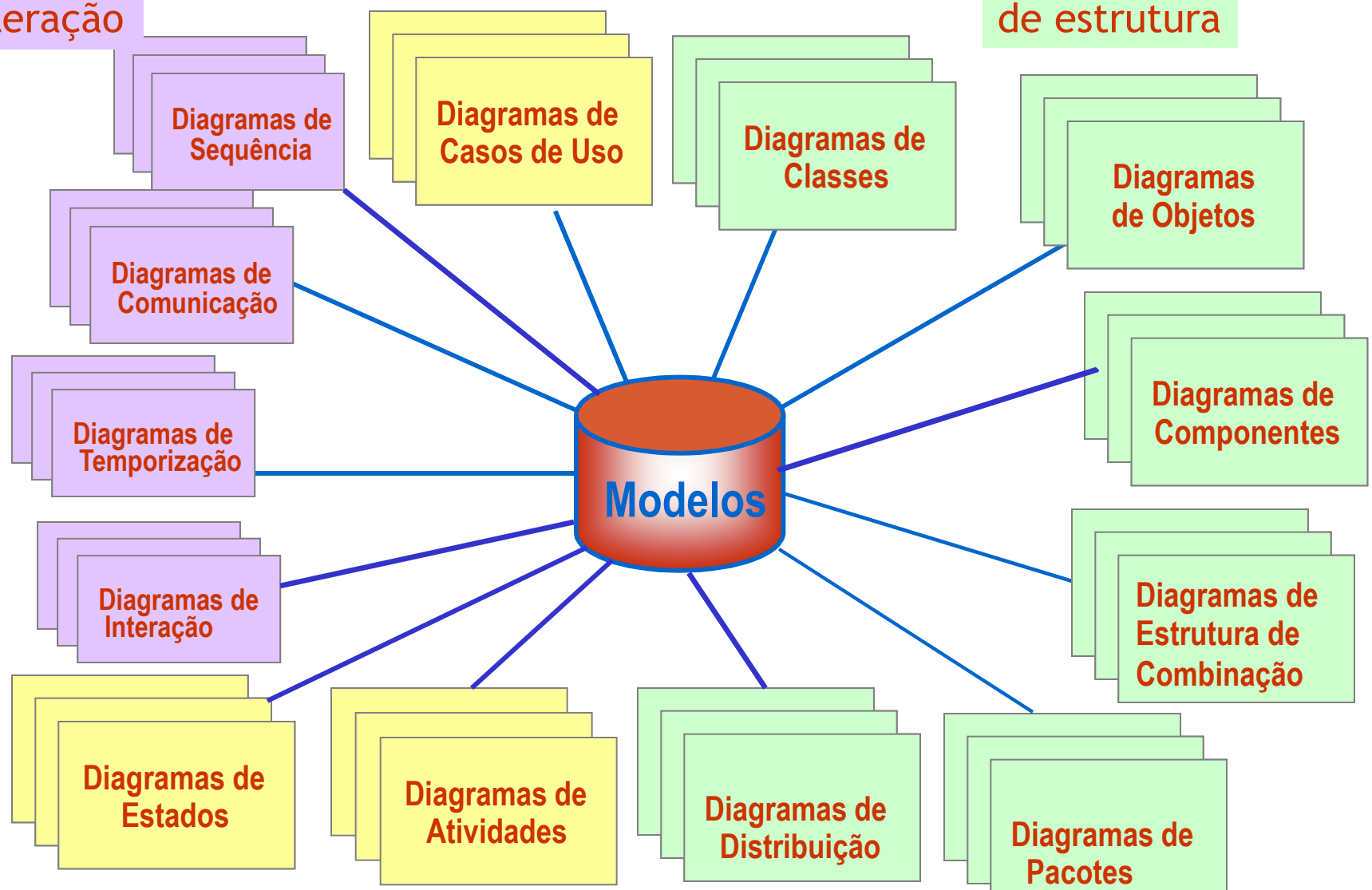
Modelos

- Um modelo é uma representação em **escala reduzida**, numa perspectiva particular, de um sistema existente ou a criar
  - Atitude de **abstração** (omissão de detalhes) é fundamental na construção de um modelo;
  - Modelos são a linguagem por excelência do projetista (*designer*)
  - Modelos são veículos para comunicação com vários interessados (*stakeholders*);
  - Modelos permitem raciocinar acerca do sistema real, sem chegar a construí-lo.
- Ao longo do ciclo de vida de um sistema são construídos vários modelos, sucessivamente refinados e enriquecidos.

Diagramas

- Um modelo pode ser constituído por um ou mais **diagramas** (desenhos) consistentes entre si, possivelmente acompanhados de descrições textuais dos elementos que aparecem no(s) diagrama(s).
  - Um **diagrama** é uma visão sobre o objeto modelado;
  - O mesmo **elemento** (exemplo: uma classe) pode aparecer em vários diagramas de um modelo.
- Na UML 2, há treze diagramas-padrão, divididos em 3 categorias:
  - Diagramas **de estrutura**: de classes, de objetos, de componentes, de estrutura de combinação (*composite structure*), de pacotes, e de distribuição (*deployment*).
  - Diagramas **de comportamento**: de casos de uso *(use cases),* de atividades e de estados *(statechart).*
  - Diagramas **de interação**: de seqüência, de comunicação, de temporização, e de interação.

Modelos e Principais Diagramas



Agora, um pouco sobre os principais diagramas da UML

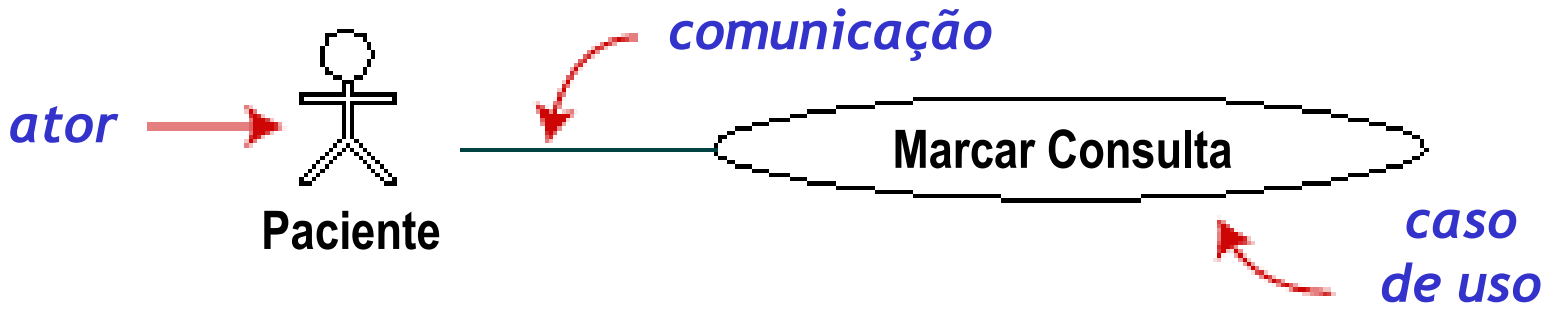
Diagramas de Casos de Uso

- Descrevem o que um sistema faz do ponto de vista de um observador externo. A ênfase está no *quê* um sistema faz ao invés do *como* ele faz.
- São fortemente conectados a cenários. Um cenário é um exemplo do que acontece quando alguém interage com o sistema.
- Exemplo de cenário para uma clínica médica:

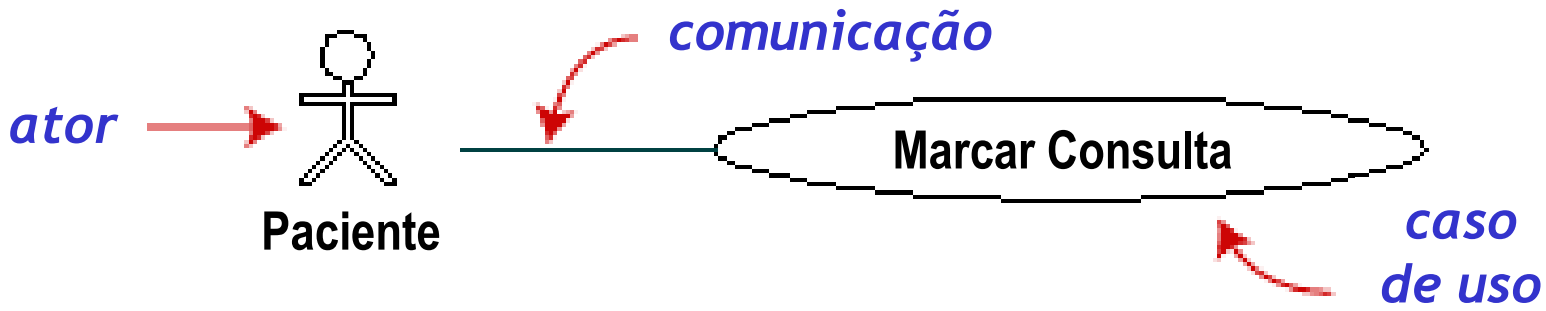
Um paciente liga para sua clínica médica para marcar uma consulta anual de rotina para "checkup". A secretária da clínica encontra o horário livre mais próximo na agenda do médico indicado pelo paciente e marca a consulta para aquele horário.

Diagramas de Casos de Uso

- Um caso de uso é um resumo dos cenários para uma tarefa ou objetivo único. Um ator é a pessoa ou aquilo que inicia os eventos envolvidos naquela tarefa/objetivo. Atores são simplesmente papéis que as pessoas ou os objetos executam.
- Abaixo o caso de uso Marcar Consulta para a clínica médica.
- O ator é o Paciente. A conexão entre ator e caso de uso é a associação de comunicação (ou comunicação, por simplicidade).



Diagramas de Casos de Uso

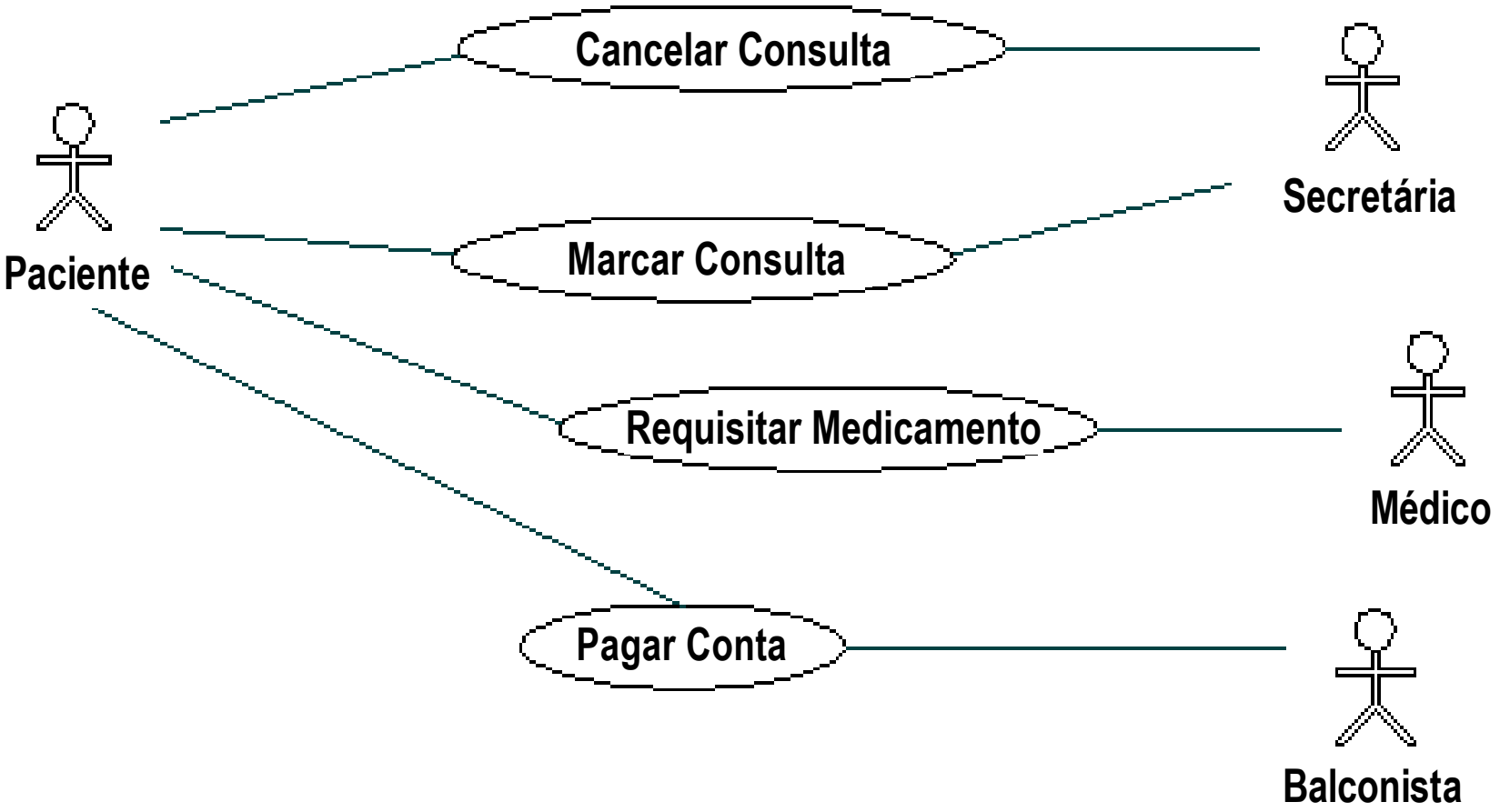


- **Atores** são **figuras estilizadas**, **casos de uso** são **ovais** e **comunicações** são **linhas** que ligam atores a casos de uso. Um **diagrama de caso de uso** é uma **coleção de atores, casos de uso e suas comunicações**.

- Marcar consulta será colocado como parte de um diagrama com quatro atores e quatro casos de uso. Perceba que um único caso de uso pode ter múltiplos atores.

Diagramas de Casos de Uso

Exemplo:



Diagramas de Casos de Uso

Os **diagramas de caso de uso** são úteis em três áreas:

- determinação de características (requisitos). Novos casos de uso geralmente geram novos requisitos enquanto o sistema é analisado e o projeto toma forma.
- comunicação entre clientes. A simplicidade de notação torna os diagramas de caso de uso uma boa forma para os desenvolvedores se comunicarem com os clientes.
- geração de casos de teste. Uma coleção de cenários para um caso de uso pode sugerir um conjunto de casos de teste para aqueles cenários.

Diagramas de Casos de Uso

RESPONDA:

Três itens de interesse nos diagramas de caso de uso são:

A. Objetos, atividades e comunicações

B. Atores, mensagens e atividades

C. Objetos, casos de uso e atividades

D. Atores, casos de uso e comunicações

Diagramas de Casos de Uso

RESPONDA:

Três itens de interesse nos diagramas de caso de uso são:

A. Objetos, atividades e comunicações

B. Atores, [illegible]

C. Objetos, casos

D. Atores, casos de uso e comunicações

✗ **Comunicações** são parte do diagrama de casos de uso, mas **objetos** e **atividades** não são.

Diagramas de Casos de Uso

RESPONDA:

Três itens de interesse nos diagramas de caso de uso são:

A. Objetos, atividades e comunicações

B. Atores, mensagens e atividades

C. Objetos, casos [illegible] e atividades

D. Atores, ca[illegible]

✗ **Atores** são as figuras estilizadas do diagrama de caso de uso. **Mensagens** não aparecem nesses diagramas e a palavra "**atividade**" é usada nos diagramas de atividade, mas não aqui.

Diagramas de Casos de Uso

RESPONDA:

Três itens de interesse nos diagramas de caso de uso são:

A. Objetos,

B. Atore

C. Objetos, casos de uso e atividades

D. Atores, casos de uso e comunicações

✗ **Objetos** não são parte dos diagramas de caso de uso (**atores** são). **Casos de uso** são, claro, mas **atividades** não são.

Diagramas de Casos de Uso

RESPONDA:

Três itens de interesse nos diagramas de ca

A.

B.

C. Objeto[illegible] de uso e atividades

D. Atores, casos de uso e comunicações

✓ **Correto! Atores** são as figuras estilizadas, **casos de uso** são as ovais rotuladas e **comunicações** são as associações que conectam atores e casos de uso. **Atores** representam as coisas do ambiente real que iniciam os eventos que configuram os casos de uso.

Diagramas de Classes

- Fornecem um “resumo” (uma visão) de um sistema, mostrando os **tipos de objetos** que o compõem e seus **relacionamentos**.

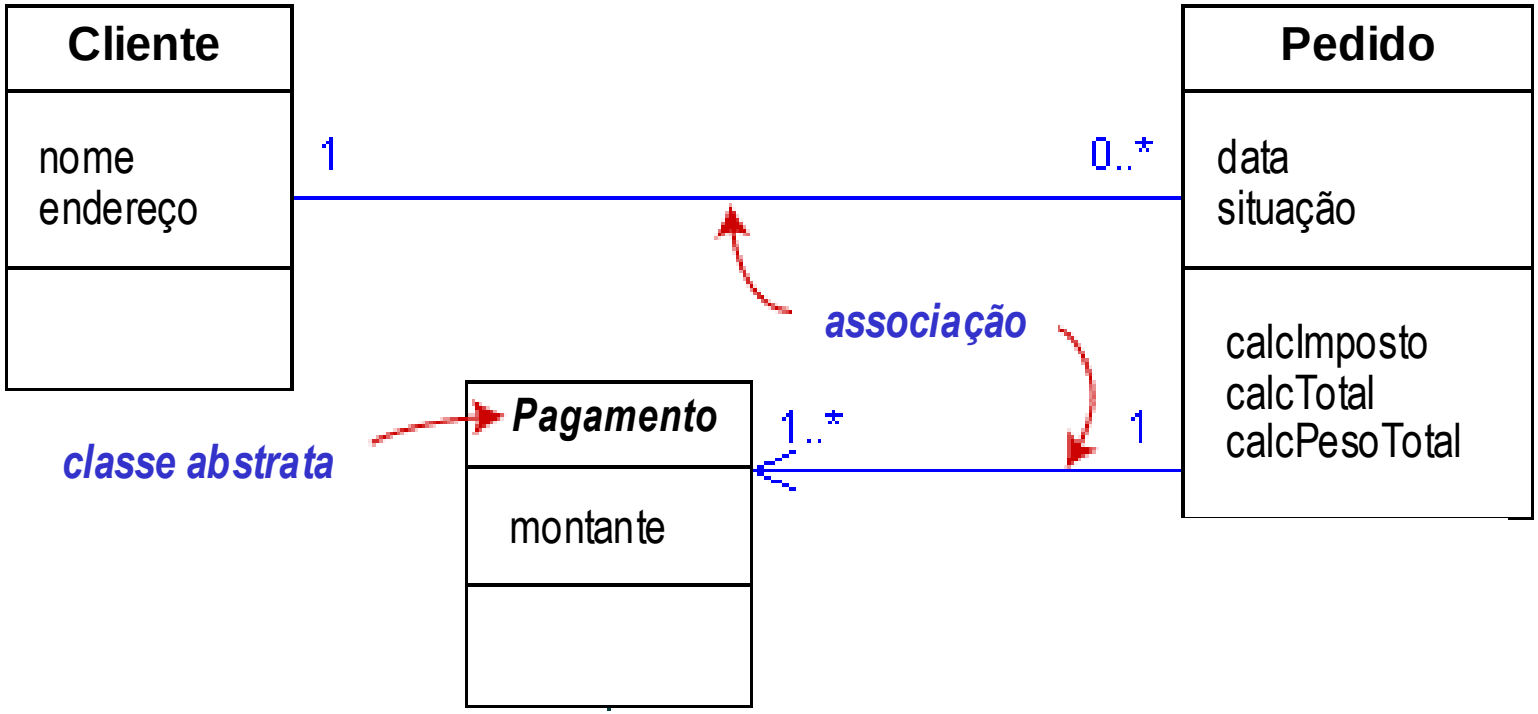
- Os **diagramas de classes** são estáticos – isto é, eles **mostram os objetos que podem interagir**, mas **não como eles interagem** ou o que acontece na sua interação.

Diagramas de Classes

- Exemplo de diagrama de classes para um pedido de um cliente a partir de um catálogo de produtos no varejo:

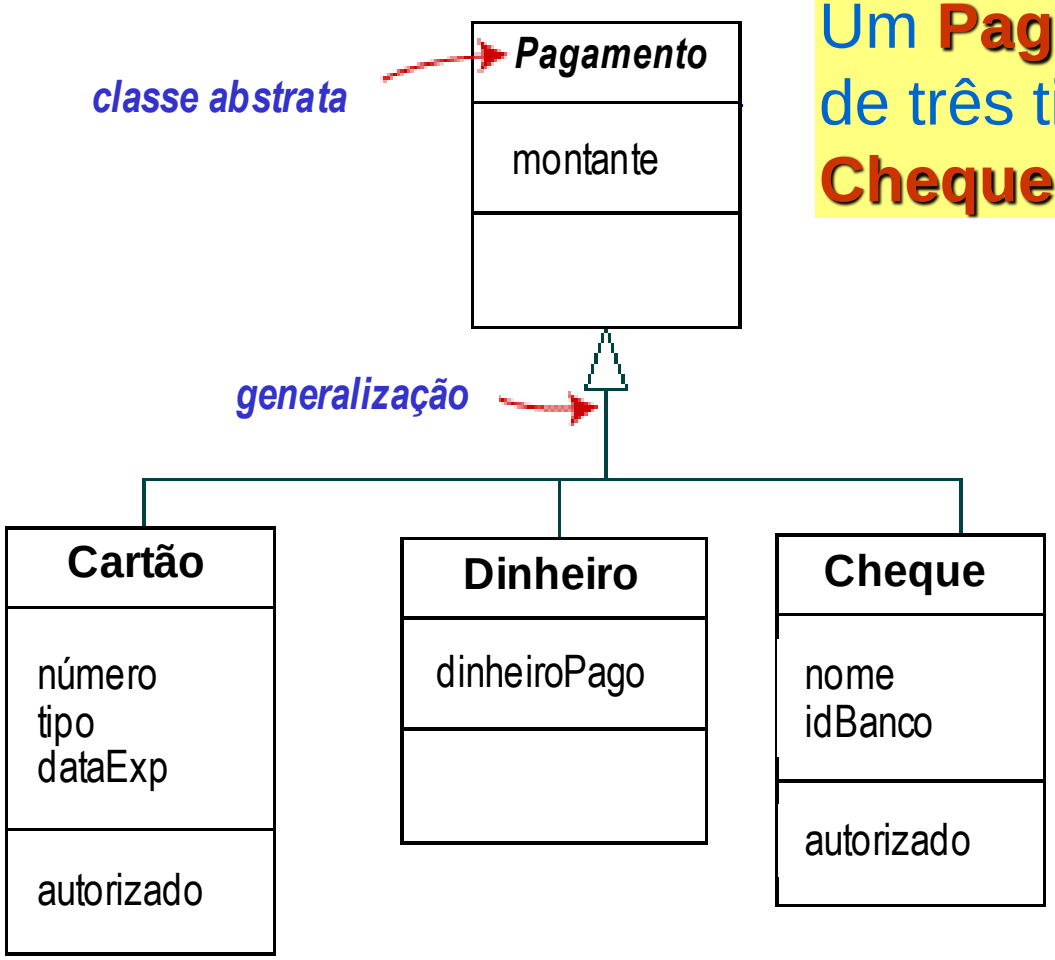
A classe central será o **Pedido**. Associado a ela estão o **Cliente** fazendo a compra e o **Pagamento**. Um **Pagamento** **é-um** de três tipos: **Dinheiro**, **Cheque** ou **Cartão**. O pedido contém **Detalhes-Pedido** (linhas de itens), cada qual com seu **Item** associado.

Diagramas de Classes



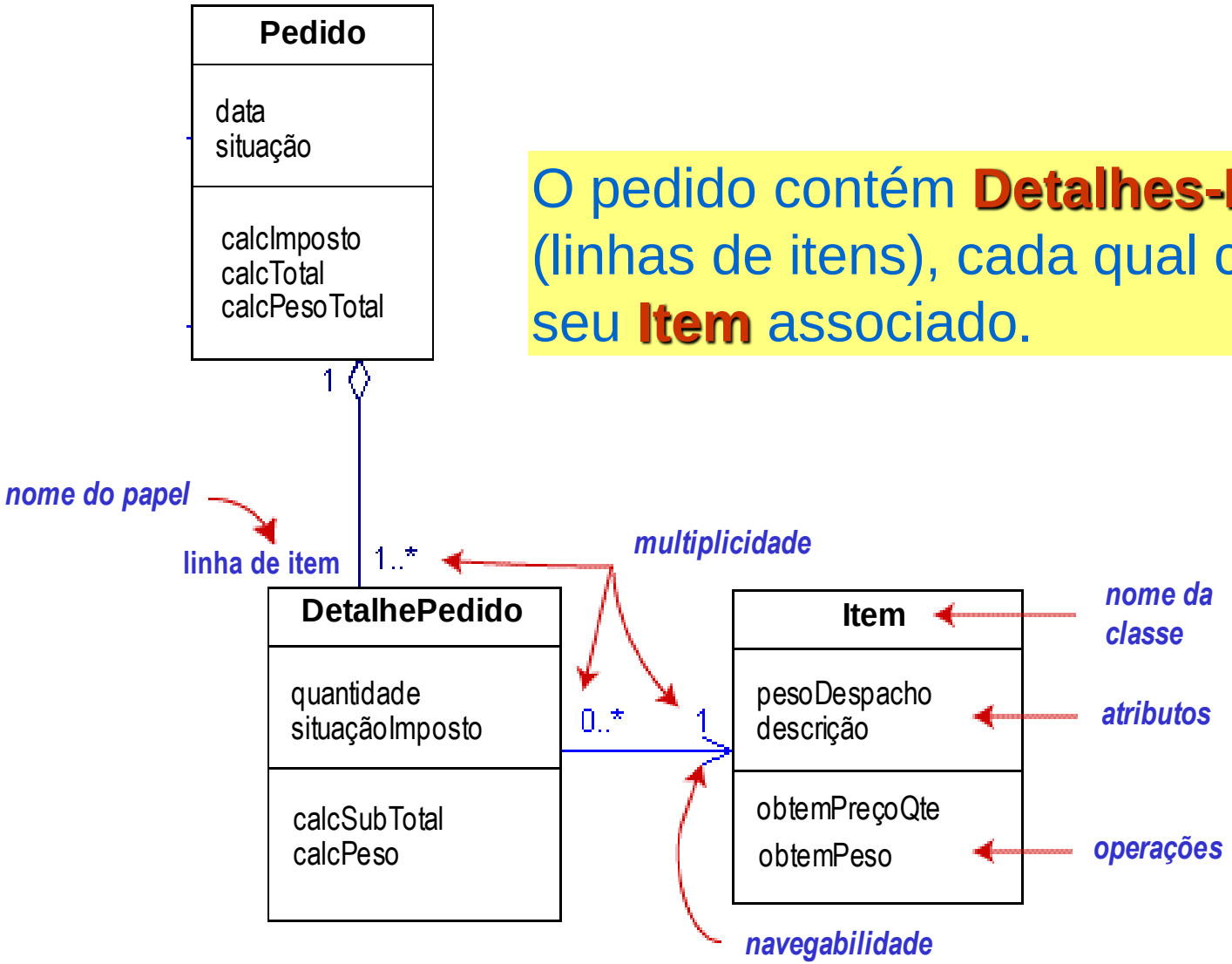
A classe central será o **Pedido**. Associado a ela estão o **Cliente** fazendo a compra e o **Pagamento**.

Diagramas de Classes



Um **Pagamento** **é-um** de três tipos: **Dinheiro**, **Cheque** ou **Cartão**.

Diagramas de Classes



O pedido contém **Detalhes-Pedido** (linhas de itens), cada qual com seu **Item** associado.

Diagramas de Classes

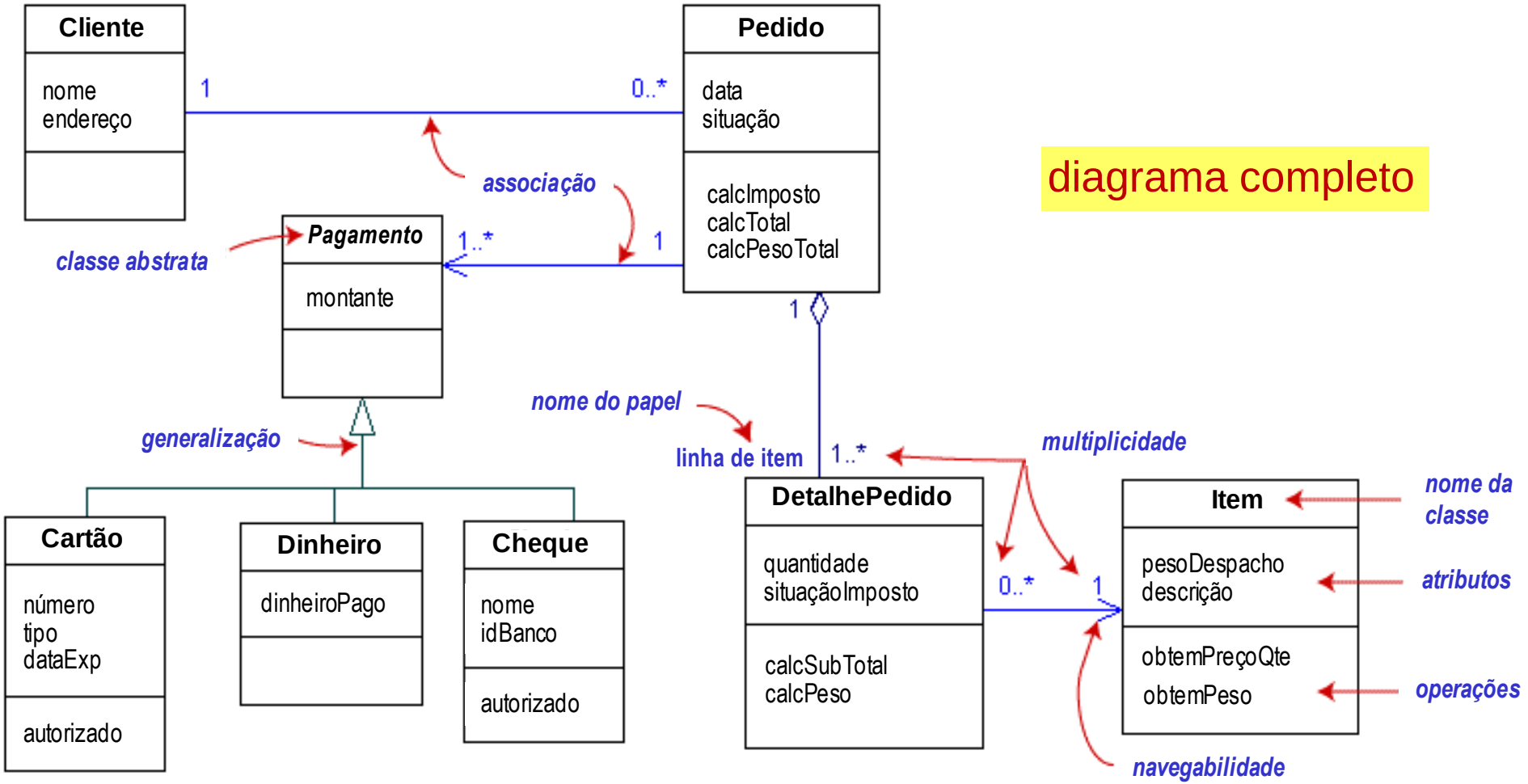


diagrama completo

Diagramas de Classes

- A notação de uma **classe** UML é um retângulo dividido em três partes:

  nome da classe, atributos e operações.

- Nomes de **classes abstratas**, tais como ***Pagamento***, são *em itálico*.

- Relacionamentos entre classes são linhas de conexão (*links*)

Diagramas de Classes

Os diagramas de classes têm três tipos de relacionamentos:

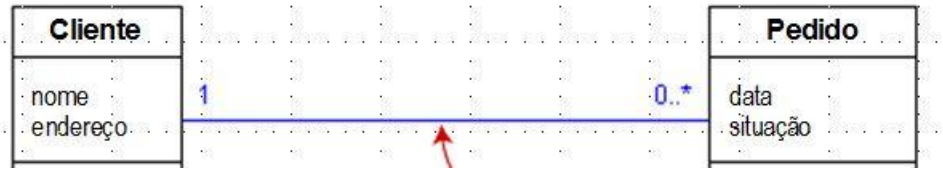
- **associação** – existe uma associação entre duas classes se uma instância de uma classe tiver que conhecer a outra para realizar o seu trabalho. No diagrama, uma associação é uma linha conectando as duas classes.
- **agregação** – uma associação na qual uma classe pertence a uma coleção. Tem um losango apontando para a classe que contém o todo (**Pedido** tem uma coleção de **DetalhePedido**).
- **generalização** – uma ligação de herança indicando que uma classe **é-uma** superclasse de outra. Tem um triângulo apontando para a superclasse (**Pagamento** é superclasse de **Dinheiro**, **Cheque** e **Cartão**).

Diagramas de Classes

- Uma associação tem duas extremidades. Uma extremidade pode ter um **nome do papel** para esclarecer a natureza da associação (um **DetalhePedido** é uma **linha de item** de cada **Pedido**).
- Uma seta de **navegabilidade** em uma associação mostra em que direção esta associação pode ser atravessada ou consultada (um **DetalhePedido** pode ser consultado sobre seus **Item**, mas não no sentido contrário!). A seta também permite conhecer quem é o "proprietário" da implementação da associação (**DetalhePedido** tem um **Item**). Associações sem setas de navegabilidade são **bi-direcionais**.

Diagramas de Classes

- A **multiplicidade** em um lado de uma associação é o número de possíveis instâncias daquela classe associadas a uma única instância da classe no outro lado. Multiplicidades são números únicos ou intervalos de números (**0..1**; **n..m**; **0..*** ou ***; **1**; **1..***). No exemplo, pode haver somente um **Cliente** para cada **Pedido**, mas um **Cliente** pode ter qualquer quantidade de **Pedido**.



- **Classes**, **associações** e **multiplicidades** são de presença OBRIGATÓRIA nos diagramas. Já a **navegabilidade** e os **papéis** são OPCIONAIS (servem para aumentar a clareza do diagrama).

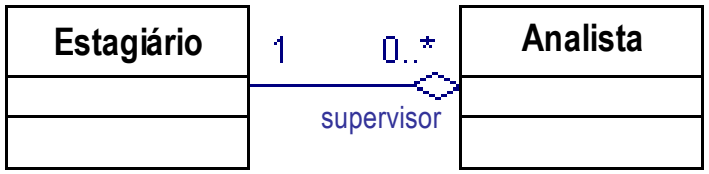
Diagramas de Classes

DC - Exerc

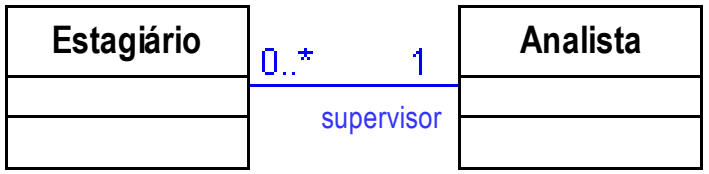
RESPONDA:

Cada estagiário do Departamento de Sistemas de Informação de uma Empresa será supervisionado por um de seus analista de sistemas. Alguns analistas supervisionam muitos estagiários e alguns não supervisionam nenhum. Qual dos diagramas seguintes mais claramente representa o relacionamento estagiário-analista?

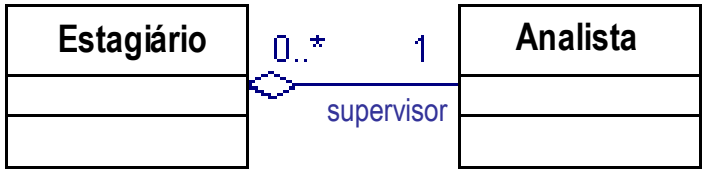
A.



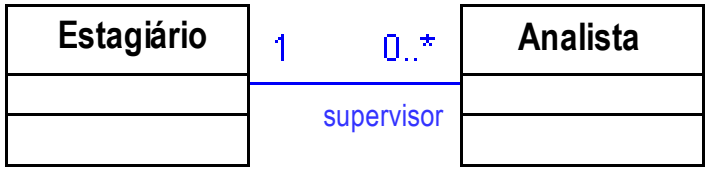
C.



B.



D.



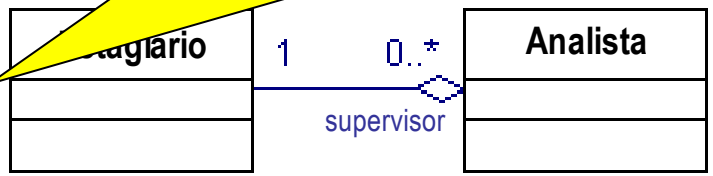
Diagramas de Classes

RESPONDA:

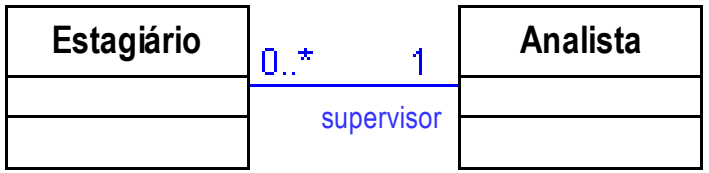
Cada estagiário do Departamento de Sistemas de Informação da
nalista de sistemas.
rios e alguns não
eguintes mais
iário-analista?

✗ **Não.** No mínimo as multiplicidades estão invertidas. Na agregação a multiplicidade "muitos" deveria ir no lado oposto ao losango.

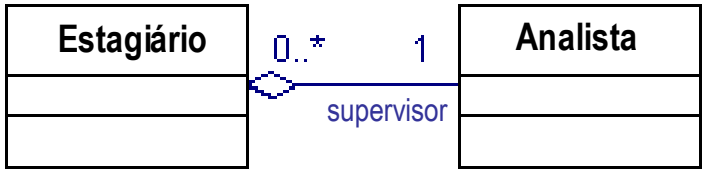
A.



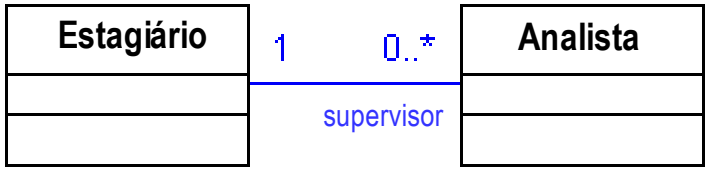
C.



B.



D.



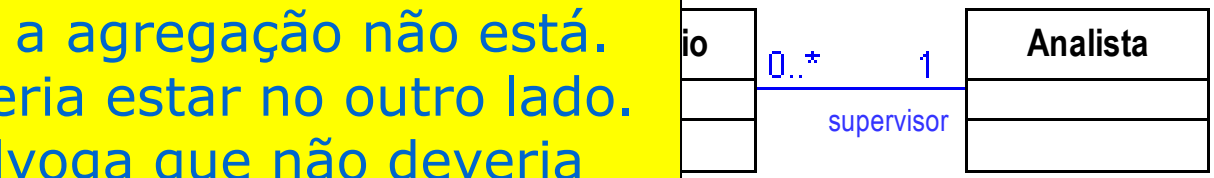
Diagramas de Classes

RESPONDA:

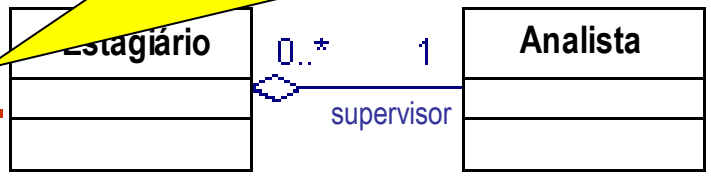
Cada estagiário do Departamento de Sistemas de Informação de uma Empresa será supervisionado por um de seus analista de sistemas. Alguns analistas supervisionam muitos estagiários e alguns não supervisionam nenhum. Qual dos diagramas seguintes

o estagiário-analista?

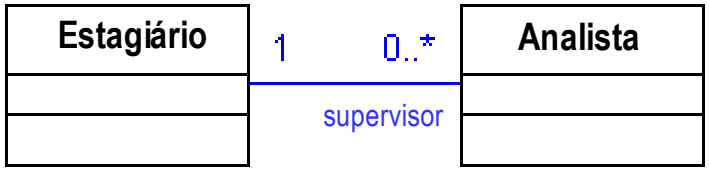
✗ **Não.** As multiplicidades estão corretas, mas a agregação não está. O losango deveria estar no outro lado. Porém, se advoga que não deveria haver qualquer agregação aqui!



B.



D.

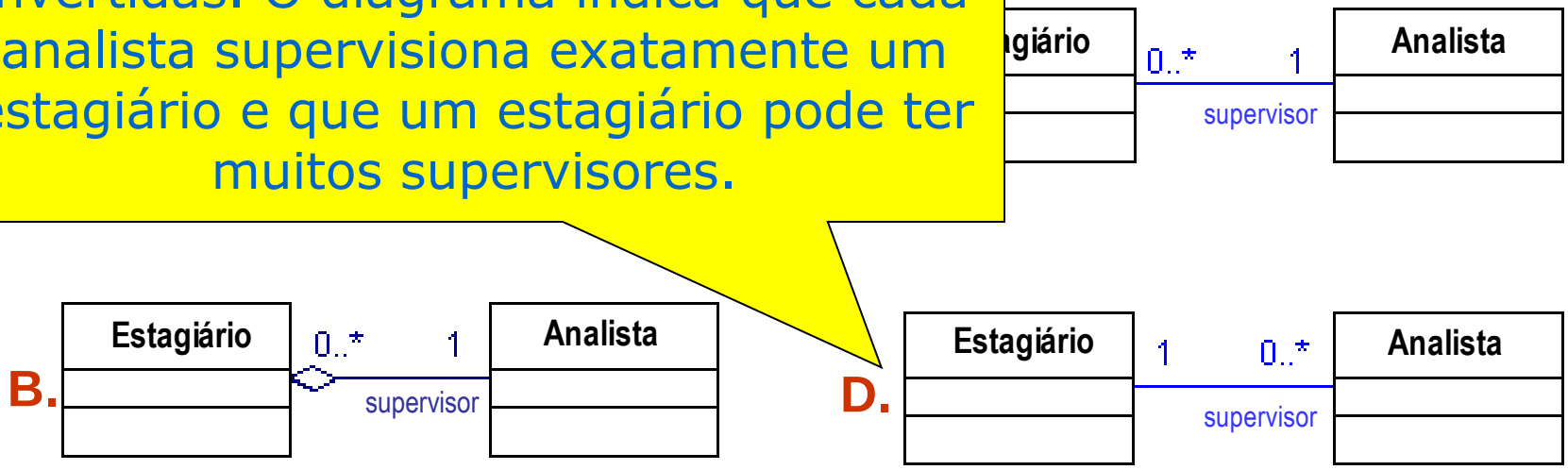


Diagramas de Classes

RESPONDA:

Cada estagiário do Departamento de Sistemas de Informação de uma Empresa será supervisionado por um de seus analista de sistemas. Alguns analistas supervisionam muitos estagiários e alguns não supervisionam nenhum. Qual dos diagramas seguintes ...ento estagiário-analista?

✗ **Não.** As multiplicidades estão invertidas. O diagrama indica que cada analista supervisiona exatamente um estagiário e que um estagiário pode ter muitos supervisores.

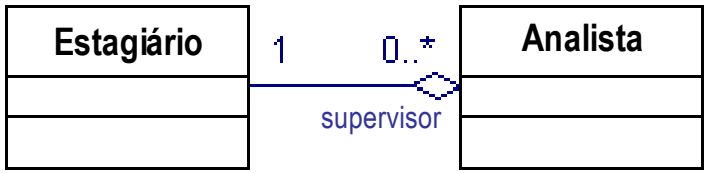


Diagramas de Classes

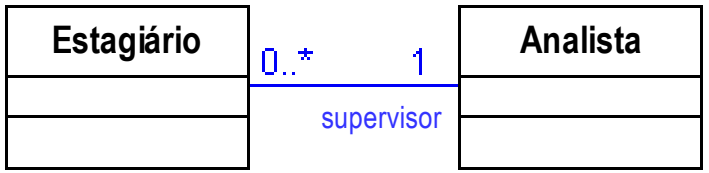
RESPONDA:

Cada estagiário do Departamento de Sistemas de Informação de uma Empresa será supervisionado por um de seus analista de sistemas. Alguns analistas supervisionam muitos estagiários e alguns não supervisionam nenhum. Qual dos diagramas seguintes mais claramente representa o relacionamento estagiário-analista?

A.



C.



✓ **Correto!** O diagrama mostra possivelmente muitos estagiários sendo supervisionados por um analista. (Obs.: Poderia ter-se usado a notação * ao invés de **0..***)