

Centro Universitário FMU

Disciplina:

Engenharia de Software I

Aula 8

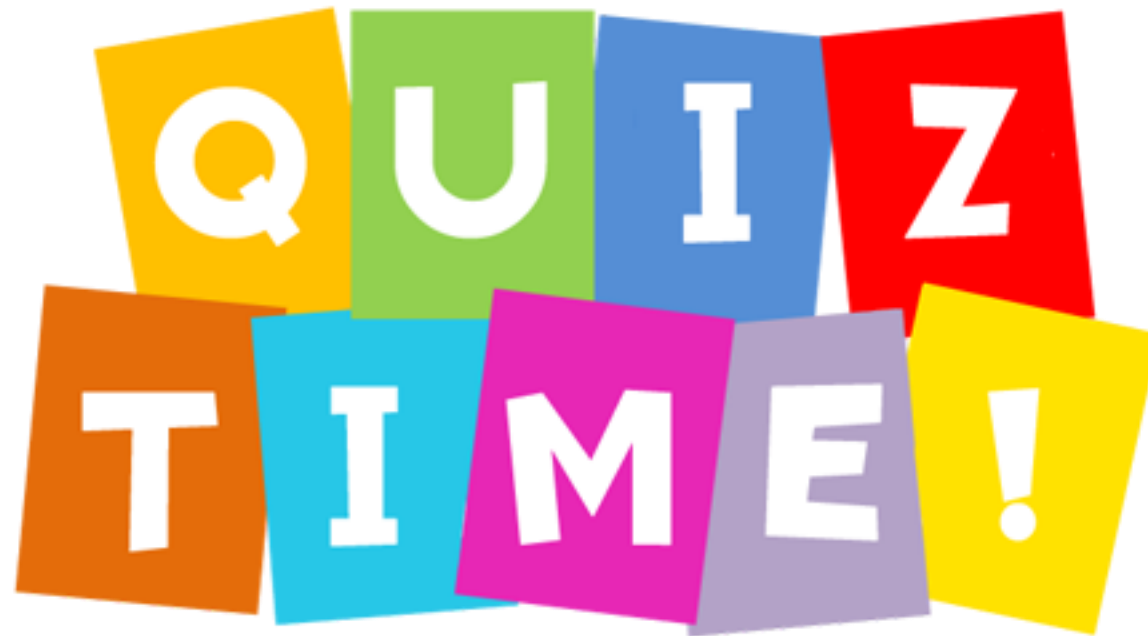
Bacharelado em Ciência da Computação

Tecnologia em Análise e Desenvolvimento de Sistemas

Prof.: Celso Eduardo Guimarães

celso.guimaraes@fmu.br

Fivou para entregar hoje:
Atividade 4
No Blackboard



www.quizizz.com/join/

Modelagem de sistemas

A modelagem de sistemas é o processo de desenvolvimento de modelos abstratos de um sistema, de maneira que cada modelo apresenta uma visão ou perspectiva diferente do sistema.

Atualmente, a modelagem de sistemas se tornou a representação de um sistema usando algum tipo de notação gráfica, que hoje em dia quase sempre são baseadas em notações em Unified Modeling Language (UML).

A modelagem de sistemas ajuda o analista a entender a funcionalidade do sistema e os modelos são usados comunicação com os clientes.

Modelagem de sistemas

Os modelos de sistemas existentes são usados durante a engenharia de requisitos para ajudar a esclarecer o que o sistema existente faz e podem ser usados como uma base discussão dos seus pontos fortes e fracos.

Os modelos de um novo sistema são usados durante a engenharia de requisitos para ajudar a explicar os requisitos propostos para outros stakeholders do sistema.

Os engenheiros usam esses modelos discutir propostas de projeto e documentar o sistema implementação.

Em um processo de engenharia dirigida a modelos, é possível gerar uma implementação do sistema completa ou parcial a partir do modelo do sistema.

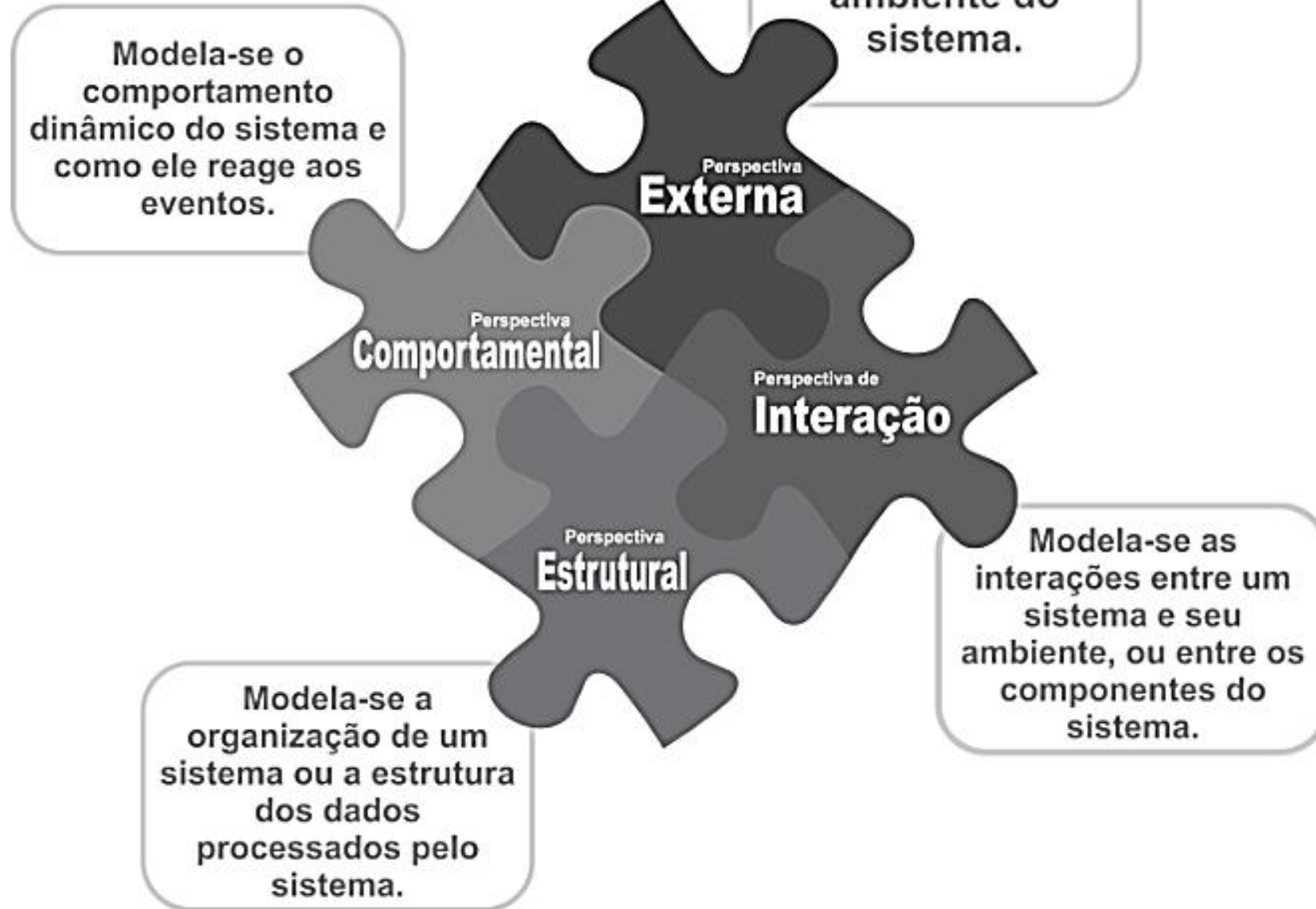
Modelagem de sistemas

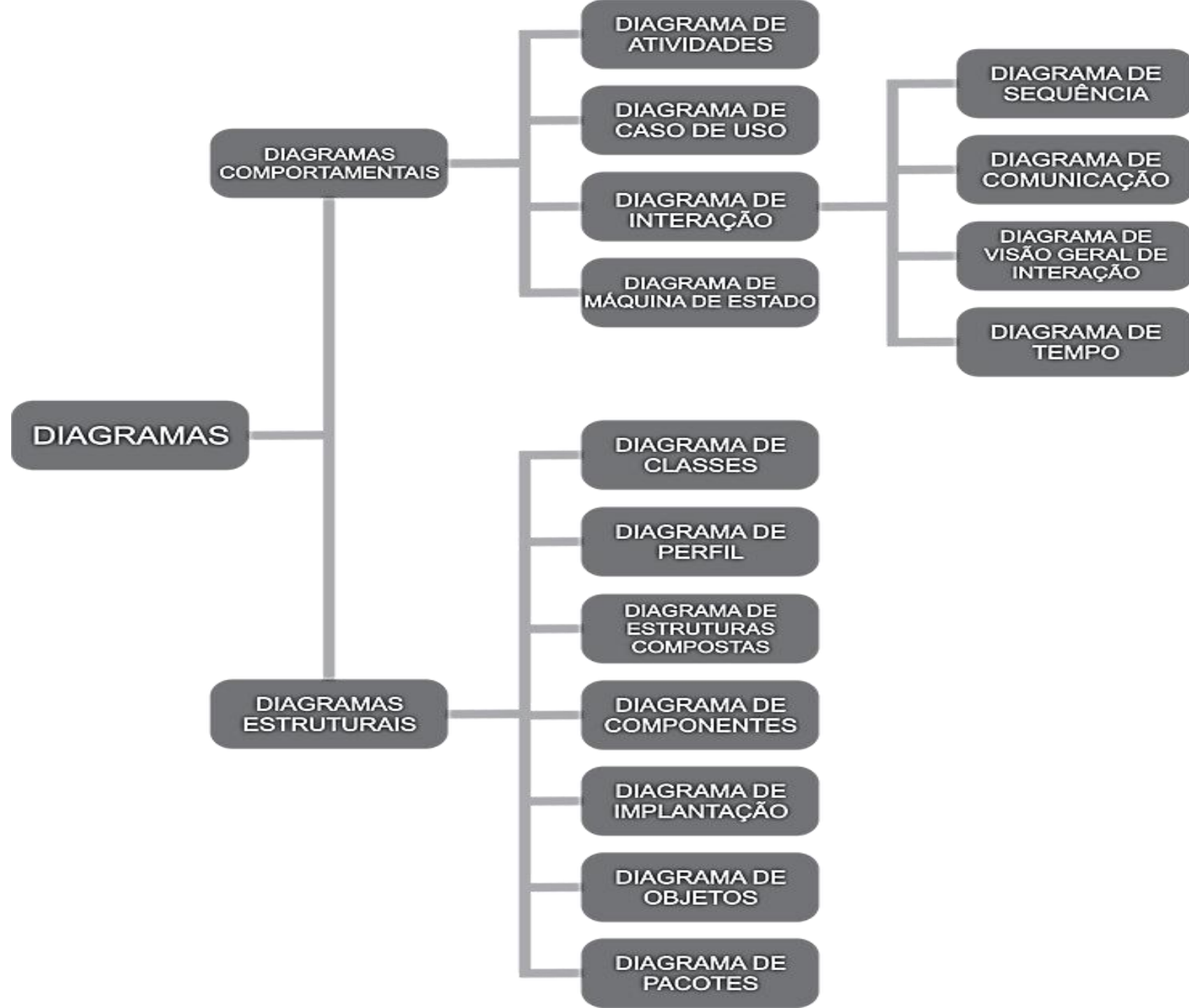
- ☐ Os modelos são usados durante o processo de engenharia de requisitos para ajudar a extrair os requisitos do sistema;
- ☐ Durante o processo de projeto, são usados para descrever o sistema para os engenheiros que o implementam;
- ☐ Após isso, são usados para documentar a estrutura e a operação do sistema.
 - ✓ Pode-se desenvolver modelos do sistema existente e do sistema a ser desenvolvido.

Modelagem de sistemas

- ☐ O aspecto mais importante de um modelo de sistema é que ele deixa de fora os detalhes. O modelo é uma abstração do sistema a ser estudado, e não uma representação alternativa dele.
- ☐ A partir de perspectivas diferentes, você pode desenvolver diversos modelos para representar o sistema.

Por exemplo:





INTRODUÇÃO À UML

Unified Modeling Language – linguagem de modelagem unificada

- É uma linguagem-padrão para descrever e documentar projeto de software.
- A UML pode ser usada para visualizar, especificar, construir e documentar os artefatos de um sistema de software.

INTRODUÇÃO À UML

Unified Modeling Language

Entendendo o vocabulário da UML (elementos visuais do diagrama e seus significados)...

... Poderá entender e especificar um sistema e explicar o projeto daquele sistema para outros interessados.

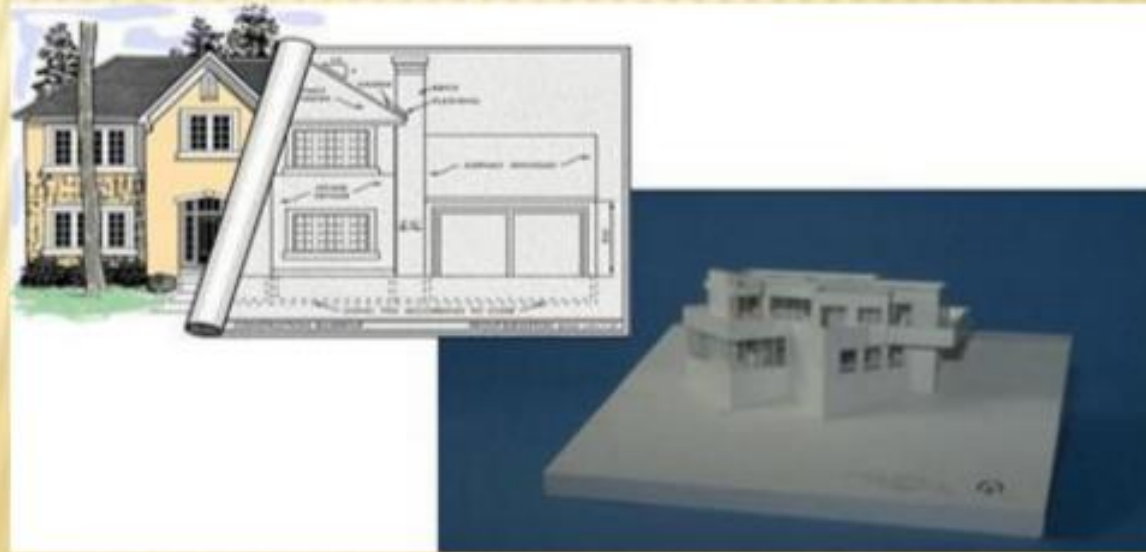
INTRODUÇÃO À UML

Breve Histórico

- Grady Booch, Jim Rumbaugh e Ivar Jacobson desenvolveram a **UML** na década de 1990:
 - **Com colaboração da comunidade de desenvolvimento de software.**
- A UML combinou um grupo de notações de modelagem concorrentes usadas pela indústria do software na época.
- Em 1997, a UML 1.0 foi apresentada ao OMG (Object Management Group), uma associação sem fins lucrativos dedicada a manter especificações para ser usadas pela indústria de computadores.
- A UML 1.0 foi revisada tornando-se a UML 1.1 e adotada mais tarde naquele ano.
- O padrão atual é a UML 2.2 e agora é um padrão ISO.

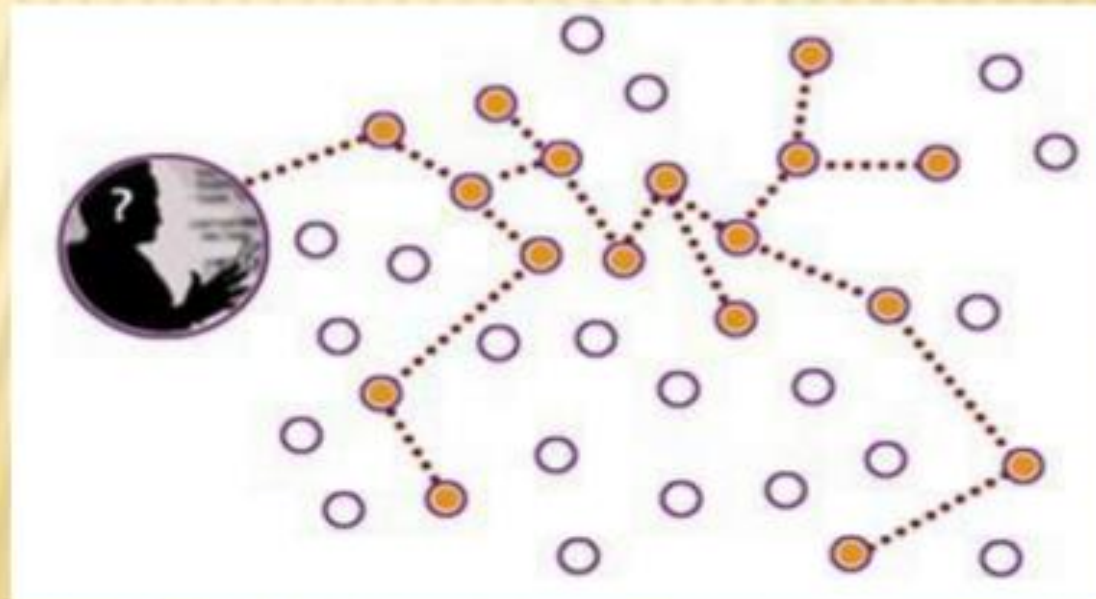
UML - MOTIVAÇÃO E NECESSIDADE DE UMA MODELAGEM VISUAL

Da mesma forma que é impossível construir uma casa sem primeiramente definir sua planta, também é impossível construir um software sem inicialmente definir sua arquitetura.



Desta forma, é extremamente importante ter uma representação visual de seu sistema antes que ele entre na etapa de implementação.

Descobrir entre as etapas a serem percorridas, aquelas mais importantes do ponto de vista do cliente.



UML - MOTIVAÇÃO E NECESSIDADE DE UMA MODELAGEM VISUAL

Necessidade de estabelecer um rumo, que deve ser

definido a partir dos requisitos de software. Uma

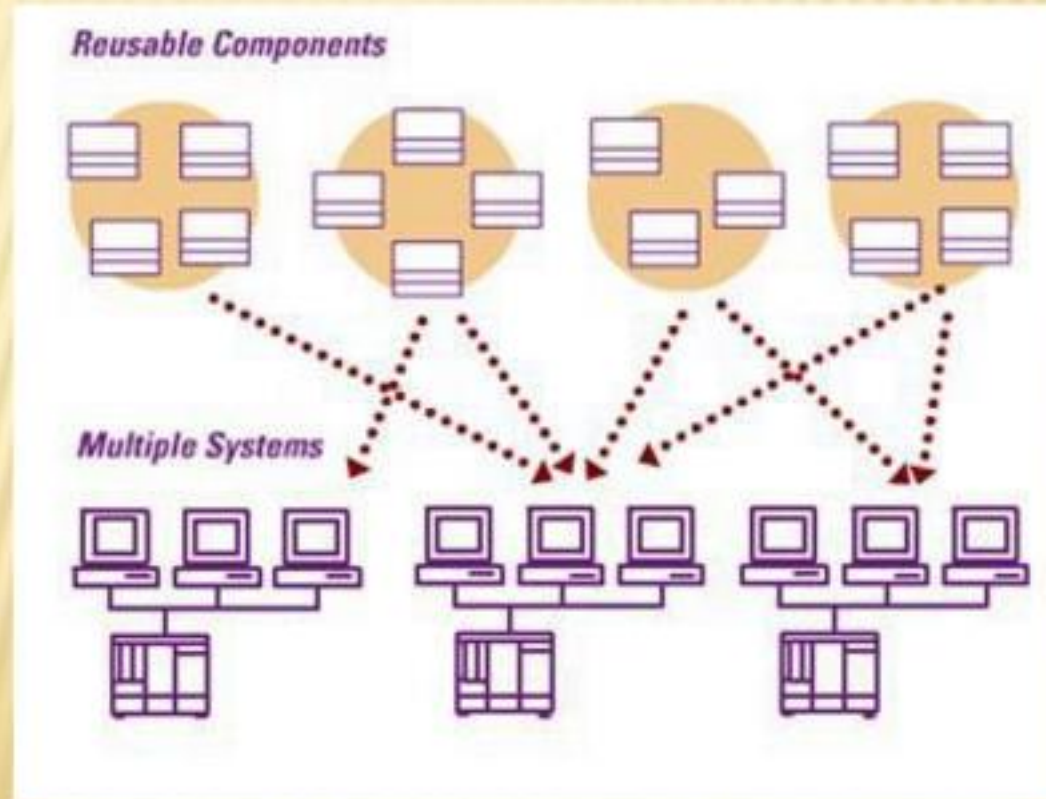
modelagem visual permite representar (especificar) estes requisitos, ou seja, facilita a captura destes requisitos.

Necessidade de estabelecer uma padronização para facilitar a comunicação entre os analistas (responsáveis pelo levantamento dos requisitos) e o time de desenvolvimento (responsáveis pela implementação).

Um sistema tem geralmente muitas classes (centenas e até milhares) e nem sempre estas classes são vistas por uma só pessoa. Isto é, dependendo do nível hierárquico desta pessoa, formas diferentes de apresentar uma diagrama de classes devem existir.

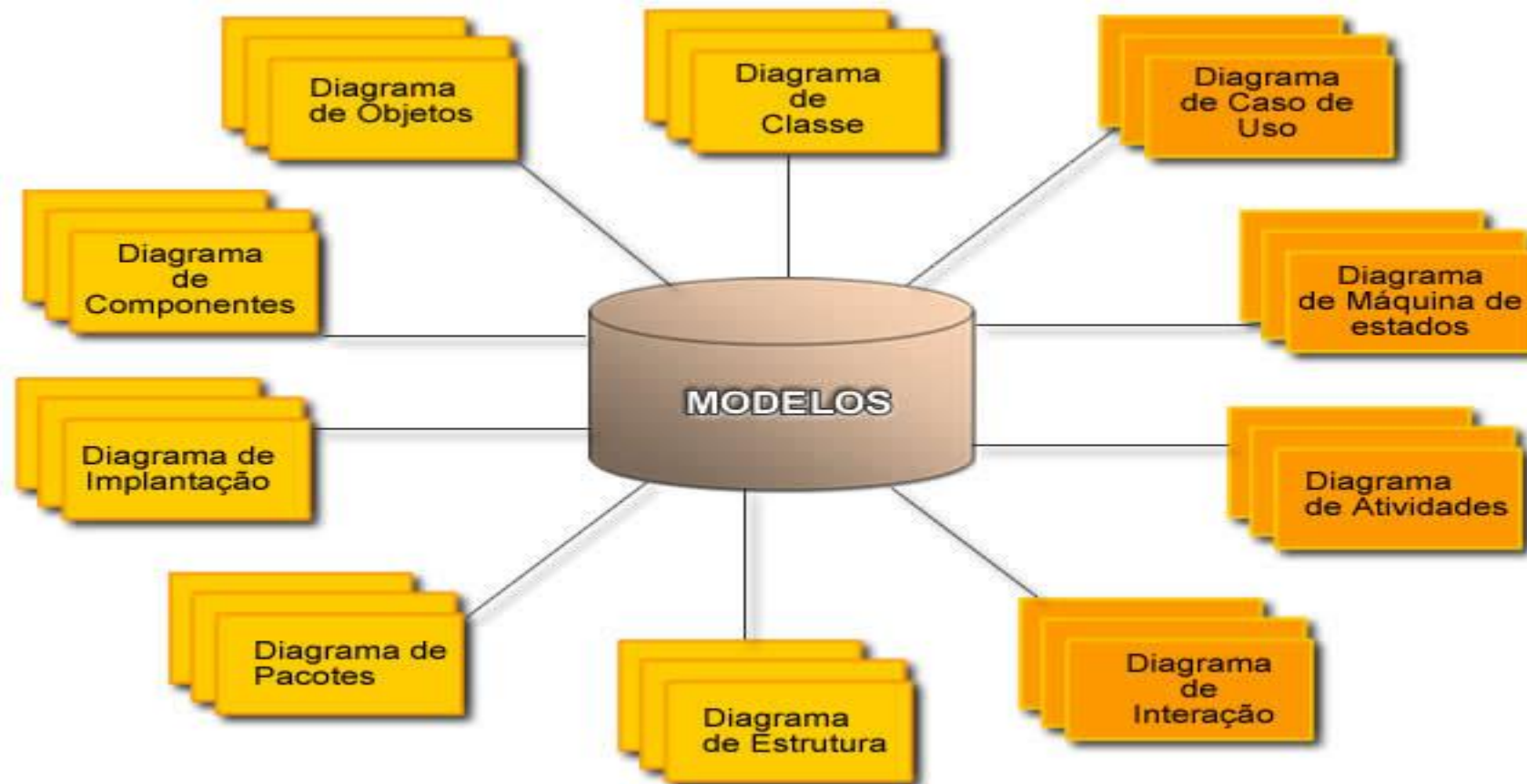
- ✗ Um cliente não quer saber o que é uma classe, mas apenas compreender determinados conceitos;
- ✗ Um gerente de um projeto não precisa ver detalhes de um modelo;
- ✗ Um time de desenvolvimento precisa ver um diagrama e compreender uma série de detalhes.

A utilização de uma modelagem visual facilita a visualização, e, por conseguinte, a criação de um melhor modelo (mais flexível, mais robusto e principalmente mais reutilizável).



UML – CONCEITOS BÁSICOS

✗ UML é uma linguagem para visualização, especificação, construção e documentação de artefatos de um software em desenvolvimento.





https://www.youtube.com/watch?time_continue=80&v=2dSq0Vu1GFo

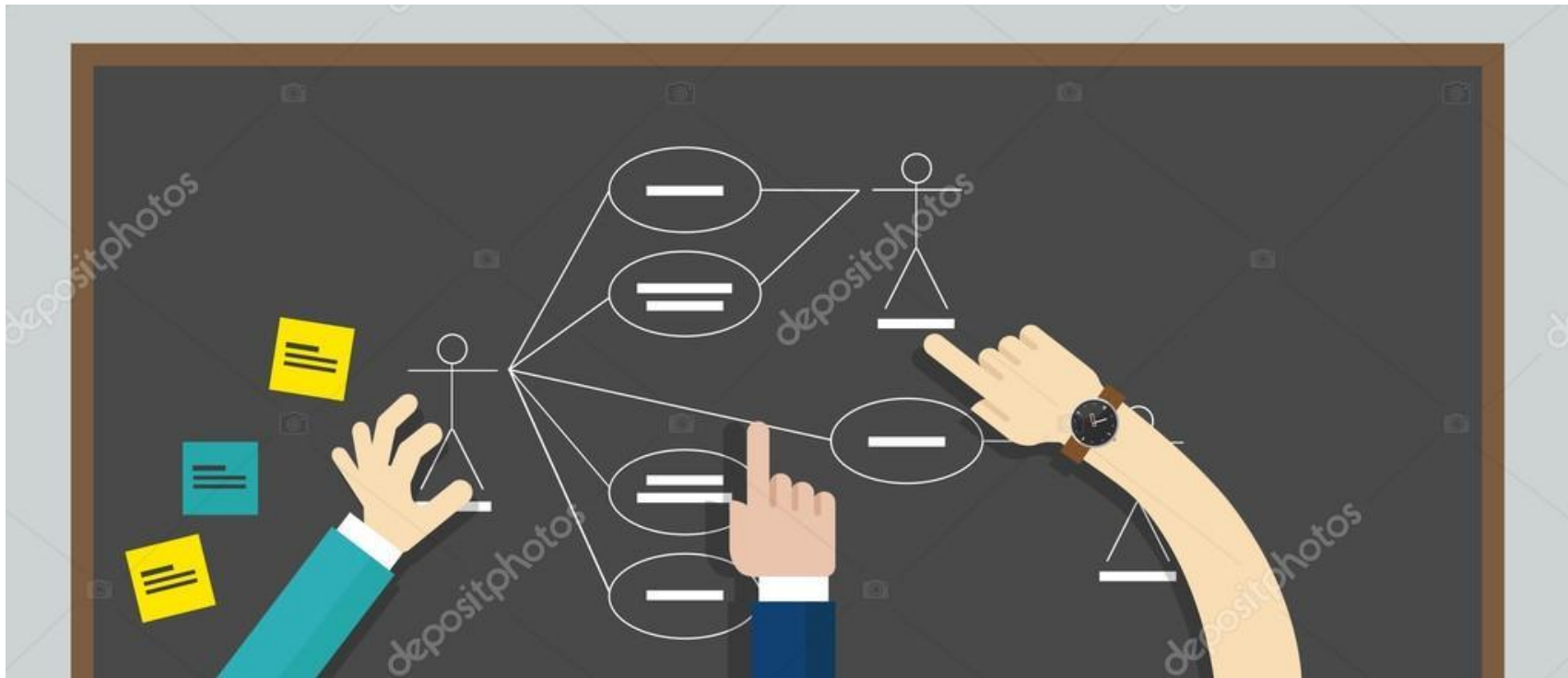
Tipos de diagramas UML

- Os diagramas de atividades, que mostram as atividades envolvidas em um processo ou no processamento de dados.
- Os diagramas de casos de uso, que mostram as interações entre um sistema e seu ambiente.
- Os diagramas de sequência, que mostram as interações entre os atores e o sistema e entre os componentes do sistema.
- Os diagramas de classe, que mostram as classes de objeto no sistema e as associações entre as classes.
- Os diagramas de estado, que mostram como o sistema reage a eventos internos e externos.

Modelagem de sistemas

- ❑ **Todo diagrama é constituído de elementos, cada um com um propósito, regras e notação diferentes para definir uma situação do processo de desenho.**
- ❑ **Embora a maioria dos diagramas seja gráfica, apresentando nós relacionados entre si por uma linha, demonstrando uma conexão, o relacionamento pode ser expresso na forma de recipiente em que um símbolo está contido em outro, ou de anexo, em que um símbolo aparece próximo a outro.**

Modelagem de Casos de Uso



Modelagem de Casos de Usos (MCU)

- ❑ O **Modelo de Casos de Uso** (MCU) é uma representação das *funcionalidades* externamente observáveis ao sistemas e dos *elementos externos* que **INTERAGEM** com ele.

Modelagem de Casos de Usos (MCU)

- ❑ O **Modelo de Casos de Uso** (MCU) é um *modelo de análise* que **representa** um **refinamento dos requisitos funcionais** do sistema em desenvolvimento.

Diagramas de Casos de Uso



Ferramenta UML utilizada na modelagem de casos de uso.

► A interação do usuário junto ao sistema pode ser ilustrada por meio do **diagrama de casos de uso**.

► Ele representa cada uma das funções do sistema e como cada ator interage com elas.

•

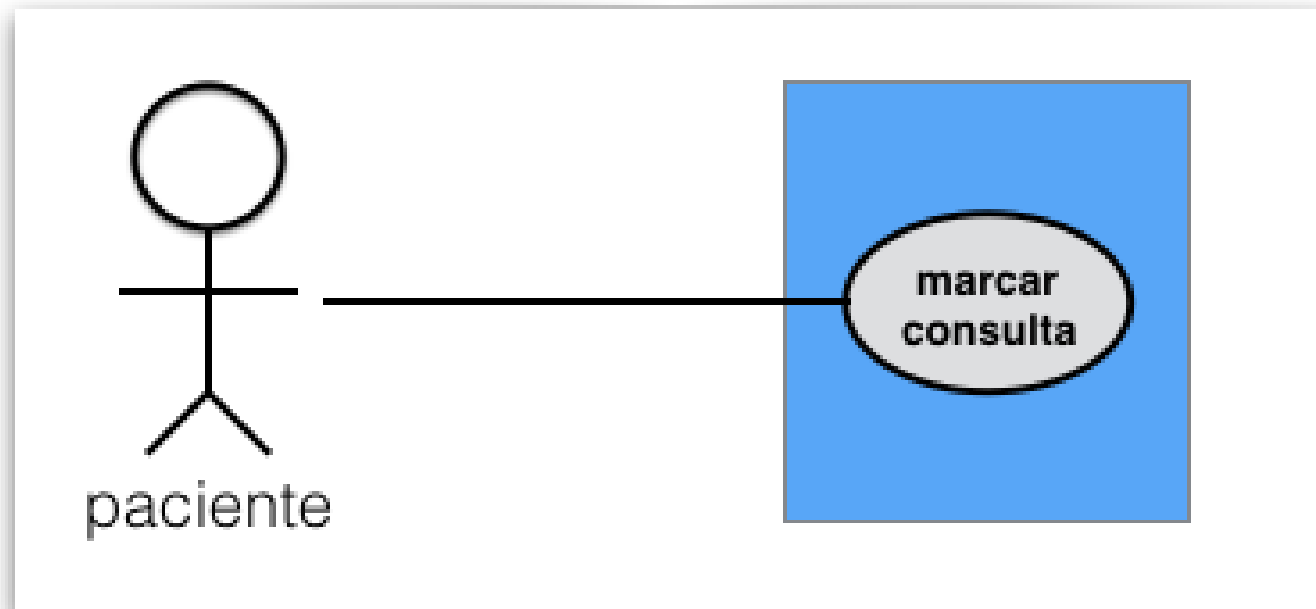


**Somente para
requisitos funcionais**

Diagramas de Casos de Uso

Ferramenta UML utilizada na modelagem de casos de uso.

► A interação do usuário junto ao sistema pode ser ilustrada por meio do **diagrama de casos de uso**.



Diagramas de Casos de Uso

- ▶ Essencialmente, um caso de uso conta uma história estilizada sobre como um usuário final (desempenhando um de uma série de papéis possíveis) interage com o sistema sob um conjunto de circunstâncias específicas.
- ▶ A história poderia ser um texto narrativo, uma descrição geral das tarefas ou interações, uma descrição baseada em gabaritos ou uma representação esquemática.
- ▶ Independentemente de sua forma, um caso de uso representa o software ou o sistema do ponto de vista do usuário final.

Diagrama de Casos de Uso

- Procura, por meio de uma linguagem simples, possibilitar a compreensão do comportamento externo do sistema por qualquer pessoa, tentando apresentar o sistema através de uma perspectiva do usuário.
- Dentre todos os diagramas da UML, é o mais abstrato e, portanto o mais flexível e informal.
- Geralmente é modelado no início da modelagem do sistema, ainda nas etapas de levantamento e análise de requisitos.
 - O que é modelado primeiro.

Diagrama de Casos de Uso

- Tem por objetivo apresentar uma visão externa geral das funções e serviços que o sistema deverá oferecer ao usuário.
 - ✓ Sem se preocupar como essas funções serão implementadas.
- Um caso de uso descreve as operações que o sistema deve cumprir para cada usuário.
 - ✓ Irá existir um caso de uso para cada tarefa que o sistema deve executar para atender aos requisitos funcionais.

Diagramas de Casos de Uso

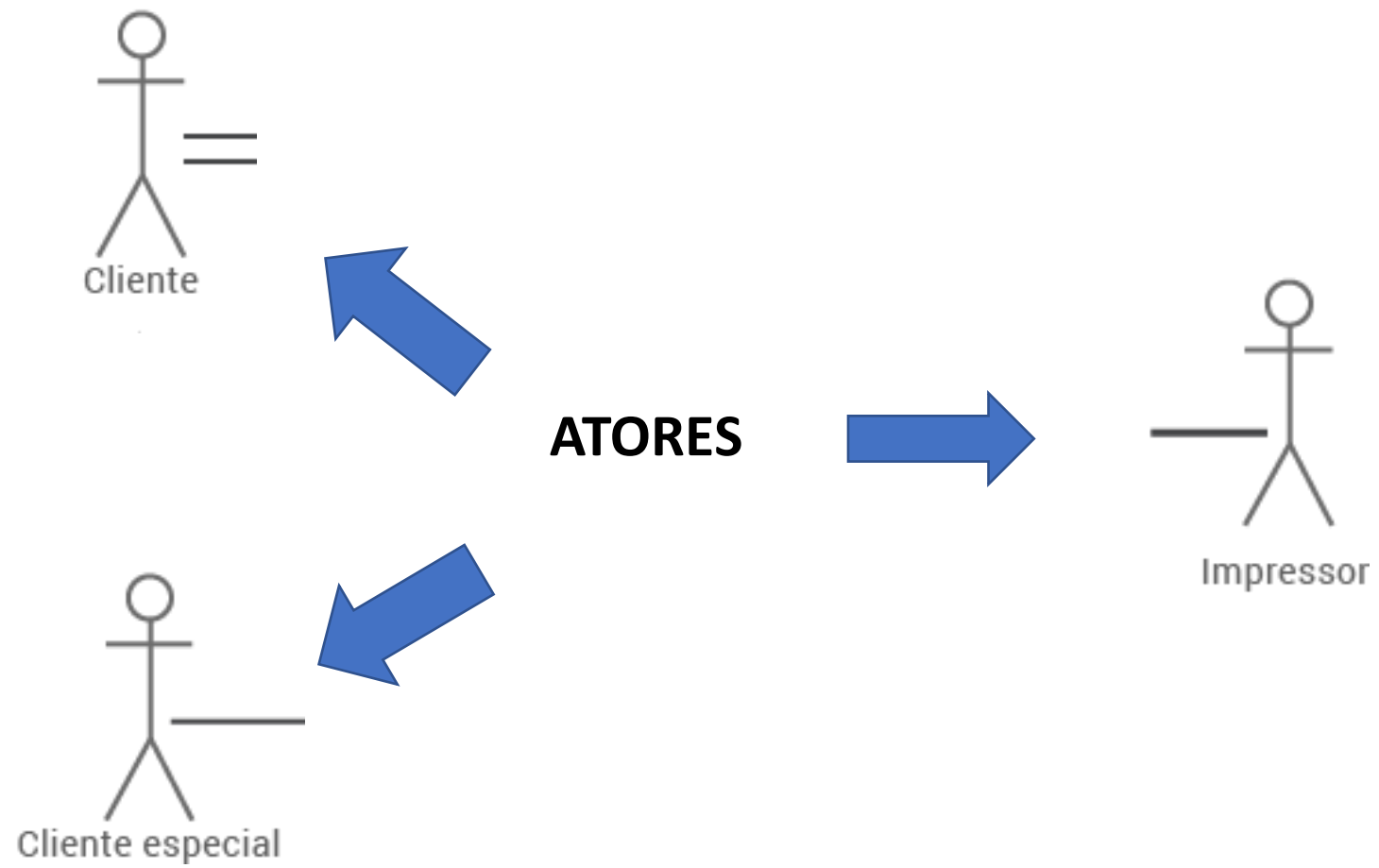
- ▶ O primeiro elemento essencial do diagrama de casos de uso é o **ator**. Esse personagem é representado por um boneco que mostra o tipo de usuário à interagir com uma ou mais funções do sistema.



- ⌘ *O ator pode representar um usuário humano que manipulará o sistema*
- ⌘ *Como pode representar sistemas integrados, como é o caso do ator “Sistema de Relatórios”.*

Diagramas de Casos de Uso

ELEMENTO	REPRESENTAÇÃO GRÁFICA	OBJETIVO
Ator		Define um conjunto coerente de papéis que os usuários do sistema podem desempenhar ao interagir com ele. Pode ser considerado tanto um indivíduo como um sistema externo.



Diagramas de Casos de Uso


- ▶ Depois de identificado os atores é preciso então ilustrar os **casos de uso** em si. Cada um deles se tornará uma função dentro do sistema.





```
graph TD; U1([Vender livros]); U2([Administrar estoque]);
```

Vender livros

Administrar
estoque

Diagramas de Casos de Uso

ELEMENTO	REPRESENTAÇÃO GRÁFICA	OBJETIVO
Caso de Uso		Define uma sequência de ações realizadas por um sistema que produz um resultado de valor observável para determinado ator.

ELEMENTO	REPRESENTAÇÃO GRÁFICA	OBJETIVO
Relacionamento Simples		Linha que interliga os atores aos casos de uso que eles têm permissão de executar
Relacionamento Herança		Ideia de generalização. Pode ser aplicado tanto entre casos de uso como entre atores. Dessa maneira, é possível aproveitar características do caso de uso ou ator mais genérico.
Relacionamento Extensão		Representa que a ação do sistema que está sendo estendida pode ser executada pelo caso de uso que a estendeu. A seta é direcionada para o caso de uso que poderá executar o caso de uso opcional.
Relacionamento Inclusão		Representa que a ação do sistema será executada pelo caso de uso que a inclui. A seta é direcionada para o caso de uso que será incluído

Include

Quando o caso de uso **A** “inclui” o caso de uso **B**, significa que sempre que o caso de uso **A** for executado o caso de uso **B** também será executado. A direção do relacionamento é do caso de uso que está **incluindo** para o caso de uso **incluído**.

Extend

Quando o caso de uso **B** estende o caso de uso **A**, significa que quando o caso de uso **A** for executado o caso de uso **B** poderá (poderá – talvez não seja) ser executado também. A direção do relacionamento é do caso de uso **extensor** (aqui o caso de uso B) para o caso de uso **estendido** (aqui o caso de uso A).

Generalization

Quando o caso de uso **B** generaliza o caso de uso **C** isso significa que, além de fazer tudo que nele está especificado (ele = B), ele **também executará tudo que está especificado** no caso de uso **C**.

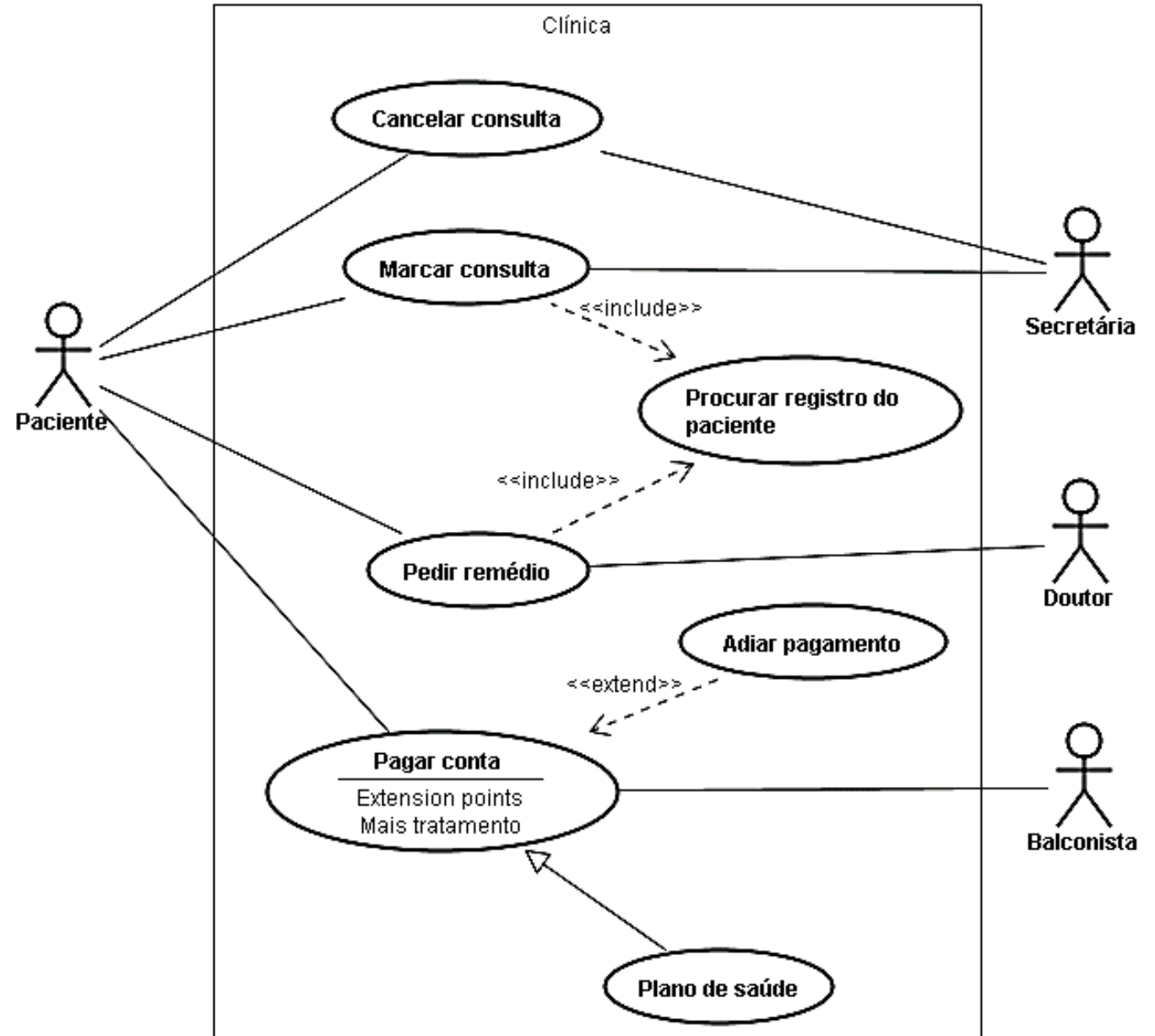
Muitos profissionais falam que isso não deve ser compreendido como a herança da orientação a objetos, mas na minha opinião deve ser sim, apenas (em tempo de modelagem de caso de uso) estamos num nível de abstração diferente, mas o produto final desta modelagem será software codificado.

A direção do relacionamento é sempre do **generalizador** (aqui o caso de uso B) para o **generalizado** (caso de uso C).

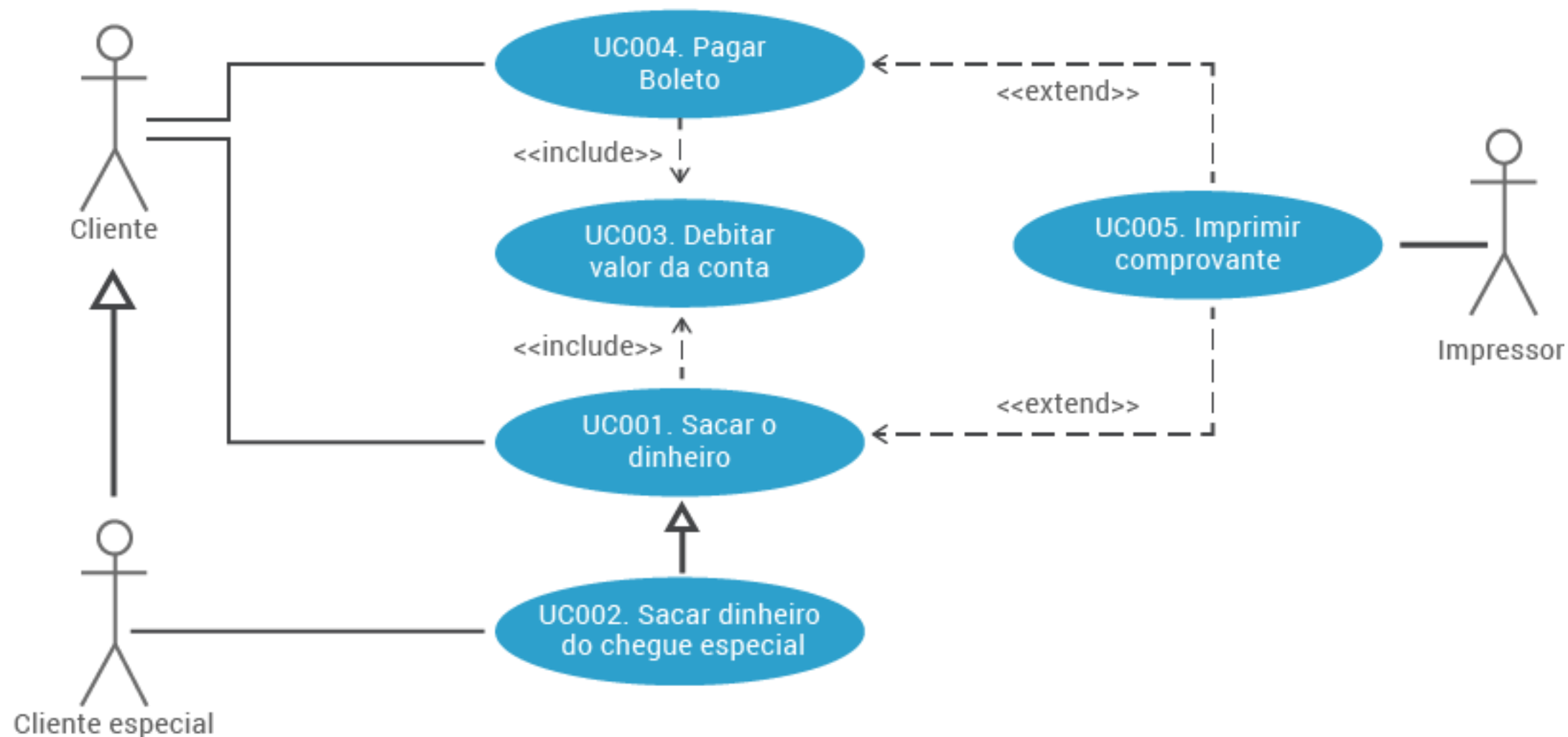
Diagramas de Casos de Uso

Colocamos um geral conhecido por **fronteira de sistema**.

– Esse bloco delimita os casos de uso e permite identificar todos os que serão implementados e disponibilizados pelo sistema.



CAIXA ELETRÔNICO

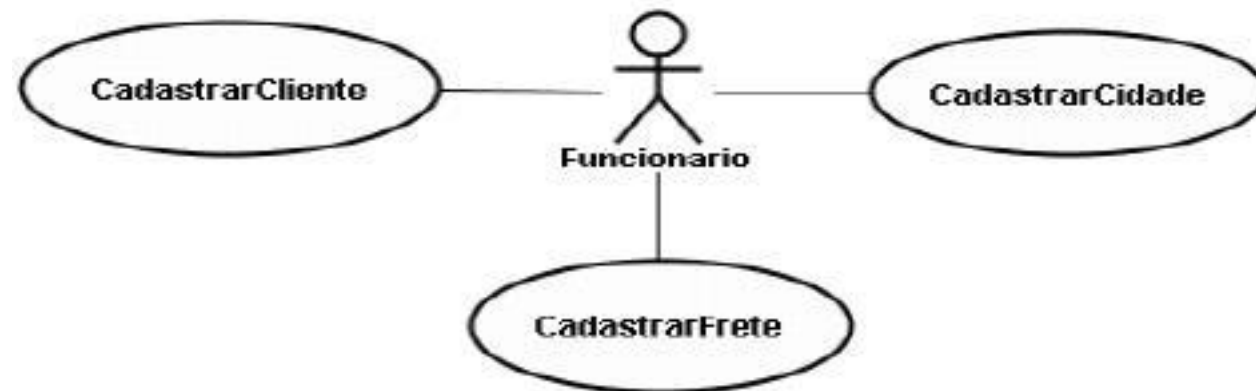


Diagramas de Casos de Uso

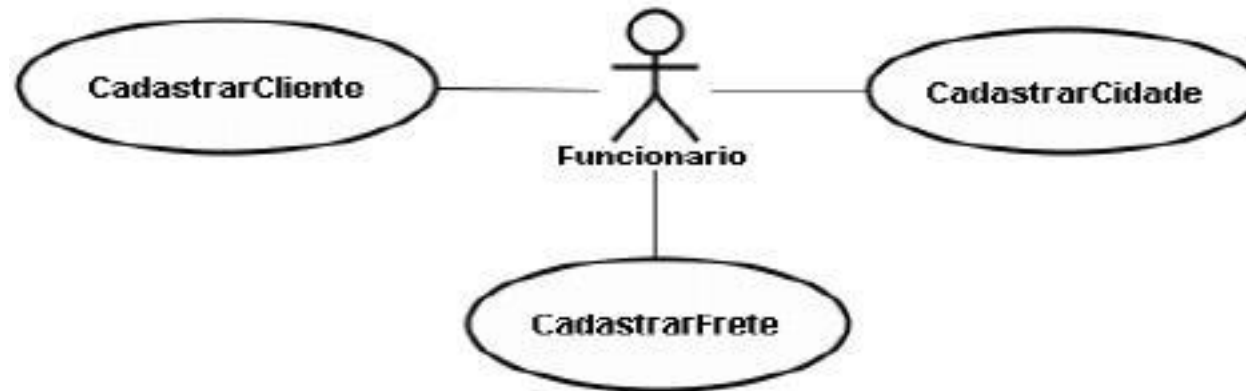
- Com o diagrama em mãos é possível entender quais as funções do sistema e como cada usuário interagirá com elas.
- Tendo em vista o contexto de uso do sistema, esse diagrama pode auxiliar na detecção de atividades e no levantamento de requisitos.
- Desse modo, sua criação e utilização se dá principalmente nas fases iniciais do projeto, mas com utilização ao longo do ciclo de vida do projeto como referência para testes de homologação e entrega.

Fragmento de um sistema para uma transportadora

- Consideremos os seguintes requisitos:
- ✓ O sistema deve permitir ao Funcionário cadastrar Clientes, contendo os dados: nome, endereço e telefone;
- ✓ O sistema deve permitir ao Funcionário cadastrar Cidades, que representam os lugares abrangidos pela empresa de transportes e contêm o nome da cidade, o estado a que pertence, e o valor para a taxa de entrega;
- ✓ O sistema deve permitir ao Funcionário cadastrar Fretes, contendo um código, uma descrição, o peso total, um cliente e a cidade de destino, não podendo haver um frete sem os dados citados. Cada frete deve ter ainda o seu valor, que deve ser calculado através do peso multiplicado por um valor fixo, acrescido da taxa de entrega da cidade de destino.



- ❑ O diagrama fornece visão geral das funcionalidades do sistema, mas não apresenta detalhamento suficiente para o entendimento total do problema, nem mesmo para estimativas relativas a tempo e custo de desenvolvimento.



- ❑ Desta forma, uma estratégia comum, apesar de não descrita formalmente na UML, é construir especificações mais detalhadas para cada um dos casos de uso do sistema.

Existem vários templates diferentes que podem ser utilizados para este fim.
Esse é somente um exemplo

Nome do Caso de Uso	CadastrarFrete	
Descrição	Este caso de uso permite ao funcionário da transportadora o cadastro de fretes de clientes.	
Ator Envolvido	Funcionario	
Interação entre Ator e Sistema	Funcionario	Sistema
	O caso de uso é iniciado quando o Funcionário seleciona a opção Cadastrar Frete.	
		Sistema apresenta janela com os campos código do frete, cliente (RIO1), cidade de destino (RIO2), descrição e peso total.
	Funcionário preenche campos e seleciona a opção Efetuar Cadastro.	
		Sistema valida as informações preenchidas pelo Funcionário (EX01).
		Sistema calcula o valor do frete e o exibe na janela (RN01).
		Sistema cadastra frete com código único gerado automaticamente.
	Funcionário seleciona opção de fechar a janela.	
		Sistema fecha a janela e o caso de uso é encerrado.
Exceções	EX01	Os campos devem estar todos preenchidos e de acordo com o domínio (tipo) do atributo. Se houver problemas no preenchimento do formulário o sistema exibe a mensagem de erro apropriada.
Alternativas	AL01	O funcionário pode cancelar a operação de cadastramento de frete a qualquer momento, fechando a janela.
Regras de Negócio	RN01	O frete deve ser calculado através da multiplicação do peso total do frete por R\$ 10,00, acrescido da taxa de entrega associada à cidade de destino.
Requisitos de Interface com o Usuário	RIO1	O sistema deve apresentar os usuários cadastrados numa caixa de combinação, para serem escolhidos pelo funcionário.
	RIO2	O sistema deve apresentar as cidades cadastradas numa caixa de combinação, para serem escolhidas pelo funcionário.

- **Nome do caso de uso:** deve conter o mesmo nome do caso de uso que está no diagrama em questão e que será detalhado;
- **Descrição:** um resumo da utilidade do caso de uso;
- **Ator envolvido:** é o ator quem executa o caso de uso, o mesmo do diagrama de casos de uso. Em determinadas situações, pode haver mais de um ator envolvido;
- **Interação entre o ator e o sistema:** descreve os passos envolvidos na realização do caso de uso, evidenciando as responsabilidades do ator e do sistema, num processo interativo;
- **Exceções:** indicam situações onde, primordialmente, o tratamento de erros deve ser efetuado;
- **Alternativas:** indicam situações opcionais que podem ocorrer durante o cenário que está sendo descrito pelo caso de uso;
- **Regras de Negócio:** são as regras impostas para a utilização do caso de uso, definidas pelo domínio da aplicação;
- **Requisitos de Interface com o Usuário:** descrevem características que devem ser implementadas na interface com o usuário.

Diagrama de Classes



Classe

Representa um conjunto de objetos com características afins. Uma classe **define o comportamento** dos objetos através de seus **métodos**, e quais **estados** ele é capaz de manter através de seus atributos.

Exemplo de classe: **Veículos**

Representa a abstração de um conjunto de **objetos** do mundo real que possuem tipos de características e de comportamentos comuns.

Atributos e Comportamentos Comuns



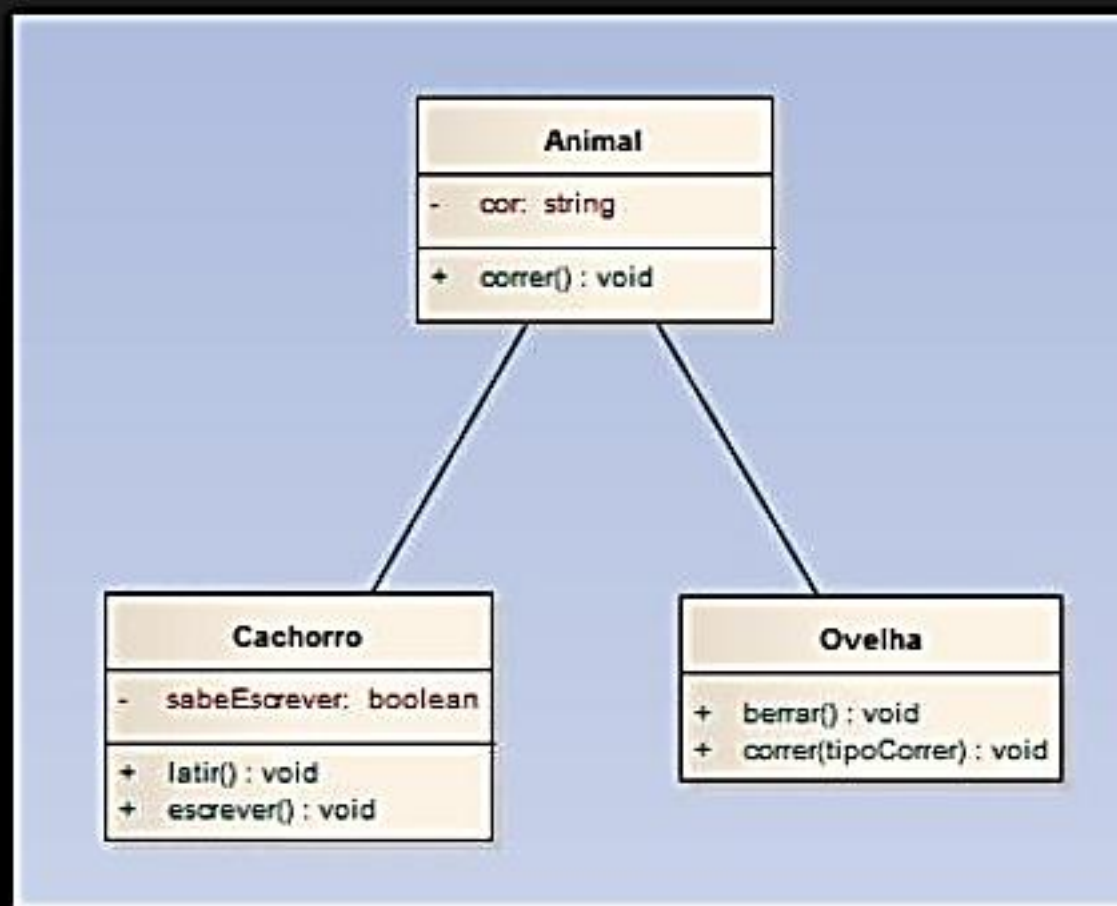
CLASSE

- O que é Classe?

- Atributos e Métodos.

Atributos = características

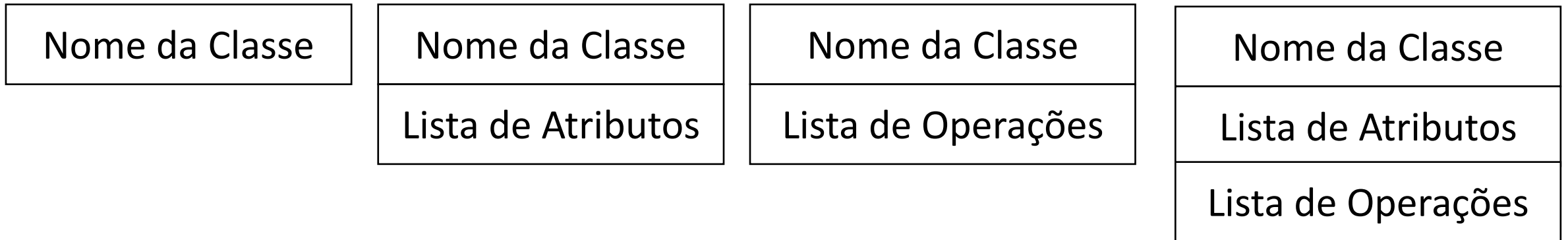
Métodos = ações



Operações caracterizam o comportamento de um objeto, e são o único meio de acessar, manipular e modificar os atributos de um objeto.

Diagrama de Classes

- ❑ Uma CLASSE é representada por, *no máximo*, três compartimentos.



Possíveis notações para uma classe na UML

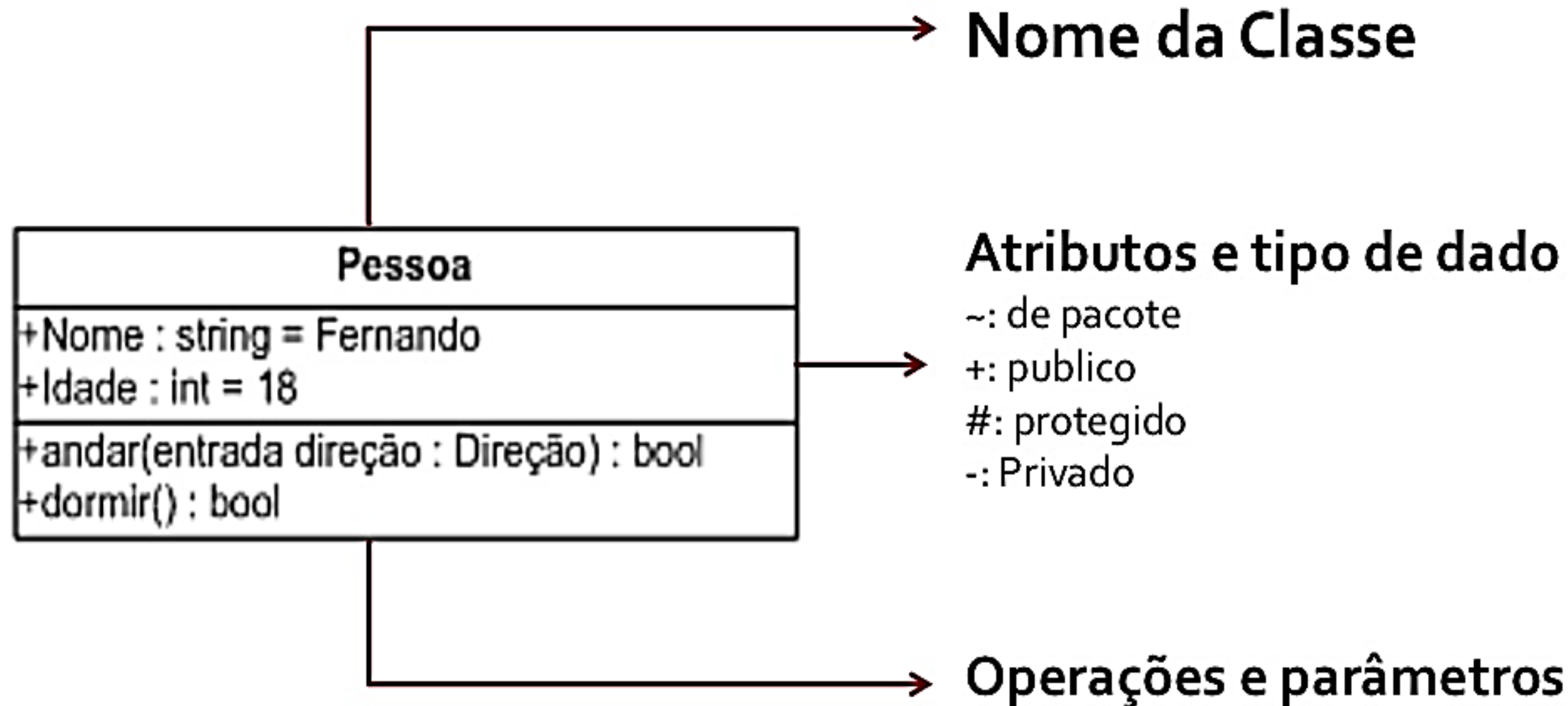


Diagrama de Classes - Visibilidade

Indica o nível de acessibilidade de um elemento, isto é, por quem uma propriedade (atributo ou método) pode ser utilizada.

+	Pública – O atributo ou método pode ser usado por qualquer objeto.
#	Protegida – O atributo ou método pode ser usado por qualquer objeto da classe e também por suas subclasses.
~	Pacote – O atributo ou método é visível por qualquer objeto dentro do pacote.
-	Privada – O atributo ou método é visível somente pela classe que o define.

Diagrama de Classes - Multiplicidade

Indica uma faixa de cardinalidade permitida a um elemento, isto é, a quantidade de instância possíveis em um relacionamento.

0..1	Zero ou um.
1..1	Um e somente um.
0..*	Zero ou muitos.
*	Muitos.
1..*	No mínimo um ou muitos.
3..5	Mínimo de três e máximo de cinco.

Exemplos de Expressões de Multiplicidade

*	T	zero ou mais; "muitos(as)"
1..*	T	um ou mais
1..40	T	um a 40
5	T	exatamente 5
3, 5, 8	T	exatamente 3, 5 ou 8

Diagrama de Classes – Atributos e Métodos

- **Atributos** são os dados das classes. Os valores dos atributos variam de uma instância de classe para outra.
- **Métodos** são as funções que um objeto pode executar.

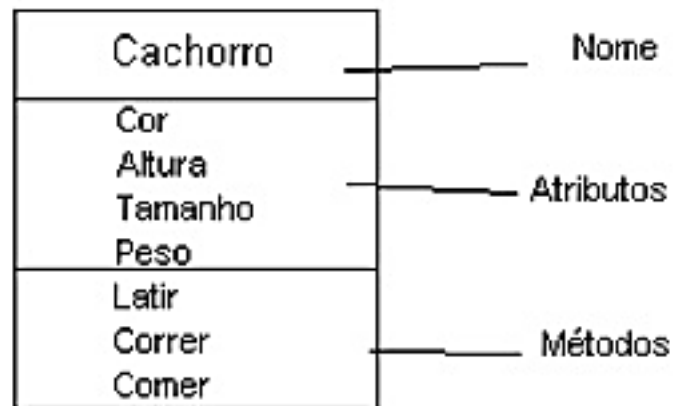


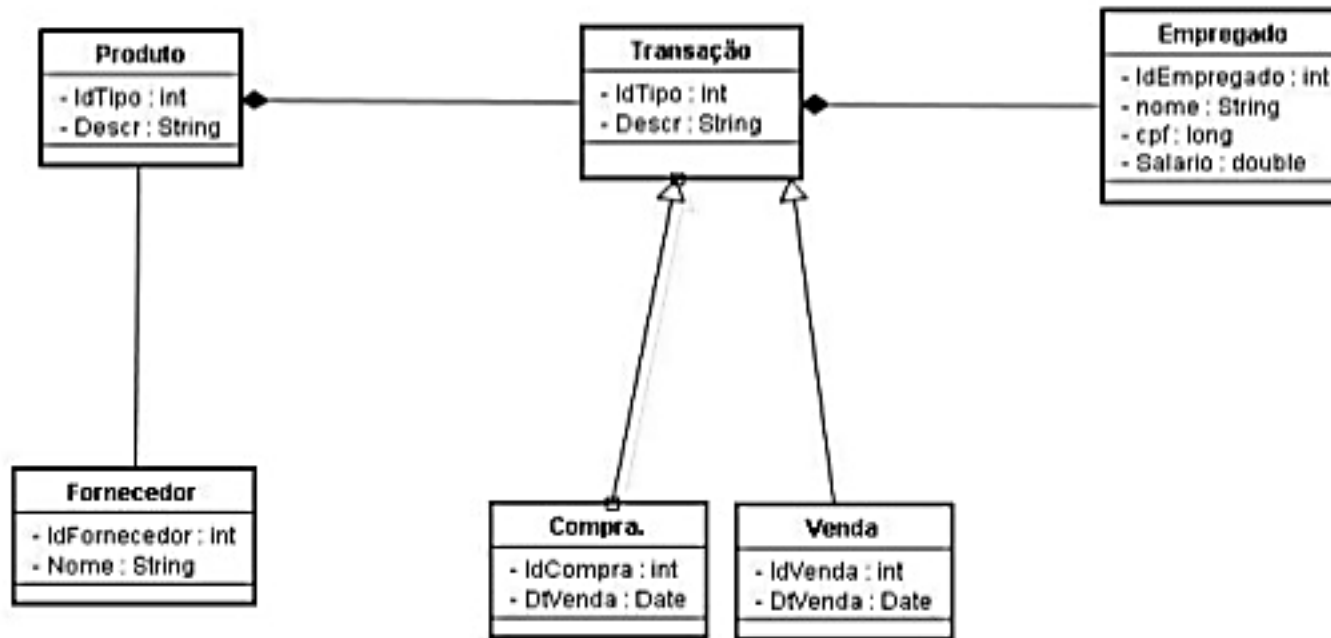
Diagrama de Classes

- O diagrama de casos de uso fornece uma visão externa do sistema.
- Ele atende as necessidades de apoio no levantamento e entendimento de requisitos junto aos usuários.
- Com posse desse diagrama, o analista precisa prosseguir no desenvolvimento do sistema.

Diagrama de Classes

- Na OO a funcionalidade é fornecida por meio de colaboração entre objetos.
- Externamente ao sistema, os atores visualizam resultados de cálculos, relatórios produzidos, confirmações de requisições realizadas etc.
- internamente, os objetos dos sistemas colaboram uns com os outros para produzir os resultados visíveis fora.

- Use os diagramas de classe para fazer a modelagem da visão estática do projeto de um sistema.



- O aspecto estático é porque não apresenta informações sobre como os objetos do sistema interagem no decorrer do tempo.
- Também é dito estrutural porque a estrutura das classes de objetos componentes do sistema e as relações entre elas são representadas.

Diagrama de Classes

- Esses aspectos (estático e dinâmico) da OO não são independentes.
- A construção de um serve para adicionar detalhes no outro.
- Exemplo: o modelador, em determinado momento, detecta a necessidade de uma mensagem entre dois objetos. Isso implica em uma referência estrutural entre eles.

Modelo de Classes

- À medida que o sistema é desenvolvido, o modelo de classes é incrementado com novos detalhes. Há 3 estágios de abstração sucessivos pelos quais o modelo de classes passa:
 - ✓ Modelo de classes de análise
 - ✓ Modelo de classes de especificação
 - ✓ Modelo de classes de implementação

Modelo de Classes

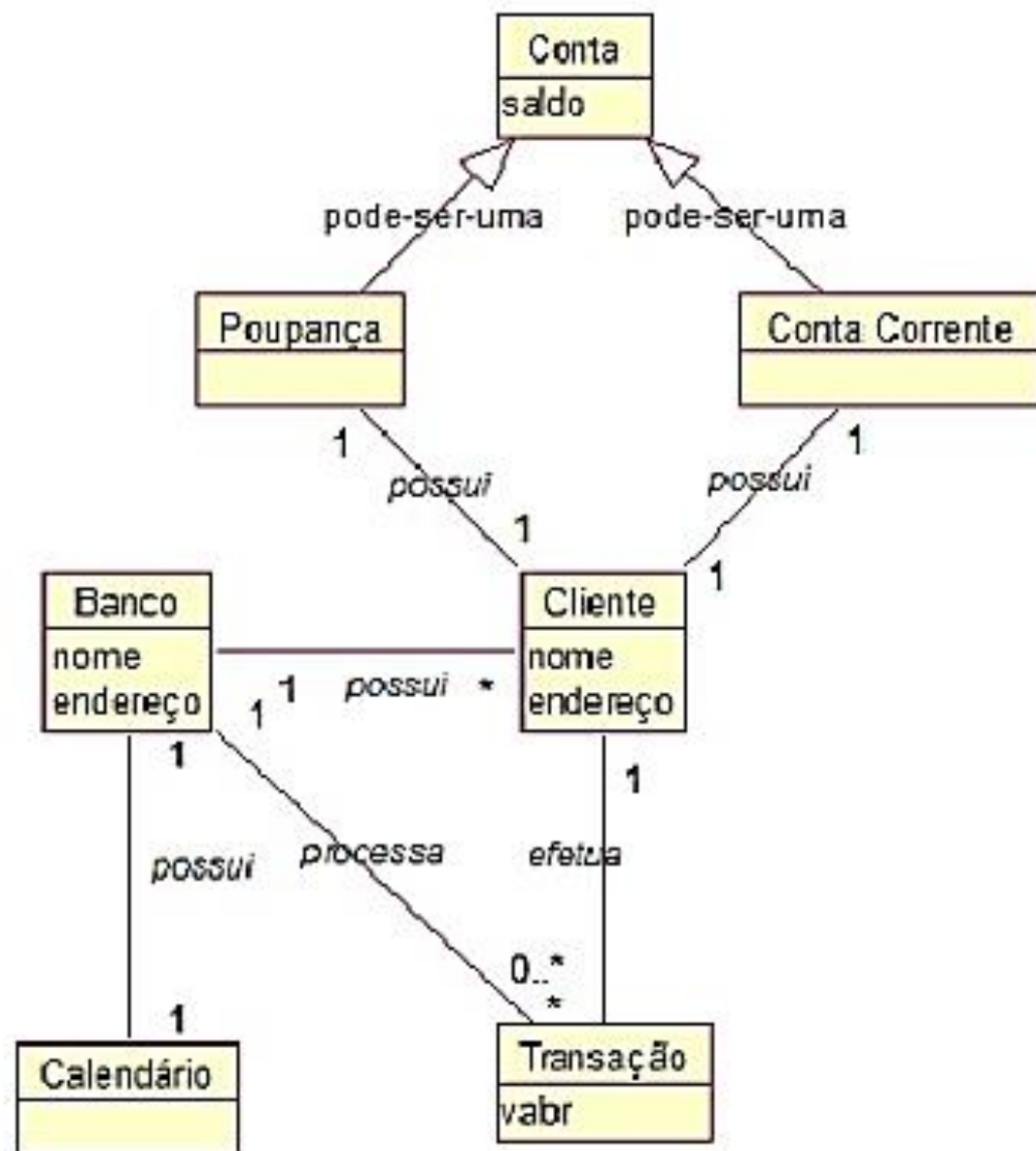
✓ Modelo de classes de análise

Focamos a atenção em “o que” o sistema deve fazer.

É construído na fase de análise e em geral não leva em consideração restrições tecnológicas.

O modelo de casos de uso, juntamente com o modelo de classes formam o conjunto principal da fase de análise do processo de desenvolvimento de software.

Conceitual



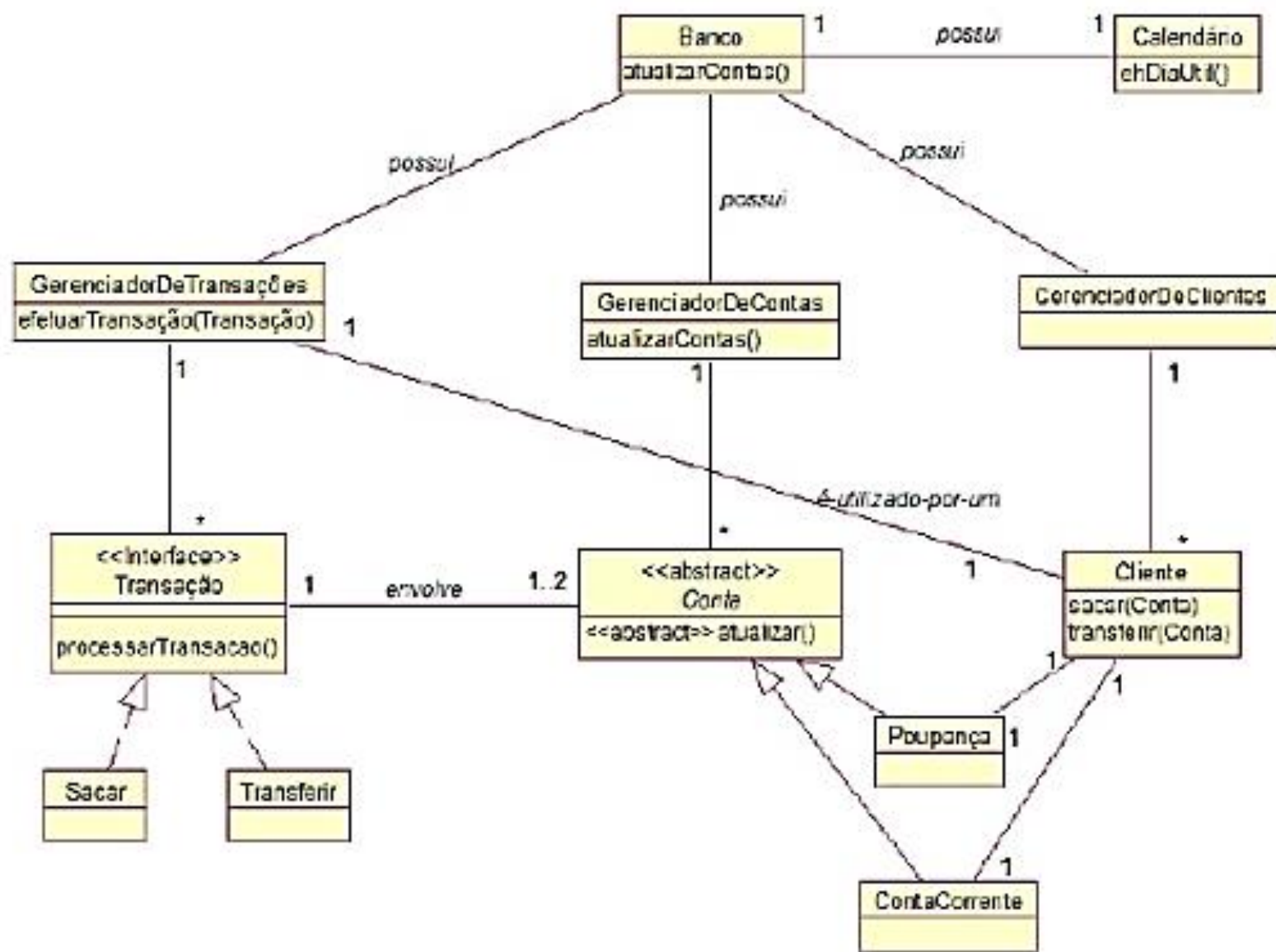
Modelo de Classes

✓ Modelo de classes de especificação

É um detalhamento do modelo de classes de análise. É também conhecido como modelo de classes de projeto.

Nesse momento descobrimos a necessidade de criar outras classes focando a atenção sobre “como” o sistema deverá funcionar.

Especificação



Modelo de Classes

✓ Modelo de classes de implementação

É a implementação da classe em uma linguagem de programação – normalmente Orientada a Objetos, como Java, C#, C++, etc.

Ele é construído na fase de implementação de um modelo iterativo.

Implementação

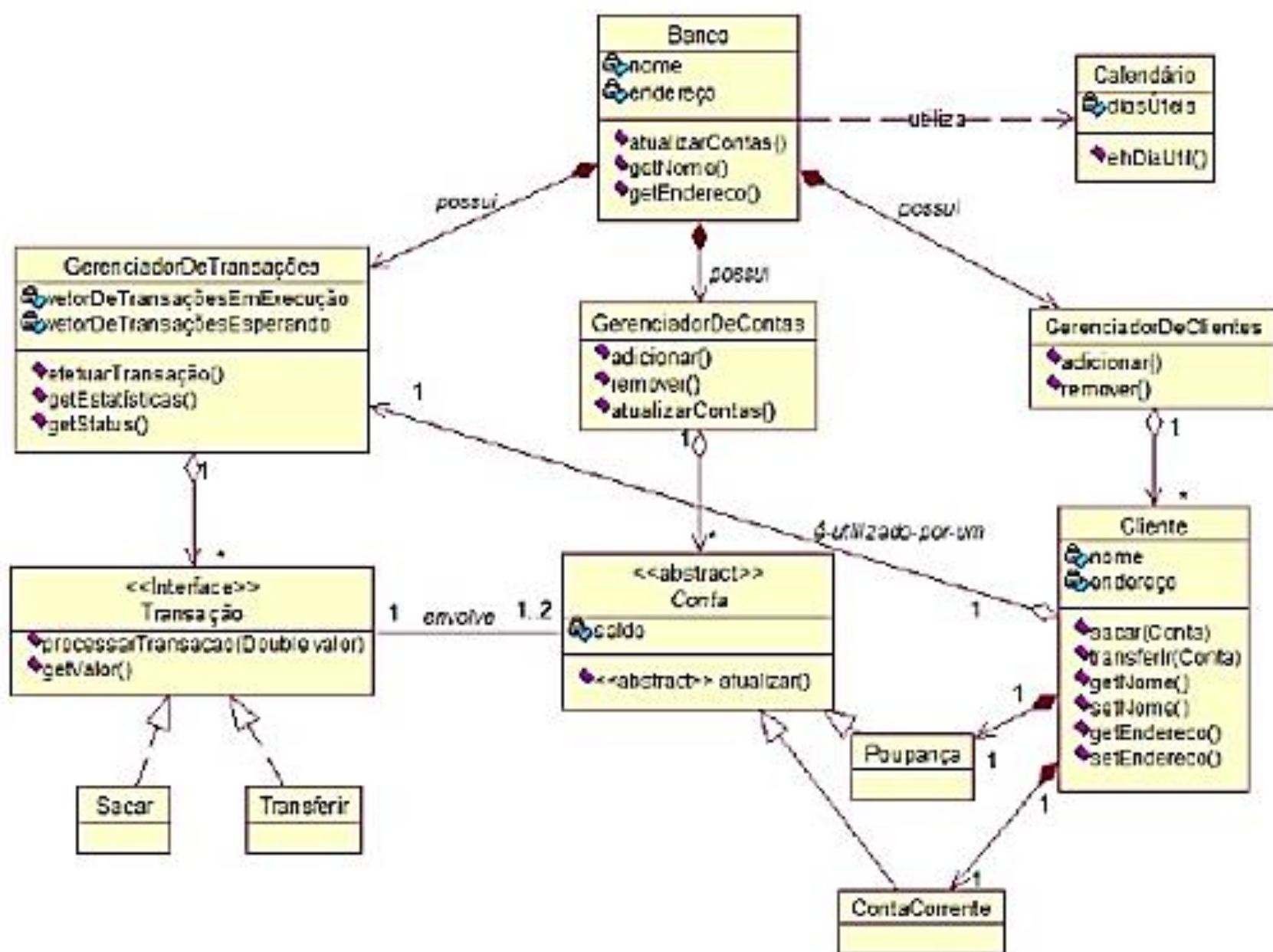


Diagrama de Classes

- ❑ O diagrama de classes é utilizado na construção do modelo de classes desde o nível de análise até o nível de especificação.

Diagrama de Classes

- ☐ Os objetos de uma classes podem se **relacionar** uns com os outros.
- ☐ A existência de relacionamento entre os objetos permitem que eles troquem mensagens uns com os outros.
- ☐ Em última análise, relacionamentos entre objetos permitem que eles colaborem entre si a fim de produzir funcionalidades do sistema.

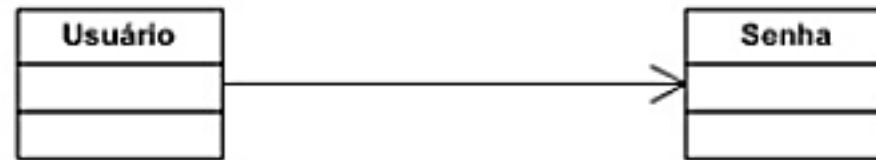
Diagrama de Classes

- ☐ Podemos representar o relacionamento entre os objetos em um Diagrama de Classes.
- ☐ Para isso, existe o elemento na notação conhecido como **associação**.
- ☐ Representa relacionamentos que são formados entre objetos durante a execução do sistema.

Relacionamento

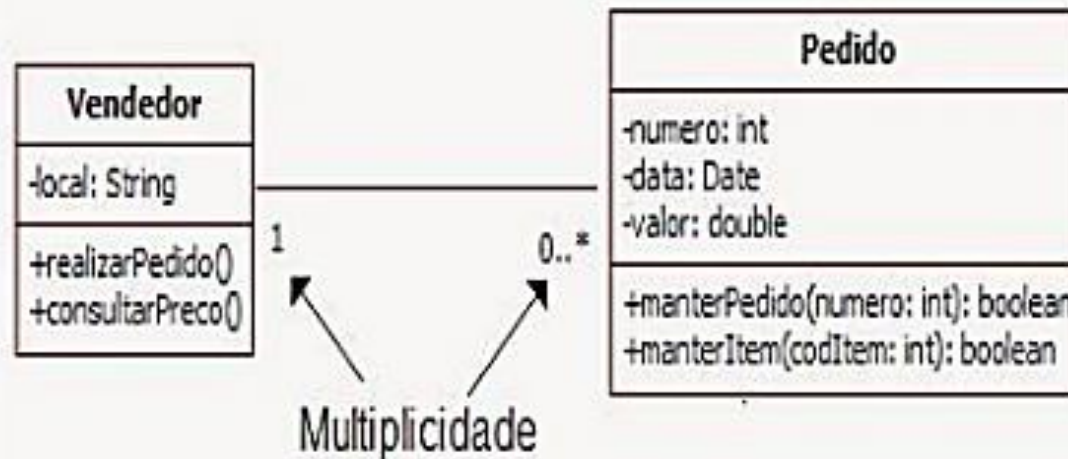
Associações

- Define relacionamento entre as classes



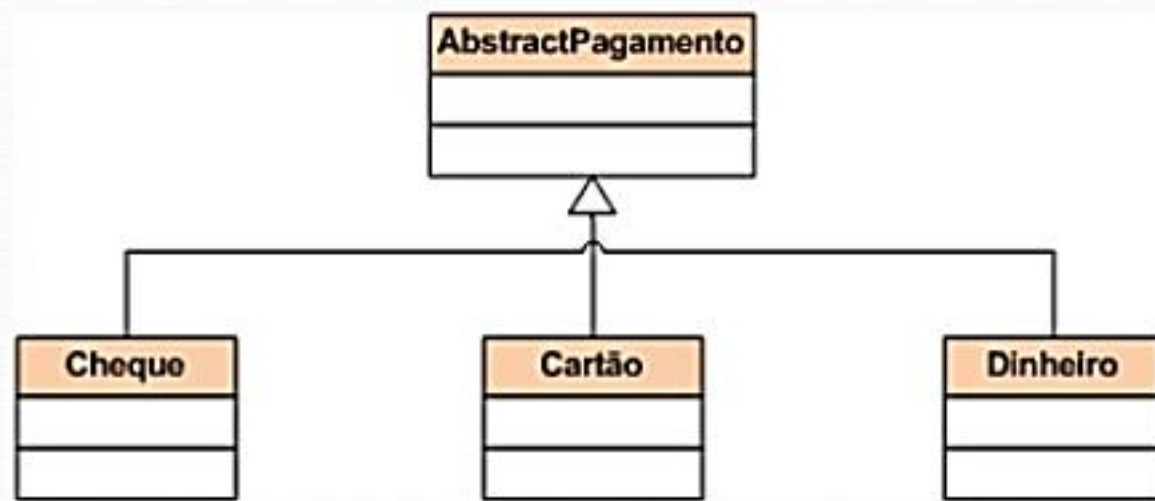
- As associações representam ligações possíveis entre objetos das classes envolvidas na associação.
- Durante a execução do programa, haverá possibilidade da troca de mensagens entre os objetos dessas classes.

Multiplicidade



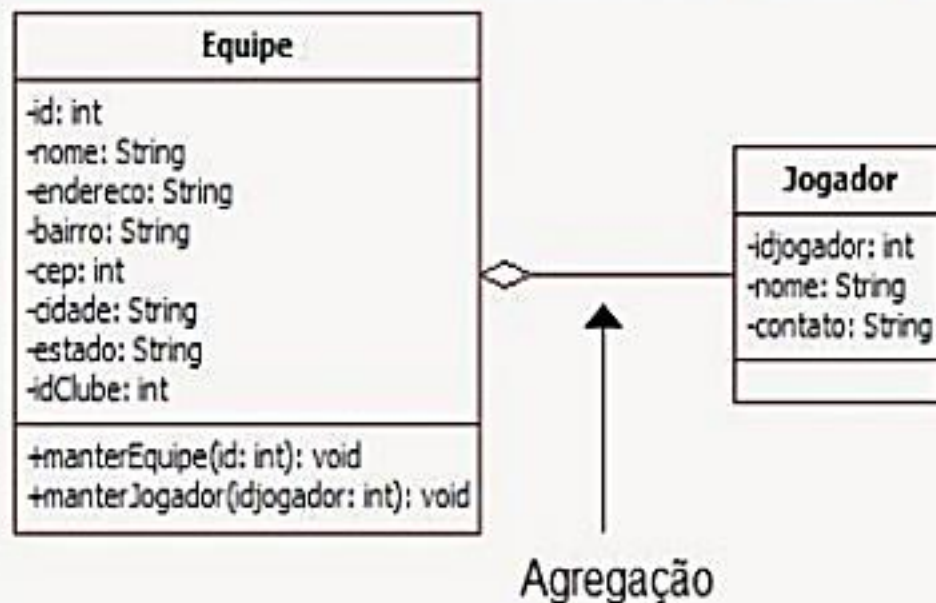
Um objeto para a classe Vendedor que esteja associado a diversos objetos da classe Pedido.

Relacionamento – Generalização



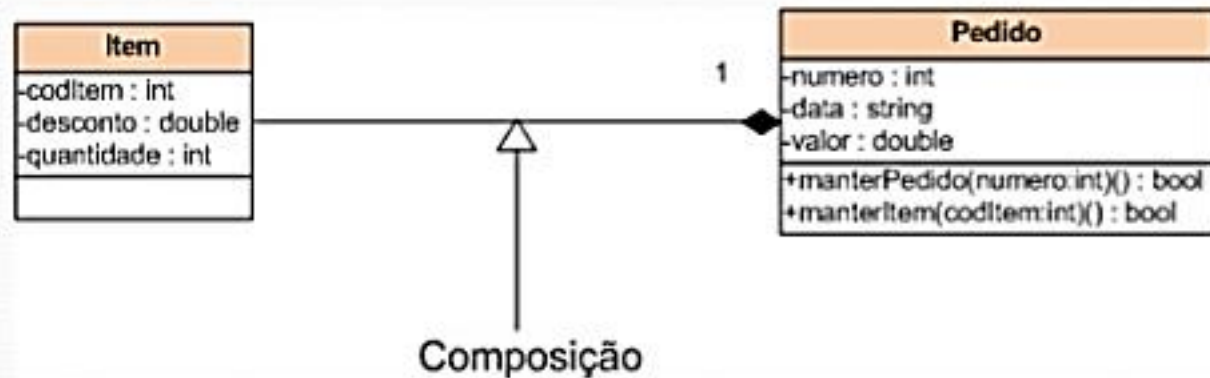
- Na implementação física corresponde a um processo de herança.

Relacionamento - Agregação



- Uma agregação representa um todo que é composto de várias partes;

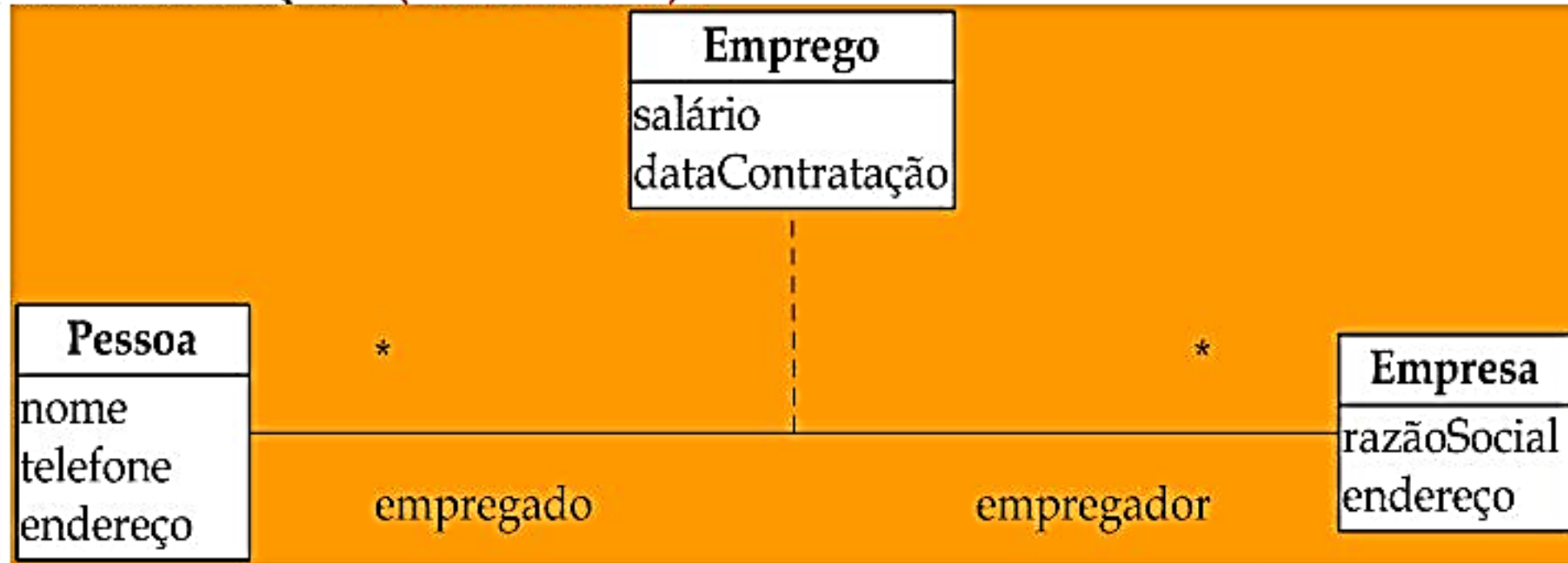
Relacionamento - Composição



- A composição, é diferente da agregação, pois é um relacionamento de contenção. Um objeto CONTÉM outros objetos(elementos). Esses elementos que estão contidos dentro de outro objeto depende dele para existir.

Classe Associativa

- É uma classe que está ligada a uma **associação**, ao invés de estar **ligada a outras classes**.
- É normalmente necessária quando **duas ou mais classes estão associadas**, e é necessário manter informações sobre esta associação (**histórico**).



Exemplo

