

Modelagem de software orientados a objetos

Fatec 2025



O paradigma da orientação a objetos

- Paradigma = forma de abordar um problema
- Princípios:
 - Qualquer coisa é um objeto
 - Objetos realizam tarefas por meio da requisição de serviços a outros objetos
 - Cada objeto pertence a uma classe
 - Uma classe agrupa objetos similares
 - A classe é um repositório para comportamento associado ao objeto
 - Classes são organizadas em hierarquias

A large orange circle on the left side of the slide, partially cut off by the edge.

Pensar orientado a objetos

- Onde você olhar, você vê objetos
 - Pessoas, carros, plantas
- Humanos pensam em objetos
 - Alto nível



Características de Objetos

- Classificação
 - Animados ou inanimados
- Possuem atributos
 - Tamanho, forma, peso, etc
- Exibem comportamentos
 - Uma bola rola, um avião voa
 - Uma pessoa anda, fala, pensa



Desenvolvimento Orientado a objetos

- Análise orientada a objetos
 - Cria um modelo de objetos para o domínio da aplicação
- Projeto orientado a objetos
 - Cria um modelo para implementar os requisitos
- Programação orientada a objetos
 - Implementa o projeto usando uma linguagem de programação



Exemplo da Orientação a objetos

- Para comprar uma pizza VOCÊ pede pelo telefone
- VOCÊ informa ao ATENDENTE seus dados e características da pizza que deseja
- O ATENDENTE atende sua solicitação e comunica o PIZZAIOLO seu pedido
- O PIZZAIOLO faz a pizza e chama o MOTOBÓY
- O MOTOBÓY entrega a pizza para VOCÊ na sua casa

Abstração

- Uma abstração é qualquer modelo que inclui os aspectos mais relevantes e essenciais de alguma coisa
- Permitem gerenciar a complexidade e concentrar a atenção nas características essenciais de um objeto
- Foca no que um objeto faz, antes de definir como implementá-lo

Por que modelar software?

- Os sistemas (mesmos os pequenos) tendem a crescer e nunca são finalizados
- São dinâmicos, em constante mudança
 - Clientes desejam melhorias
 - O mercado muda
 - Governo promulga novas leis, novos impostos
- Portanto, um sistema precisa de documentação detalhada, precisa e atualizada

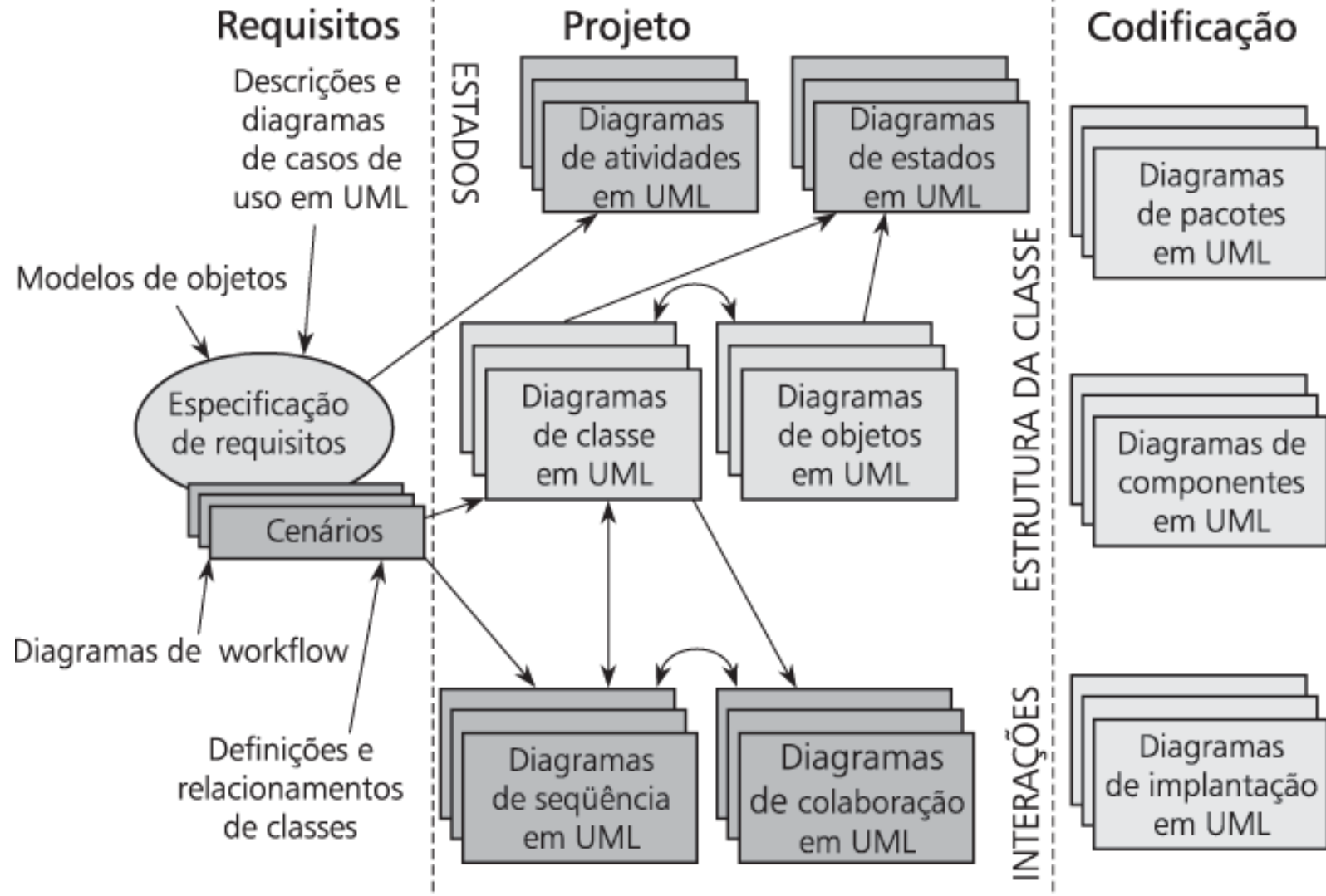


Modelagem do projeto de software

- Após a definição do diagrama de caso de usos e a sucessiva validação pelo cliente é possível dar continuidade no processo de modelagem
- Não há uma sequência lógica para a elaboração dos diagramas UML
- Também não há uma quantidade definida de cada um



Como a UML apóia o Desenvolvimento



OO: Classe de objetos



- Uma classe representa uma categoria, e os objetos são os membros (ou exemplos) dessa categoria
- Não trabalhamos diretamente com as classes e sim, com suas instâncias

OO: Atributos ou propriedades

- Representam características de uma classe
- Contém duas informações: nome e tipo de dado



OO: métodos,
operações,comportamento



- Representa uma atividade que um objeto de uma classe pode executar
- Conjunto de instruções que são executadas quando um método é chamado

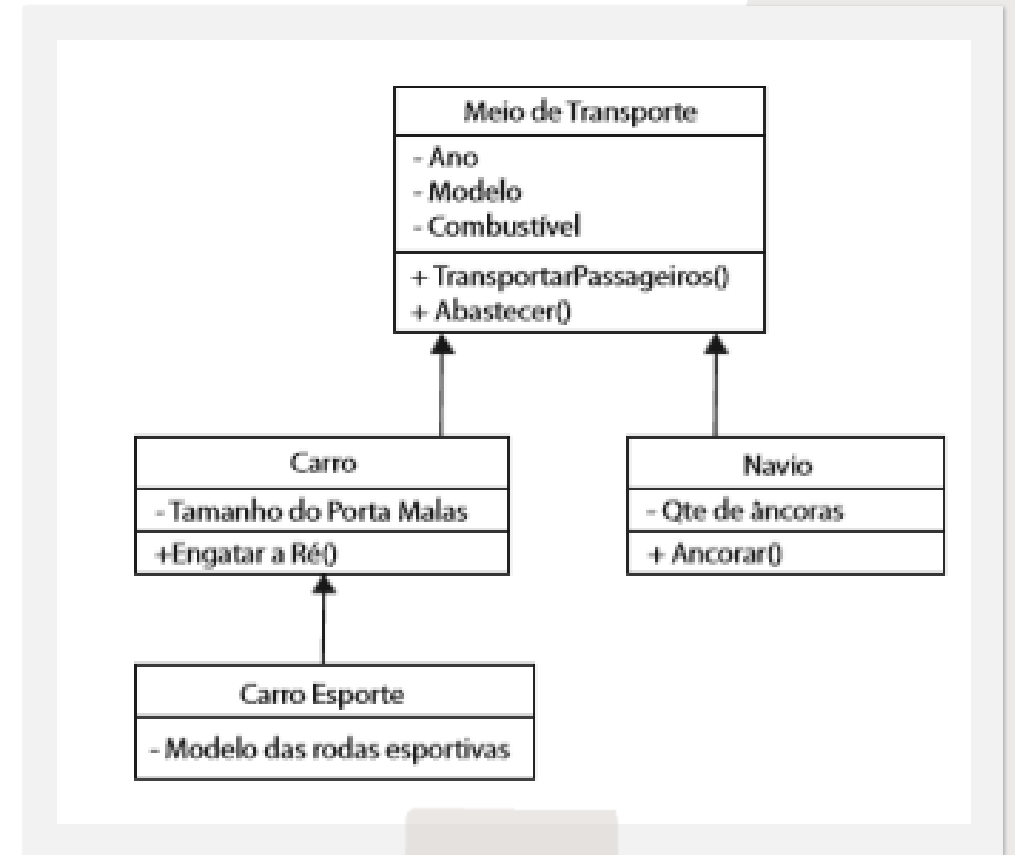
Visibilidade



- Indica o nível de acessibilidade de um determinado atributo ou método
- (-) somente os objetos da classe detentora do atributo ou método poderão enxergá-lo ou utilizá-lo
- (#) determina que, além dos objetos da classe detentora, também os objetos de suas subclasses poderão ter acesso ao mesmo
- (+) visibilidade pública para qualquer objeto

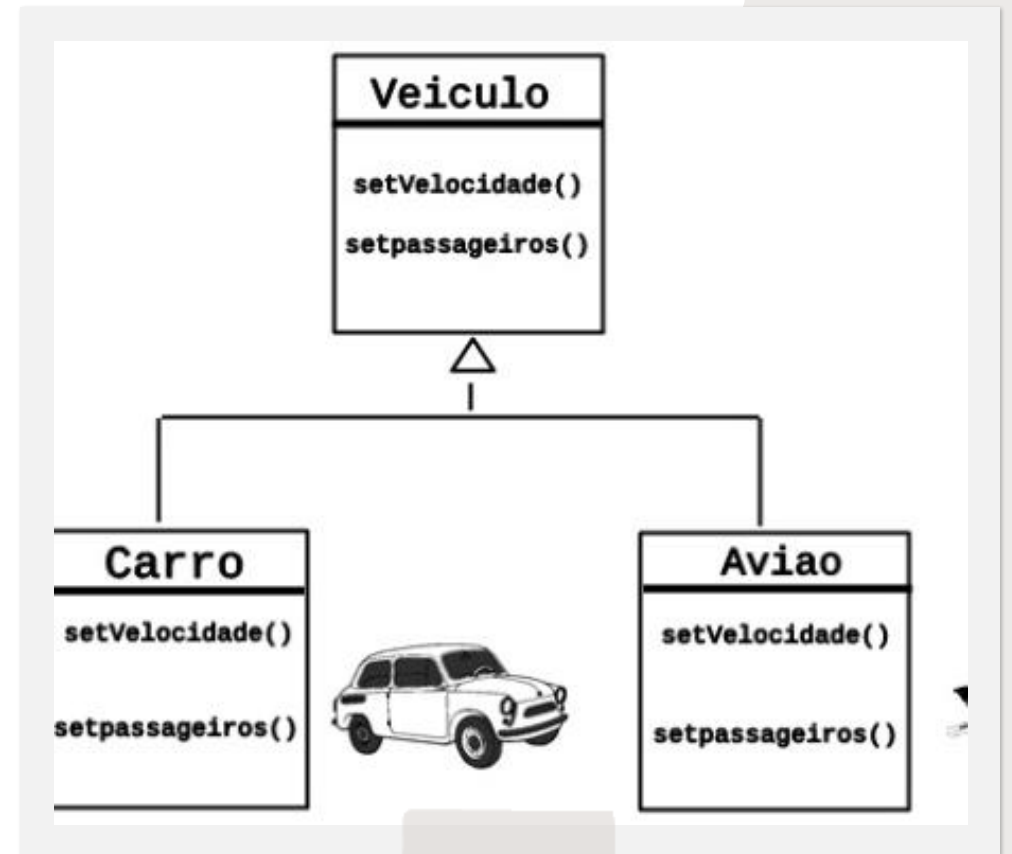
Herança

- Permite o reaproveitamento de atributos e métodos
- Superclasse, classe-mãe = contém classes derivadas
- Subclasse, classe-filha = herdam atributos e métodos
- Permite trabalhar com especializações



polimorfismo

- Associado à herança
- Redecaração (como mesmo nome) de métodos previamente herdados
- Embora semelhantes, diferem de alguma forma da implementação usada na superclasse



Casos de uso

- Definem os serviço, tarefas ou funções do sistema
- Nomes indicam ação (verbos)
- Ex: cadastrarVenda, Sacar, ConsultarFilme



Casos de uso

- Capturam requisitos do sistema
- Necessidades (serviços, tarefas, funcionalidades)
- Casos de uso primários:
 - Processo importante
 - Requisitos funcionais
- Casos de uso secundários
 - Processos periféricos
 - Ex: manutenção em cadastros
- Eventualmente, pode ser um formulário, pois um UC pode abranger vários formulários

Diagramas de Caso de Uso

- Parte mais importante
- Único instrumento que acompanha o software do seu início até a sua conclusão
- Uma macro atividade que encerra diversas tarefas ou atividades
- Uma representação descrita de variadas ações para a realização dessa macro atividade



Diagramas de caso de uso

- Possibilita a compreensão do comportamento externo do sistema (em termos de funcionalidade)
- Mais abstrato, flexível e informal dos diagramas UML
- Visão externa geral das funcionalidades, sem se preocupar com sua implementação
- Recomendável: que seja apresentado junto com um protótipo aos clientes
- Concentra-se em 2 itens principais: atores e casos de uso

Diagrama de casos de uso

- Adota uma linguagem simples, mais flexível e informal
- Permite a compreensão do comportamento externo do sistema por qualquer stakeholder
- Apresenta o sistema através da perspectiva do usuário
- Usado no início da modelagem, durante a especificação de requisitos
- Técnica de elicitação de requisitos

Diagrama de caso

- Apresenta uma visão geral das funções e serviços do sistema
 - Define o que o sistema faz
 - Não se preocupa em como o sistema faz
- Indica uma funcionalidade que o sistema deve oferecer
 - AbrirConta, Sacar, VerificarSaldo, etc.

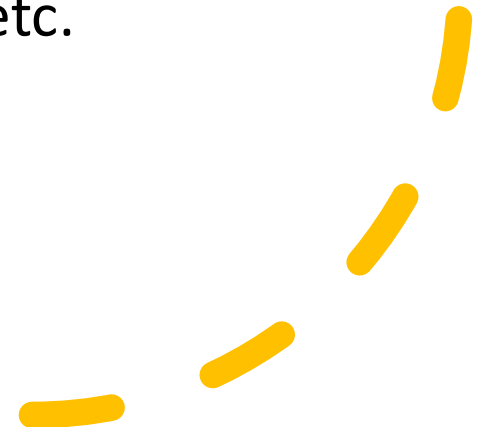
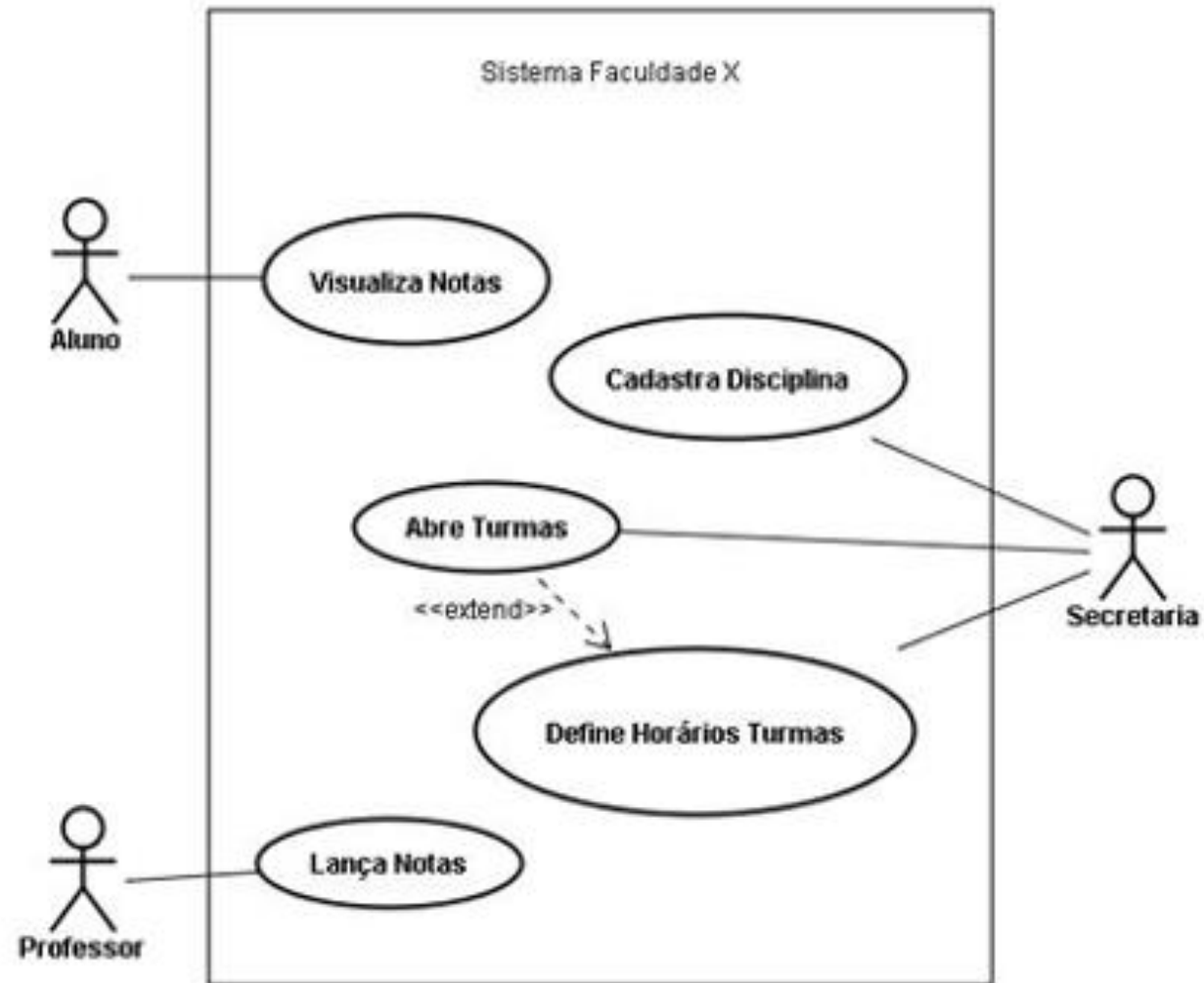
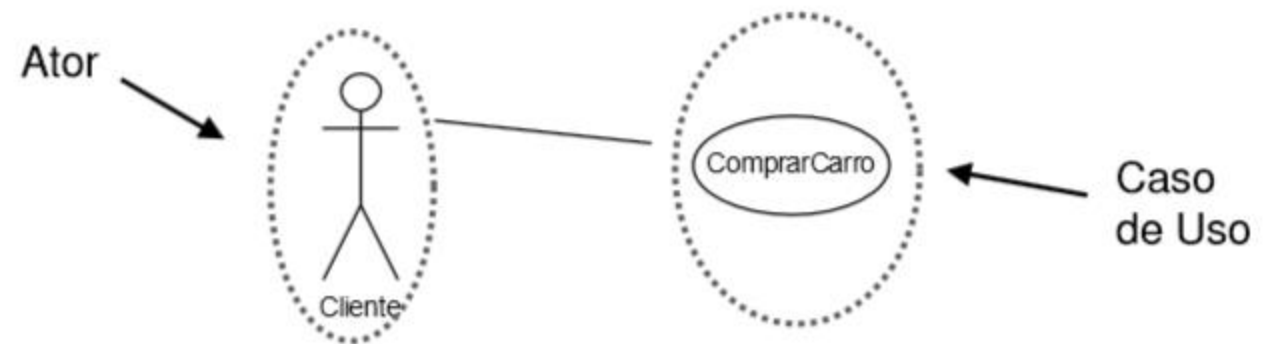


Diagrama de caso de uso



Componentes do Diagrama de caso de uso

- Atores: executam uma funcionalidade
- Casos de uso: informam qual a funcionalidade
- Relacionamentos: identificam como os atores se conectam aos casos de uso de uso
- Fronteiras do sistema ou assunto, casos de uso, atores, relacionamento
- Podem conter notas e pacotes



Representação de casos de uso

- Representados por uma elipse e seu texto descreve a funcionalidade
- Descrição curta e direta



Atores

- Representam os papéis desempenhados pelos diversos usuários que poderão utilizar os serviços e funções do sistema
- Pode representar algum hardware ou software



Atores

- Representam os papéis desempenhados pelos diversos usuários
- Os casos de uso descrevem as interações entre o sistema e estes atores
- Atores podem ser:
 - Pessoas que interagem com o sistema
 - Hardware especial que dispara uma interação
 - Um software que se comunica com o sistema

Atores

- Ator é algo (usuário, software ou hardware) que não faz parte do sistema mas que interage com ele em algum momento
- Representação: homem palito + papel representado
- Ex: Pessoas (empregado, cliente, gerente, vendedor, etc)
- Organizações (empresa Fornecedora, Agencia De Impostos, Administradora De Cartões, etc)
- Equipamentos (Leitor De Código De Barras, Sensor, etc)



documentação ou descrição do caso de uso

- Linguagem natural
- Elementos que podem ser documentados:
 - Quais atores interagem com o caso de uso
 - Quais etapas devem ser executadas
 - Quais parâmetros devem ser fornecidos
 - Quais restrições e
 - validações, etc
- Varias formas podem ser usadas para esta documentação: descrição passo a passo, tabelas, fluxogramas, Sequencia Típica, e principalmente, CENÁRIOS

Documento de caso de uso

- Fluxo principal
 - Apresenta as ações que devem normalmente ser realizadas quando o serviço for realizado
 - Ações do ator
 - Ações do sistema
- Fluxo alternativo
 - Situações que fogem da situação ideal
 - Opções que podem ser executadas conforme uma condição
- Fluxos de exceção
 - Situações em que o fluxo não possa ser concluído

CENÁRIOS

- Descrevem uma situação de uso do sistema
- Inclui: nome, ator, pré-condição, fluxo normal, fluxo alternativo e pós-condição

Caso de Uso:	Visualizar Avaliações
Sumário:	Aluno visualiza avaliação que recebeu – notas e frequência – nas turmas de um semestre letivo.
Ator Primário:	Aluno
Ator Secundário:	Professor
Pré-Condições:	O aluno está identificado pelo sistema.
Fluxo Principal:	<ol style="list-style-type: none">1. O aluno solicita a visualização das avaliações para as disciplinas em que participou.2. O sistema exibe os semestres letivos nos quais o aluno se inscreveu em pelo menos uma disciplina.3. O aluno seleciona os semestres letivos cujas avaliações deseja visualizar.4. O sistema exibe uma lista de avaliações agrupadas por semestres letivos selecionados e por turma.5. O aluno visualiza as avaliações e o caso de uso termina.
Fluxo de Exceção (2):	<p>Aluno sem inscrição</p> <ol style="list-style-type: none">1. Não há semestre letivo no qual o aluno tenha participado em alguma disciplina.2. O sistema reporta o fato e o caso de uso termina.
Pós-Condições:	O aluno obteve as avaliações que desejava visualizar.

Cenário

- Nome do cenário: Sacar
- Ator: cliente
- Pré-condição: conta e senha validadas
- Fluxo normal:
 - 1) informar o valor do saque
 - 2) confirmar dados e operação
 - 3) debitar valor da conta do cliente
- Fluxo alternativo
 - Apresentar aviso ao cliente
- Pós-condição: valor sacado é debitado do saldo do cliente

Potenciais objetos:
CONTA
CLIENTE

Potenciais atributos:
VALOR
DADO

Potenciais métodos:
EntrarComValor
DebitarValor

Documento de caso de uso

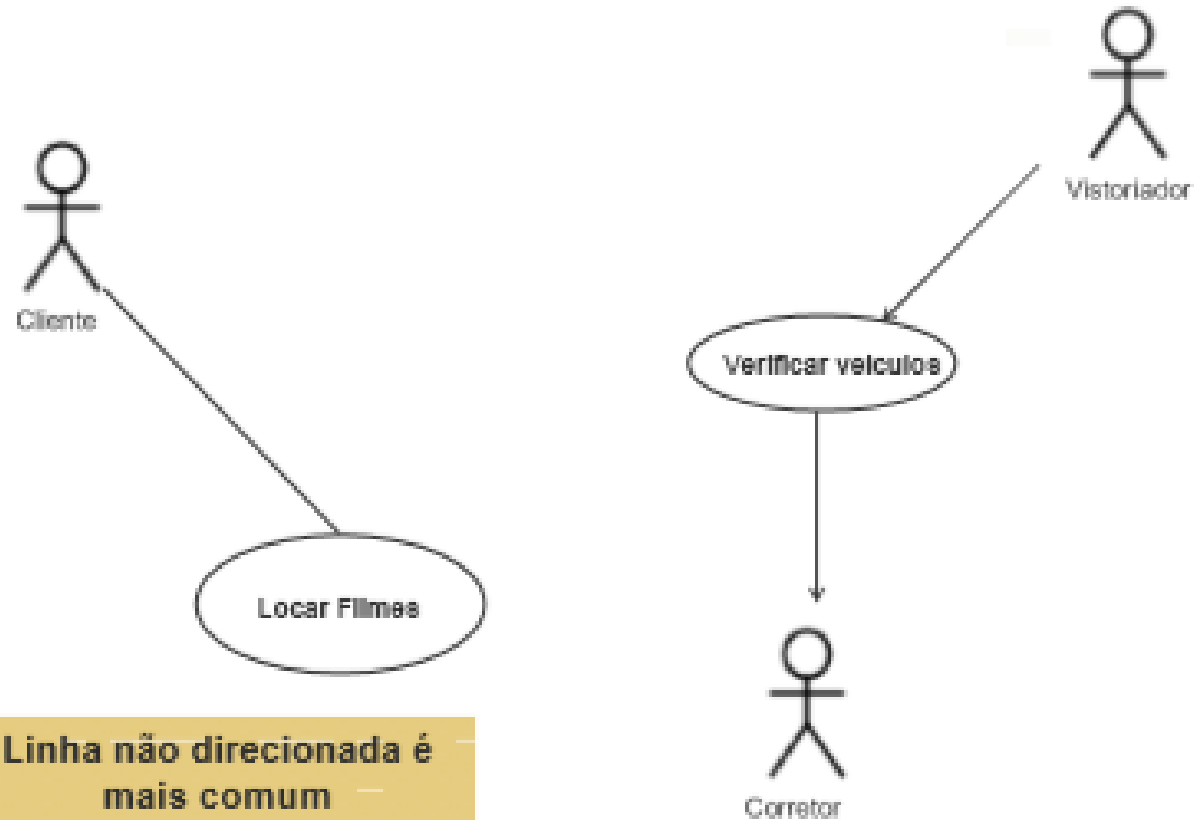
Campo	Descrição
NOME	
CASO DE USO GERAL	Geral ou especializado? Caso a partir do qual ele foi derivado.
ATOR PRINCIPAL	Ator que mais itera ou o mais interessado nos resultados
ATORES SECUNDARIOS	Interagem menos ou não tem interesse nos resultados.
RESUMO	
PRÉ-CONDIÇÕES	Ex: “o pedido de abertura precisa ser aprovado”
PÓS-CONDIÇÕES	Ex: “é necessário realizar um depósito inicial”

Associações

- Representam as interações ou relacionamentos entre os atores e casos de uso
- Demonstra que o ator utiliza a funcionalidade
- Requisita a execução
- Recebe resultado produzido pelo UC a pedido de outro ator

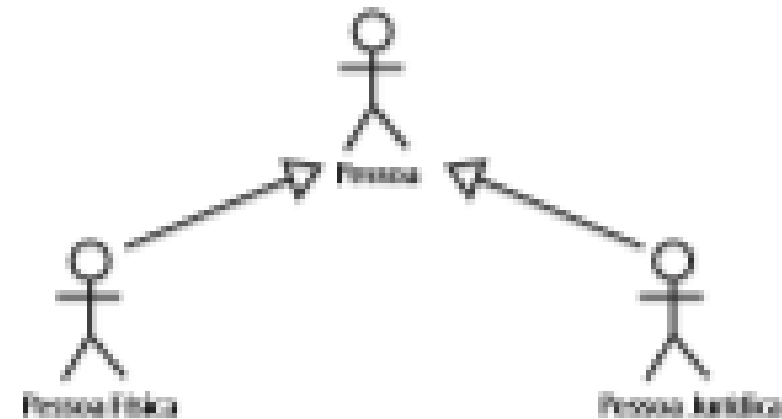


Associação



Generalização/Especialização

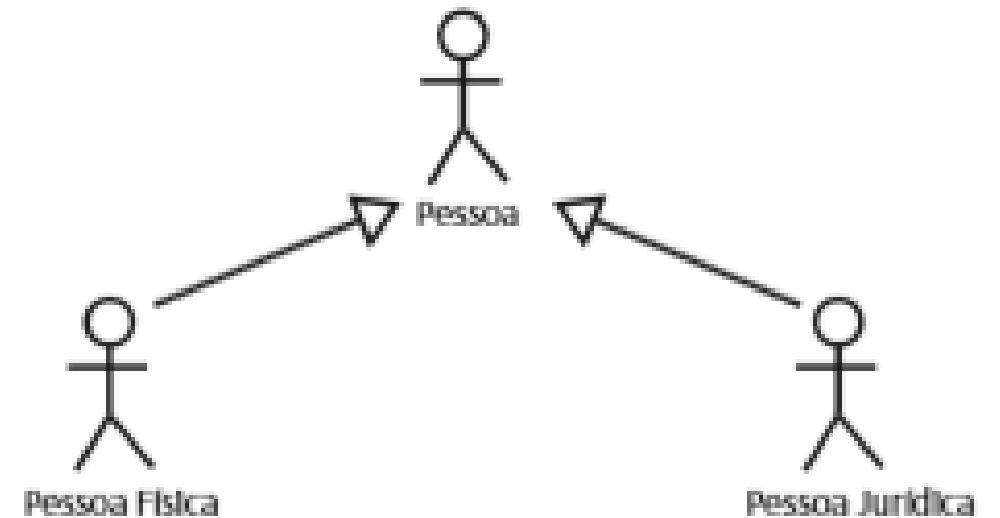
- Relacionamento entre casos de uso na qual existem 2 ou mais UC com características semelhantes
- Caso de uso geral que descreve características compartilhadas



Especialização / Generalização

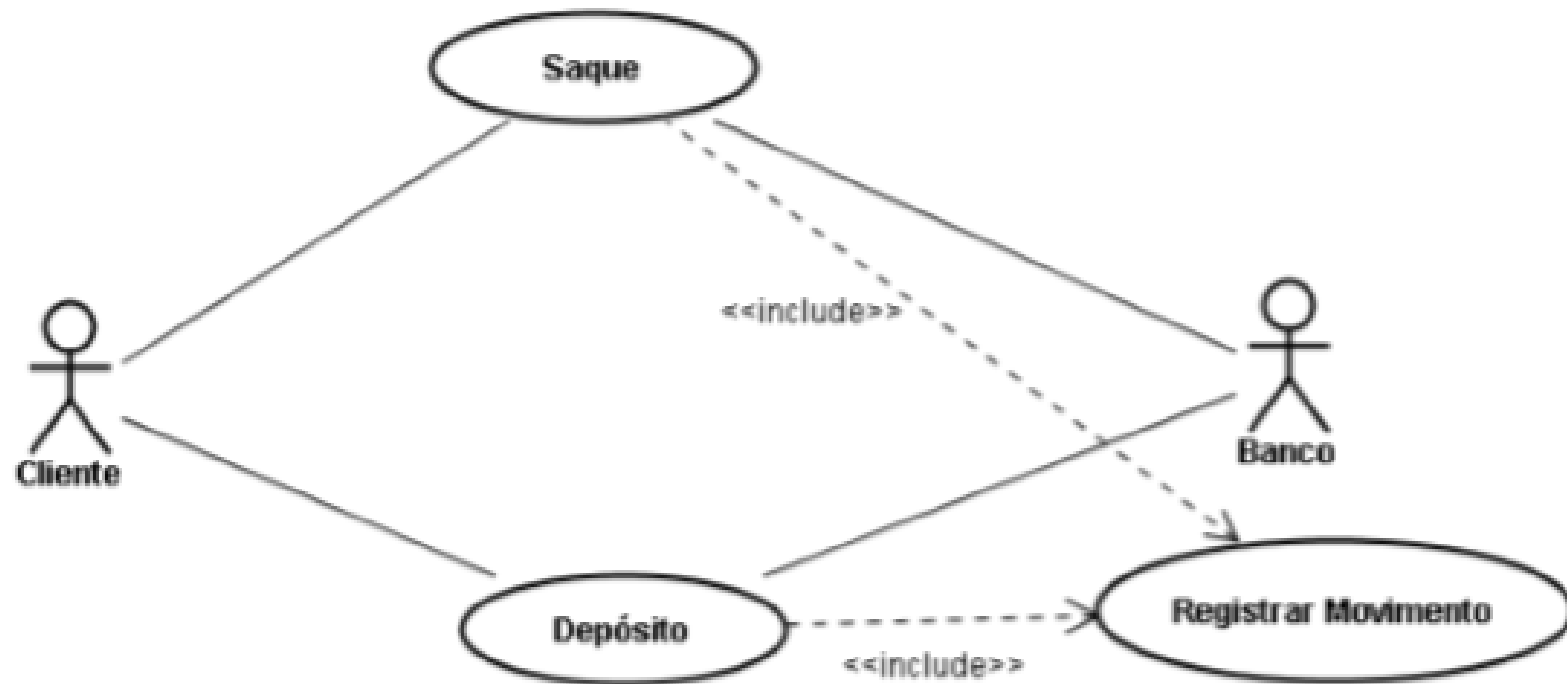
- Acontece quando 2 ou mais casos de uso possuem características semelhantes
 - Reuso
- O caso de uso geral que descreve as características compartilhadas por todos os outros Casos de Uso envolvidos, cuja documentação conterá somente as características peculiares de cada um
- Não há a necessidade de colocar a mesma documentação para todos os casos de uso envolvidos, pois toda estrutura de um caso de uso generalizado é herdada pelos casos de uso especializados

Especialização/Generalização



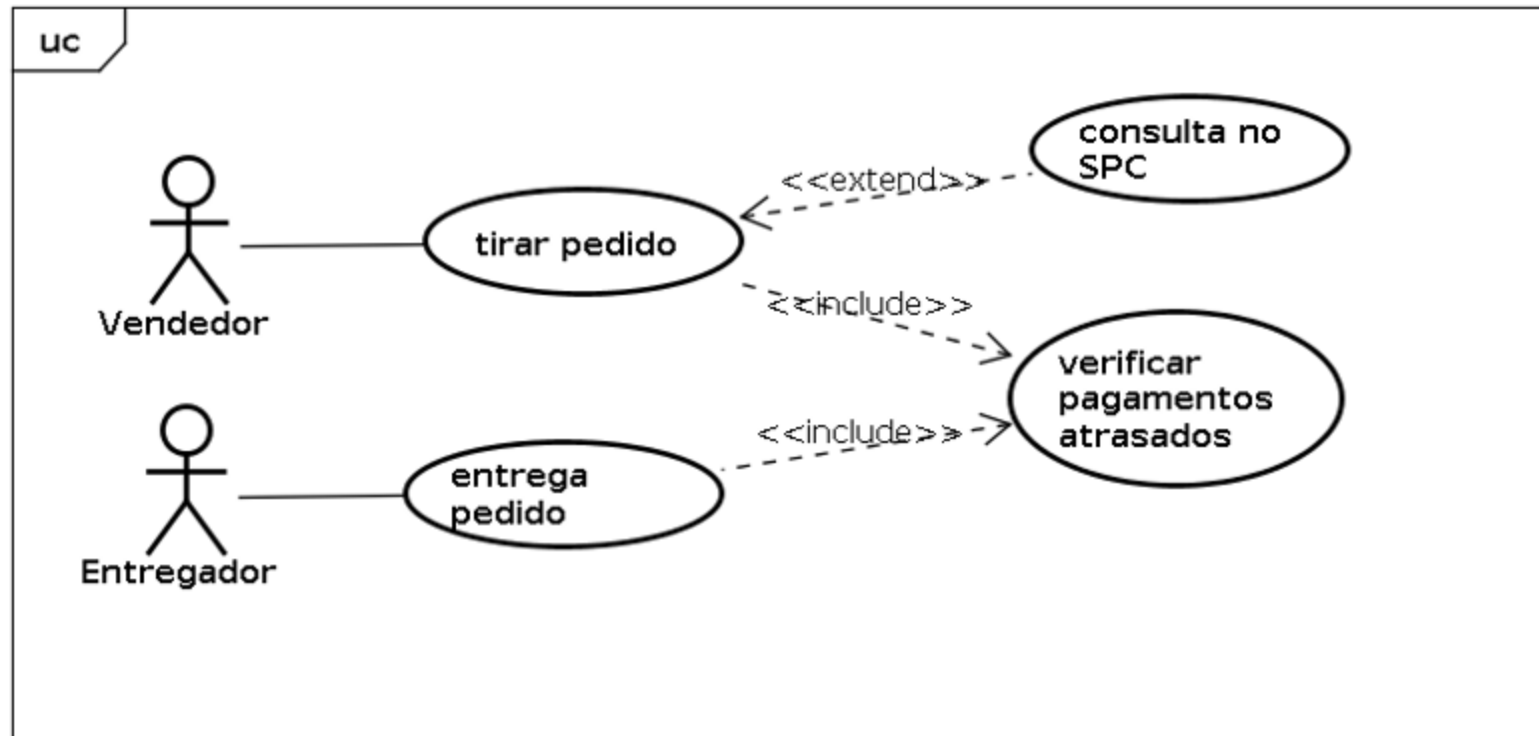
Inclusão <<include>>

- Utilizada quando existe um serviço, situação ou rotina comum a mais de um caso de uso
- Evita a descrição de uma mesma sequência de passos em diversos casos de uso
- Quando um caso de uso possui um relacionamento de inclusão com outro, a execução do primeiro obriga a execução do segundo



Inclusão

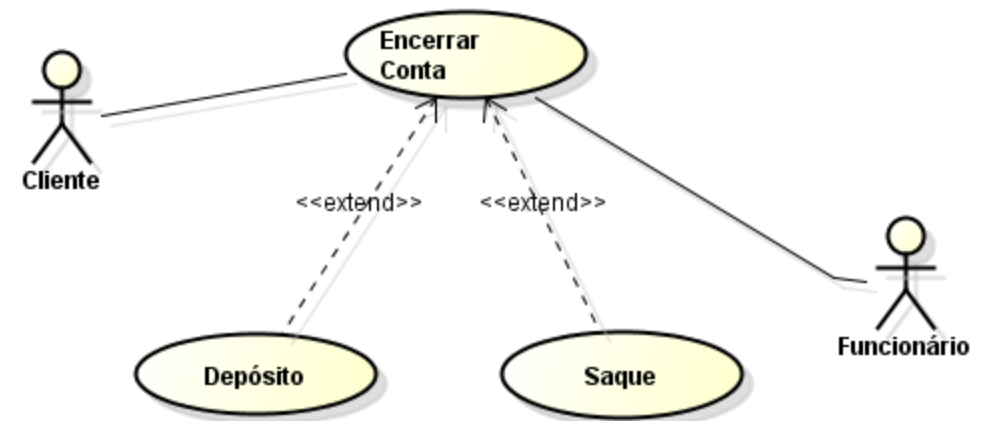
- Quando existe um cenário, situação ou rotina comum a mais de um caso de uso
- Indicam uma obrigatoriedade. Quando um UC tem um relacionamento de inclusão com outro, a execução do primeiro obriga também a execução do segundo
- Comparação: chamada de subrotinas



Extensão

- Descrevem cenários opcionais de um caso de uso
- Descrevem cenários que apenas ocorrerão em uma situação específica se determinada condição for satisfeita
- Um teste determina se é necessário executar o caso de uso estendido
- Representam eventos que não correm sempre

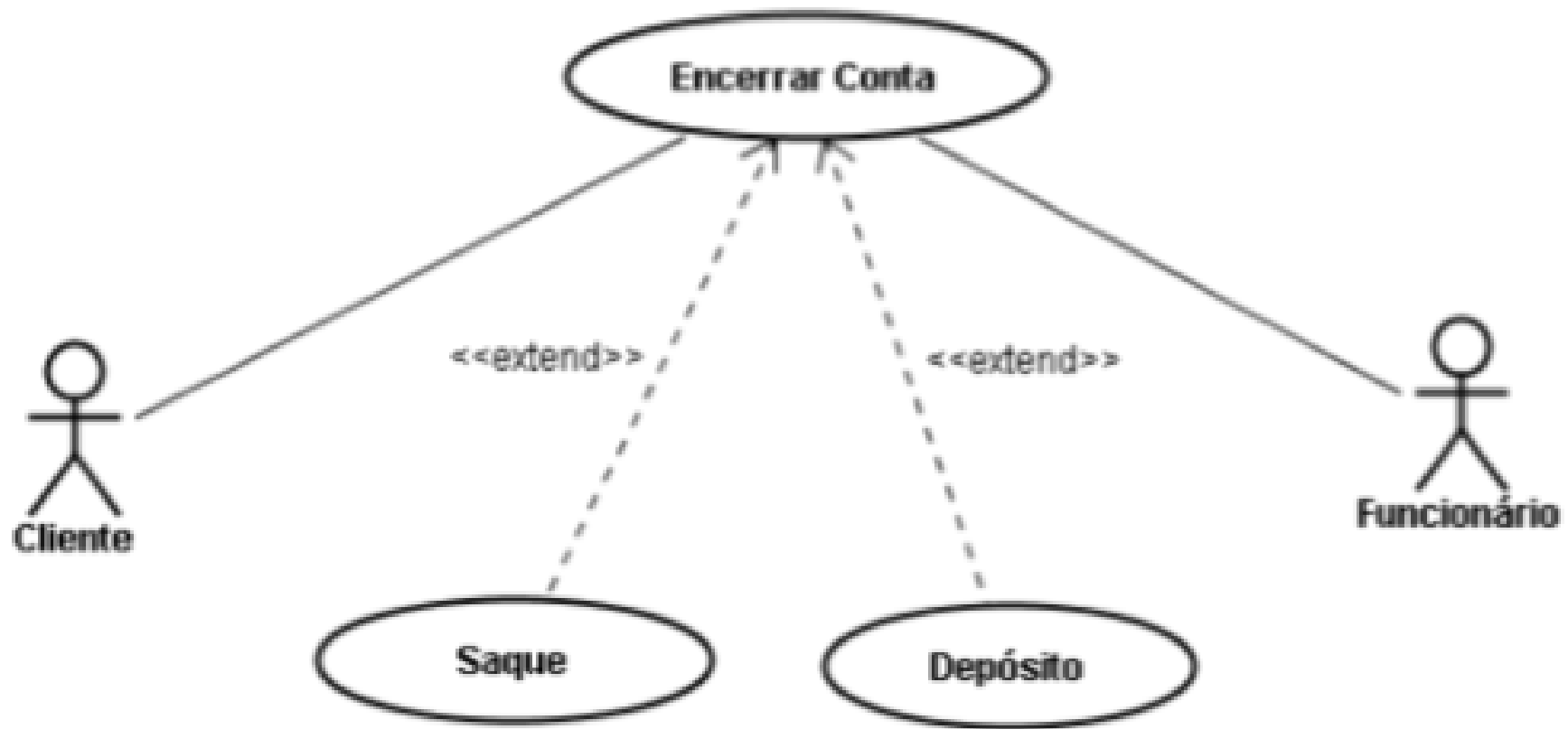
Condição: o saldo deve estar zerado



Extensão de caso de uso <<extend>>

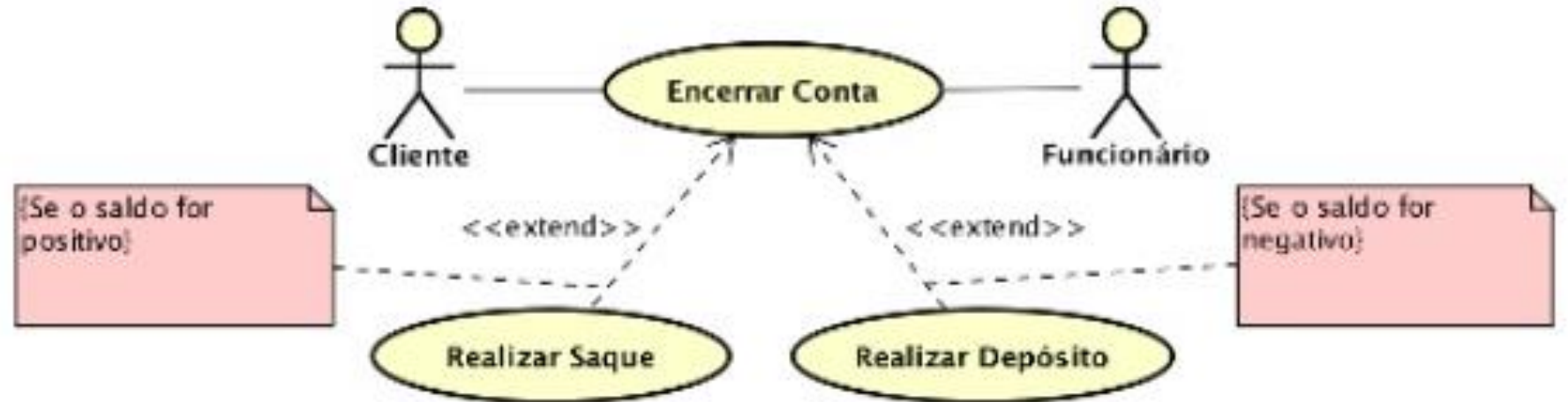
- Utilizado em funcionalidades opcionais
 - Ex: cenários que somente acontecerão em uma situação específica, se uma determinada situação for satisfeita
- Pode necessitar de um teste para determinar se o caso de uso será estendido

<<extend>>



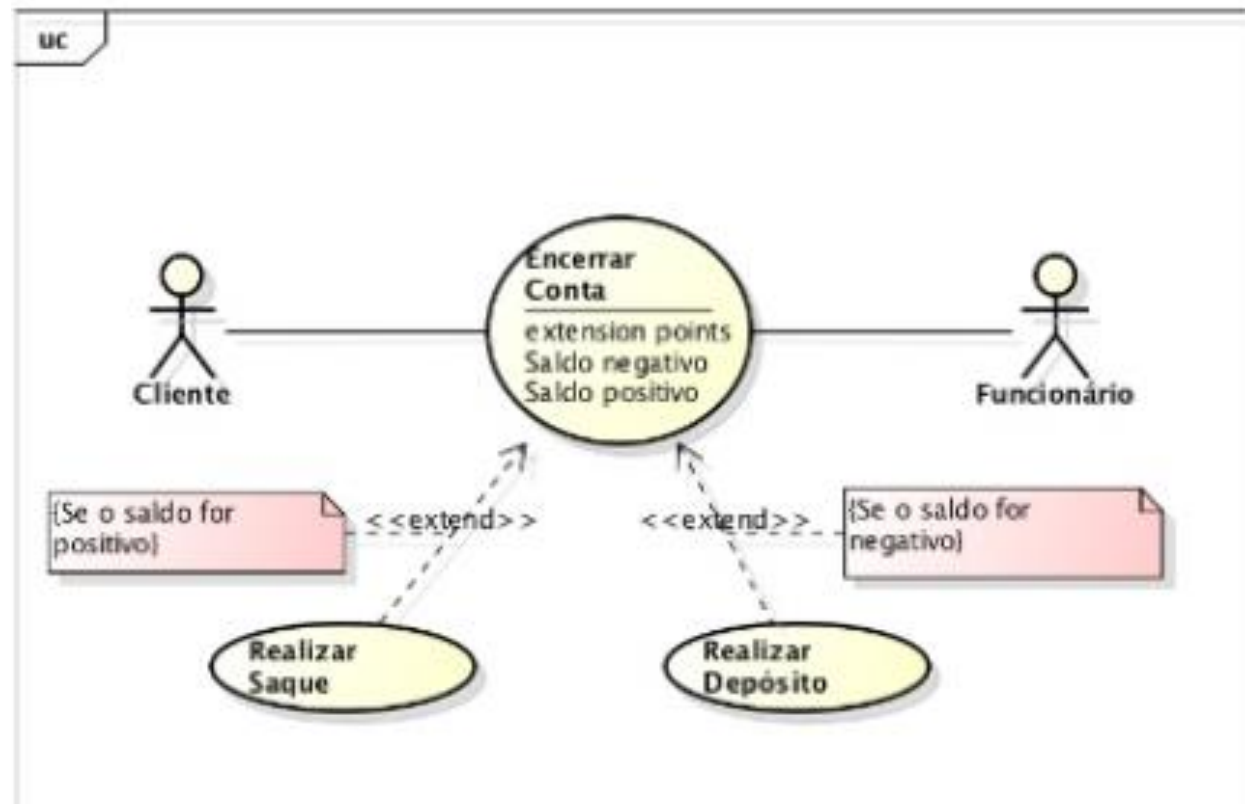
Restrições em extensões

- São textos entre chaves e utilizados para definir validações, consistências, condições, etc que devem ser aplicadas



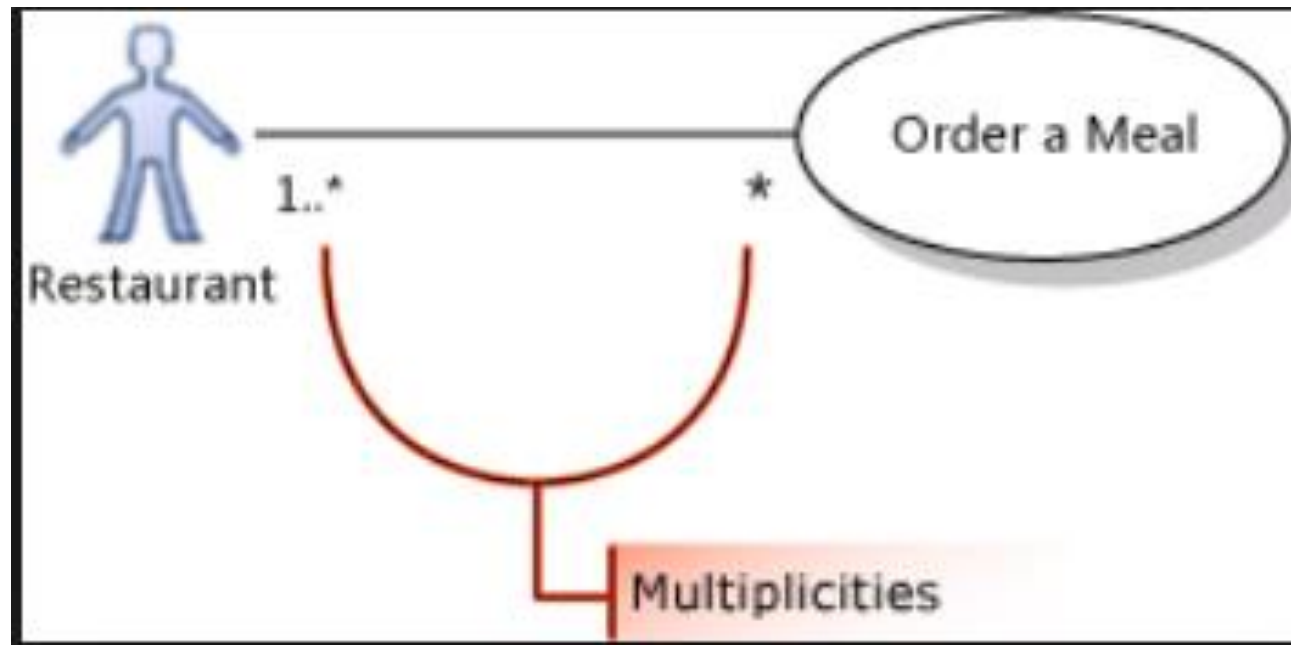
Pontos de extensão

- Identifica um ponto no comportamento de um UC a partir do qual esse comportamento pode ser estendido pelo comportamento de outro UC, se a condição para isso ocorrer



Multiplicidade

- Especifica o número de vezes que um ator pode utilizar um determinado UC



Fronteira do sistema

- Classificador que contém um conjunto de casos de uso
- Identifica um subsistema ou mesmo um sistema completo

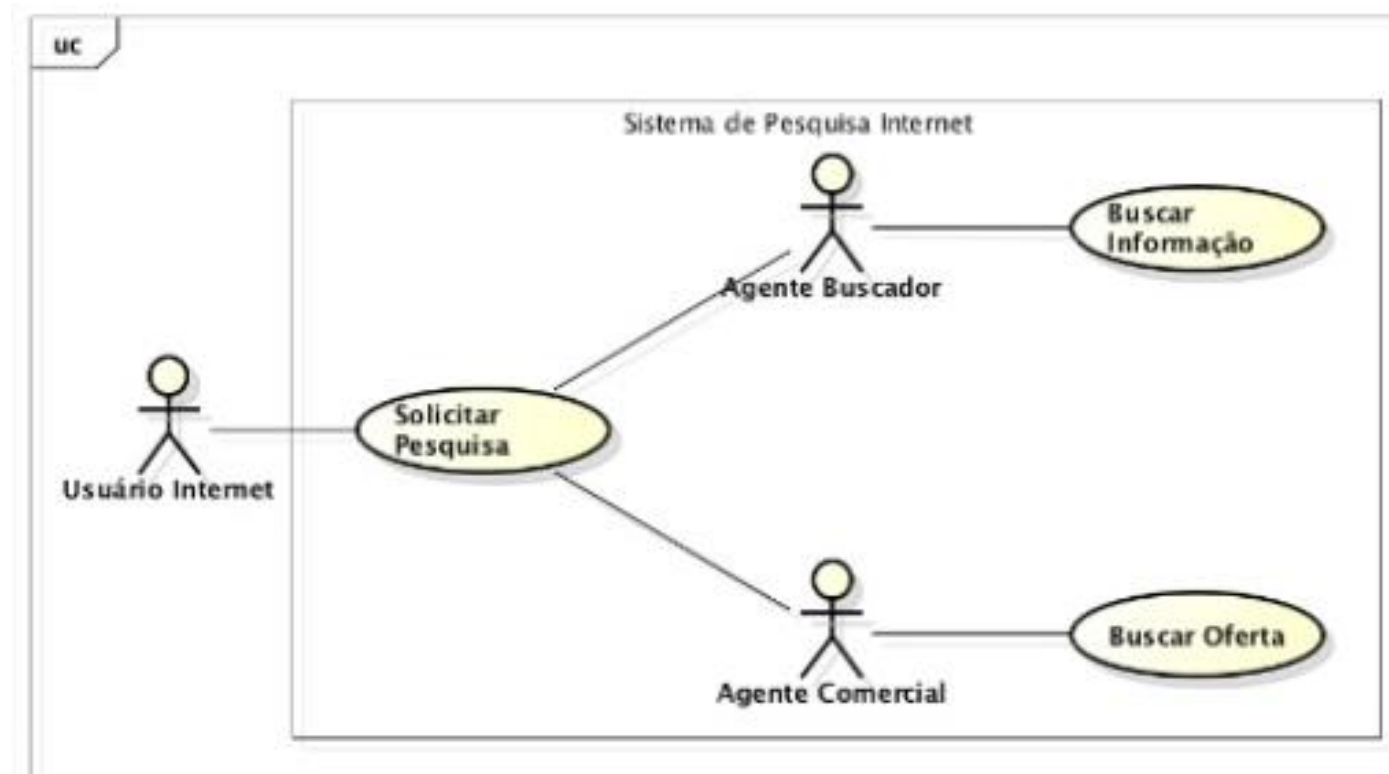


Diagrama de Caso de Uso

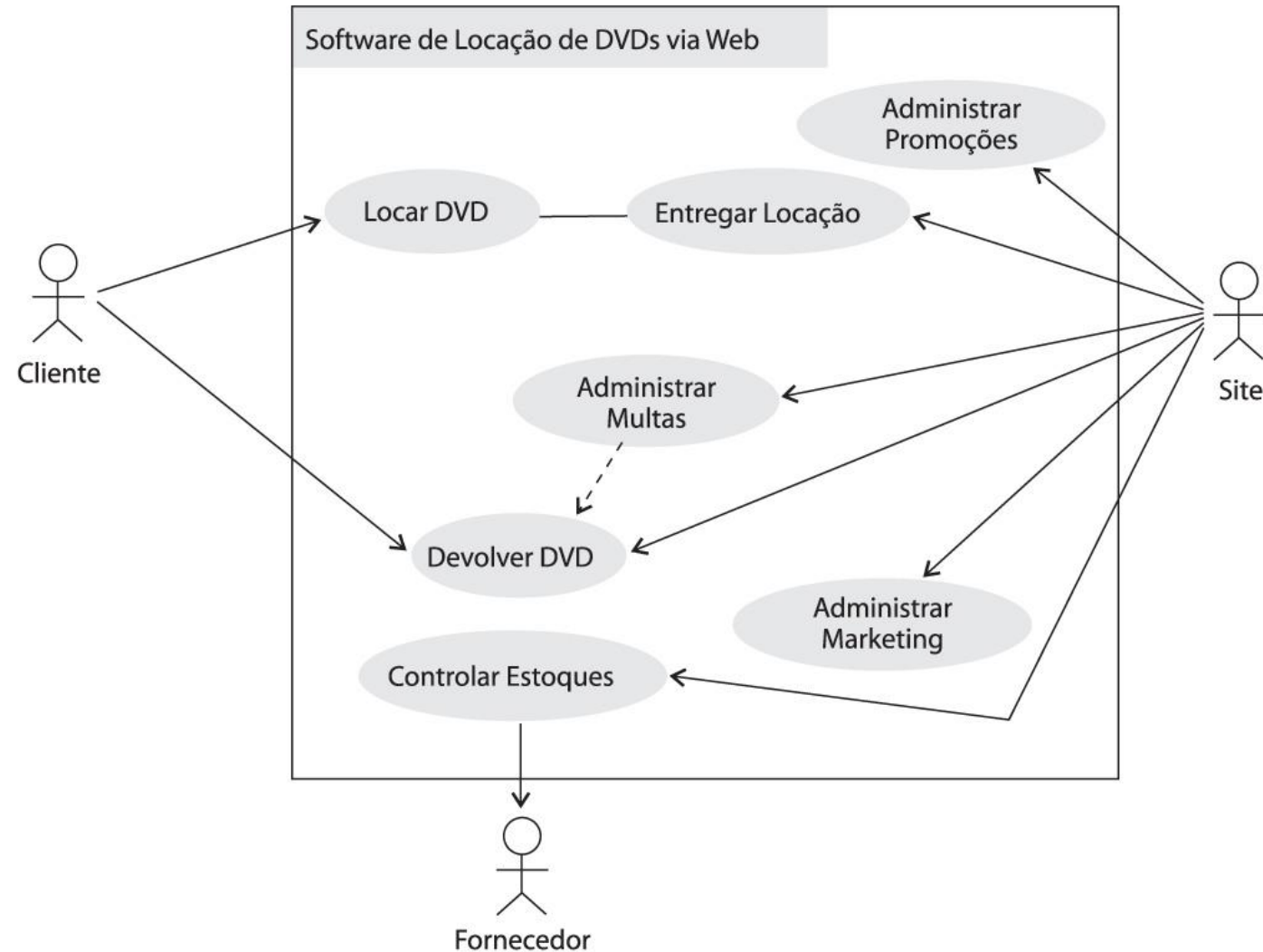






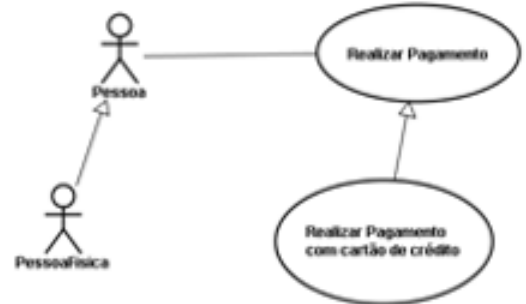


Figura 2.2 Caso de Uso relativo ao documento Visão.

Resumo: Casos de uso

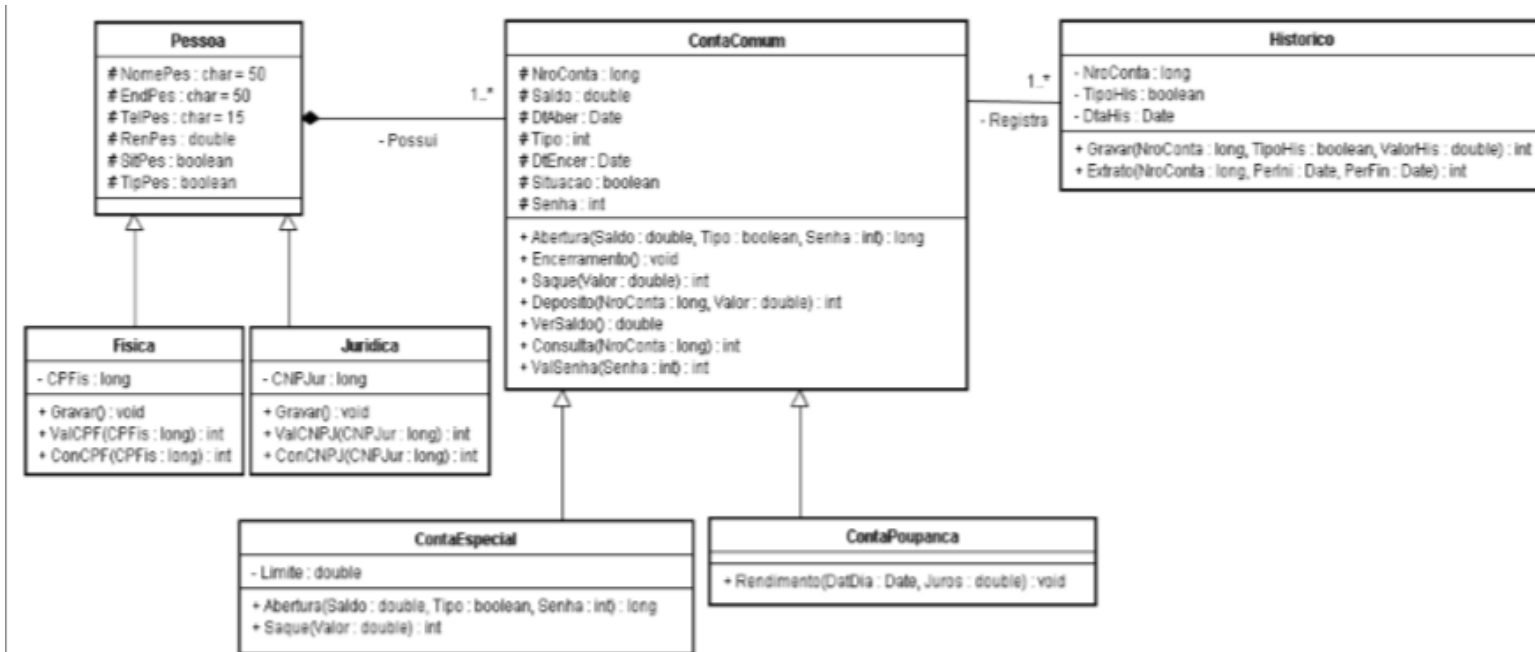
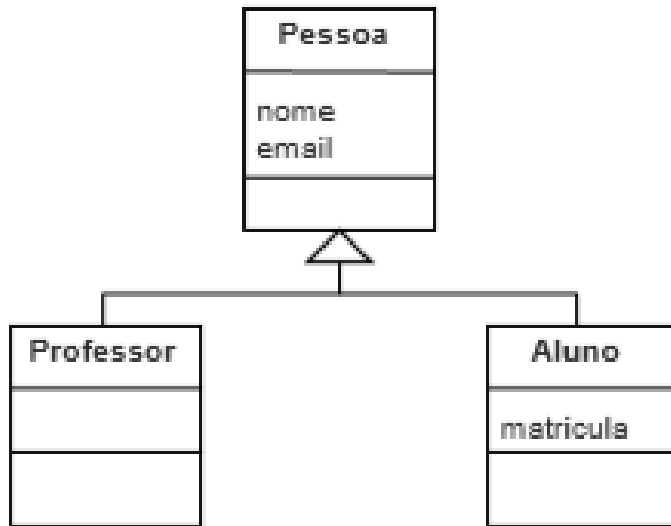
Nome	Representação
Fronteira do Sistema	<p>Elemento gráfico: retângulo que agrupa os casos de uso e os relacionamentos</p> 
Ator	<p>Elemento gráfico: boneco palito</p> 
Caso de Uso	<p>Elemento gráfico: elipse</p> 

Resumo: Casos de uso

Nome	Representação
Relacionamento de Comunicação	Elemento gráfico: linha simples 
Relacionamento de Inclusão – somente entre casos de uso	Elemento gráfico: seta tracejada com estereótipo: <<include>> 
Relacionamento de Extensão – somente entre casos de uso	Elemento gráfico: seta tracejada com estereótipo: <<extend>> 
Relacionamento de Generalização	Elemento gráfico: seta fechada 

Classe

- Descrição de um conjunto de objetos que compartilham os mesmos atributos, operações e relacionamentos
- Classes são matrizes de objetos e descrevem um conceito abstrato do domínio do problema



Diagramas de classe

- Considerado o mais importante e mais utilizado
- Serve de apoio para os demais diagramas
- Exibe conjunto de classes, interfaces e colaborações
- Modelagem da visão estática
- Oferece suporte aos requisitos funcionais
- Representam os dados (atributos) e funções (métodos) do sistema
- Representam a forma como os dados são armazenados

Diagrama de Classes

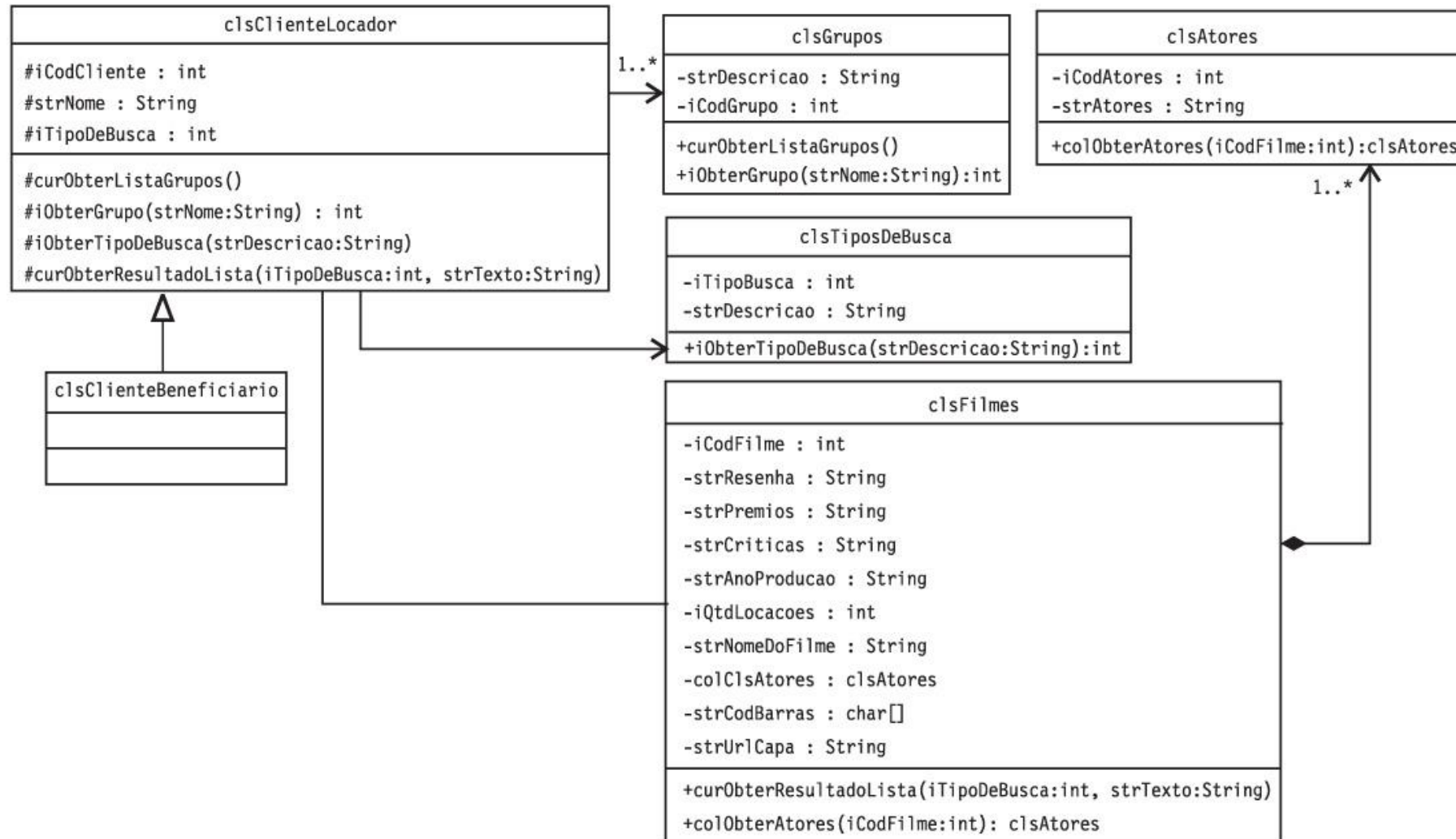
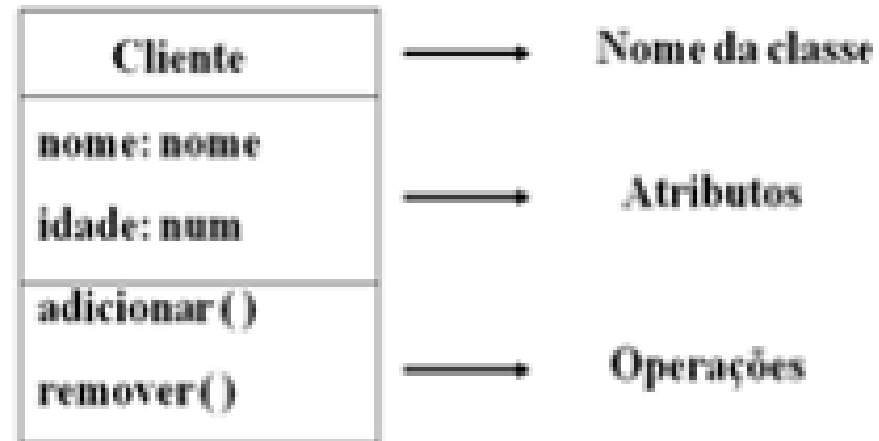


Figura 6.6 Diagrama de classes, fase 5.

Formato de uma classe



Atributos

- Permite a identificação de cada objeto de uma classe
- Os valores dos atributos podem variar de instância para instância
- Atributos podem conter os tipos de dados armazenados
 - Byte, int, String, double, etc

Métodos

- São apenas declarados
 - Diagrama de classe não define a implementação
- Outros diagramas permitem modelar o comportamento interno dos métodos
 - Diagrama de atividades
 - Diagrama de sequência

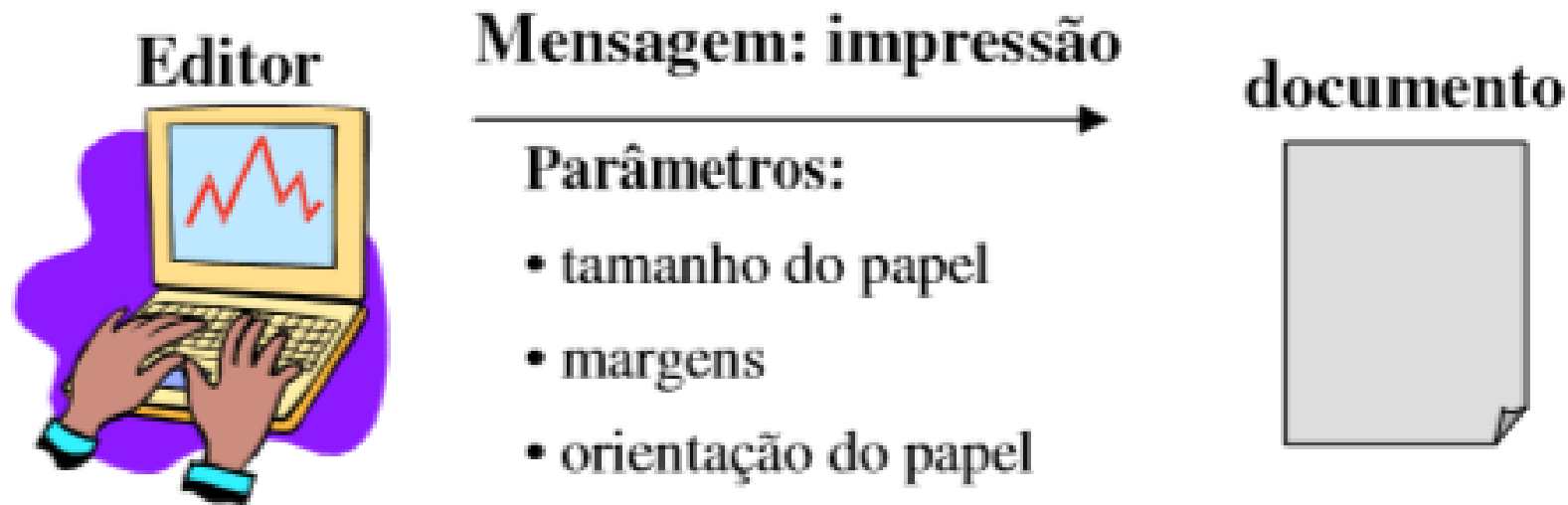


Tipos de visibilidade

- Pública(+)
 - O atributo ou método pode ser utilizado por qualquer classe
- Protegida(#)
 - Somente a classe, sub-classe e classes amigas(ex:pacotes) tem acesso
- Privada(-)
 - Somente a classe terá acesso

Comunicação

- Quem envia a mensagem a um objeto não necessita saber como o receptor irá tratá-la, deve apenas conhecer o resultado final do tratamento e o que é necessário para obtê-lo



Comunicação entre objetos

- Objetos se comunicam através da troca de mensagens
- Mensagens definem:
 - Nome do serviço requisitado
 - Informação necessária para a execução do serviço
 - Nome do requisitante
- Na prática, mensagens são implementadas através de chamadas de métodos
 - Nome: nome do método
 - Informação: parâmetros
 - Requisitante: método/objeto que realizou a chamada

Relacionamentos

- Ligam classes e interfaces
- Representado por um caminho
- Principais tipos:
 - Associação
 - Agregação/Composição
 - Herança
 - Dependência



Associação

- Relacionamento estrutural, determina que as instâncias de uma classe estão de alguma forma ligadas às instâncias de outras
- Associação binária
- Agregação – tipo especial de associação

Elemento gráfico: linha simples

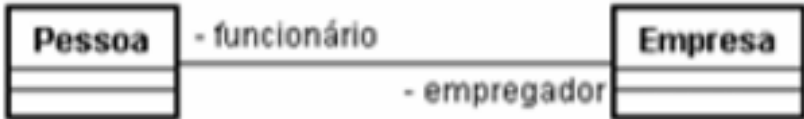
Exemplo:




Notação:

- Um departamento possui um ou mais funcionários na sua hierarquia.

Características de associação

Característica	Função	Representação
Papel	Quando uma classe está em uma associação, possui um papel específico neste relacionamento.	 <pre>classDiagram Pessoa -- Empresa : - funcionário, - empregador</pre>

Características de associação

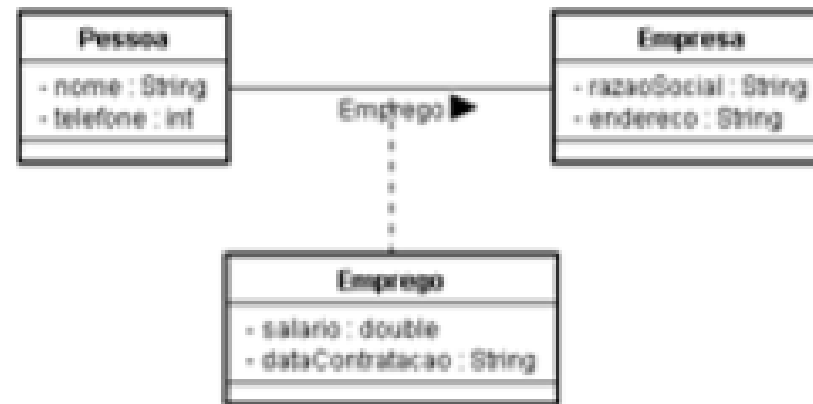
Característica	Função	Representação
Nome	Uma associação pode ter um nome, que pode ser utilizado para descrever a natureza do relacionamento.	 <pre>graph LR; Pessoa -- "Trabalha para" --> Empresa</pre>

Classes associativas

- Classes ligadas a associações, em vez de estarem ligadas a outras classes

Elemento gráfico: classe ligada a uma associação por uma linha tracejada

Exemplo:



Classe associativa = Emprego

Características de associação

Característica	Função	Representação
Multiplicidade	É importante determinar a quantidade (multiplicidade) de objetos que podem ser conectados pela instância de uma conexão. A tabela 4 ilustra a simbologia para representar as multiplicidades.	<pre>classDiagram Pessoa "1" -- "1..*" Empresa</pre>

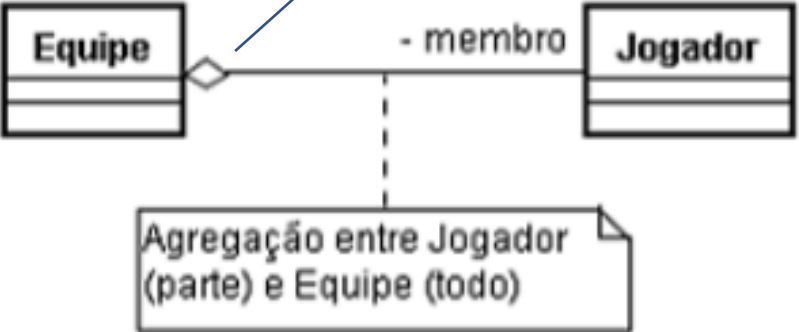
Simbologia para representar a multiplicidade

Nome	Simbologia
Apenas um	1
Zero ou Muitos	0..*
Um ou Muitos	1..*
Zero ou um	0..1

Agregação

Demonstra que as informações de um objeto precisam ser complementadas por um objeto de outra classe

Fica do lado
do objeto-
todo

Característica	Função	Representação
Agregação	É um tipo especial de associação que ilustra a associação todo/parte. Toda ação realizada sobre a classe todo afetará as classes partes.	

Composição

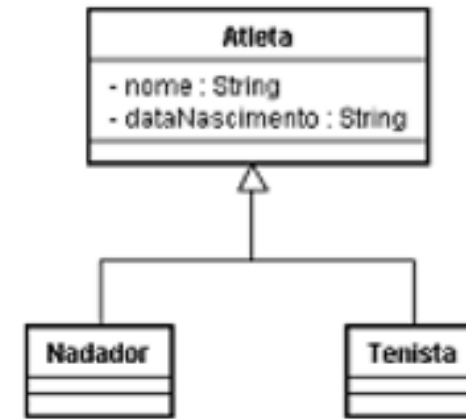
Característica	Função	Representação
Composição	<p>A composição é um tipo especial de agregação - agregação por valor. Semanticamente equivalente a um atributo.</p> <p>A remoção do todo implica na remoção das partes e o acesso às partes é restrito ao todo. Significa que as classes/objetos indicadas como "partes" de uma composição não estarão acessíveis externamente, somente por meio da classe/objeto "todo".</p>	<p>Classe Pessoa</p> <p>Classe Pessoa depois de utilizar composição</p>

Generalização/Especialização

- Os objetos da classe filha podem ser utilizados em qualquer lugar onde a classe mãe ocorra, mas não o contrário
- Identificar super-classes (geral) e sub-classes (especializadas)
 - Semântica “é um”
 - Tudo que a classe geral pode fazer, as classes específicas também podem
- Atributos e métodos definidos na classe-mãe são **herdados** pelas classes filhas

Elemento gráfico: seta fechada

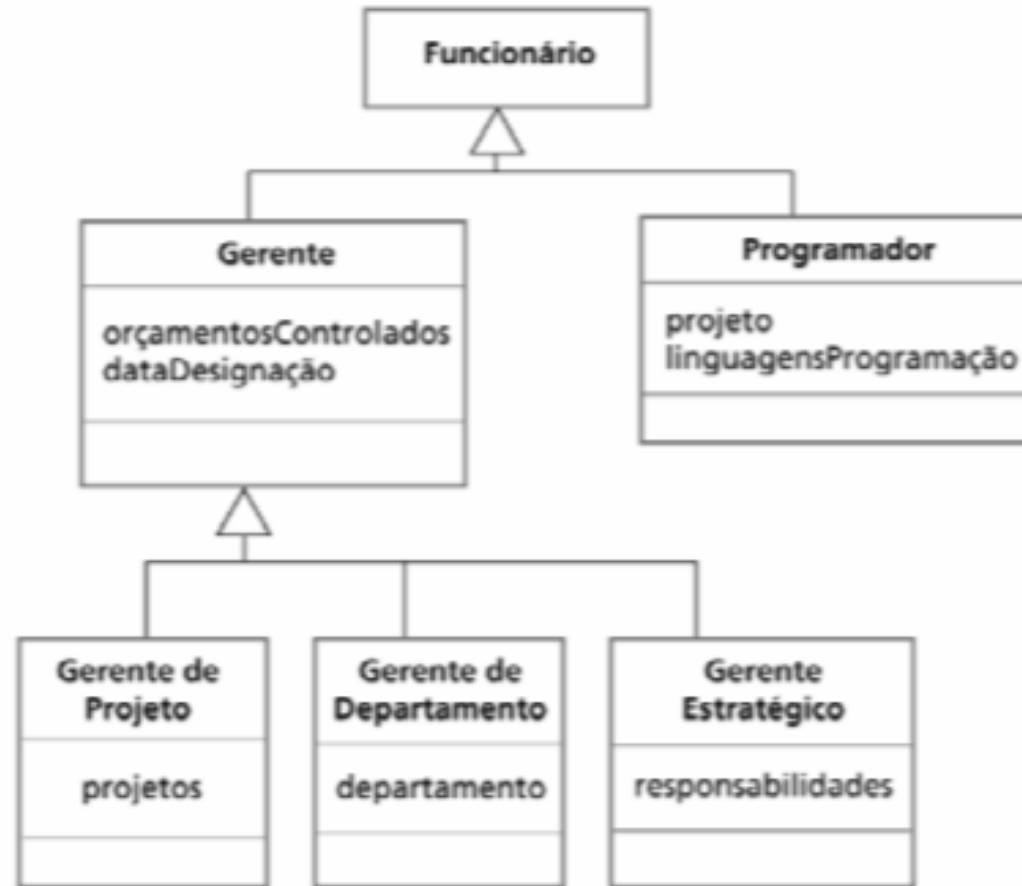
Exemplo:



Notação:

- A classe: Atleta é a classe-mãe.
- As classes: Nadador e Tenista herdam a estrutura – atributos e operações – da classe Atleta.

Hierarquia



Dependência

- Entre 2 itens
- Tipo menos comum de relacionamento
- Identifica uma relação fraca entre objetos de duas classes
- Uma alteração no item dependente pode afetar o funcionamento do item dependente

Elemento gráfico: seta aberta tracejada **Exemplo:**



Notação:

- A classe Dependente depende estruturalmente da classe Independente.



Diagrama de classes

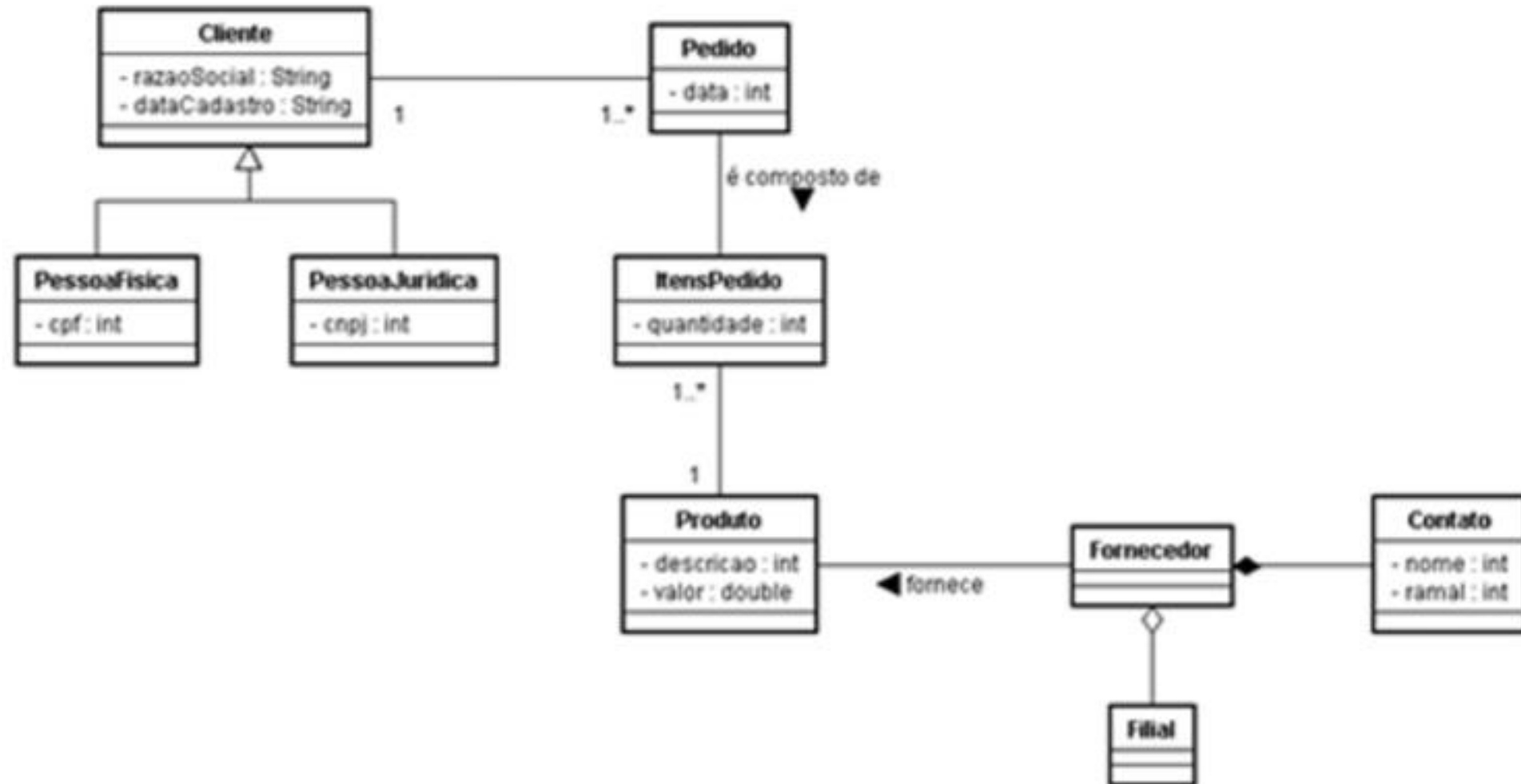


Diagrama de classes

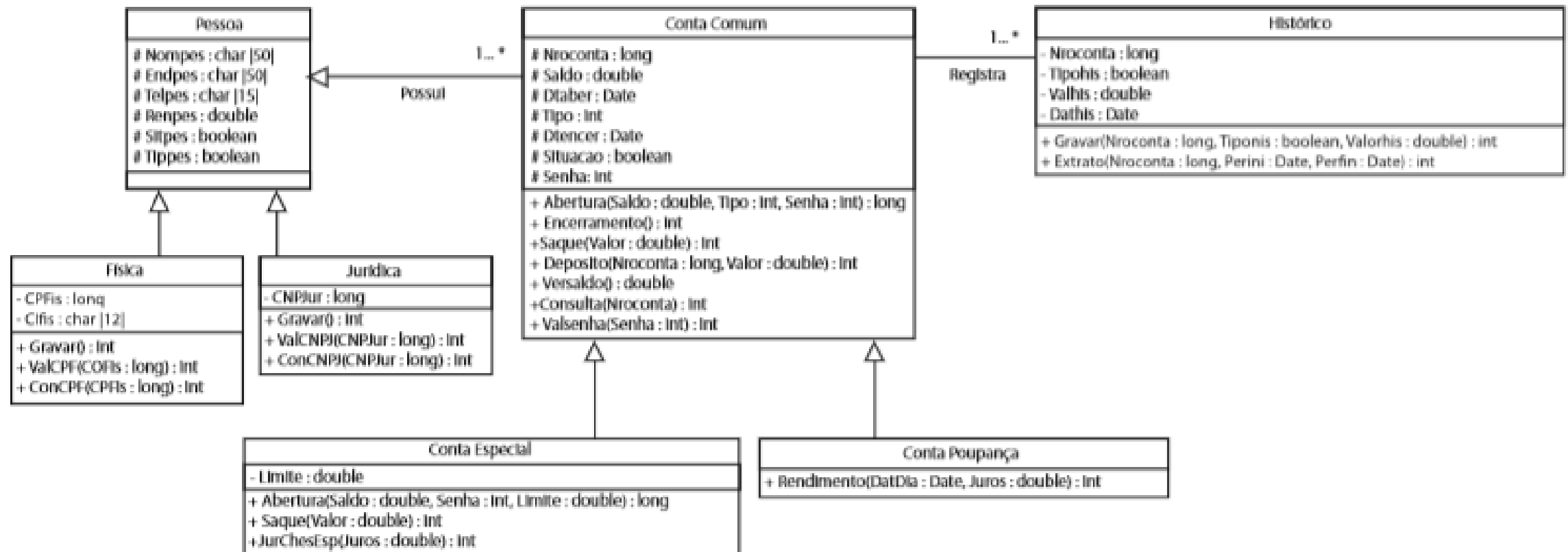


Figura 8 – Diagrama de Classes de um Sistema Bancário
Fonte: GUEDES, 2004, p.27

Diagrama de Objetos

- Fazem a modelagem de instâncias de itens contidos em diagramas de classes
- Mostra um conjunto de objetos e seus relacionamentos em um determinado ponto no tempo
- Modela a visão estática, uma foto



Diagrama de objetos

- Complemento do diagrama de classes
- Exibe os valores armazenados pelos objetos de um diagrama de classes

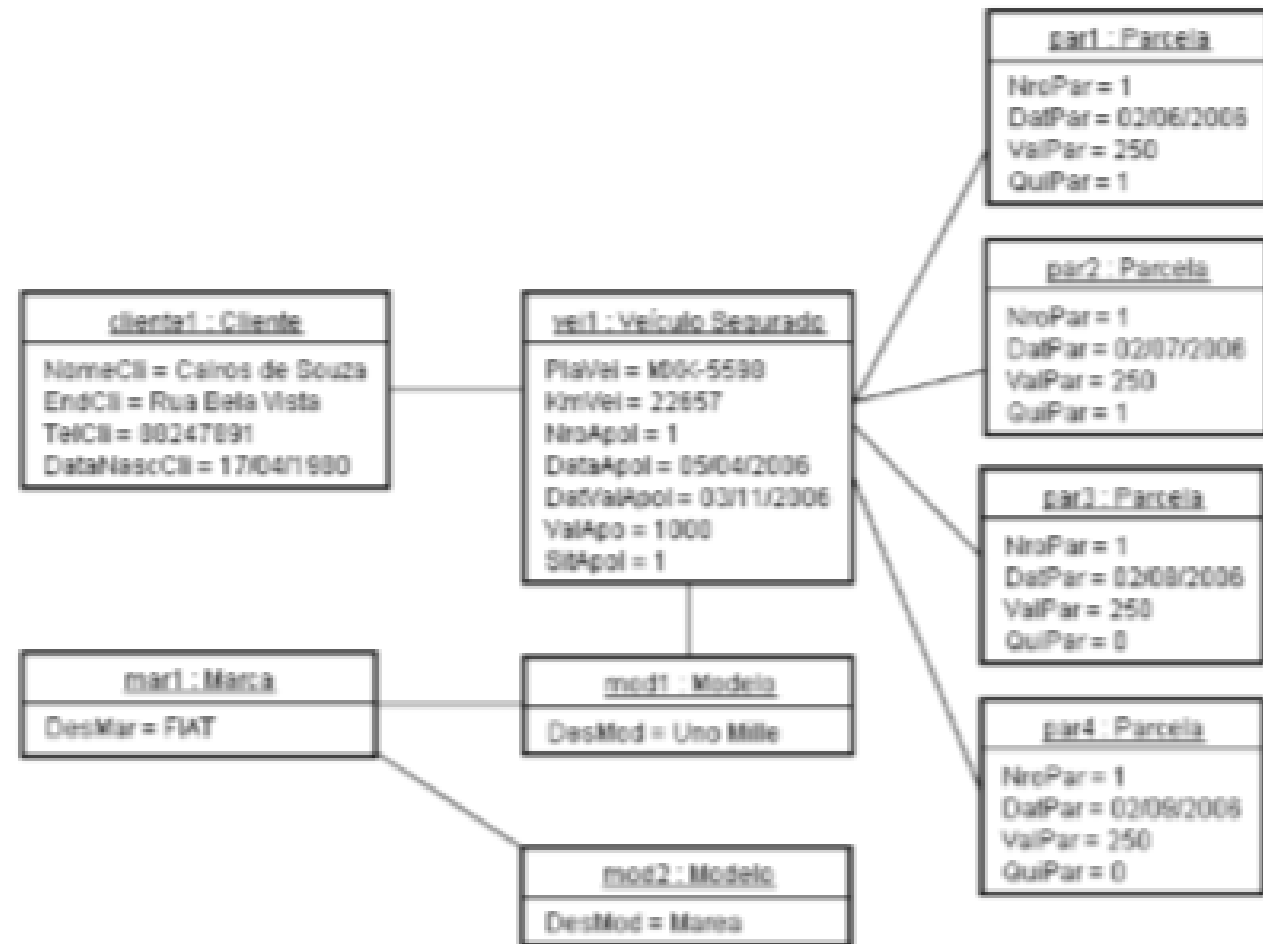


Diagrama de Objetos

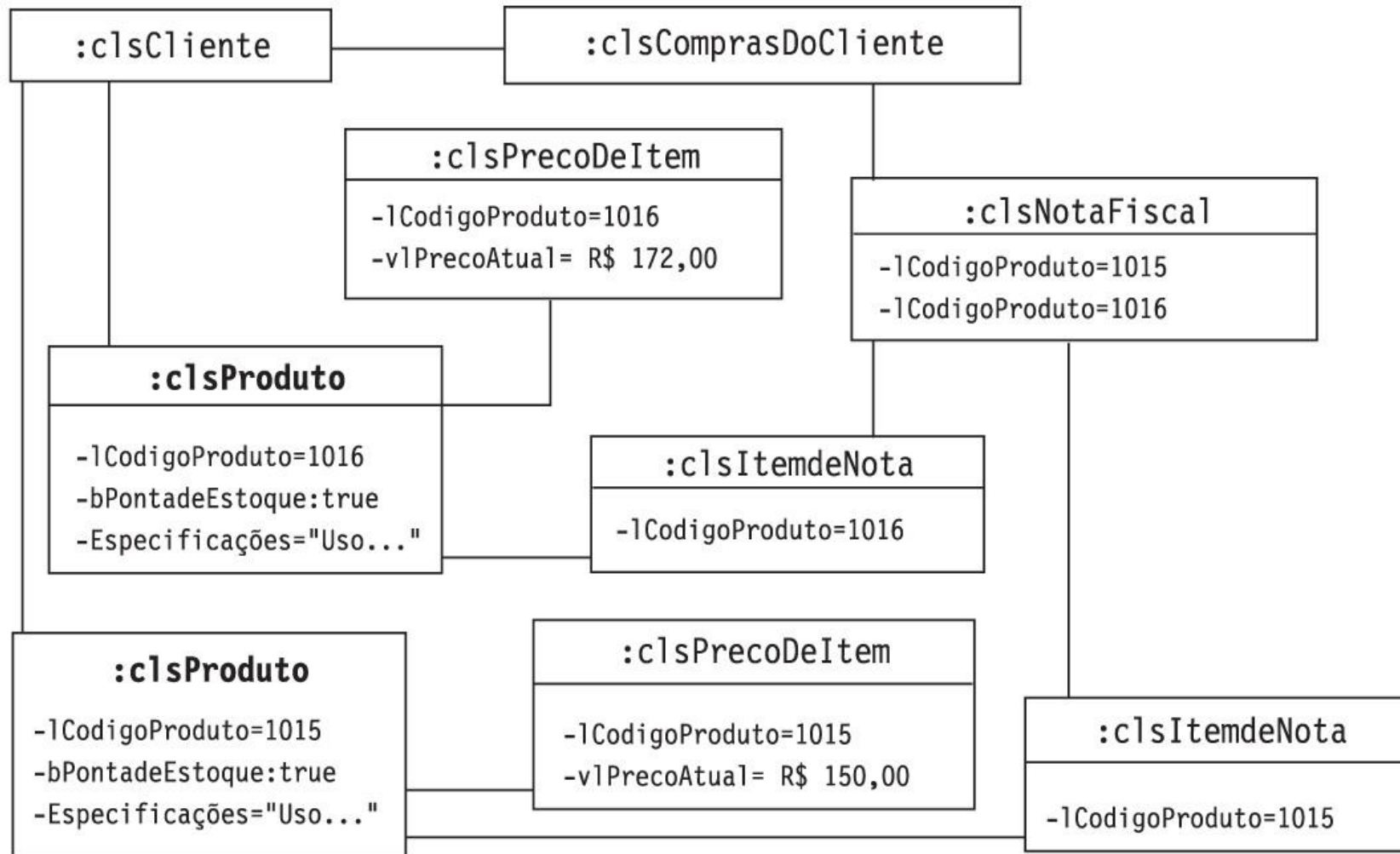


Figura 13.1 Diagrama de objetos – exemplo.

Diagrama de objetos

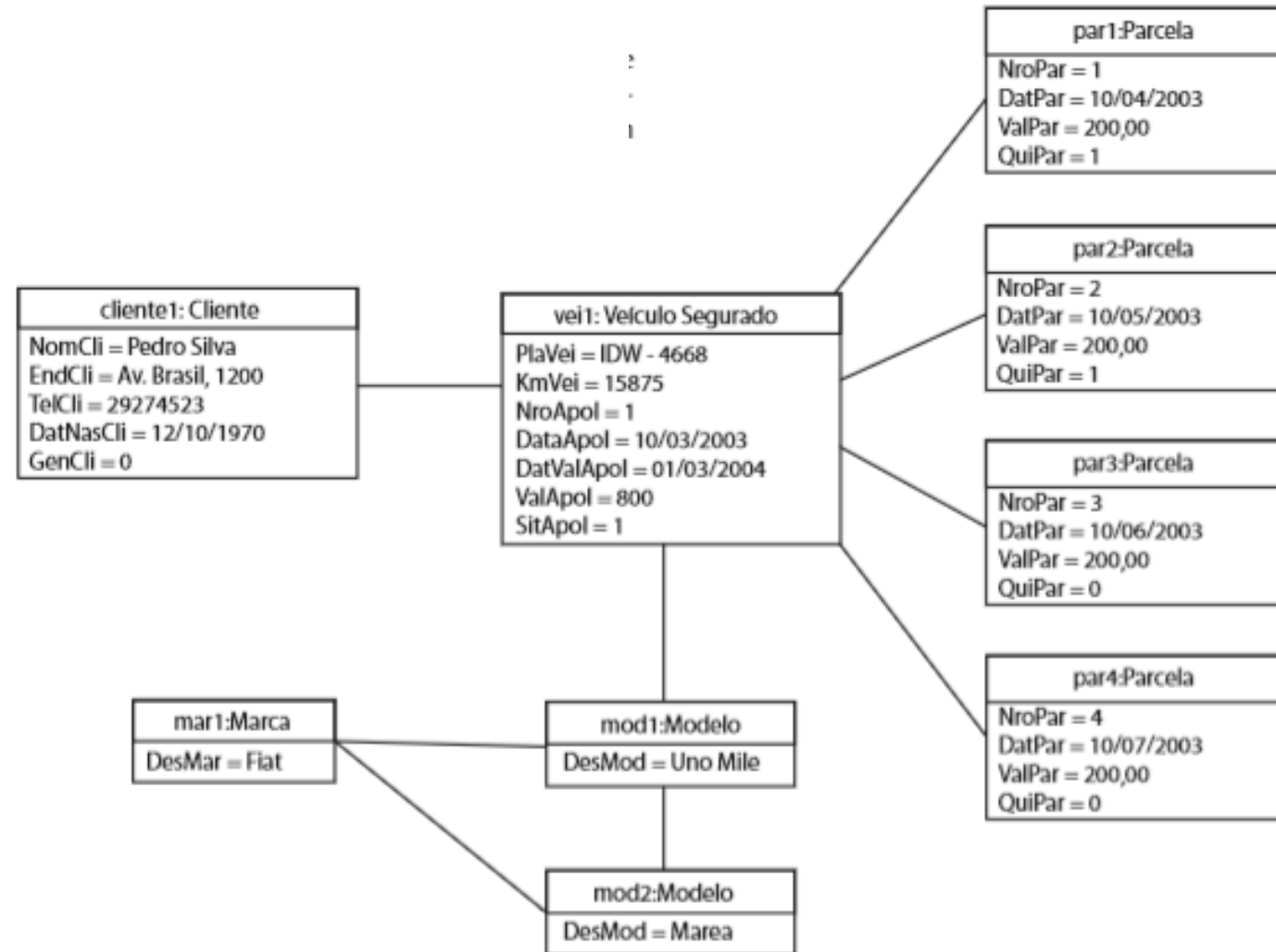
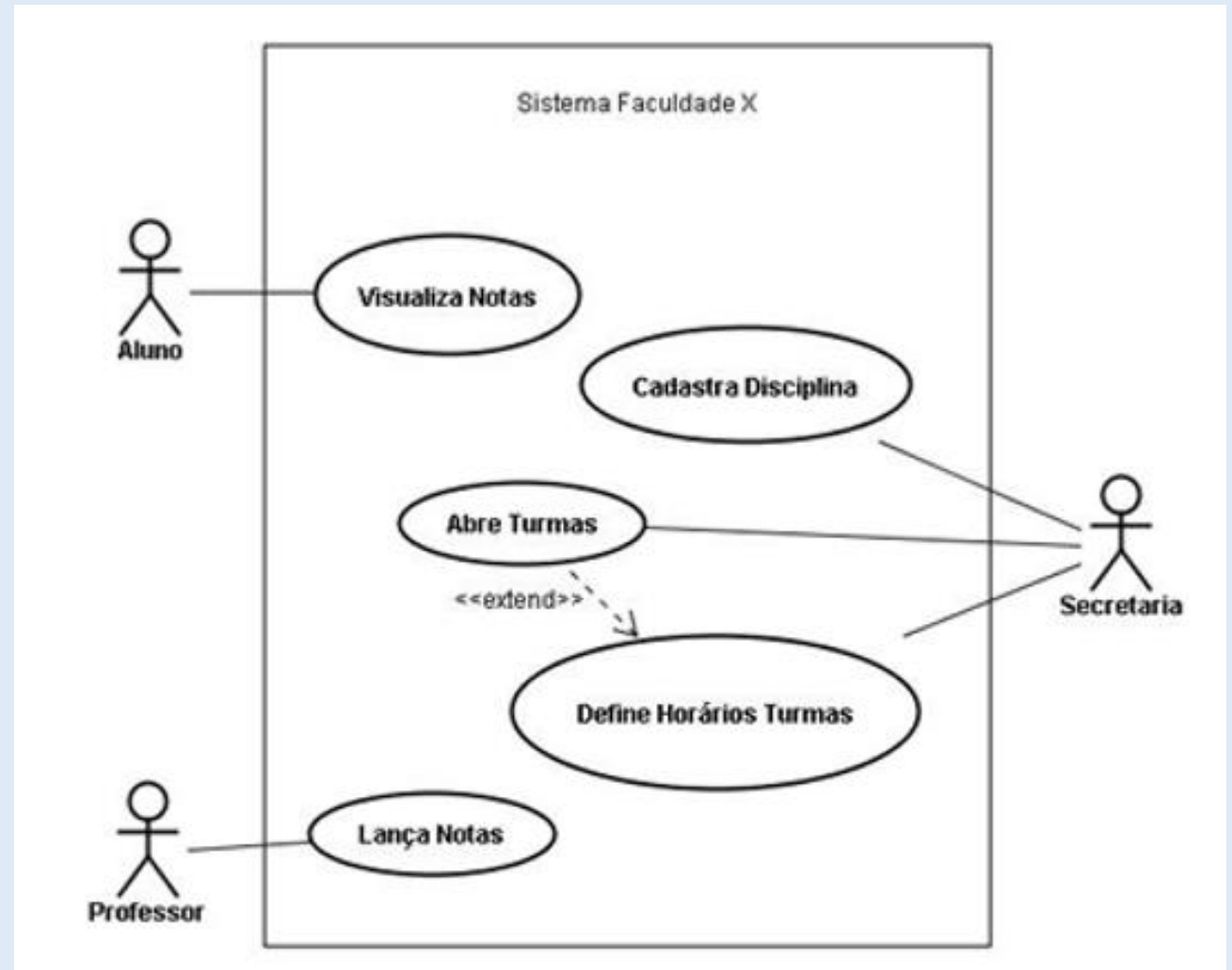


Figura 9 – Diagrama de Objetos de um Sistema para Seguradora de Veículos
Fonte: GUEDES, 2004, p.28

Atividade 1: Identificando as classes

Deve-se usar o conhecimento sobre o domínio do problema e a especificação de requisitos

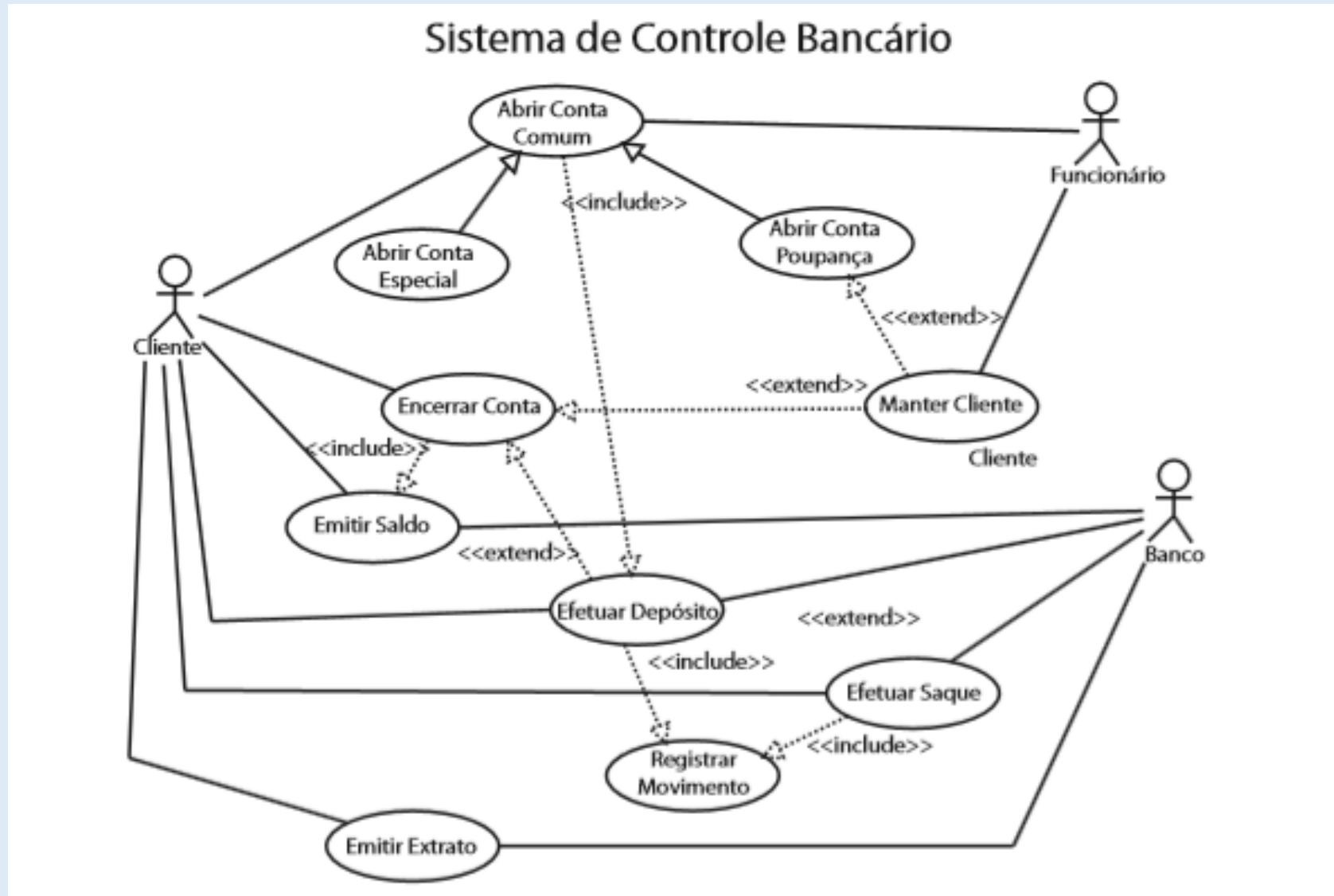


Atividade 2

A) identifique as classes do sistema



Atividade 3: Elabore o diagrama de classes



Atividade 4

A) identifique os atores do sistema;

B) identifique os casos de uso do sistema;

C) elabore o diagrama de casos de uso

d) Cenários

#	Descrição	Tipo
1	O sistema deve permitir ao Funcionário cadastrar Clientes contendo os dados: nome, endereço, e telefone	Requisito Funcional
2	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	Requisito Funcional
3	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 por meio do peso multiplicado por um valor fixo, acrescido da taxa de entrega da cidade de destino	Requisito Funcional
4	O sistema deve ser para plataforma WEB	Requisito Não-Funcional

Atividade 5

A) identifique os atores do sistema;

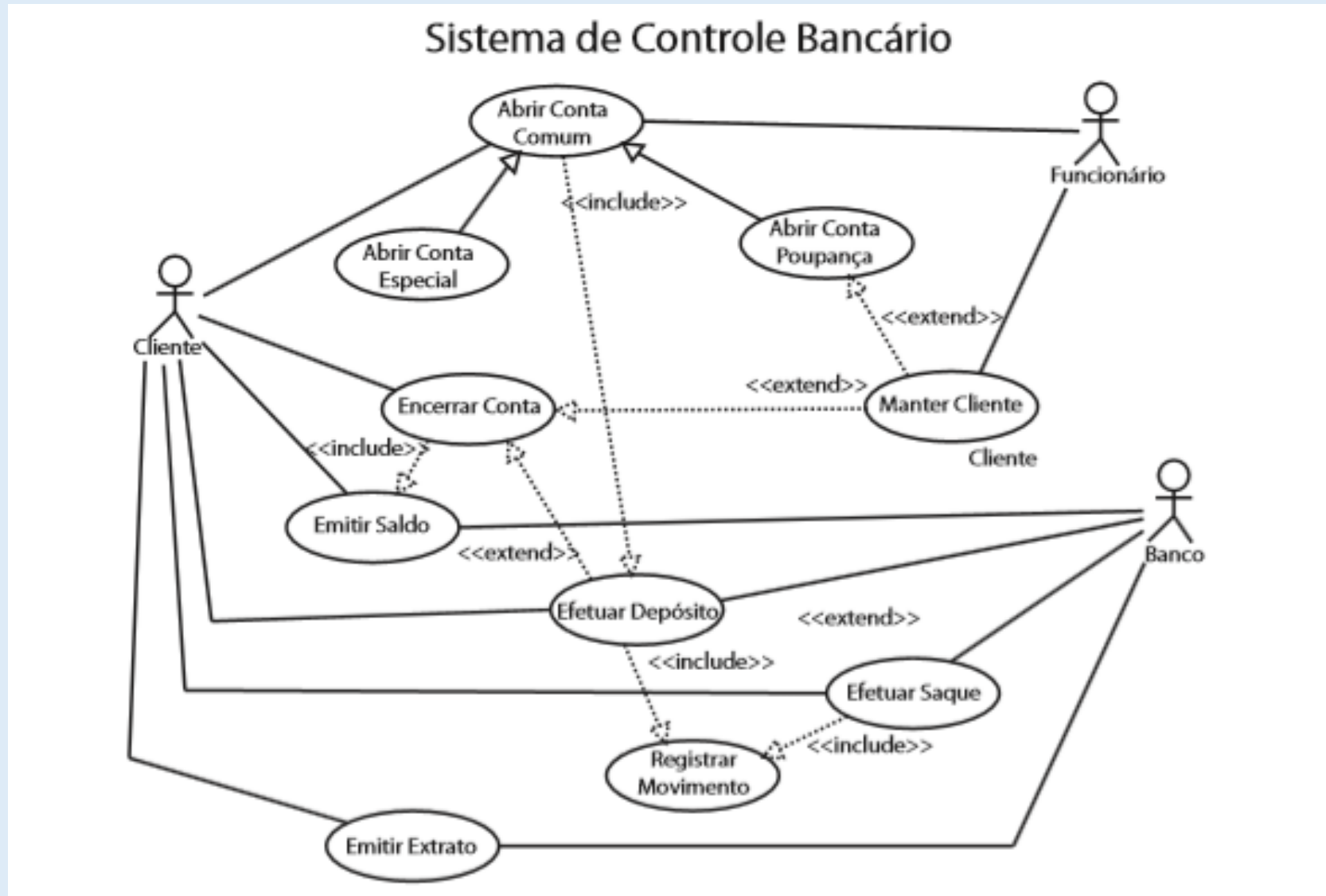
B) identifique os casos de uso do sistema;

C) elabore o diagrama de casos de uso

d) Cenários

#	Descrição	Tipo
1	O sistema deve permitir que o gerente cadastre salões de festas	Requisito Funcional
2	O sistema deve permitir que o gerente cadastre profissionais	Requisito Funcional
3	O sistema deve permitir que o gerente realize o cadastro de serviços utilizados em uma festa de casamento, como buffet e carros utilizados	Requisito Funcional
4	O sistema deverá permitir que o cliente realize o cadastro da lista de convidados	Requisito Funcional
5	O sistema deverá permitir que o cliente elabore a configuração da festa	Requisito Funcional
6	O sistema deverá permitir a geração do orçamento para o cliente	Requisito Funcional
7	O sistema deverá permitir que o gerente realize a impressão de relatório de festas	Requisito Funcional
8	O sistema deverá permitir que o cliente cadastre a lista de presentes	Requisito Funcional
9	O sistema deverá permitir que o gerente realize o cadastro de usuários	Requisito Funcional
10	O sistema deverá permitir a autenticação dos usuários	Requisito Funcional
11	O sistema deve ser compatível com os navegadores Internet Explorer e Firefox	Requisito Não-Funcional
12	O sistema deve garantir que o tempo de retorno das consultas seja inferior a 5 segundos	Requisito Não-Funcional

Elabore os cenários deste diagrama



Atividade 7

Desenvolva o diagrama Caso de usos e de classes para um sistema de controle de cinema, com base nos seguintes requisitos:

- Um cinema pode ter muitas salas, sendo necessário, registrar informações a respeito de cada sala, como sua capacidade;
- O cinema apresenta muitos filmes. Um filme tem informações como título e duração;
- Um filme pertence a um único gênero, mas um gênero pode se referir a muitos filmes;
- Um filme pode ter muitos atores atuando nele, e um ator pode atuar em muitos filmes. Em cada filme, um ator interpretará um ou mais papéis diferentes. Por força do Marketing, é útil anunciar os principais atores do filme e que papéis eles representam;
- Um mesmo filme pode ser apresentado em diferentes salas e horários. Cada apresentação é chamada de Sessão. Um filme sendo apresentado em sessão tem um conjunto máximo de ingressos, determinado pela capacidade da sala;
- Os clientes do cinema podem comprar ou não ingressos para assistir a uma sessão. O funcionário deve intermediar a compra do ingresso. Um ingresso deve conter informações como o tipo de ingresso(meia entrada ou inteira). Além disso um cliente só pode comprar para sessões não encerradas.