

Princípios da Orientação a Objetos: Classes e 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

Modelagem do projeto de software

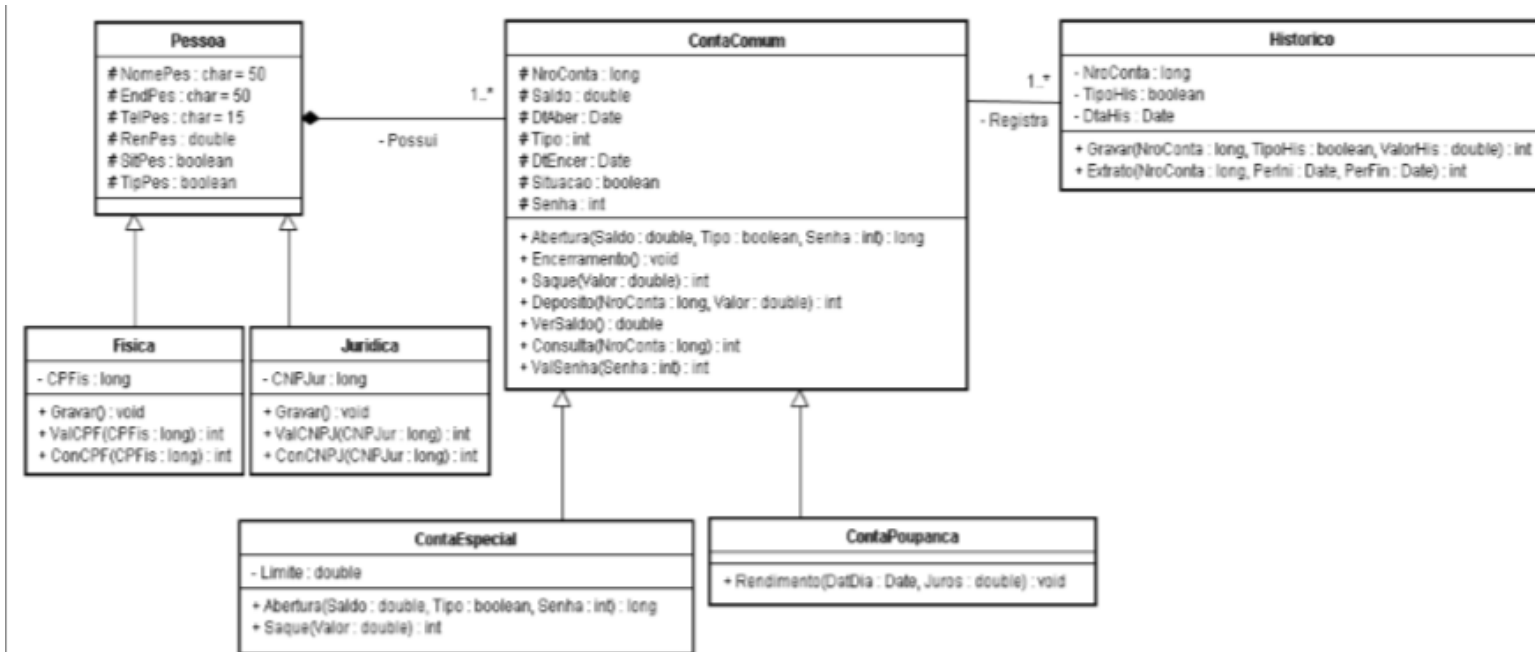
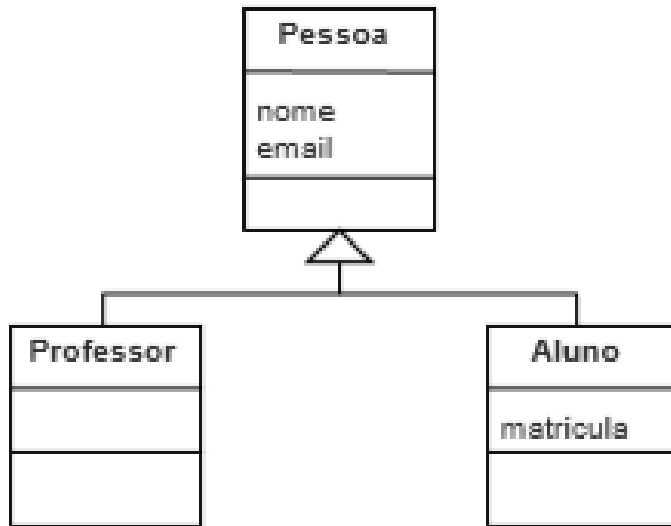
Após a definição do diagrama de caso de usos e a sucessiva validação pelo cliente é possível dar continuidade no processo de modelagem

Não há uma sequência lógica para a elaboração dos diagramas UML

Também não há uma quantidade definida de cada um

Classe

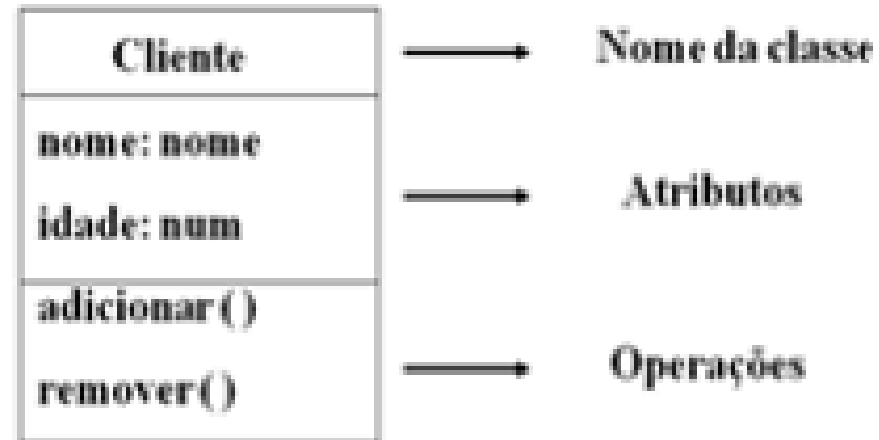
- Descrição de um conjunto de objetos que compartilham os mesmos atributos, operações e relacionamentos
- Classes são matrizes de objetos e descrevem um conceito abstrato do domínio do problema



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

Formato de uma classe



Classes em Python

```
class Carro:

    def __init__(self, marca, modelo, ano):

        self.marca = marca

        self.modelo = modelo

        self.ano = ano


    def descrever(self):

        print(f"{self.marca} {self.modelo}, Ano: {self.ano}")


# Criando objetos da classe Carro

carro1 = Carro("Toyota", "Corolla", 2020)

carro2 = Carro("Honda", "Civic", 2018)


# Chamando métodos

carro1.descrever()

carro2.descrever()
```

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

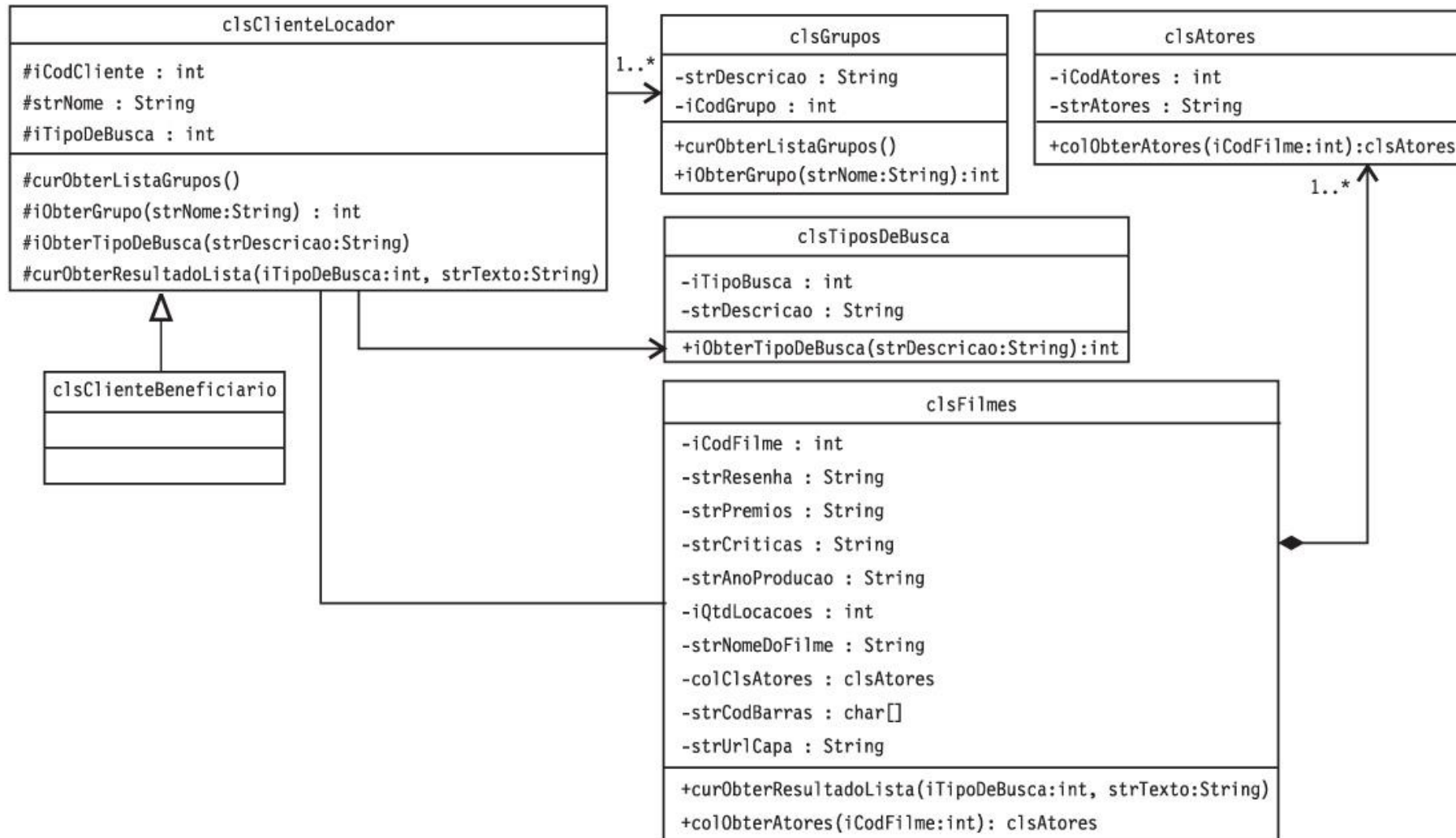


Métodos

- 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



Diagrama de Classes



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 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

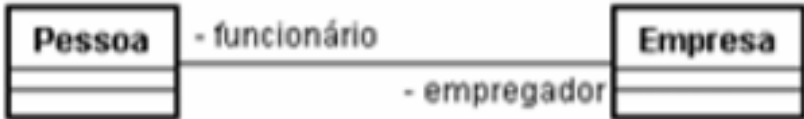
Exemplo:




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 class Pessoa class Empresa Pessoa -- Empresa : - funcionário, - empregador</pre>

Características de associação

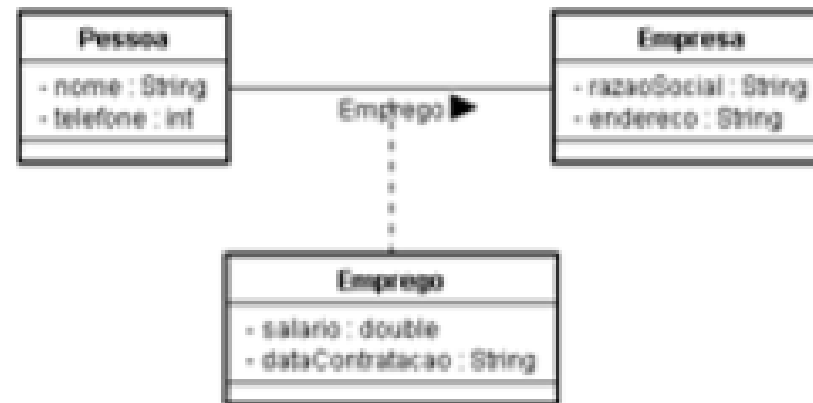
Característica	Função	Representação
Nome	Uma associação pode ter um nome, que pode ser utilizado para descrever a natureza do relacionamento.	 <pre>graph LR; Pessoa -- "Trabalha para" --> Empresa</pre>

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>

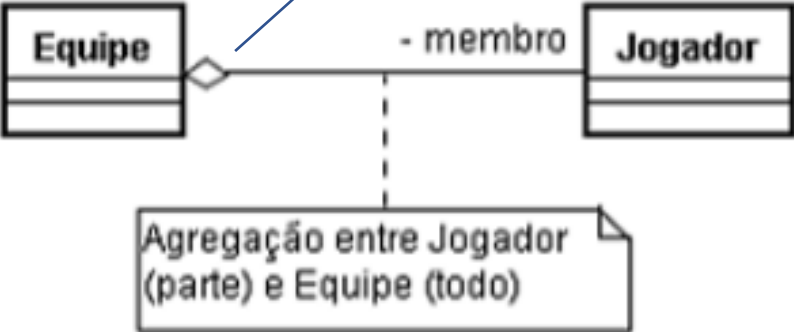
Simbologia para representar a multiplicidade

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

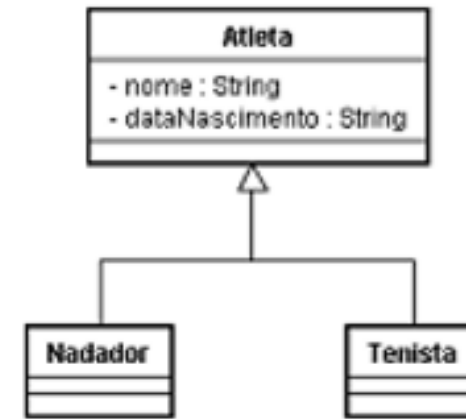
Característica	Função	Representação
Composição	<p>A composição é um tipo especial de agregação - agregação por valor. Semanticamente equivalente a um atributo.</p> <p>A remoção do todo implica na remoção das partes e o acesso às partes é restrito ao todo. Significa que as classes/objetos indicadas como "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

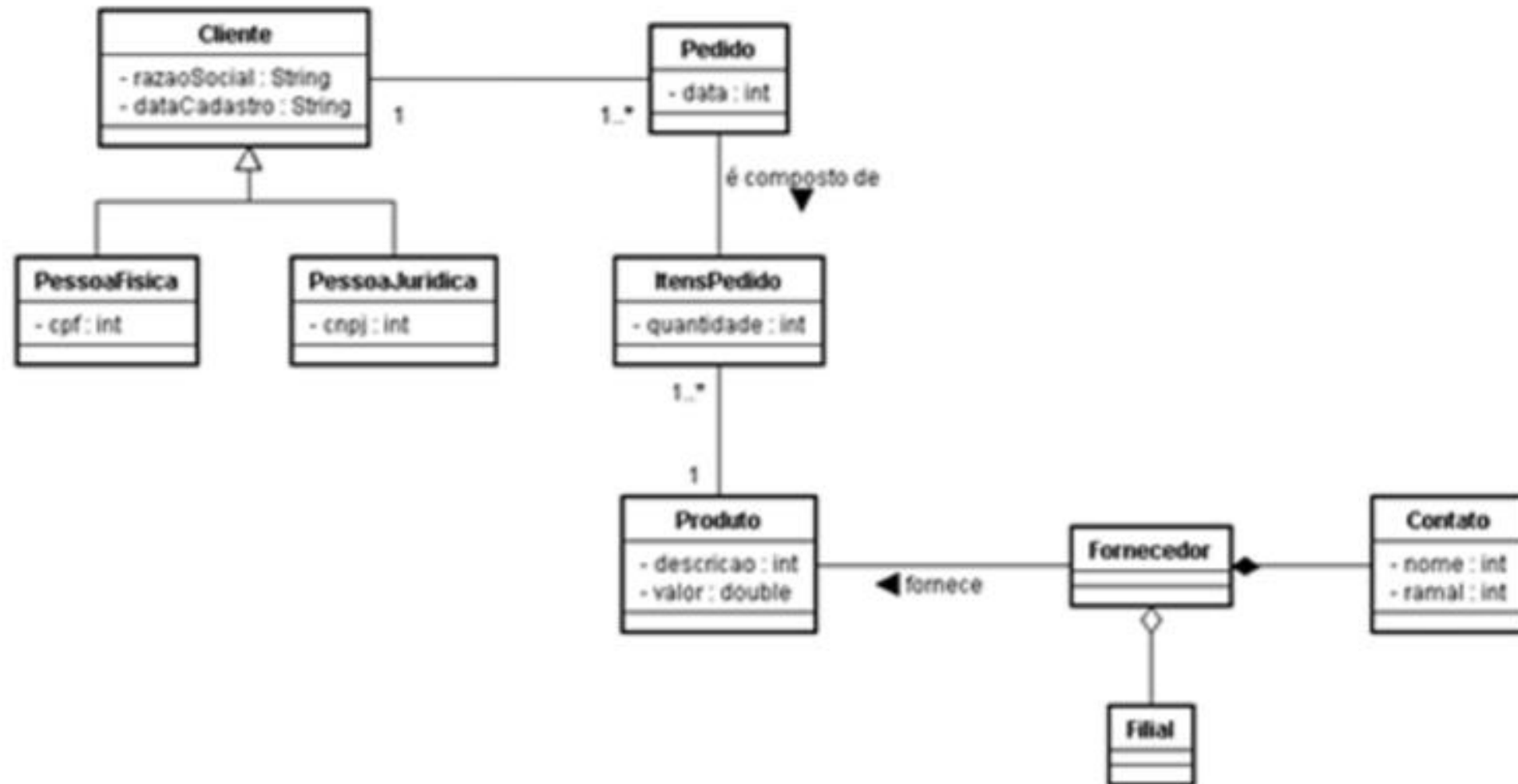


Diagrama de classes

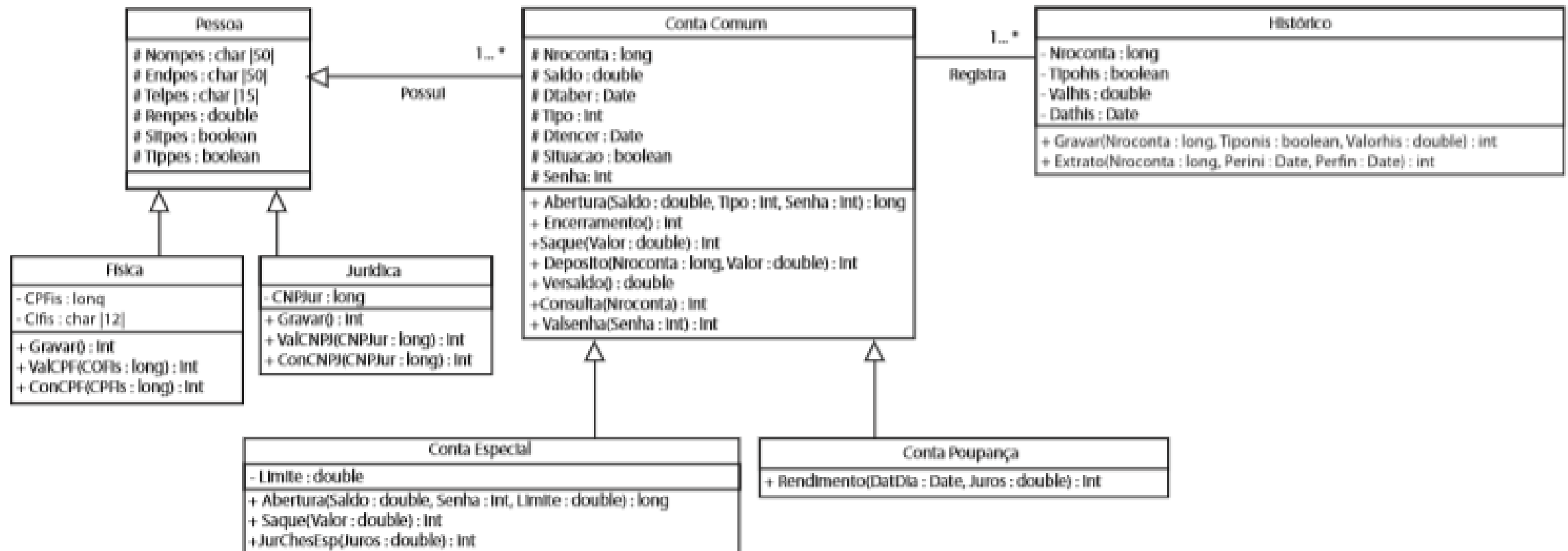


Figura 8 – Diagrama de Classes de um Sistema Bancário
Fonte: GUEDES, 2004, p.27

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

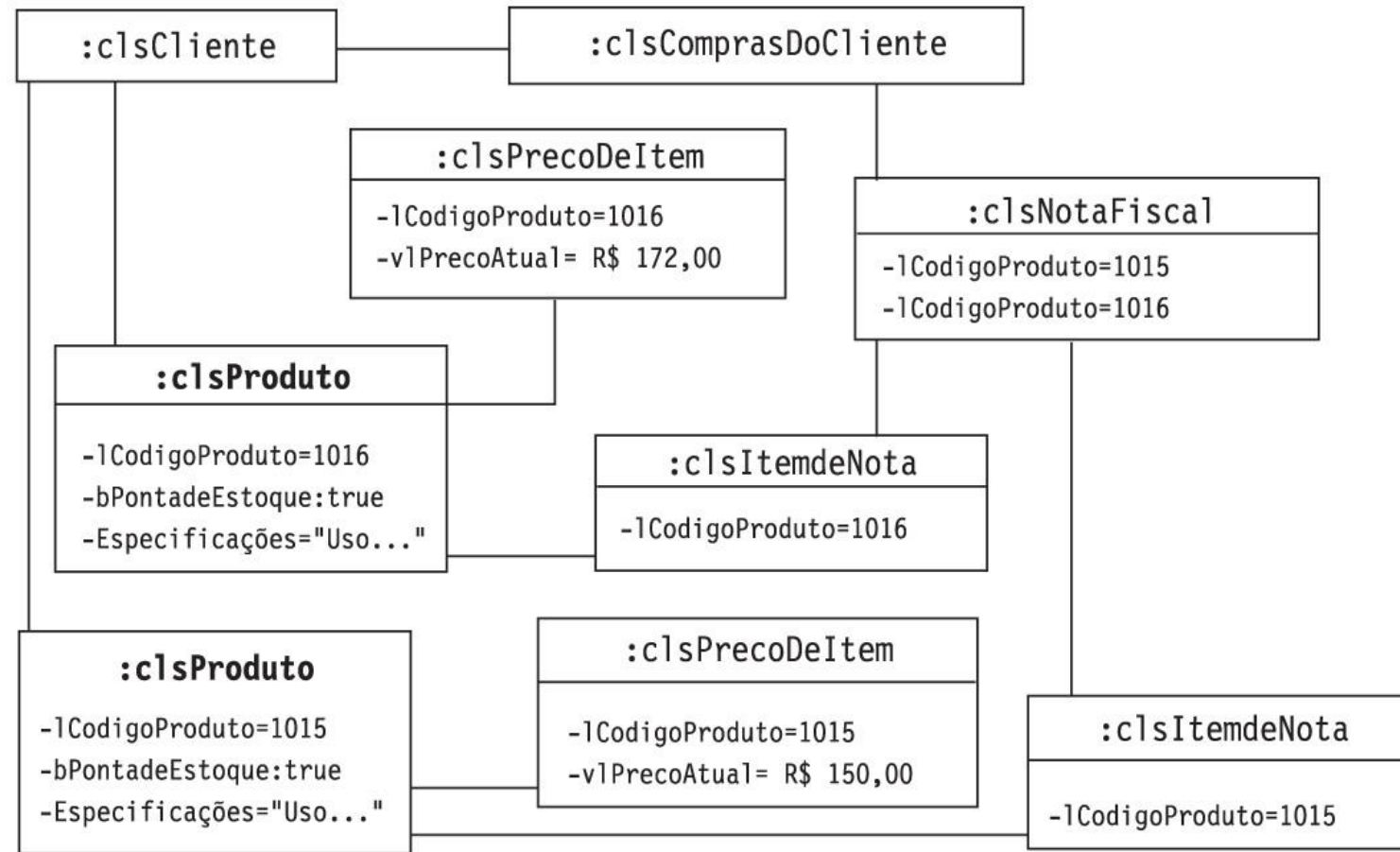


Figura 13.1 Diagrama de objetos – exemplo.

Diagrama de objetos

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

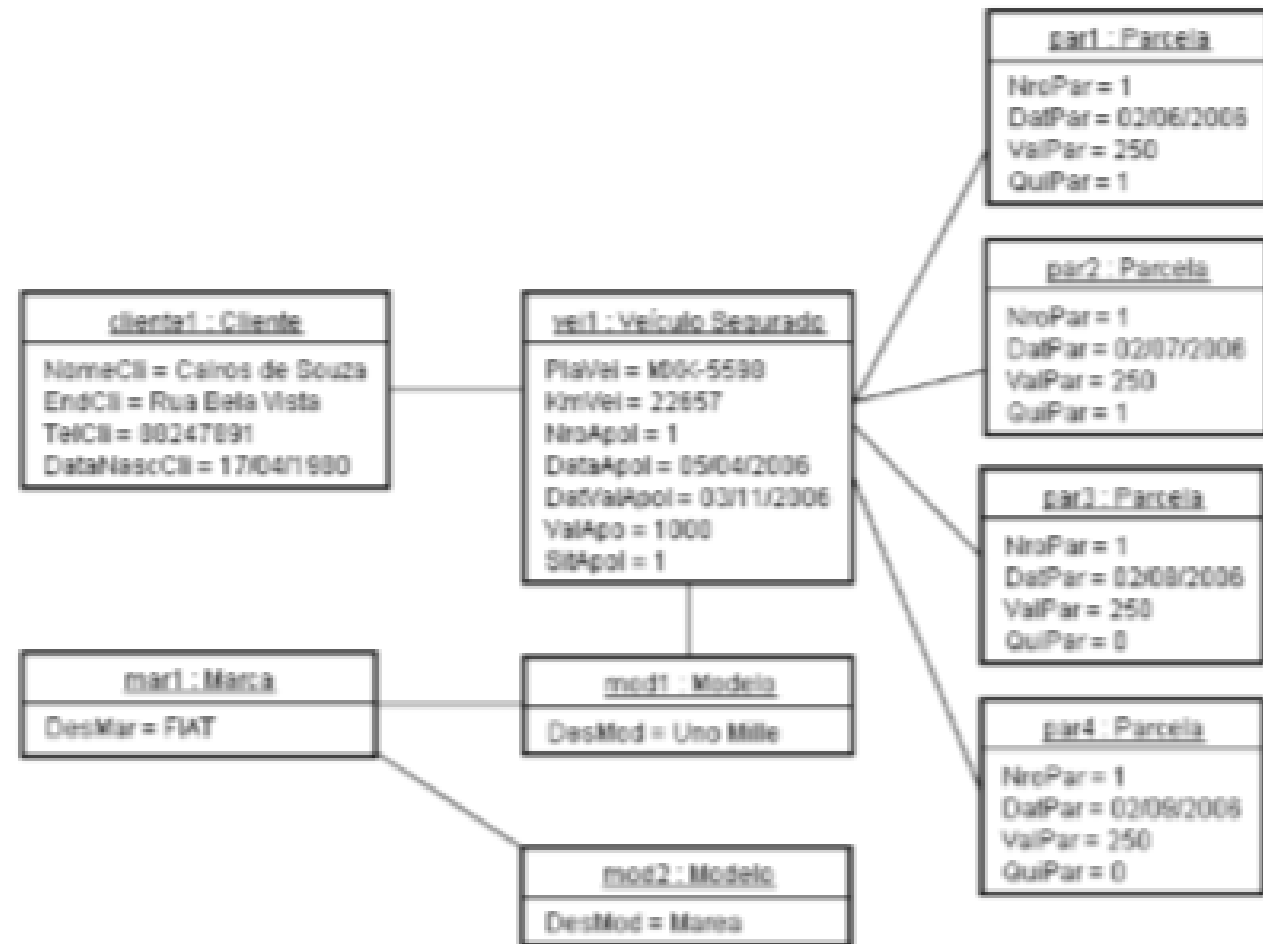


Diagrama de objetos

O Diagrama de Objetos está intrinsecamente associado ao Diagrama de classes. Este diagrama (Figura 9) oferece uma visão dos valores armazenados pelos objetos obtidos a partir do Diagrama de classes em um dado momento na execução do sistema.

5.4 - Diagrama de Seqüência

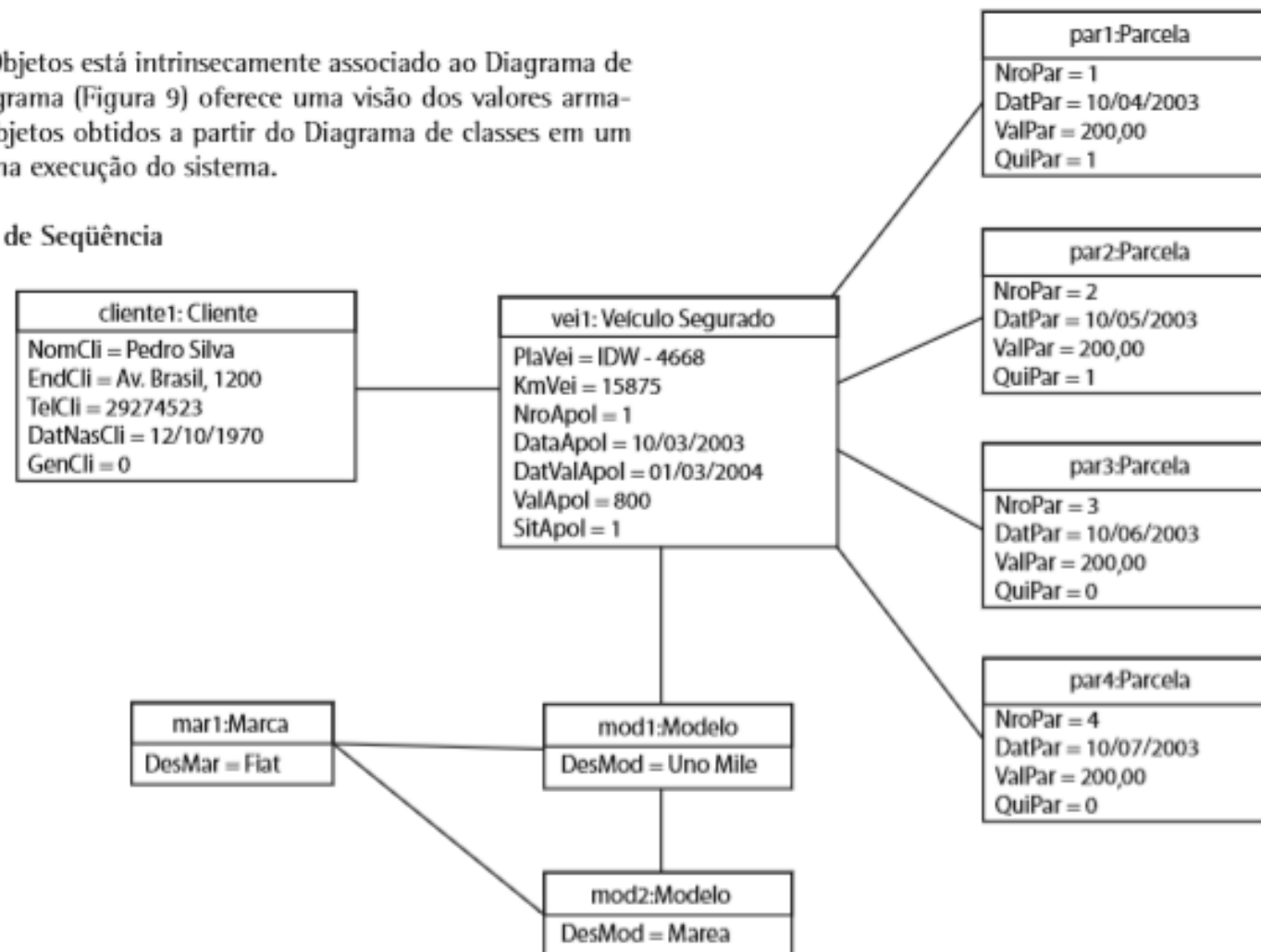
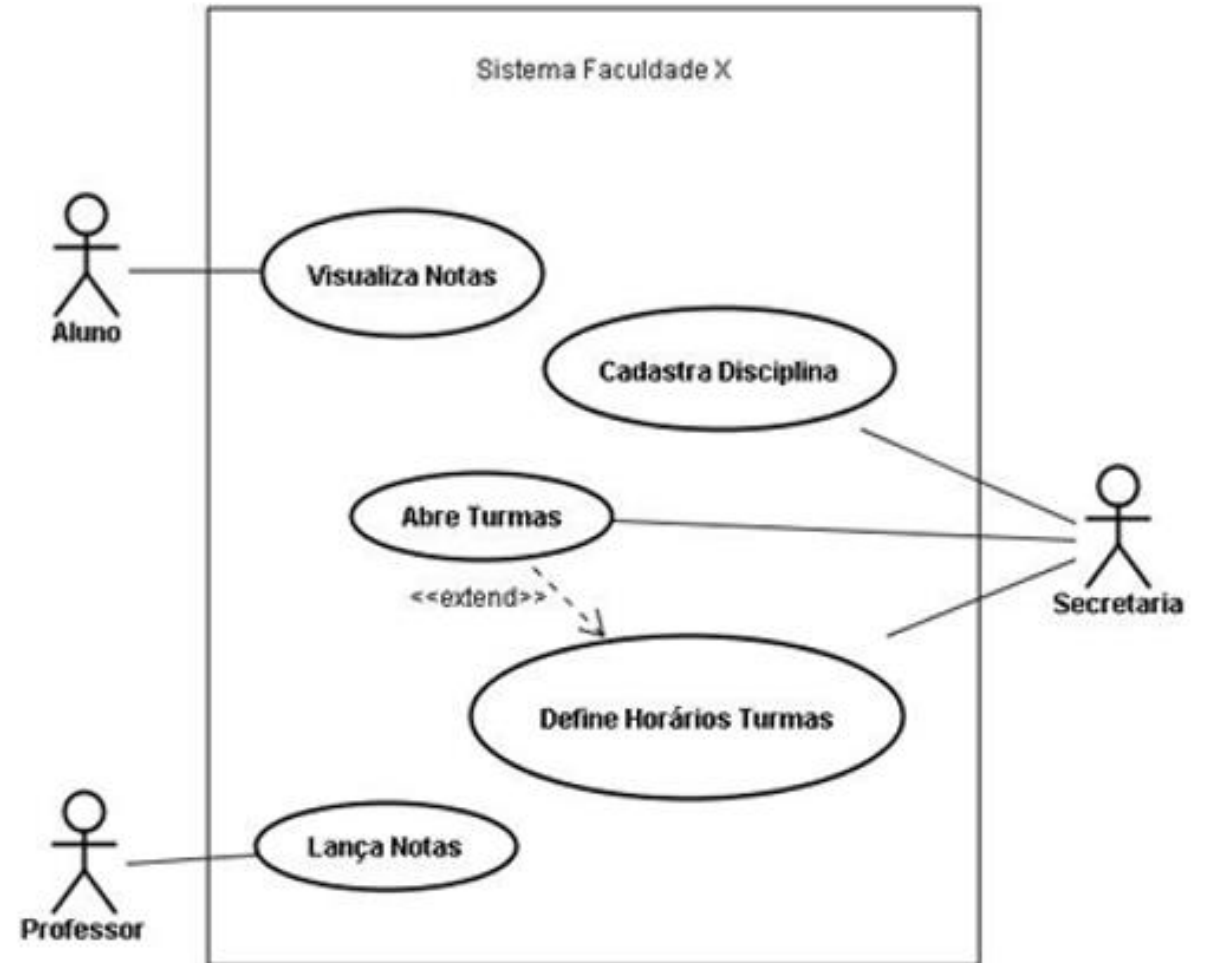


Figura 9 – Diagrama de Objetos de um Sistema para Seguradora de Veículos
Fonte: GUEDES, 2004, p.28

Atividade 1: Identificando as classes

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

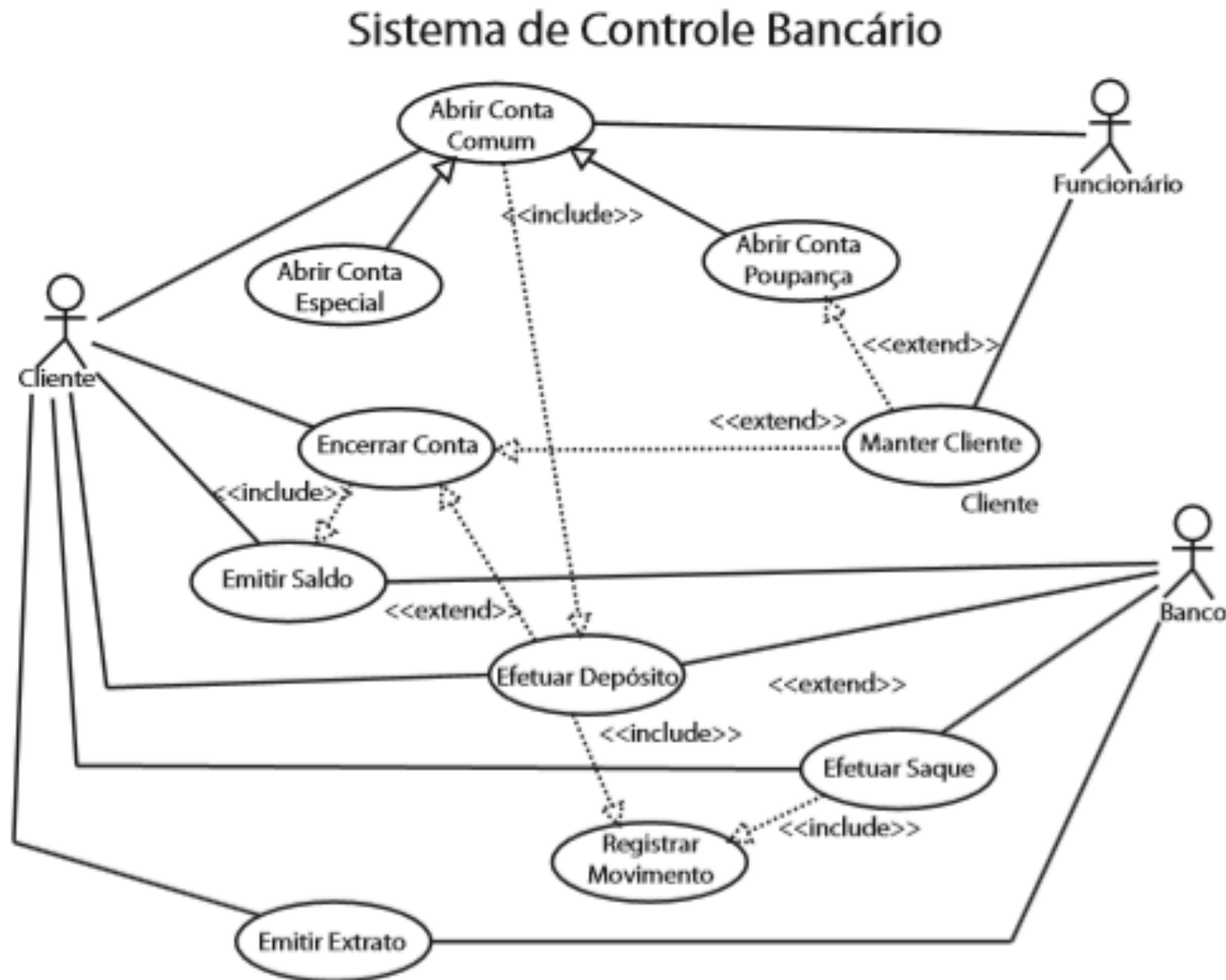


Atividade 2

A) identifique as classes do sistema

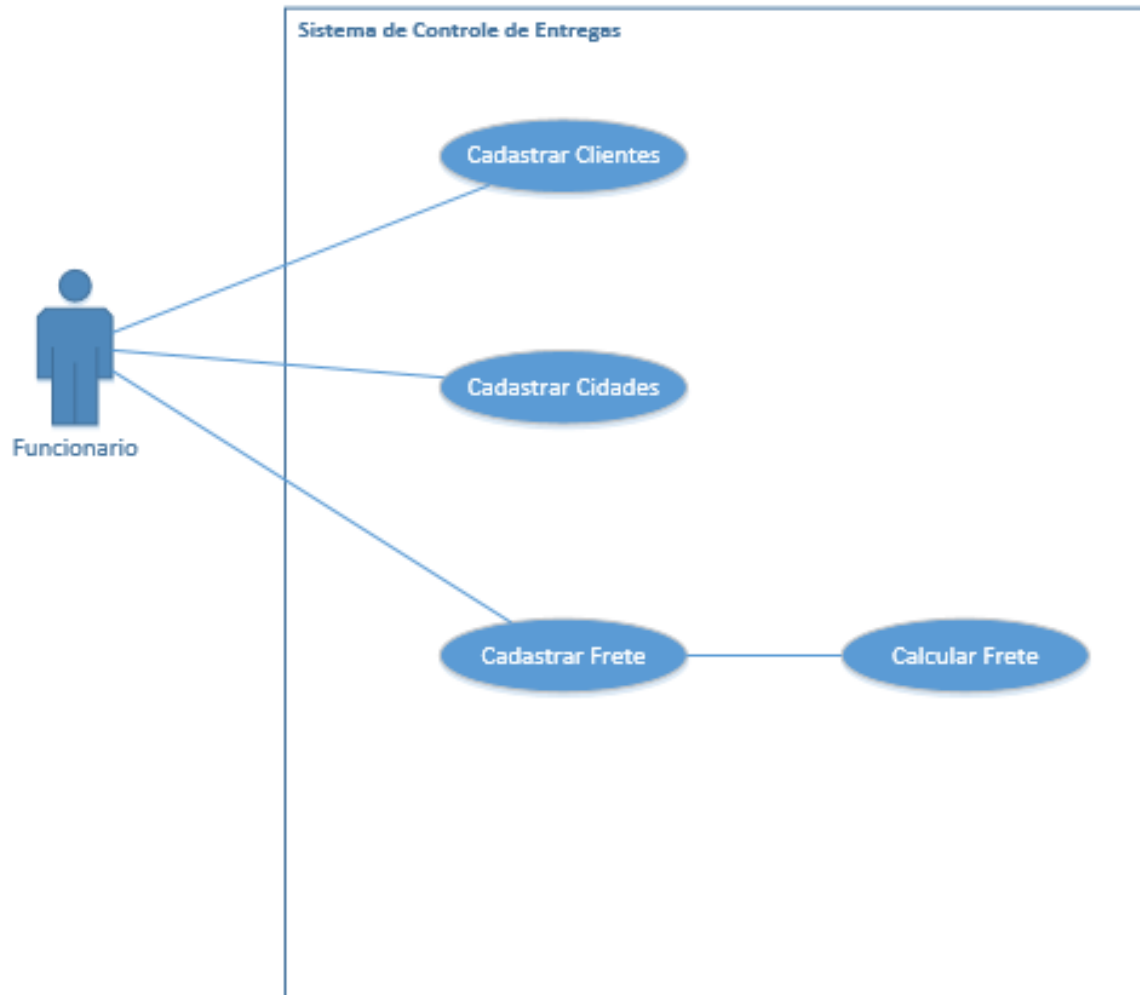


Atividade 3: Elabore o diagrama de classes



Atividade 4

A) identifique as classes do sistema



#	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

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.