

## Orientação a Objetos

Trata-se de uma maneira de pensar os problemas utilizando modelos organizados a partir de conceitos do mundo real, sendo o principal componente o **objeto**.

A O.O. é uma estratégia de desenvolvimento do software que é organizada através de uma **coleção de objetos**.

**Exemplo:** Num sistema escolar temos os seguintes **objetos**: Funcionários (professores, porteiros, administrativos, coordenadores, etc), Alunos, Turmas, Disciplinas, etc.

## As Linguagens Orientadas a Objetos – Evolução

- 1966 – Simula
- 1980 – Smalltalk
- 1986 – C++
- 1988 – Eiffel
- 1989 – Turbo Pascal 5.5
- 1995 – Java
- 2001 – C#
- 2003 – VB Net

Exercício – Pesquisa

## Ferramenta Case

Com o advento da Orientação a Objeto, surgiu também uma nova maneira de documentar sistemas, que é a UML ( Unified Modeling Language ), desenvolvida por Ivar Jacobson, Grady Booch e James Rumbaugh no início da década de 90. A UML como o próprio nome diz, unificou a notação de sistemas, focando na orientação a objetos.

As ferramentas CASE até então suportavam notações advindas da técnica Estruturada ( Análise Estruturada ), surgida no final da década de 70 com Tom De Marco. (Em seguida vieram Chris Gane, Trish Sarson e Edward Yourdon).

Existem inúmeras ferramentas CASE disponíveis no mercado. Entre elas poderia citar: Poseidon, Rational Rose 2000, ErWin, Oracle Designer, Genexus, Clarify, Dr. Case, Multicase, Paradigm, PowerDesigner, Together, Cognos, CoolGen, Smart, Theseus, BPWin, Arena, Visio, Brio, Microstrategy, StarUML, Astah, etc.

## Características da Orientação a Objetos

- **Reusabilidade**
  - Reutilizar componentes de software
  - Diminuir tempo de desenvolvimento
- **Manutibilidade**
  - Mudanças localizadas
- **Confiabilidade (encapsulamento)**
  - Maior controle e segurança das classes dos objetos
- **Extensibilidade (herança e polimorfismo)**
  - Facilidade em adicionar novas operações à um componente de uma modelagem existente

## O que são objetos?

“É um elemento do mundo real” [José Davi Furlan]

Possui um **comportamento** e **atributos**. O comportamento é a forma com que reage a estímulos e os atributos são suas características, que podem mudar de acordo com seu comportamento.

Trata-se de qualquer coisa do mundo real que tem identidade bem definida, contendo características e comportamentos.

Exemplos:

- **Concretos:** pessoa, lápis, carro, etc.
- **Intangíveis:** hora, projeto, organização, etc.
- **Personagem:** médico, professor, etc.
- **Relacional:** casamento, parceiro, propriedade.
- **Evento:** venda, defeito do sistema.
- **Interface Gráfica:** janela, ícone, botão.

Exemplo de objetos:



## Objeto – suas características e comportamentos

Comportamento (Operações ou Métodos)

Lógica contida em um objeto para designar-lhe um comportamento. É como o objeto responde a outro objeto. Também chamado de Método, Comportamento, Procedimento, Função, etc.

Objeto João



Nome: João  
DataNasc: 10/10/1969  
Endereço: Rua das Flores 23  
Telefone: 3794-2903

**Características**

Andar  
Estudar  
Falar

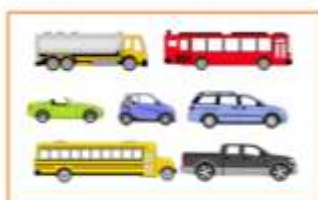
**Comportamento**

## O que é Classe?

Representa um conjunto de objetos que possuem características (**atributos**) e comportamentos (**operações**) semelhantes.

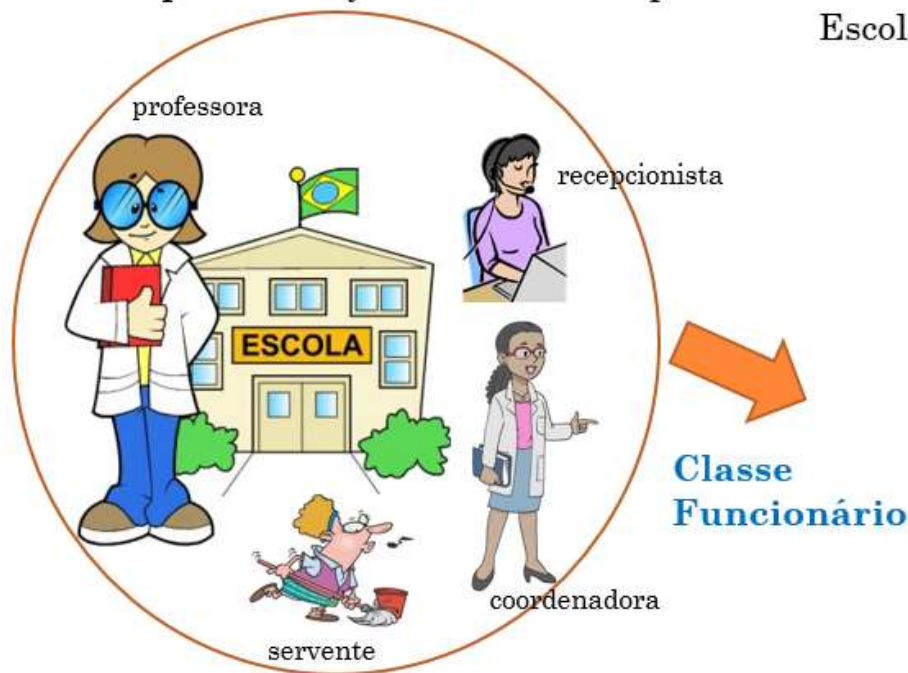
Exemplo 1:

Conjunto de objetos



**Classe Veiculo**

## Exemplo 2: Conjunto de Pessoas que trabalham na Escola



Classe Funcionário e seus **atributos**

Funcionário
+ Matrícula
- Nome
- Datanasc
- CPF
- Endereço
- Cidade
- CEP
- Função
- Telefone

**Atributos:** são as informações que os objetos armazenam.

**Visibilidade:**

- + público: visível em qualquer classe de qualquer pacote
- # protegido: visível para classes do mesmo pacote
- privado: visível somente para classe

**Exemplo:**

+ nome : String

Padrões para Nomes de Atributos

- Sem hífen, tracinhos ou espaços
- Sem Preposições (de, para, do, com, etc)
- 1ª letra minúscula (para o caso de vários nomes, a 1ª letra dos demais nomes devem ser maiúsculas).

**Exemplos: idade**

**nomeCliente**

**sldDevedor**

**qtdAlunosMatriculados**

Classe Funcionário e suas **operações**

Funcionário
+ Matrícula
- Nome
- Datanasc
- CPF
- Endereço
- Cidade
- CEP
- Função
- Telefone
+ gravar()
+ alterar()
+ excluir()
+ consultar()

**Operações:** são as ações que os objetos executam.

**Observação:** A implementação dessas operações são chamadas de método.

## Visibilidade:

- + público: visível em qualquer classe de qualquer pacote
- # protegido: visível para classes do mesmo pacote
- privado: visível somente para classe

## Exemplo:

- getName() : String

**Mensagens e Métodos:** forma de comunicação entre os objetos.



## Padrões para Nomes de Operadores

- Sem hífen, tralinhos ou espaços
- Sem Preposições (de, para, do, com, etc)
- Verbos no infinitivo (...ar, ...er, ...ir, etc)
- Sempre colocar Verbo + Sujeito
- 1ª letra minúscula (para o caso de vários nomes, a 1ª letra dos demais nomes devem ser maiúsculas).

**Exemplos:** calcularSaldoDevedor  
imprimirBoleto

Cliente
nome
dataNascimento
imprimirCliente()
calcularIdade()

## O que é uma Instância?

Representa cada ocorrência de um objeto formada a partir de uma classe.

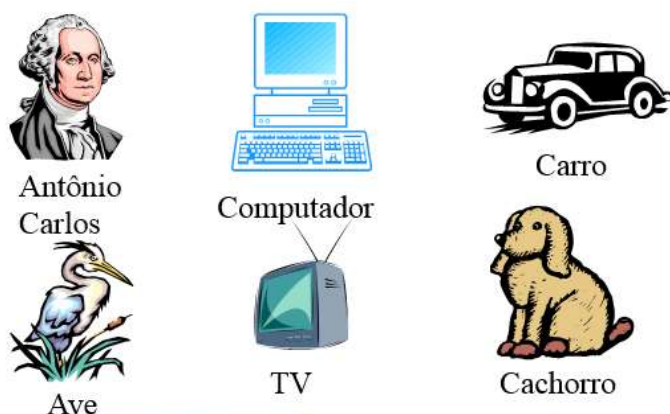


Exemplos:

Classe	Instâncias (Objetos)
Funcionário	Funcionário Ana Cláudia Funcionário Gabriel
Empresa	Empresa C&A Empresa Kibon
Veículo	Veículo Santana Veículo Eco Esport

## Exercício

1-Quantas classes podemos definir abaixo?



2- Das classes que você criou, crie:

Classe	Atributos	Métodos(comportamento)	Duas Instancias

Visto: \_\_\_\_\_

## O que é UML?

A **Unified Modeling Language (UML)** é uma linguagem de **modelagem** usada para auxiliar na visualização e desenho de sistemas de informação.

Linguagem Gráfica de Modelagem para:

- ✓ Visualizar
- ✓ Especificar
- ✓ Comunicar
- ✓ Construir
- ✓ Documentar

Artefatos de sistemas complexos

\_ Linguagem: vocabulário + regras de Combinação

## Objetivos da Modelagem

- Compreender melhor o sistema que estamos desenvolvendo
- Visualizar o sistema
- Documentar decisões tomadas
- Especificar comportamento ou a estrutura de um sistema

## Modelos e Diagramas

Um modelo é uma representação em pequena escala, numa perspectiva particular, de um sistema existente ou a criar

- Atitude de abstracção (omissão de detalhes) fundamental na construção de um modelo
- Modelos são a linguagem por excelência do projectista (designer)
- Modelos são veículos para comunicação com vários interessados (stakeholders)
- Modelos permitem raciocinar acerca do sistema real, sem o chegar a construir

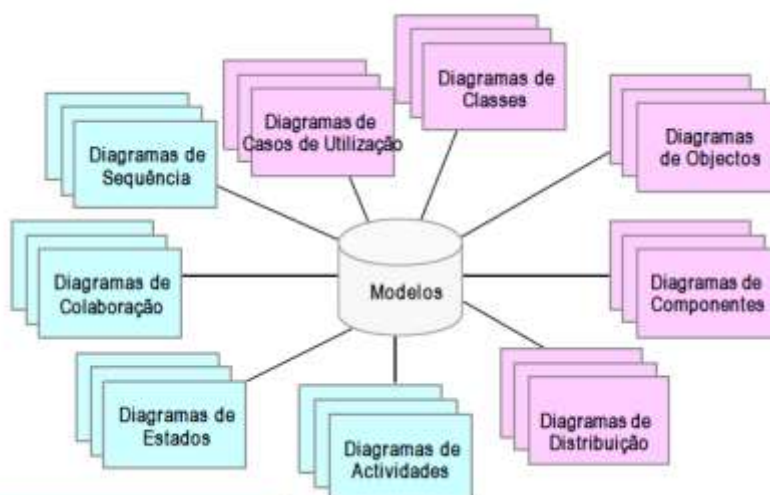
Ao longo do ciclo de vida de um sistema são construídos vários modelos, sucessivamente refinados e enriquecidos.

Um modelo é constituído por um conjunto de diagramas (desenhos) consistentes entre si, acompanhados de descrições textuais dos elementos que aparecem nos vários diagramas

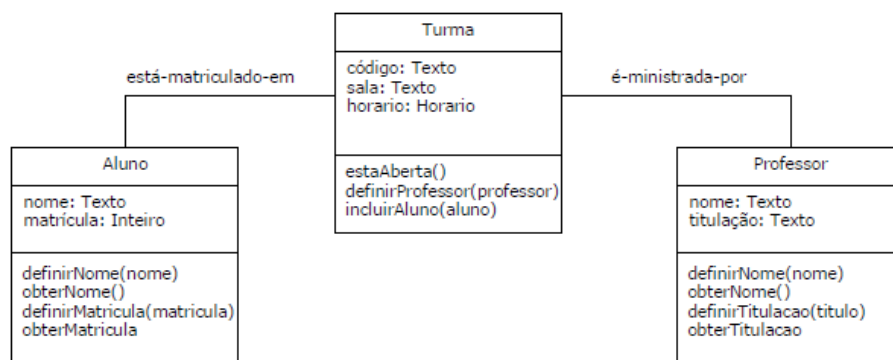
- Um diagrama é uma vista sobre um modelo
- O mesmo elemento (exemplo: classe) pode aparecer em vários diagramas de um modelo

## No UML, há nove diagramas standard

- Diagramas de visão estática:
  - casos de utilização (use case),
  - classes,
  - objetos,
  - componentes,
  - distribuição (deployment)
- Diagramas de visão dinâmica:
  - sequência,
  - colaboração,
  - estados (statechart),
  - atividades



O diagrama é usado para enfatizar os aspectos lógicos das classes, em vez de sua implementação. É o diagrama central da modelagem orientada a objetos.



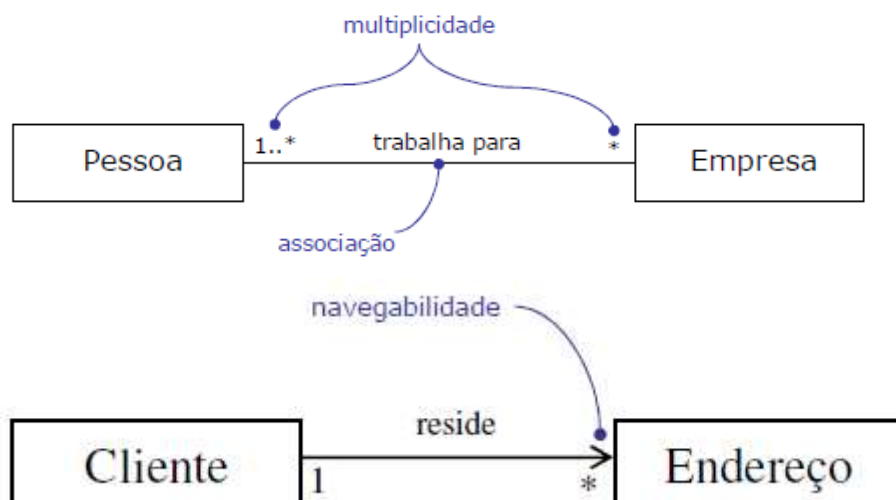
### Elementos de um diagrama de classes

- Classes
- Relacionamentos
  - Associação
  - Agregação
  - Composição
  - Generalização
  - Dependência

### Relacionamentos

Os relacionamentos possuem:

- **Nome:** descrição dada ao relacionamento (faz, tem, possui,...)
- **Sentido de leitura**
- **Navegabilidade:** indicada por uma seta no fim do relacionamento
- **Multiplicidade:** 0..1, 0..\*, 1, 1..\*, 2, 3..7
- **Tipo:** associação (agregação, composição), generalização e dependência
- **Papéis:** desempenhados por classes em um relacionamento





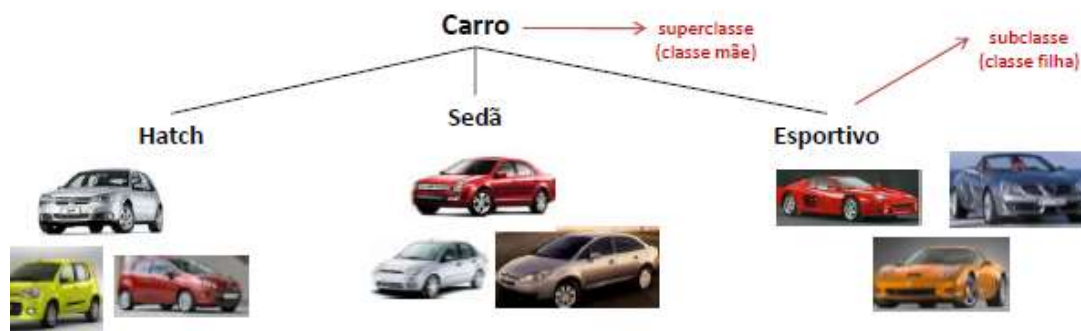
O cliente sabe quais são seus endereços, mas o endereço não sabe a quais clientes

**Ligações e Associações:** relacionamentos entre objetos e classes (respectivamente).



pertence.

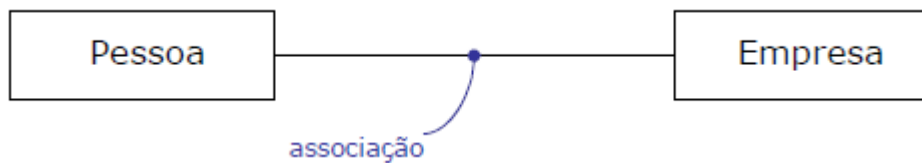
**Herança:** descrevem um conjunto de objetos com as mesmas propriedades (atributos e associações) e o mesmo comportamento (operações).



## Associação entre Classes

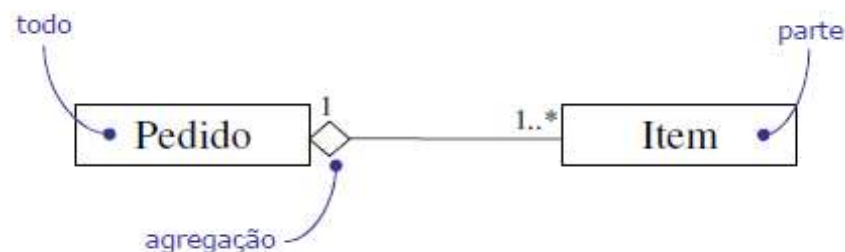
Uma associação é um relacionamento estrutural que indica que os objetos de uma classe estão vinculados a objetos de outra classe.

- Uma associação é representada por uma linha sólida conectando duas classes.



## Relacionamentos- Agregação

É um tipo especial de associação  
– Utilizada para indicar “todo-parte”.

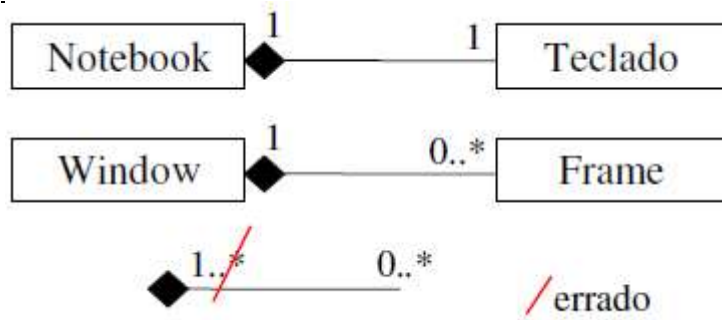


- Um objeto “parte” pode fazer parte de vários objetos “todo”.

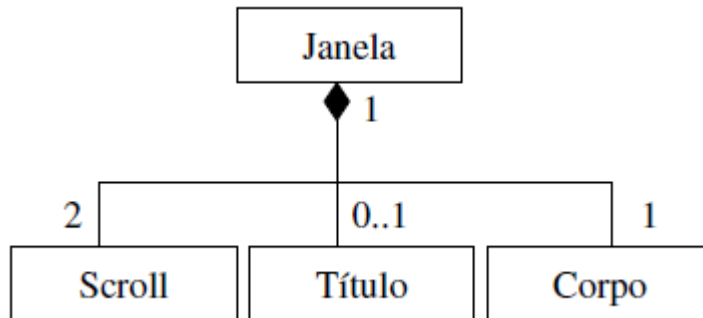
## Relacionamento: Composição

- É uma variante semanticamente mais “forte” da agregação.
- Os objetos “parte” só podem pertencer a um único objeto “todo” e têm o seu tempo de vida coincidente com o dele.

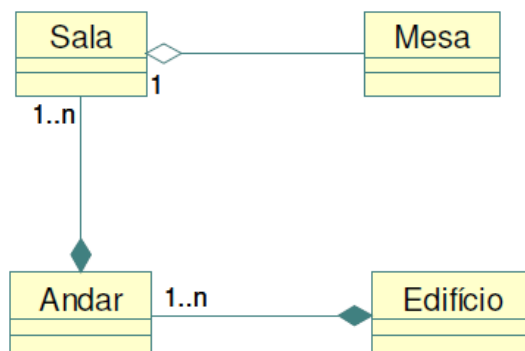




– Quando o “todo” morre todas as suas “partes” também morrem  
Exemplo:

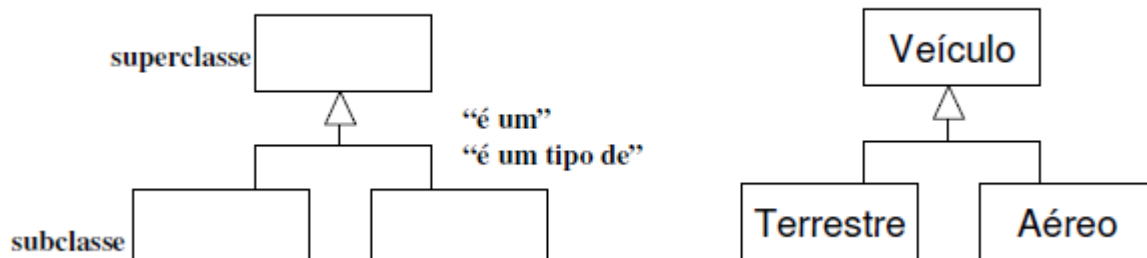


Agregação X Composição



Relacionamento: Generalização

• É um relacionamento entre itens gerais (superclasses) e itens mais específicos (subclasses)



Relacionamento: Dependência

• Representa que a alteração de um objeto (o objeto independente) pode afetar outro objeto (o objeto dependente)



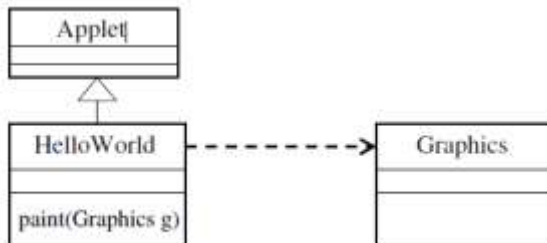
Obs:

- A classe cliente depende de algum serviço da classe fornecedor
- A mudança de estado do fornecedor afeta o objeto cliente
- A classe cliente não declara nos seus atributos um objeto do tipo fornecedor
- Fornecedor é recebido por parâmetro de método

## No Java

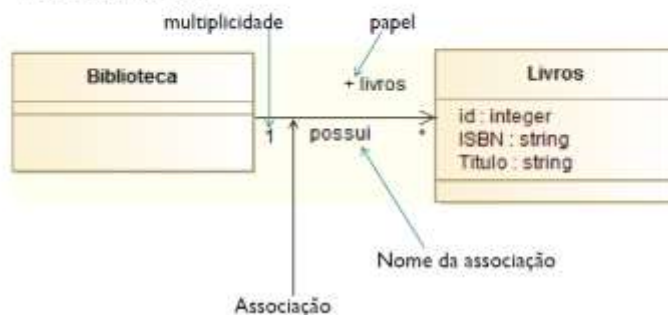
```

import java.awt.Graphics;
class HelloWorld extends java.applet.Applet
{
    public void paint (Graphics g)
        g.drawString("Hello, world!", 10, 10);
}
  
```

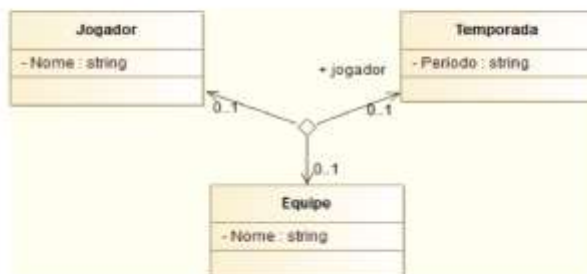


## Associação Binária

- Uma associação binária é representada por uma linha unindo as duas classes relacionadas

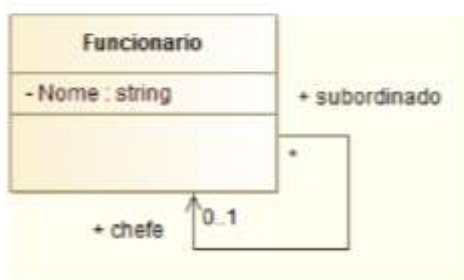


## Associação N-ária



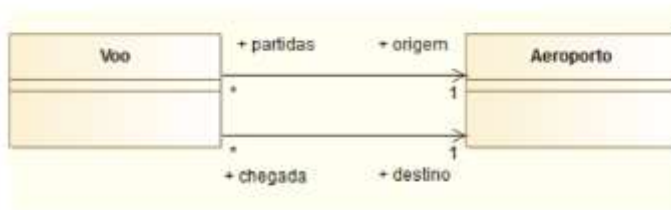
Temporada	Jogador	Equipe
00/01	Junino	Vasco
00/01	Diego	Fluminense
01/02	Diego	Vasco
01/02	Juninho	Vasco

## Associação Reflexiva



Matricula	Nome	Chefe
1	Ricardo	
2	Marcelo	1
3	Marcia	1
4	Tiago	2

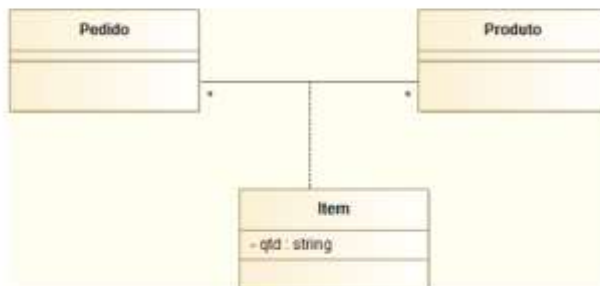
## Múltiplas Associações



Na utilização de mais de uma associação é obrigatório realizar a distinção entre elas, seja utilizando papéis ou com nome na associação.

## Classes associativas

- É uma associação que também é uma classe, ou seja, uma associação que possui atributos e operações.

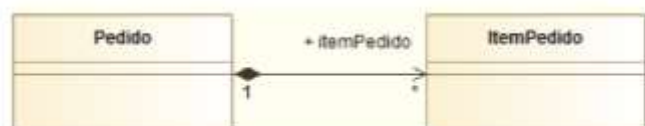


## Composição

### Classes associativas

- ID do objeto associativo é uma composição dos ID's dos objetos associados.
  - Não pode haver dois objetos da classe associativa relacionados com o mesmo par de objetos
- Exemplo:
  - Pedido e Produto
    - Uma instância produto só pode aparecer uma vez na instância de um determinado pedido.
    - Não é histórico

- A “parte” pode compor apenas 1 “todo” em determinado instante do tempo
- O “todo” é responsável pela vida de suas partes (criação e destruição)
- A “parte” não tem vida independente do “todo”



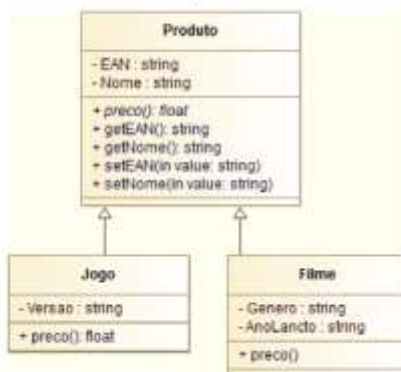
## Agregação

- A “parte” pode estar em mais de um “todo”
- A “parte” tem vida independente do “todo”



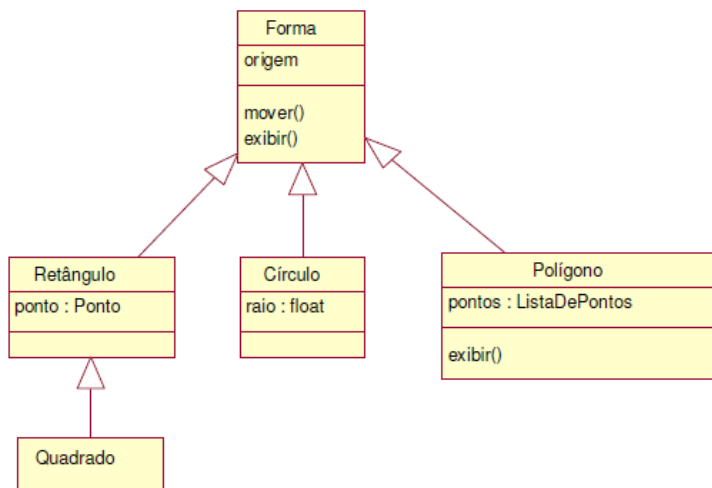
## Herença ou Generalização

- Define o relacionamento entre classes, onde uma classe compartilha a estrutura e comportamento de uma ou mais classes



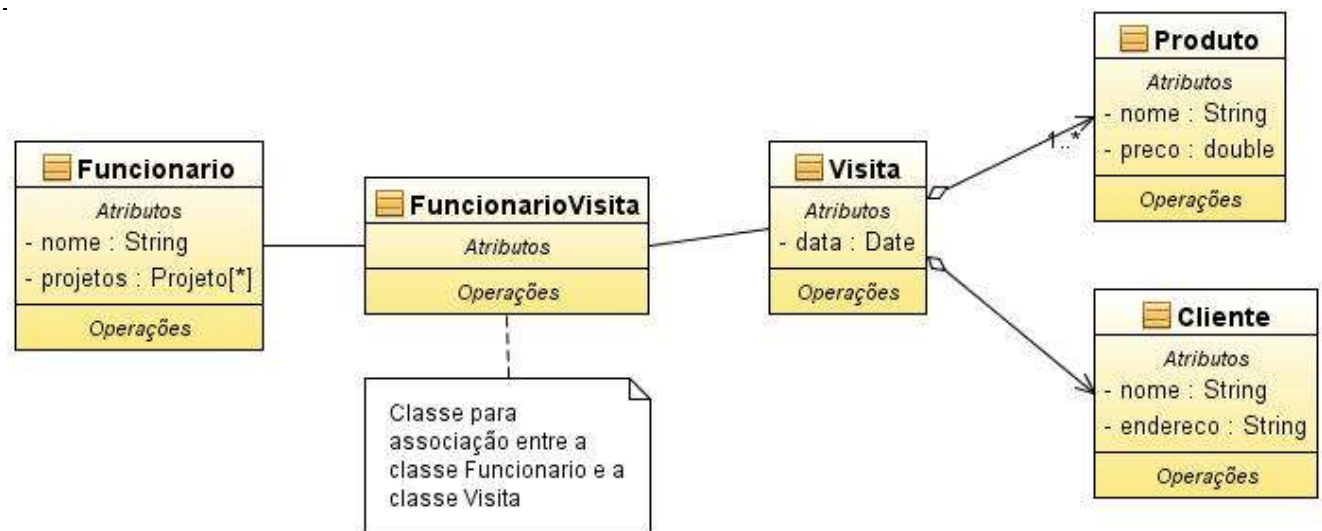
- Uma classe também pode ser abstrata.

Outro exemplo:



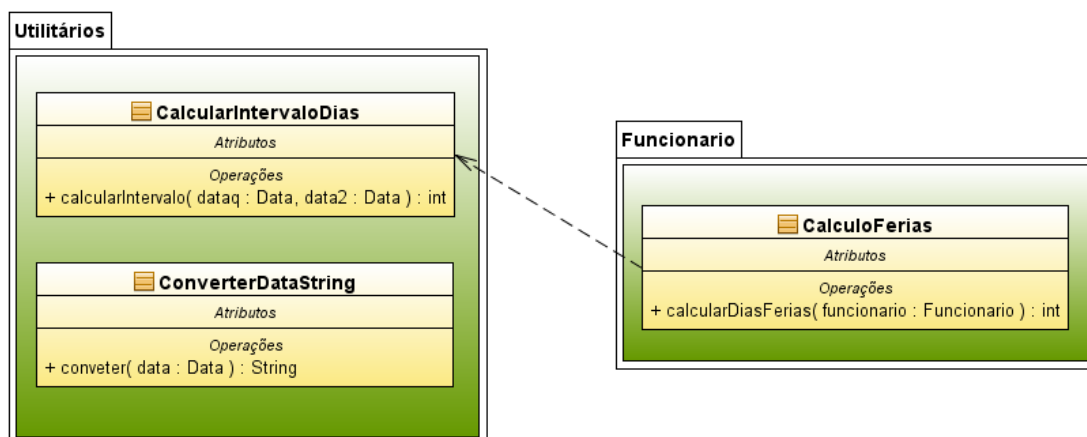
## Exemplo Prático

- Crie um diagrama de classes UML para o sistema de Agendamento de Entrega de Produtos:
  - O sistema deve permitir o agendamento de uma visita por parte de um funcionário, ao cliente de nossa empresa. Esta visita deverá ser realizada em hora e local definido pelo cliente para poder lhe entregar os produtos que foram comprados em nossa loja.



## Pacote

- Utilizamos para organizar as classes:



## Exercícios

1- Organize hierarquicamente em um diagrama as seguintes classes:

MembroDaUniversidade, Professor, Coordenador, ~~Empregado~~, Ex-aluno, CorpoDiscente, AlunoDeGraduacao, AlunoDePosGraduacao, Funcionario, CorpoDocente.

Visto: \_\_\_\_\_

2. Organize hierarquicamente em um diagrama as seguintes classes:

VeiculoTerrestre, VeiculoAquático, VeiculoAutoMotor, Carro, Moto, Barco, NavioCargueiro, Caminhão, VeiculoAnfibio, Submarino, VeiculoAereo, Aviao, HidroAviao.

Visto: \_\_\_\_\_

3- Fazer o Diagrama de Classes do Sistema Bancário abaixo:

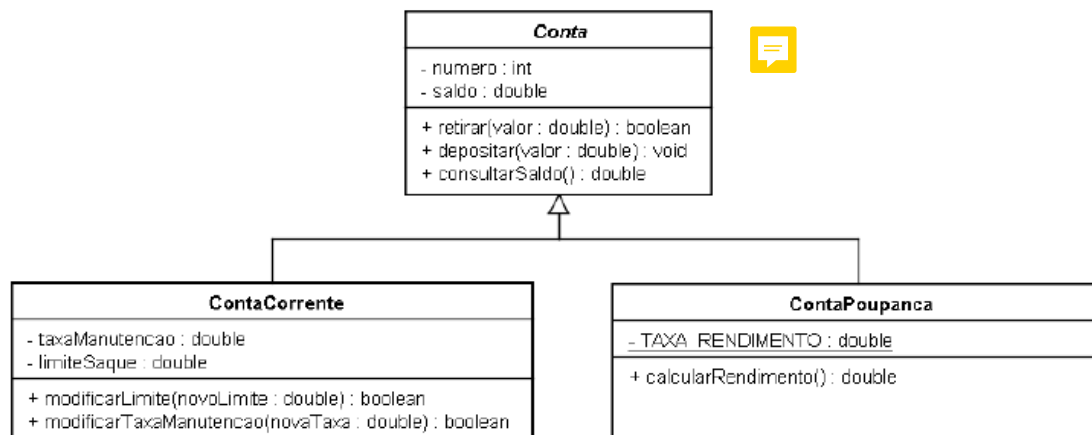
Um cliente poderá abrir uma conta, depositar, fazer saques (manual e automático), consultar saldo, alterar sua senha e aplicar em fundos (em terminais ou Internet).

O gerente fará a abertura da conta e a consulta ao saldo do cliente.

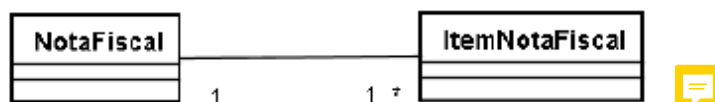
O caixa fará o saque manual, a alteração da senha e a consulta ao saldo.

Visto: \_\_\_\_\_

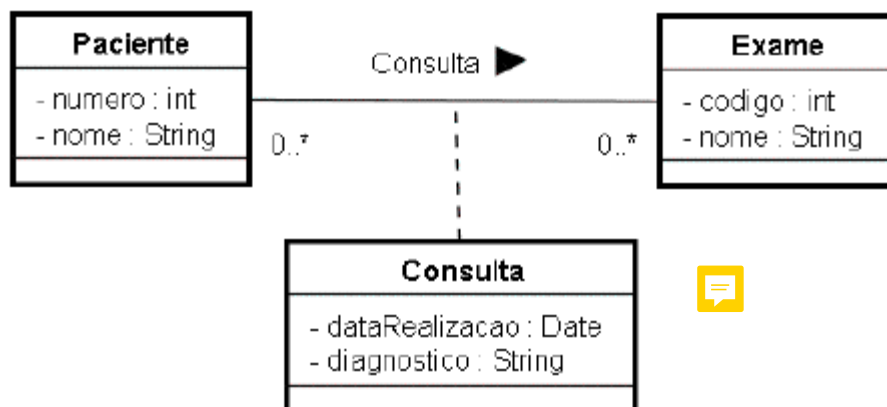
4-Identifique os tipos de associação:



a) \_\_\_\_\_

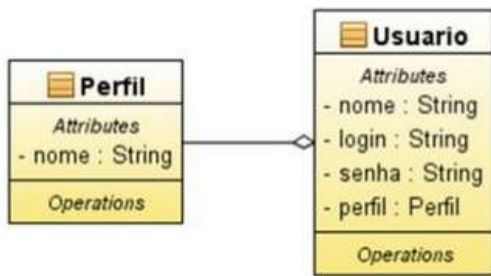


b) \_\_\_\_\_



c) \_\_\_\_\_





d) \_\_\_\_\_

5-Explique a multiplicidade dos Relacionamentos abaixo:



a)




b)




c)


6-Faça os Diagramas de Classe para os Estudos de Caso abaixo:

a)

A gráfica ABC trabalha com diversos autores que escrevem os livros que ela publica. Alguns autores escreveram apenas um livro, enquanto outros já escreveram vários. Além disso, alguns livros foram escritos por vários autores, porém um livro deve possuir pelo menos um autor. A gráfica ABC trabalha também com diversas impressoras, porém um livro só pode ser impresso em uma única impressora. Deve ser possível buscar os livros de um autor, saber qual foi a impressora em que o livro foi impresso, saber a quantidade de páginas que uma impressora já imprimiu e a quantidade de páginas que ela imprimiu de um determinado livro, entre outros.

Faça em folha separada ou no verso da apostila.

Visto: \_\_\_\_\_

b)

Uma pessoa possui cpf e nome. Um professor é uma pessoa que também possui uma titulação e a IES (Instituição de Ensino Superior) vinculada. Um aluno também é uma pessoa e possui a informação de qual período se encontra e qual o seu curso (um curso possui um registro do MEC, um nome e sua área de concentração).

Uma palestra possui um nome e um assunto e pode ser ministrada por diversos professores.

Para cada professor ministrando cada palestra, deve constar a data e a localização (rua, número, bairro, cidade, estado e telefone) deste evento e, um aluno, pode assistir vários destes eventos.

Pelo menos as seguintes operações devem existir:

- ☐ Alterar o nome de uma pessoa;
- ☐ Alterar a titulação de um professor;
- ☐ Alterar a IES de um professor;
- ☐ Alterar o período em curso de um aluno;
- ☐ Alterar o curso de um aluno;
- ☐ Alterar o registro do MEC de um curso;
- ☐ Alterar o nome de uma palestra;
- ☐ Alterar a data de um evento;
- ☐ Alterar a localização de um evento;
- ☐ Alterar o telefone de uma localização

*Faça em folha separada ou no verso da apostila.*

Visto: \_\_\_\_\_

c)

Um imóvel possui um endereço, área (em m2) e um proprietário. O imóvel pode ser uma casa ou um apartamento. Caso seja casa possuirá também o número do registro do lote e, caso seja apartamento, possuirá o número do andar e um flag indicando se possui ou não elevador. Um proprietário possui cpf, nome e telefone.

Pelo menos as seguintes operações devem existir:

- ☐ Alterar a área de um imóvel
- ☐ Alterar o proprietário de um imóvel
- ☐ Alterar o flag se possui ou não elevador do apartamento
- ☐ Recuperar a quantidade de imóveis de um proprietário
- ☐ Alterar o telefone de um proprietário
- ☐ PS: Atenção com os parâmetros e com o tipo de retorno das operações

*Faça em folha separada ou no verso da apostila.*

Visto: \_\_\_\_\_

## Diagramas de Caso de uso (use cases)

- São utilizados para modelar os requisitos de um sistema. Ele descreve as principais funcionalidades do sistema e a interação dessas funcionalidades com os usuários do mesmo sistema. Nesse diagrama não nos aprofundamos em detalhes técnicos que dizem como o sistema faz.

## Exemplo pratico

Vamos criar um cenário de exemplo para vermos a notação de um diagrama de caso de uso:

“A clínica médica Saúde Perfeita precisa de um sistema de agendamento de consultas e exames. Um paciente entra em contato com a clínica para marcar consultas visando realizar um check-up anual com seu médico de preferência. A recepcionista procura data e hora disponível mais próxima na agenda do médico e marca as

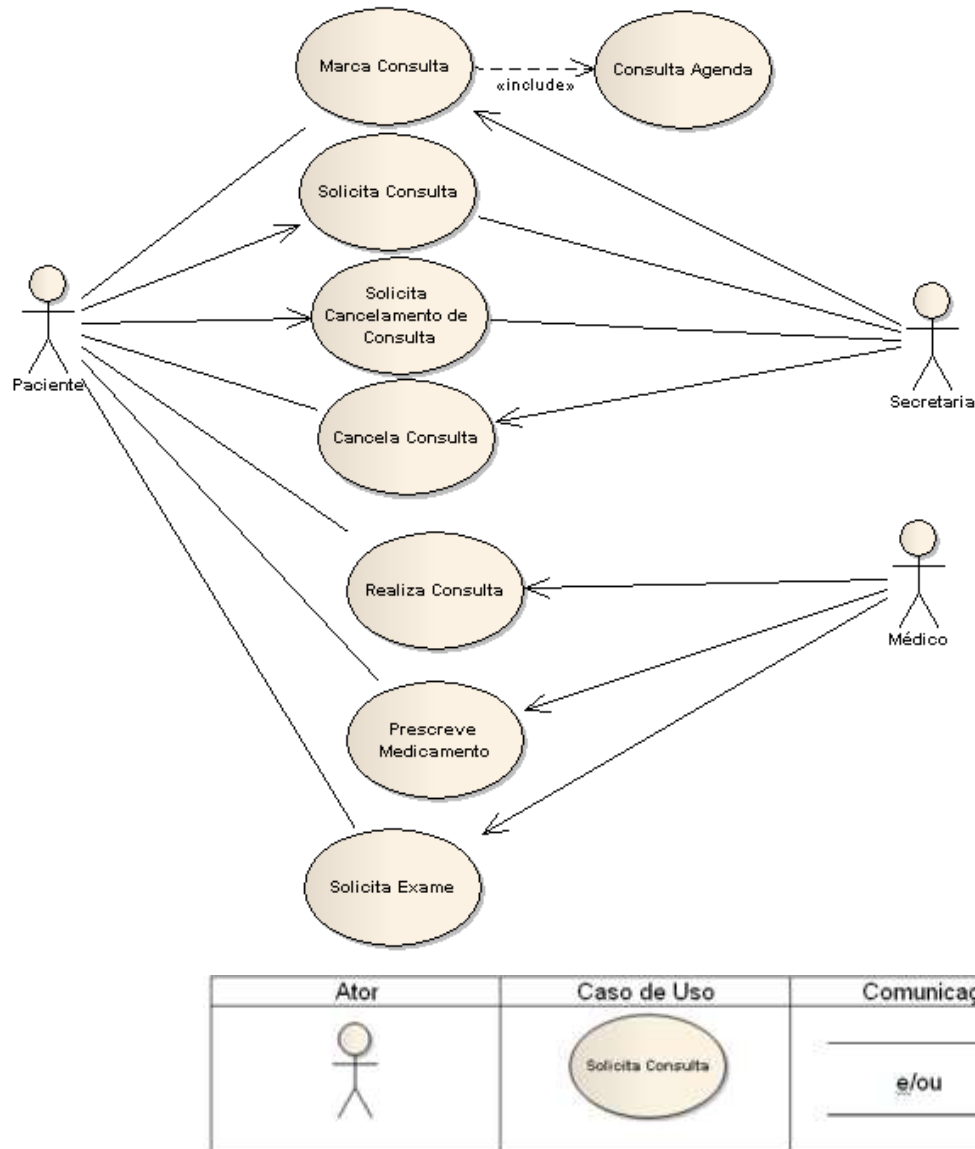
consultas. Posteriormente o paciente realiza a consulta, e nela o médico pode prescrever medicações e exames, caso necessário”.

Com esse cenário simples podemos começar a criar nosso diagrama. Inicialmente vamos definir nossos atores:

- a) Paciente
- b) Secretária
- c) Médico

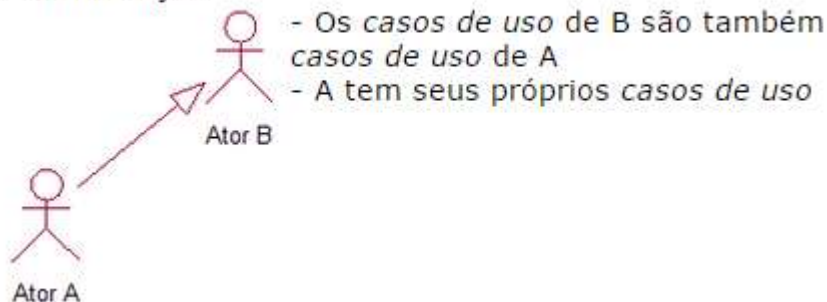
Agora vamos definir algumas ações de cada usuário:

- a) Paciente
  - Solicita Consulta
  - Solicita Cancelamento de Consulta
- b) Secretária
  - Consulta Agenda
  - Marca Consulta
  - Cancela Consulta
- c) Médico
  - Realiza Consulta
  - Prescreve Medicação
  - Solicita Realização de exames



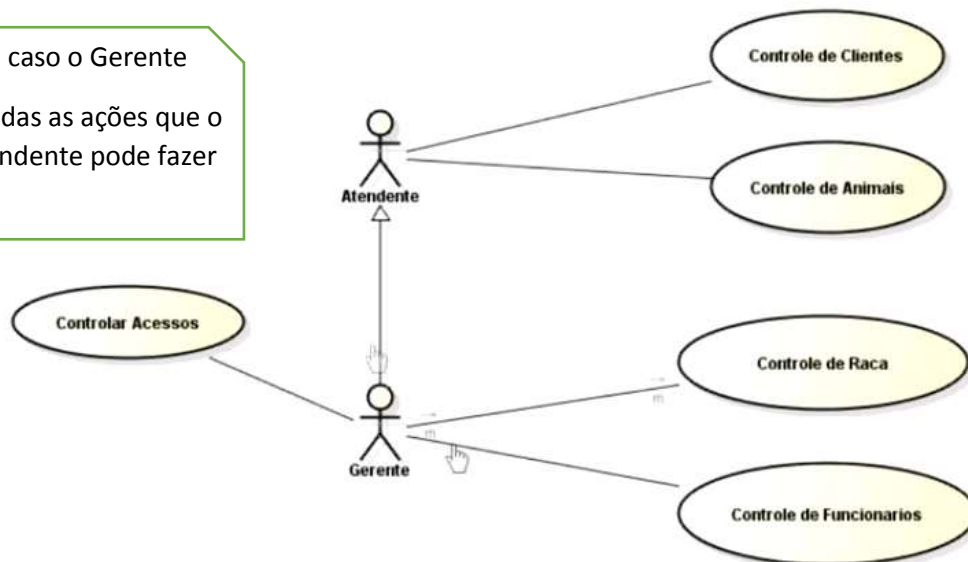
- **Ator** – um ator representa uma entidade que interage com o sistema que está sendo modelado.
- **Caso de uso** – descreve uma sequência de ações de um sistema.
- **Associação** – é representada por uma linha sólida ligando o caso de uso
- **Pode acontecer também a Generalização>>**

## Generalização



Nesse caso o Gerente

**Herda** todas as ações que o ator Atendente pode fazer



## Exercícios

1-Faça o Diagrama de Caso de Uso para o Estudo de Caso abaixo:

O 3D colégios oferecerá um curso on-line, onde os **professores** poderão publicar material de aula, visualizar, cadastrar, corrigir e excluir tarefas, associar alunos em determinadas disciplinas e criar tópicos no fórum de discussão.

Os **alunos** poderão enviar tarefas, visualizar o material disponibilizado, criar tópicos e interagir no fórum.

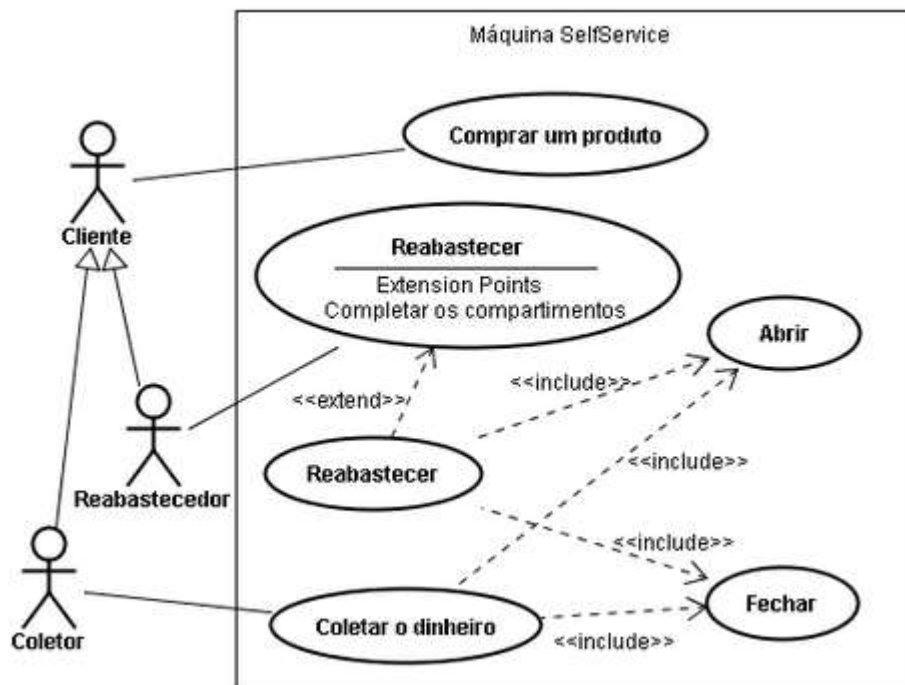
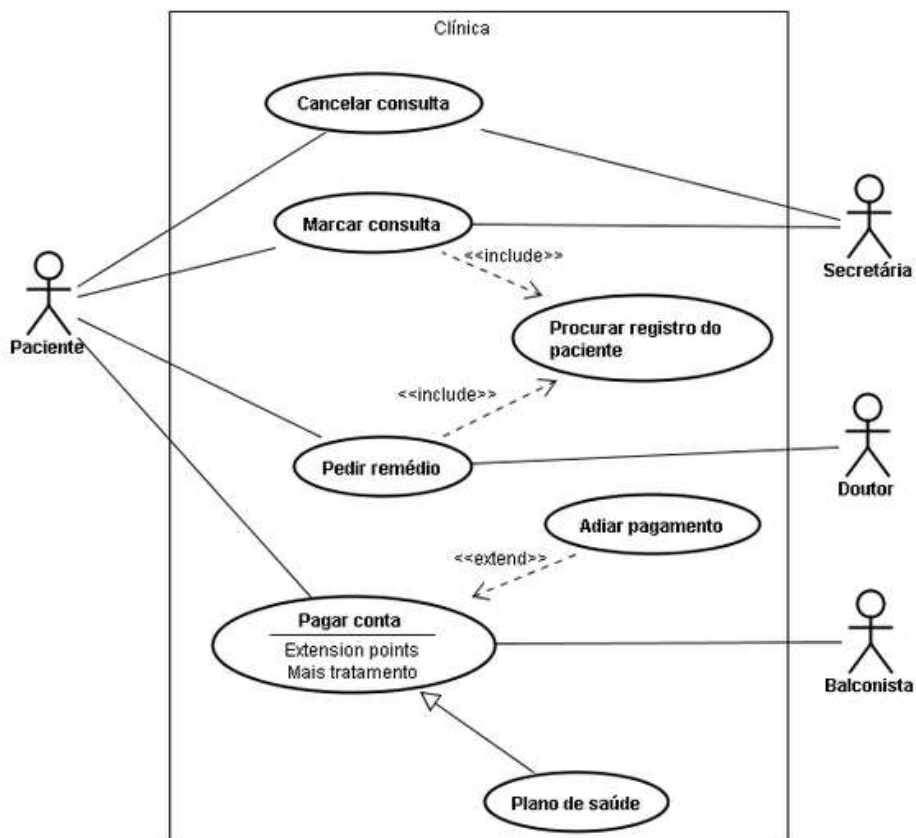
Os **professores** e **alunos** poderão se logar no site e obter a senha.

Visto: \_\_\_\_\_

## Melhorando o Diagrama de Caso de Uso

Para melhorar um pouco mais esse diagrama vamos ver o conceito de include>>. Include e extend são relações entre os casos de uso.

- **Include:** seria a relação de um caso de uso que para ter sua funcionalidade executada precisa chamar outro caso de uso.
- **Extend:** Esta relação significa que o caso de uso extendido vai funcionar exatamente como o caso de uso base só que alguns passos novos inseridos no caso de uso extendido.



## «extend» X «include»

<u>Include</u>	<u>Extend</u>
Aumentam o comportamento do caso de uso básico.	
O caso de uso incluído <b>sempre</b> é utilizado.	O caso de uso de extensão <b>pode</b> ser usado.
A seta do relacionamento é desenhado do caso de uso em execução para o caso de uso incluído. A seta do relacionamento indica que o caso de uso básico direciona o caso de uso incluído para a execução.	A seta do relacionamento é desenhada do caso de uso de extensão para o caso de uso em execução. A seta indica que o caso de uso de extensão está tomando a decisão se deverá interromper o caso de uso em execução.

### Exercícios – Prático (Deve ser feito no Laboratório)

1-Desenhe os diagramas de Caso de Uso da Clínica e da Máquina de SelfService no programa de Modelagem:

Visto: \_\_\_\_\_

2- Atividade do estudo de Caso abaixo:

- a) Identificar e relacionar os casos de usos e os atores
- b) Desenvolver o Diagrama de Casos de Uso

O Clube "ACME" é um clube familiar onde qualquer pessoa pode ser sócia desde que seja indicado por algum outro sócio e seja aprovado pela Diretoria do Clube. O solicitante deve entregar uma solicitação preenchida, anexando os documentos necessários e duas cartas de recomendação de algum sócio.

Ao ser aceito como sócio em reunião ordinária da Diretoria, o solicitante deve pagar a primeira anuidade. Feito isso, o novo sócio receberá sua carteira de sócio, que deverá apresentar na portaria sempre que ingressar nas dependências do clube. A cada 12 meses, evidentemente, uma nova anuidade deve ser paga. Para isso, o sócio recebe em casa um boleto de cobrança bancária.

O clube mantém chalés para uso exclusivo dos sócios. O sócio deve requisitar o chalé com pelo menos 15 dias de antecedência. Não havendo chalé disponível na data solicitada, o sócio entrará em uma fila, sendo avisado quando houver disponibilidade. Sempre que se hospedar num chalé, o sócio deverá pagar o que foi consumido. Não o fazendo, será impedido de solicitar outro chalé até a quitação dos débitos.

Visto: \_\_\_\_\_



## Especificação do Caso de Uso

---

**Nome:** Comprar produto.

**Pré-condições:** Cliente com acesso a internet na página inicial do comércio eletrônico.

**Cenário Principal de Sucesso:**

1. O cliente navega pelo catálogo e seleciona itens para comprar.
2. O cliente vai para o caixa, isto é, fecha o carrinho de compra.
3. O cliente preenche formulário de remessa (forma de pagamento; endereço da entrega; opção de entrega imediata ou em três dias).
4. O sistema apresenta a informação completa do faturamento, incluindo a remessa.
5. O cliente autoriza a compra.
6. O sistema confirma imediatamente a venda.
7. O sistema envia uma confirmação para o cliente por e-mail.

**Extensões:**

**3a. Cliente regular:**

- .1: O sistema mostra a informação atual da remessa, a informação do preço e a informação de cobrança.
- .2: O cliente pode aceitar ou escrever por cima desses padrões, retomando ao CPS no passo 6.

**6a. O sistema falha na autorização de compra a crédito**

- .1: O cliente pode inserir novamente a informação do cartão de crédito ou cancelar.

**Pós-condições:** Cliente finaliza sua compra.

### Detalhar cada Caso de Uso

- Nome do caso de uso
- Resumo
- Pré-condição
- Descrição detalhada
  - Fluxo normal
- Fluxos Alternativos
  - Sub-fluxo 1:
  - Sub-fluxo 2:
- Exceções:
  - Exceção 1:
  - Exceção 2:
- Requisitos não funcionais (específicos do caso de uso)
- Requisitos de interface do usuário
- Regras de negócio (pode ser utilizado um documento à parte)

### Exemplo

Fluxo Principal	Fluxos Alternativos
1. O cliente chega no balcão com os livros que pretende emprestar.	<b>3a. O cliente não possui cadastro</b> 3a.1 O cliente informa seus dados para cadastro. 3a.2 O funcionário registra o cadastro. 3a.3 Retorna ao Fluxo Principal no passo 3.
2. O cliente se identifica e entrega os livros para o funcionário.	<b>3b. O cliente está com sit. pendente (Pgto em aberto)</b> 3b.1 O cliente efetua o pagamento. 3b.2 O funcionário registra pagamento, eliminando a pendência. 3b.3 Retorna ao Fluxo Principal no passo 3.
3. O funcionário registra o cliente e inicia o empréstimo.	<b>4a. O livro está reservado</b> 4a.1 O funcionário informa que o livro não pode ser emprestado pois possui reserva. 4a.2 Retorna ao Fluxo Principal no passo 4 sem registrar o livro reservado.
4. O funcionário registra cada livro.	<b>4b. O livro está danificado</b> 4b.1 O funcionário informa que o livro precisará de manutenção e não poderá ser emprestado. 4b.2 O funcionário registra que a fita está danificada. 4b.3 Retorna ao Fluxo Principal no passo 4, sem registrar o livro danificado.
5. O funcionário finaliza o empréstimo, entrega os livros de volta para o cliente avisando a data de devolução.	
6. O cliente vai embora com os livros.	

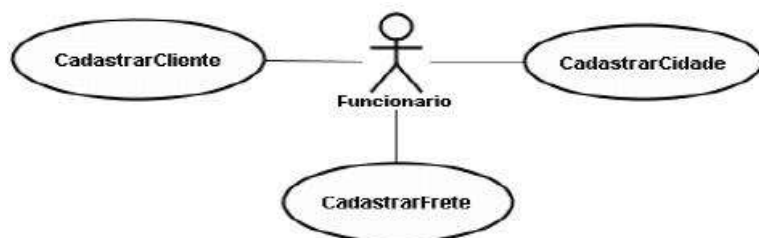
## Requisitos Especiais

Nos Requisitos Especiais de um caso de uso, você descreve todos os requisitos no caso de uso que não são abordados pelo fluxo de eventos. Esses requisitos não funcionais influenciarão o modelo do design. Você pode organizar esses requisitos em categorias como Usabilidade, Confiabilidade, Desempenho e Capacidade de Substituição, mas geralmente há tão poucos deles que esses agrupamentos não são particularmente úteis.

Exemplo:

- No Sistema da Máquina de Reciclagem, um requisito especial do caso de uso Devolver Itens de Depósito poderia ser:
- A máquina deve ser capaz de reconhecer os itens de depósito com uma confiabilidade de mais de 95%.

## Outro Modelo



Nome de 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 (EXO1).
		Sistema calcula o valor do frete e o exibe na janela (RNO1).
		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	EXO1	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	ALO1	O funcionário pode cancelar a operação de cadastramento de frete a qualquer momento, fechando a janela.
Regras de Negócio	RNO1	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.

## Contém:

**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 iterativo;
- **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.

## Exercícios

1-Para o estudo de Caso abaixo:

## Controle de alocação horas em projetos

A empresa ACME S/A necessita realizar o controle das horas trabalhadas pelos seus funcionários em cada projeto contratado.

Os funcionários devem informar o projeto em que trabalharam, as atividades desenvolvidas e o período de alocação.

Os gestores (gerentes e coordenadores) de cada projeto, devem aprovar ou reprovar as horas lançadas pelos funcionários nos seus respectivos projetos.

- A. Faça o Diagrama de Caso de uso
- B. Faça a especificação dos Casos de Uso

Visto: \_\_\_\_\_

- C. Faça no Programa de Modelagem o Diagrama Completo:

Visto: \_\_\_\_\_

**Agora que aprendemos alguns exercícios serão feitos em folha a parte. Tente resolve-los e alguns serão montados na Ferramenta de Modelagem.**

Tabela de Anotações de Exercícios

Descrição	Data de Entrega	Visto

### Ferramentas IDE

- Eclipse UML - <http://www.eclipseuml.com>
- Jude - <http://jude.change-vision.com>
- NetBeans – <http://www.netbeans.org>
- Magic Draw - <http://secure.nomagic.com>
- Microsoft Office Visio - <http://office.microsoft.com/pt-br/visio/default.aspx>
- Poseidon - <http://www.gentleware.com>
- Enterprise Architect - <http://www.sparxsystems.com.au/>