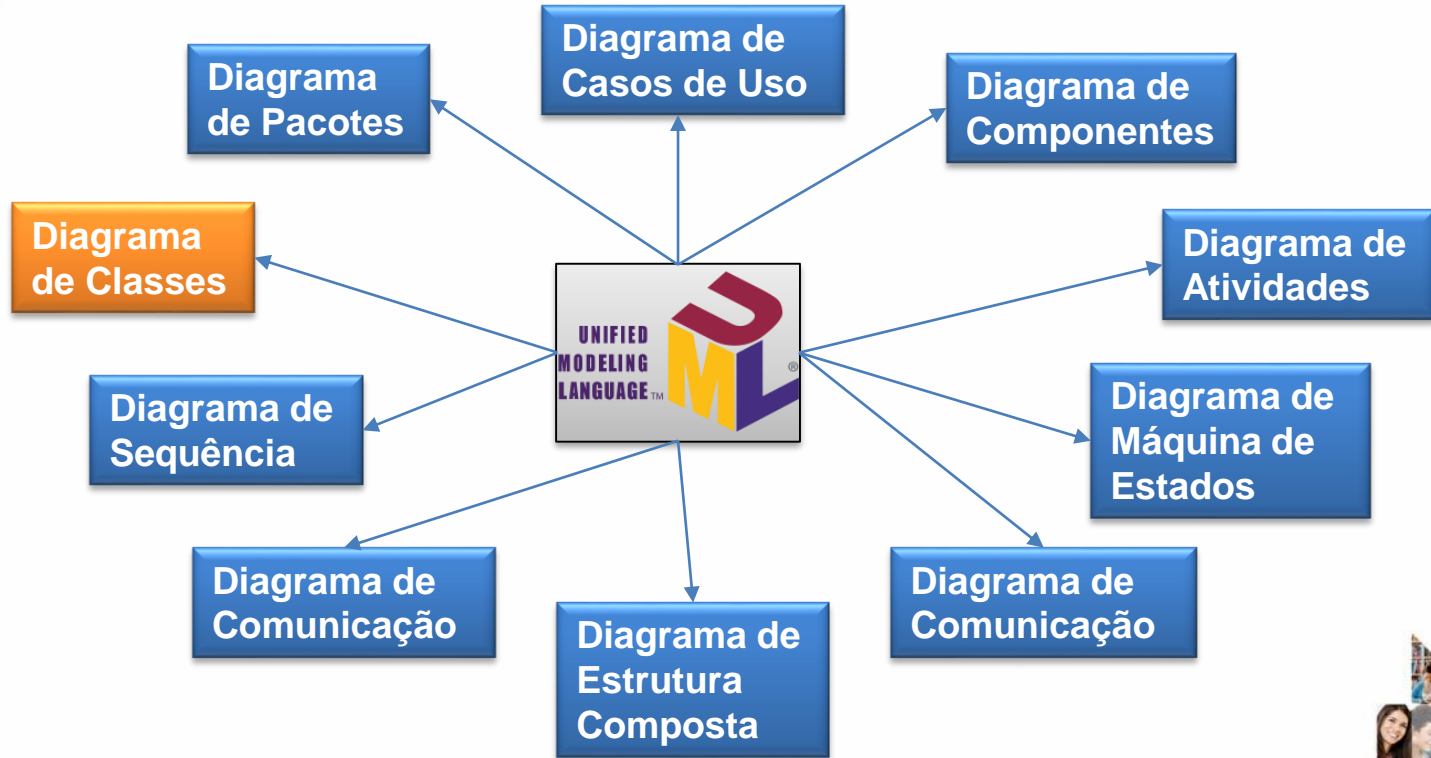


# Modelagem e Desenvolvimento Orientado a Objeto



## Modelos



## UML Modeladores

Open Source - Java

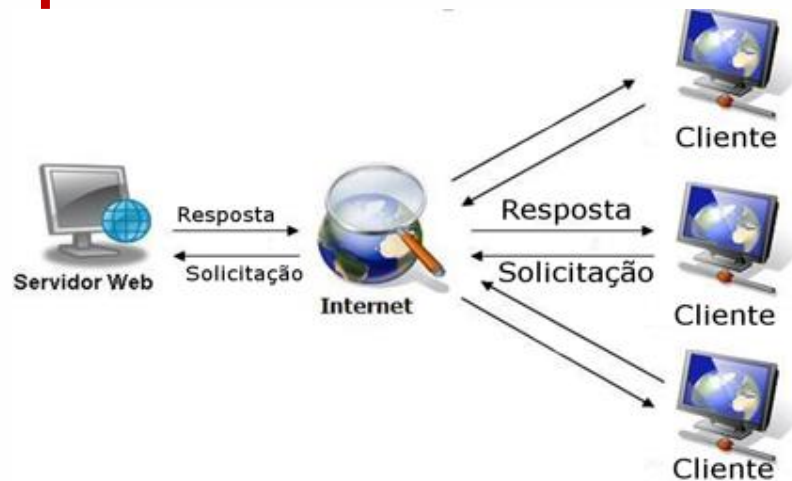


Versão Free – Recursos Bloqueados  
(descontinuada – set/2018)

astah  
community

# Modelagem e Desenvolvimento Orientado a Objeto

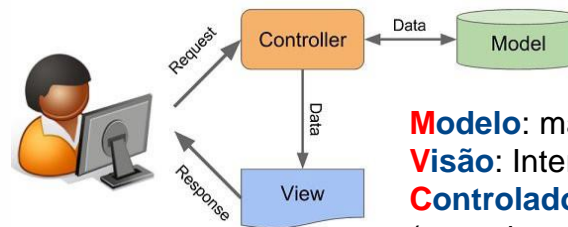
## Arquitetura Cliente / Servidor



### Arquitetura em Camadas



### Arquitetura MVC



**Modelo:** manipulação de dados.  
**Visão:** Interface gráfica do usuário.  
**Controlador:** tratamento de dados (pesquisar, listar, calcular, etc.)  
Interpretação de eventos.

CRUD (Create, Read, Update, Delete)

# Modelagem e Desenvolvimento Orientado a Objeto

## Diagrama de Classes

Na Orientação a Objetos, a representação de um conjunto de objetos com características (propriedades) e comportamentos (métodos) semelhantes é chamada de Classe. Sendo assim, a classe nada mais é que uma descrição do que o objeto possui e como o objeto atua.

### NomeDaClasse

- + PropriedadeClasse\_01 : int
- + PropriedadeClasse\_02 : String
- + PropriedadClasse\_03 : float

- + MetodoDaClasse\_01( ) : void
- + MetodoDaClasse\_02( ) : int
- + MetodDaClasse\_03(parametro01 : int) : float

# Modelagem e Desenvolvimento Orientado a Objeto

## Diagrama de Classes



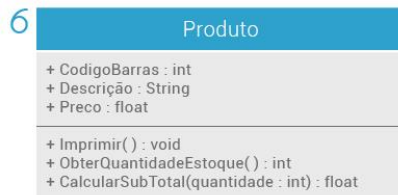
2 Defina o nome da classe, representada por um substantivo, com a primeira letra em maiúsculo.

3 As propriedades são definidas no segundo bloco da classe. Cada propriedade pode ter um tipo de dado específico, inclusive referenciado a outra classe. A propriedade também é representada por um substantivo.

5 Os sinais representam o encapsulamento de cada atributo / método da classe.



Os métodos são definidos no terceiro bloco da classe. Assim como as funções existentes na Análise Estruturada, eles podem retornar ou não algum resultado e podem ter parâmetros recebidos. Os métodos, por serem ações possíveis da classe, são representados por um verbo.



```
class Produto
{
    public int Codigobarras { get; set; }

    public string Descricao { get; set; }

    public float Preco { get; set; }

    public void Imprimir ()
    {
        // TODO - Escrever o método de impressão
    }

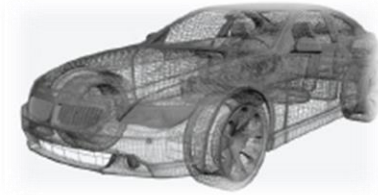
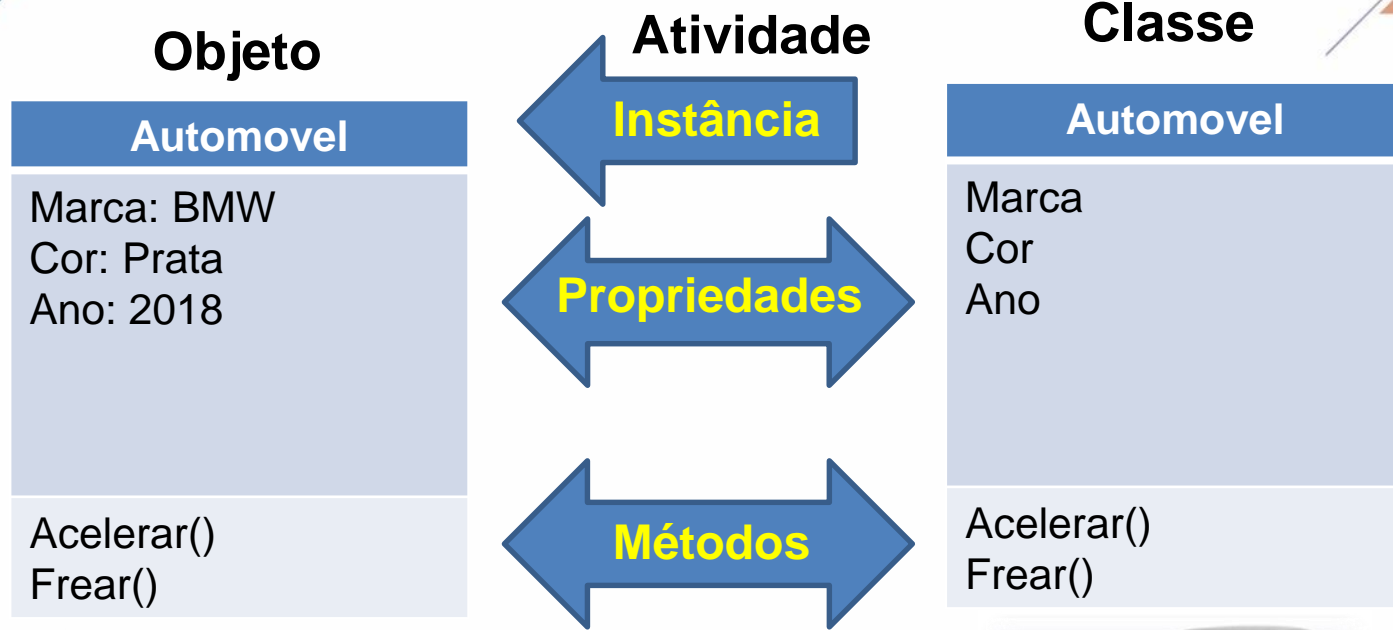
    public int ObtemQuantidadeEstoque()
    {
        // TODO - Escrever o método para obter a quantidade em estoque
        return 0;
    }

    public float CalcularSubTotal(int quantidade)
    {
        return quantidade * Preco;
    }
}
```



# Modelagem e Desenvolvimento Orientado a Objeto

## Diagrama de Classes



**Atividade**

**Abstração**

**Classe**

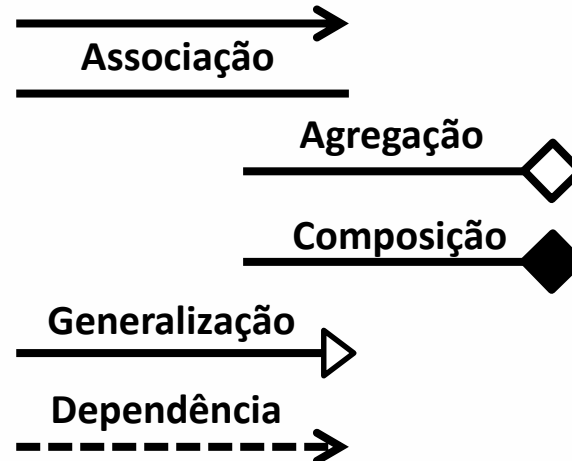
**Modelo ou Protótipo**

# Modelagem e Desenvolvimento Orientado a Objeto

## Diagrama de Classes

### Associações

Um sistema orientado a objeto é formado pela associação de diversas classes. Dessa maneira, representar essas associações se faz necessário para termos uma visão completa da estrutura do sistema. Existem diferentes tipos de associações entre classes. Cada uma delas tem um propósito específica e deve ser utilizada de acordo com o cenário que está sendo modelado.

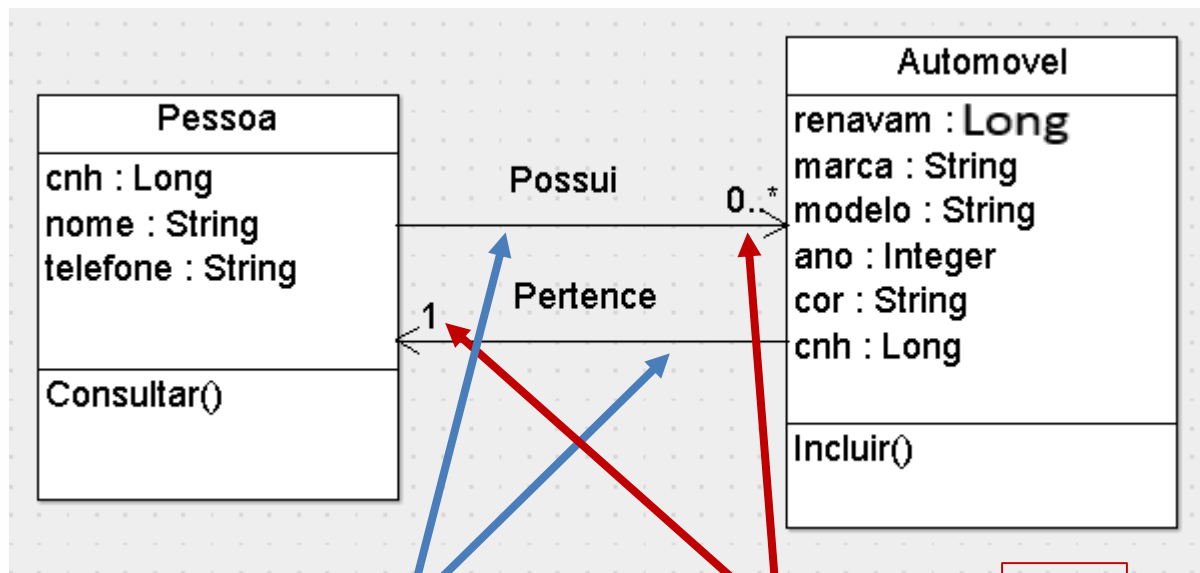




# Modelagem e Desenvolvimento Orientado a Objeto

## Diagrama de Classes

### Associação Binária ou Simples



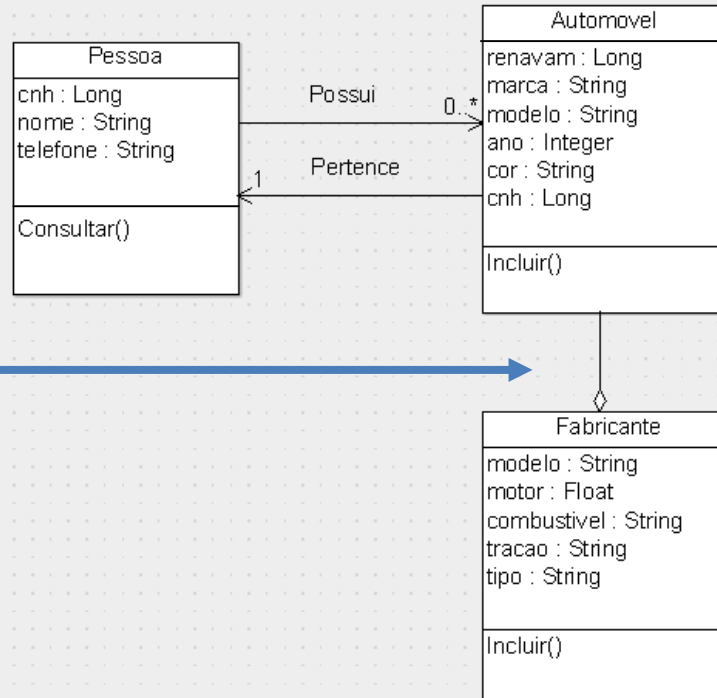
Associação

Multiplicidade  
ou Cardinalidade

1..1  
0..1  
1..\*  
0..\*

## Diagrama de Classes

### Associação - Agregação



Agregação

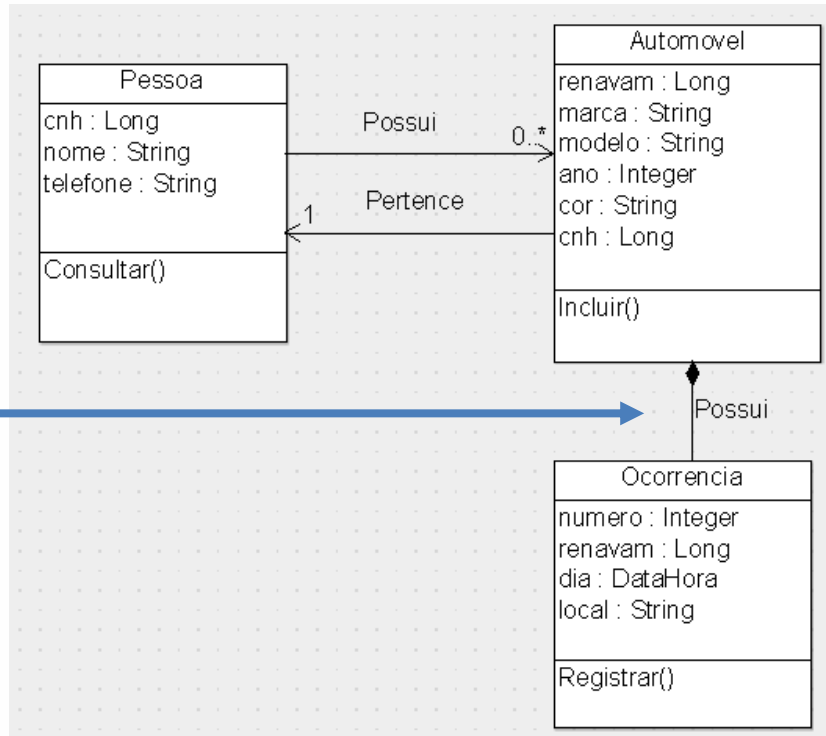
Uma classe faz parte da outra de forma não exclusiva.

Representação de relacionamentos do tipo "todo-parte".

# Modelagem e Desenvolvimento Orientado a Objeto

## Diagrama de Classes

### Associação - Agregação - Composição



Composição

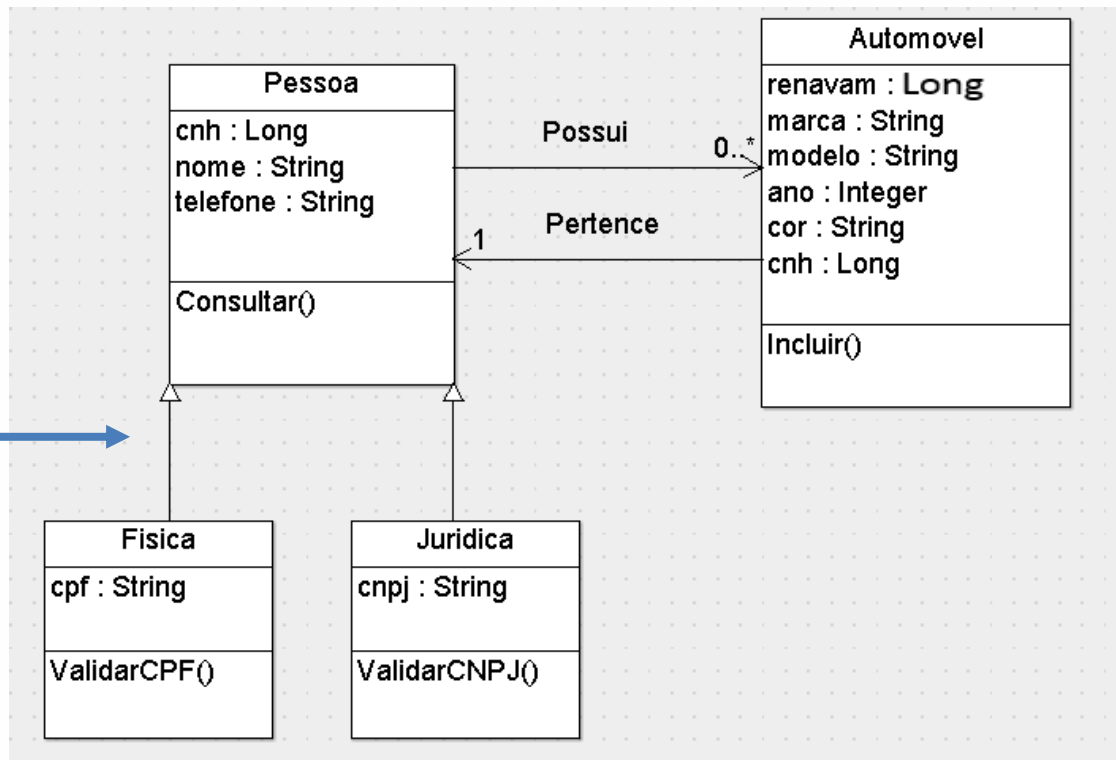
A composição é uma especialização da agregação, que indica que a classe que está compondo a entidade que se deseja representar será responsável pelo controle da classe que está sendo composta.

O todo (**Automovel**) é responsável pela existência da parte (**Ocorrencia**).

## Diagrama de Classes

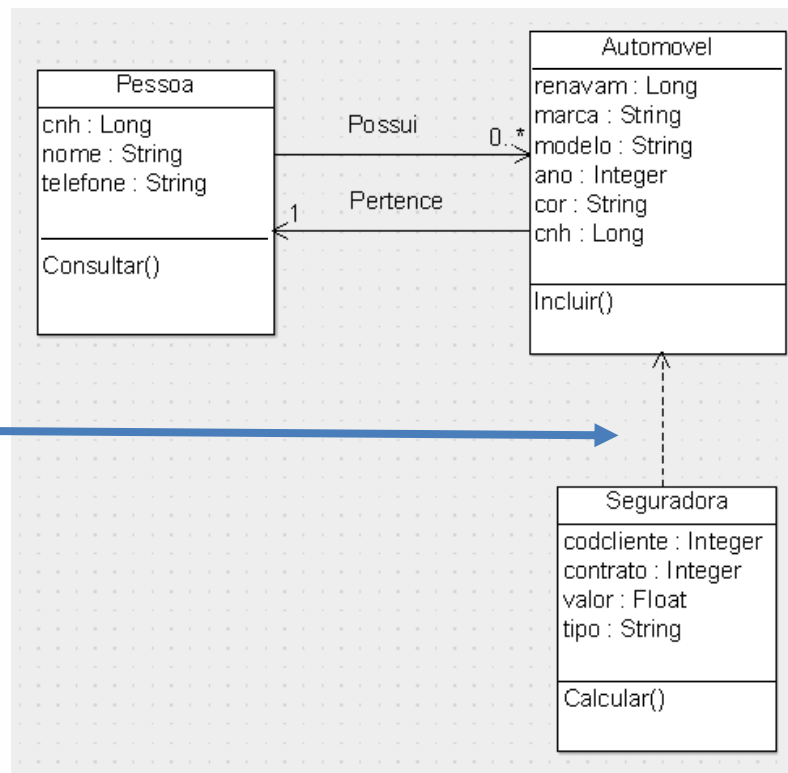
### Generalização

Generalização



## Diagrama de Classes

### Dependência



Dependência

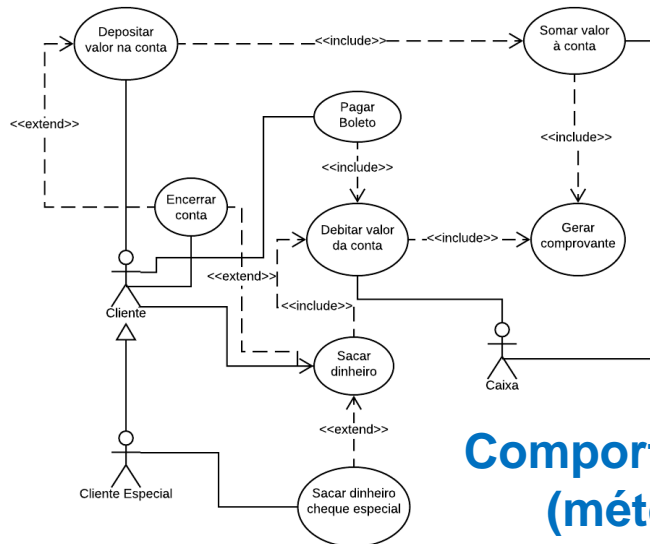
Uma dependência indica a ocorrência de um relacionamento entre dois ou mais elementos do modelo, onde uma classe cliente (Seguradora) é dependente de alguns serviços da classe fornecedora (Automovel), mas não tem uma dependência estrutural interna com esse fornecedor.

# Modelagem e Desenvolvimento Orientado a Objeto

## Diagrama de Casos de Uso e ER

### Atendimento Bancário

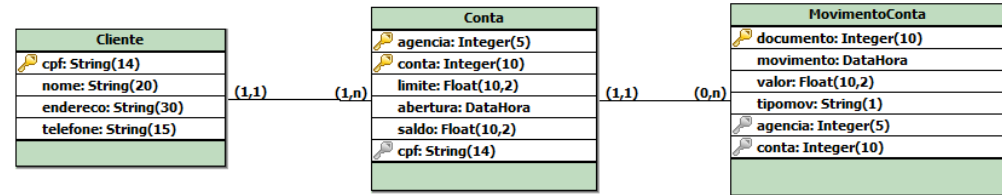
Para o desenvolvimento de um sistema, além do diagrama de Casos de Uso, será necessário uma perspectiva mais detalhada, assim, o diagrama de **Classes**, deverá vir acompanhado do diagrama de **Entidade Relacionamento (ER)** do Banco de Dados. Mais adiante será necessário o diagrama de **Atividades**.



**Comportamentos  
(métodos)**

### Entidade Relacionamento (ER) - Banco de Dados

#### Modelo Lógico ou Relacional

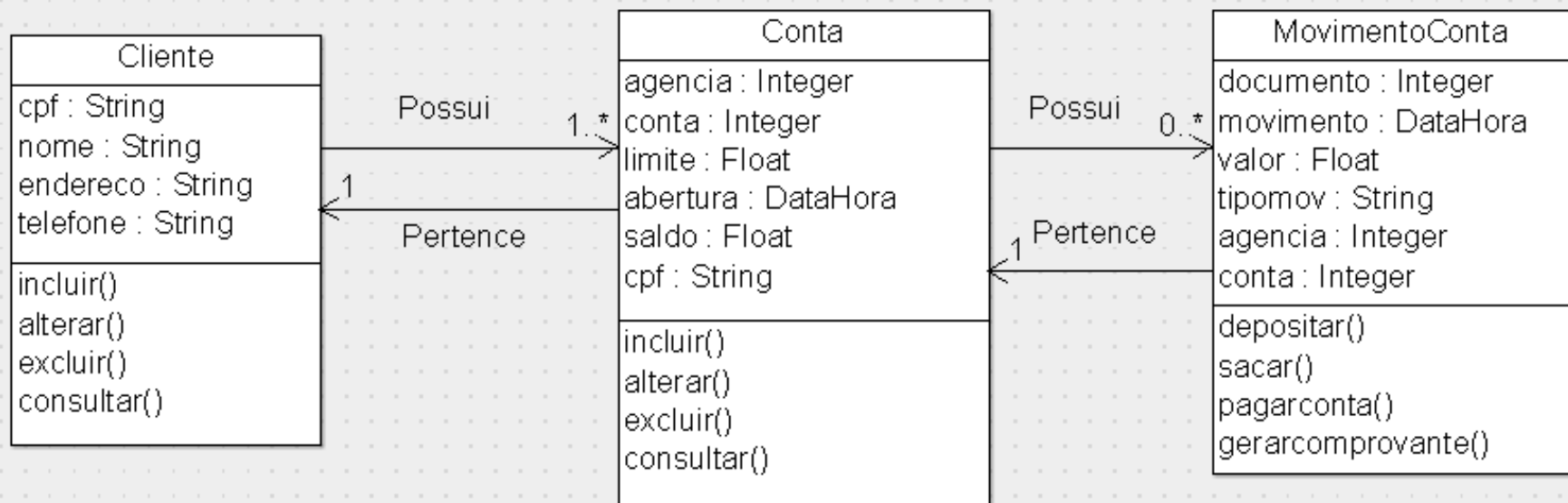


**Características  
(propriedades)**



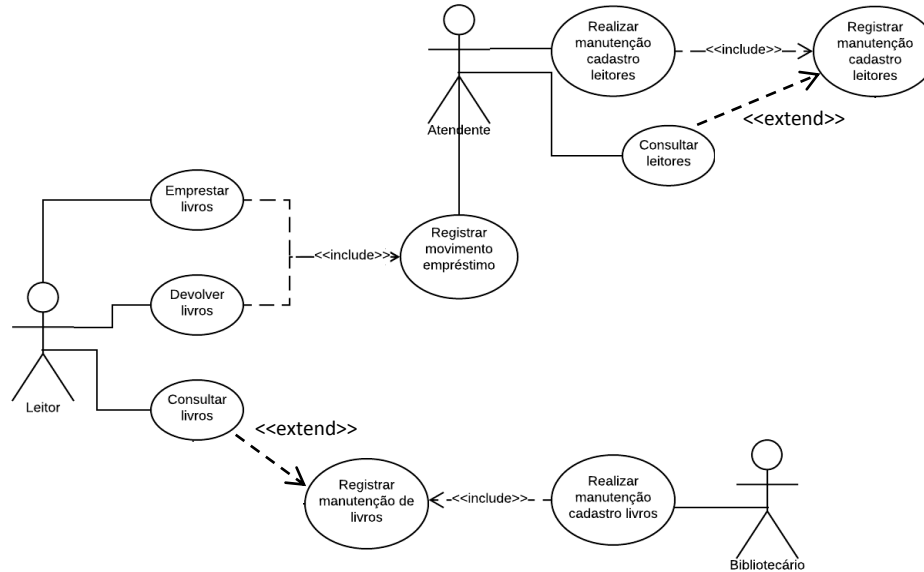
## Diagrama de Classes

### Atendimento Bancário



## Diagrama de Casos de Uso e ER

### Biblioteca



## Classes

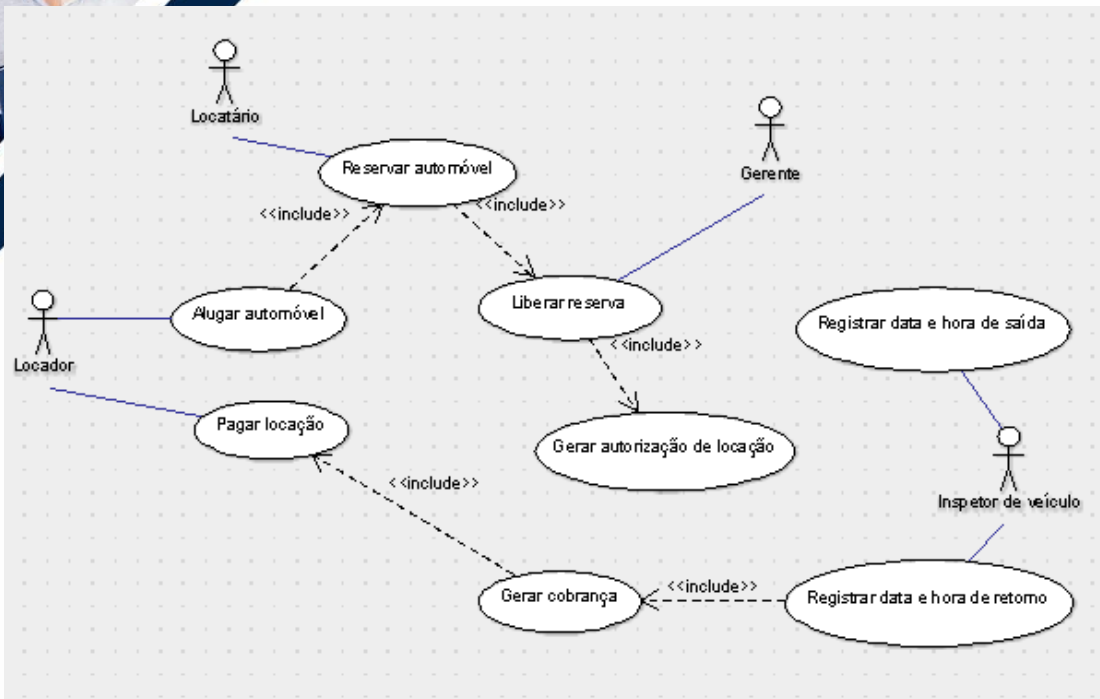


## Métodos

# Modelagem e Desenvolvimento Orientado a Objeto

## Diagrama de Casos de Uso e ER

Locadora de Automóvel



## Classes

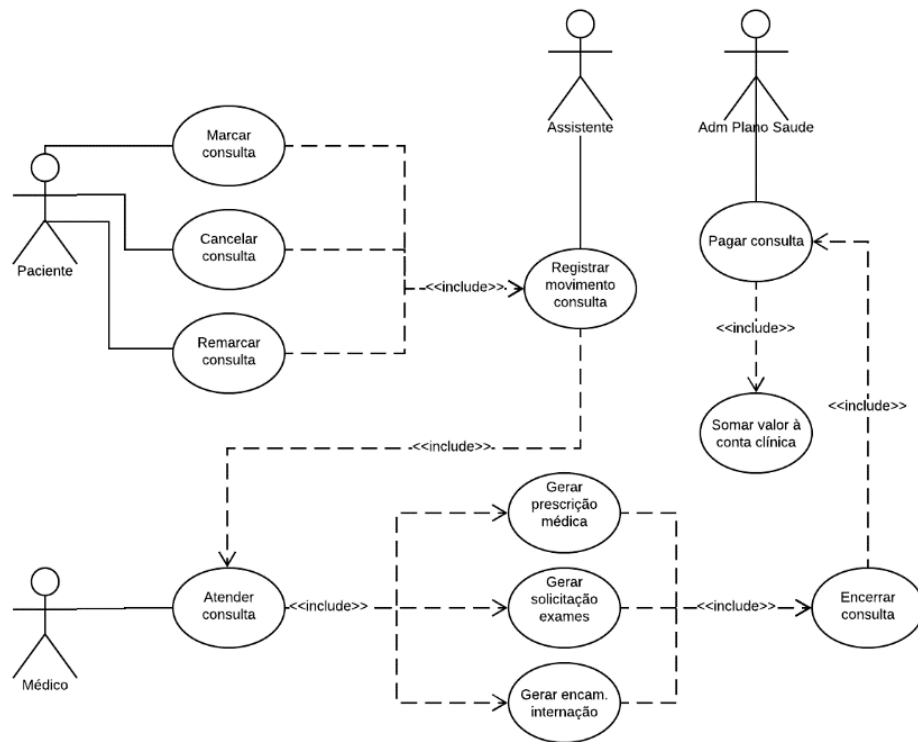


## Métodos

# Modelagem e Desenvolvimento Orientado a Objeto

## Diagrama de Casos de Uso e ER

### Clínica Médica



## Classes

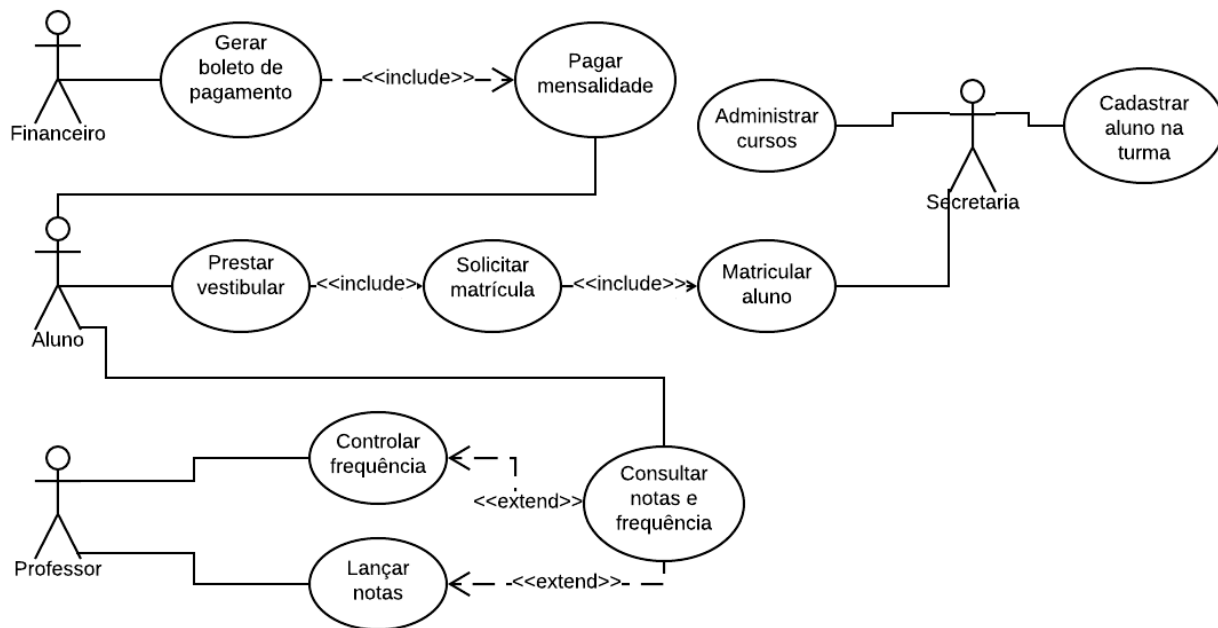


## Métodos

# Modelagem e Desenvolvimento Orientado a Objeto

## Diagrama de Casos de Uso

Faculdade



## Classes



## Métodos

# Modelagem e Desenvolvimento Orientado a Objeto

## Diagrama de Classes

### Implementação



| NomeDaClasse   |
|--|
| + PropriedadeClasse_01 : int<br>+ PropriedadeClasse_02 : String<br>+ PropriedadClasse_03 : float             |
| + MetodoDaClasse_01() : void<br>+ MetodoDaClasse_02() : int<br>+ MetodDaClasse_03(parametro01 : int) : float |

2 Defina o nome da classe, representada por um substantivo, com a primeira letra em maiúsculo.

3 As propriedades são definidas no segundo bloco da classe. Cada propriedade pode ter um tipo de dado específico, inclusive referenciado a outra classe. A propriedade também é representada por um substantivo.

5 Os sinais representam o encapsulamento de cada atributo / método da classe.

| NomeDaClasse   |
|--|
| + PropriedadeClasse_01 : int<br>+ PropriedadeClasse_02 : String<br>+ PropriedadClasse_03 : float             |
| + MetodoDaClasse_01() : void<br>+ MetodoDaClasse_02() : int<br>+ MetodDaClasse_03(parametro01 : int) : float |

4 Os métodos são definidos no terceiro bloco da classe. Assim como as funções existentes na Análise Estruturada, eles podem retornar ou não algum resultado e podem ter parâmetros recebidos. Os métodos, por serem ações possíveis da classe, são representados por um verbo.

| 6 Produto   |
|---|
| +CodigoBarras : int<br>+ Descrição : String<br>+ Preço : float  |
| + Imprimir() : void<br>+ ObterQuantidadeEstoque() : int<br>+ CalcularSubTotal(quantidade : int) : float |

```
class Produto
{
    public int CodigoBarras { get; set; }

    public string Descricao { get; set; }

    public float Preco { get; set; }

    public void Imprimir ()
    {
        // TODO - Escrever o método de impressão
    }

    public int ObtemQuantidadeEstoque()
    {
        // TODO - Escrever o método para obter a quantidade em estoque
        return 0;
    }

    public float CalcularSubTotal(int quantidade)
    {
        return quantidade * Preco;
    }
}
```



# Modelagem e Desenvolvimento Orientado a Objeto

