

# Engenharia de Software 2

Clávison M. Zapelini

[clavison.zapelini@unisul.br](mailto:clavison.zapelini@unisul.br)

# Diagrama de Classes

Clávison M. Zapelini

[clavison.zapelini@unisul.br](mailto:clavison.zapelini@unisul.br)

# Diagrama de Classes

- É com certeza o mais importante e o mais utilizado diagrama da UML.
- Permite a visualização das classes que comporão o sistema com seus respectivos atributos e métodos, bem como os relacionamento entre as classes.

# Diagrama de Classes

- Apresenta uma visão estática de como as Classes estão organizadas;
- Preocupação apenas com a estrutura lógica.
- Serve como base para outros diagramas da UML.

# Persistência

- Em muitos casos é necessário preservar de forma permanente os objetos de uma Classe.
  - A Classe precisa ser Persistente.
- Uma Classe Persistente apresenta muitas semelhanças com uma entidade como as definidas no MER.
  - Modelo utilizado para definir as tabelas em banco de dados Relacional.

# Persistência

- Deve ficar claro que nem toda Classe é persistente, não sendo muitas vezes necessário preservar (armazenar) suas informações.

# Classes, Atributos e Métodos

- Classes costumam possuir atributos e atributos armazenam os dados dos Objetos da Classe.
- Métodos que são as funções que uma instância da Classe pode executar.

# Atributos

- Os valores dos Atributos podem variar de instância para instância.
  - É exatamente essa característica, que permite a identificação de cada Objeto.



# Atributos

- Cada atributo deverá conter um tipo de dados, ou seja a forma como a informação deverá ser armazenada.
  - Byte:
    - Tamanho em bits: 8
    - Faixa de valores: -128 a 127
  - Boolean:
    - Tamanho em bits: 8
    - Faixa de valores: true ou false
  - Int:
    - Tamanho em bits: 32
    - Faixa de valores: -2.147.482.648 a 2.147.843.467
  - Long:
    - Tamanho em bits: 64
    - Faixa: -9.223.372.036.854.775.802 a +9.223.372.036.854.775.802
  - Double:
    - Tamanho em bits: 64
    - Faixa: -1.79769313486231570E+308 a +1.79769313486231570E+308

# Atributos

- Char:
  - Texto.
- Date:
  - Data.

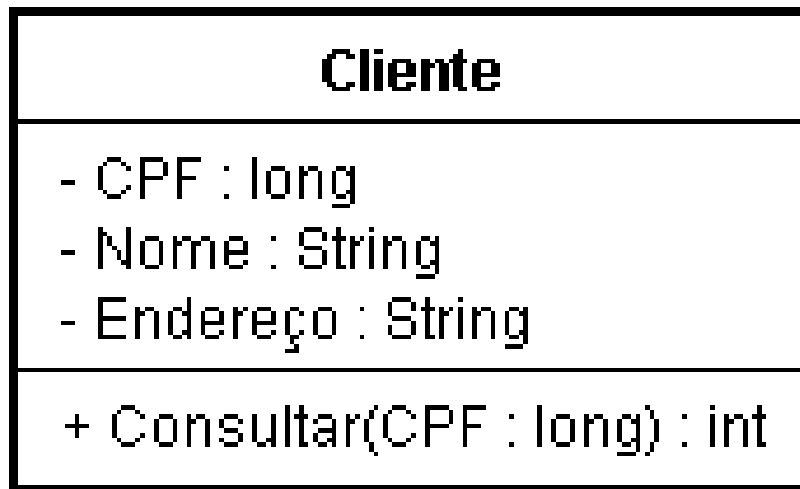
# Métodos

- Embora os Métodos sejam declarados no Diagrama de Classes, não é uma preocupação desse Diagrama, definir as etapas que estes métodos deverão percorrer quando forem chamados.
  - Essa função é atribuída a outros Diagramas como:
    - Diagrama de Seqüência e
    - Diagrama de Atividades

# Representação de uma Classe

- Como já mostrado, anteriormente, uma Classe é representado por um retângulo com três divisões:
  - Na primeira parte → Nome da Classe;
  - Na segunda parte → Os Atributos da Classe;
  - Na terceira parte → Os Métodos da Classe.

# Representação de uma Classe



← Classe

← Atributos

← Métodos

# Tipos de visibilidade

- Visibilidade Pública
  - O atributo ou método que possuir essa visibilidade pode ser utilizado por qualquer Classe.
    - Símbolo (+), sinal de mais.
- Visibilidade Protegida
  - O atributo ou método que possuir essa visibilidade somente a classe possuidora ou as sub-classes terão acesso.
    - Símbolo (#), sustenido.
- Visibilidade Privada
  - Somente a Classe possuidora desse atributo ou método poderá utilizá-lo.
    - Símbolo (-), sinal de menos.

# Relacionamento

- As Classes costumam possuir relacionamento entre si, com o intuito de compartilhar informações e colaborarem umas com as outras para permitir a execução dos diversos processos executados pelo sistema.

# Associações

- Descreve um vínculo que ocorre normalmente entre duas Classes, chamado neste caso de **Associação Binária**.
- Em uma Associação determina-se que as instâncias de uma Classe estão de alguma forma ligadas às instâncias das outras Classes.



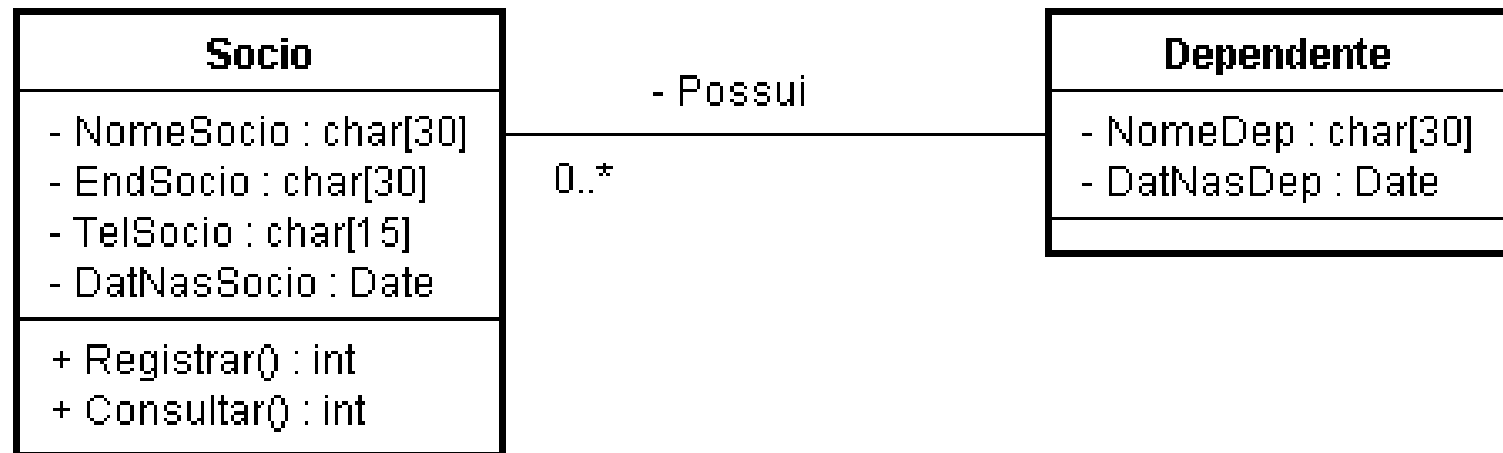
# Multiplicidade

0..1	No mínimo zero (nenhum) e no máximo um. Indica que os Objetos da classe associada não precisam obrigatoriamente estar relacionados.
1..1	Um e somente um. Indica que apenas um objeto da classe se relaciona com os objetos da outra classe.
0..*	No mínimo nenhum e no máximo muitos. Indica que pode não haver não instâncias da classe participando do relacionamento.
*	Muitos. Indica que muitos objetos da Classe estão envolvidos no Relacionamento.
1..*	No mínimo um e no máximo muitos. Indica que há pelo menos um objeto envolvido no relacionamento, podendo haver muitos.
3..5	No mínimo 3 e no máximo 5. Indica que há pelo menos 3 instâncias envolvidas no relacionamento e que pode ser 4 ou 5 as instâncias envolvidas, mas não mais do que isso.

# Associação Binária

- Ocorre quando são identificados relacionamentos entre duas classes.
- Este tipo de Associação constitui-se na mais comum encontrada nos Diagramas de Classe.

# Representação da Associação Binária



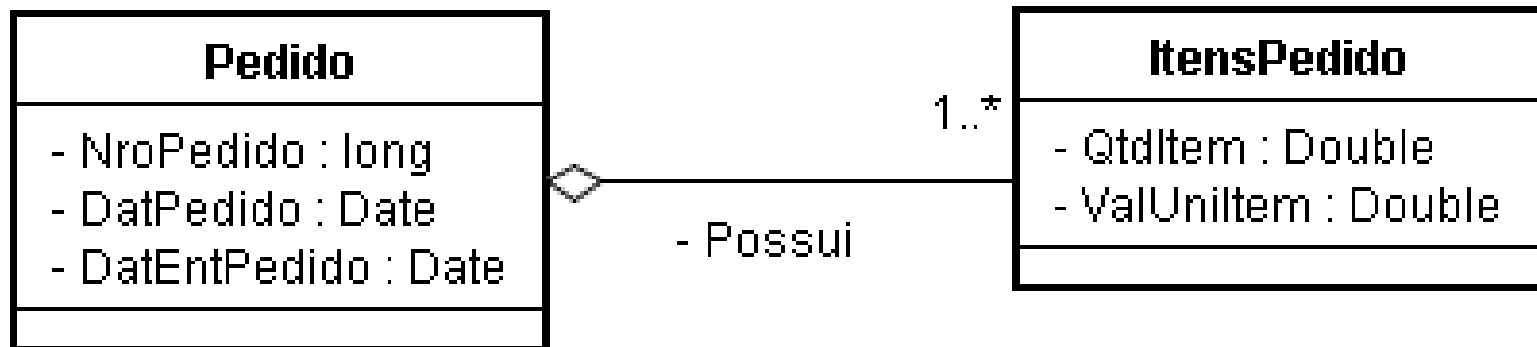
# Agregação

- É um tipo especial de associação onde tenta-se demonstrar que as informações de um objeto (chamado objeto-todo) precisam ser complementadas pelas as informações contidas em um objeto de outra classe (chamado objeto-parte).

# Representação de Agregação

- O símbolo de agregação difere do de associação por conter um losango na extremidade da classe que contém os *objetos-todo*.

# Representação de Agregação



# Composição

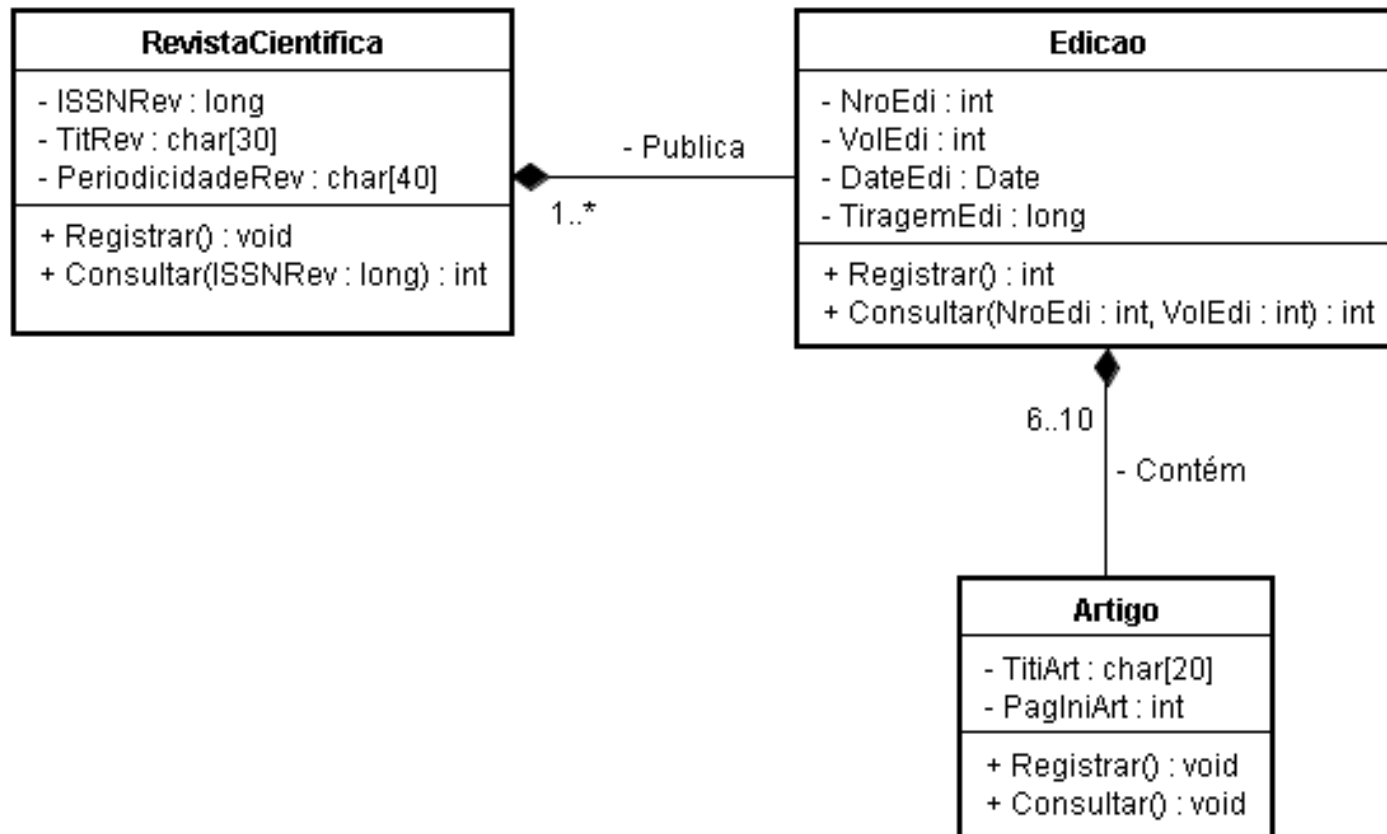
- Constitui-se em uma variação do tipo agregação. Uma associação do tipo Composição tenta representar um vínculo mais forte entre os objetos-todo e objetos-parte.
- Tenta mostrar que os objetos-parte têm que pertencer exclusivamente a um único objeto-todo.

# Representação da Composição

- O símbolo usado para a associação de Composição é um losango preenchido, e da mesma forma que na Agregação, deve ficar ao lado do objeto-todo.



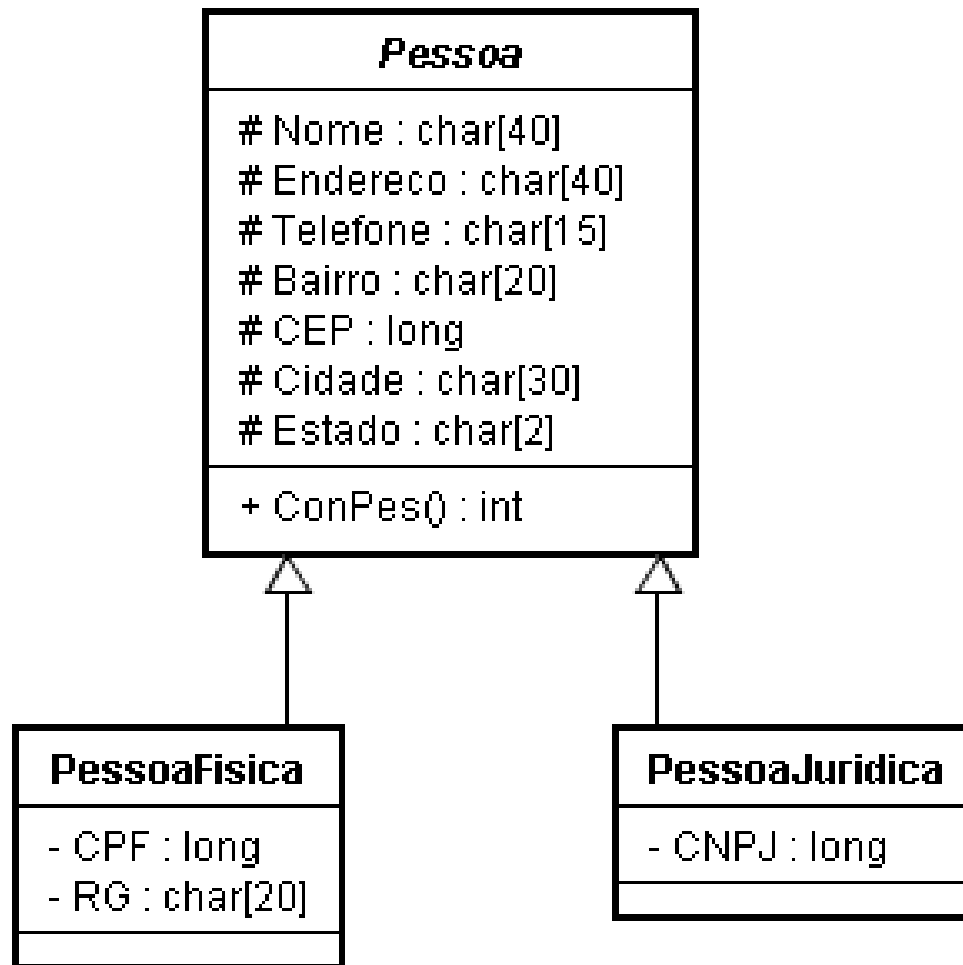
# Representação da Composição



# Especialização / Generalização

- Similar à associação de mesmo nome utilizado no Diagrama de Casos de Uso. Seu objetivo é identificar classes-mãe (gerais) e classes filhas (especializadas).
- Permite também demonstrar a ocorrência de métodos polimórficos nas classes especializadas.

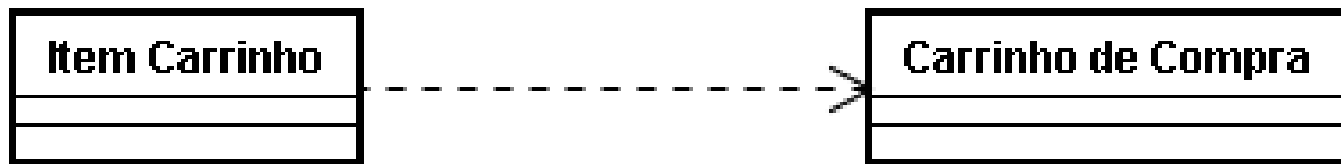
# Especialização / Generalização



# Dependência

- Não é um tipo comum de relacionamento, como o próprio nome diz, identifica um certo grau de dependência de uma classe em relação a outra.
- Representado por uma reta tracejada entre duas classes, contendo uma seta na extremidade do relacionamento que é dependente de alguma forma.

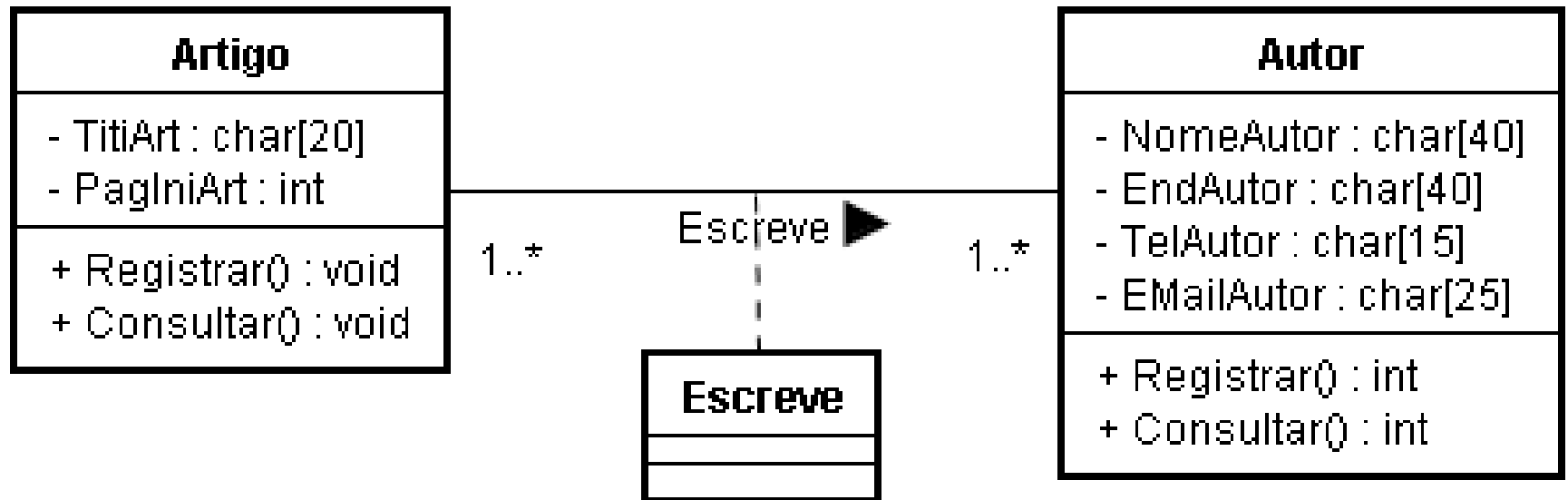
# Dependência



# Classe Associativa

- São classes produzidas quando da ocorrência de associações que possuem multiplicidade muitos (\*) em todas as suas extremidades.
- As Classes Associativas são necessárias nesses casos porque não existe um repositório que possa armazenar as informações produzidas pelas as associações.

# Classe Associativa

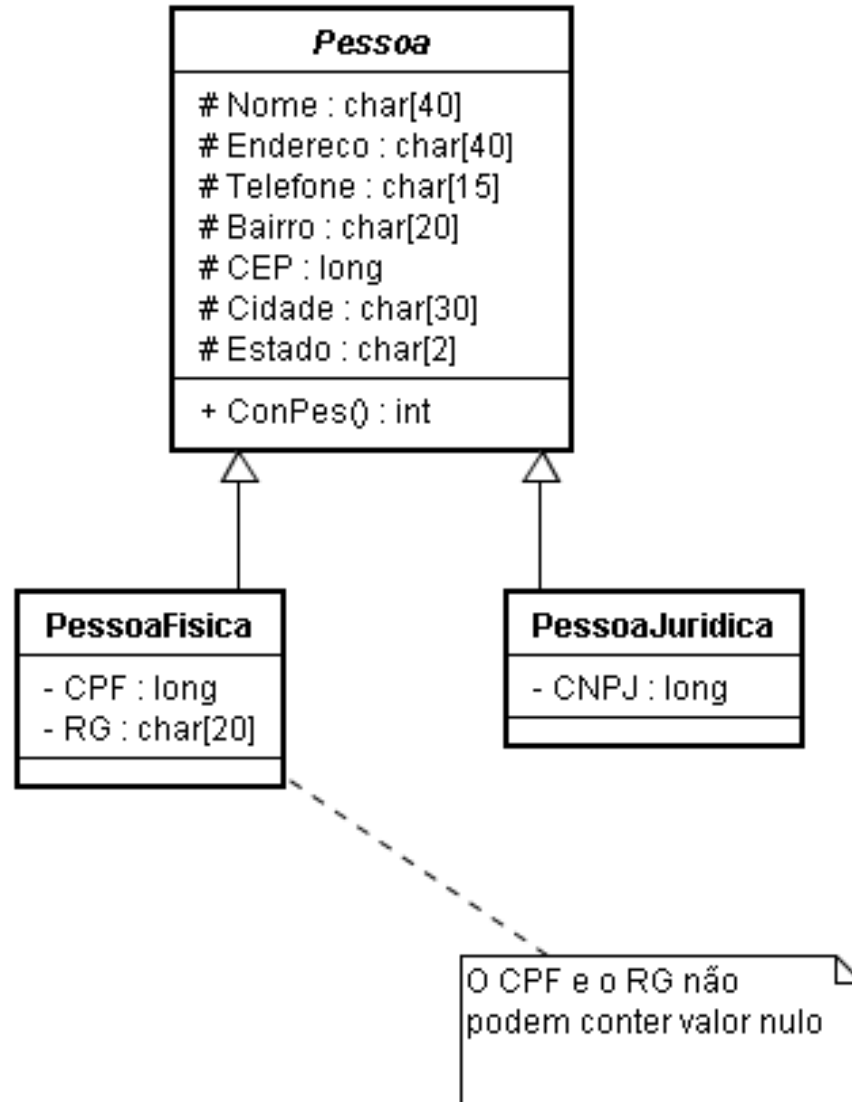


# Notas

- São importantes para informar algum comentário necessário a classe, método ou atributo, fazendo com que, todos tomem conhecimento de forma imediata a observação feita, seja essa observação feita para validar ou simplesmente informar como o objeto notificado se comporta.



# Notas





# Criar o Diagrama de Classes do trabalho da Disciplina