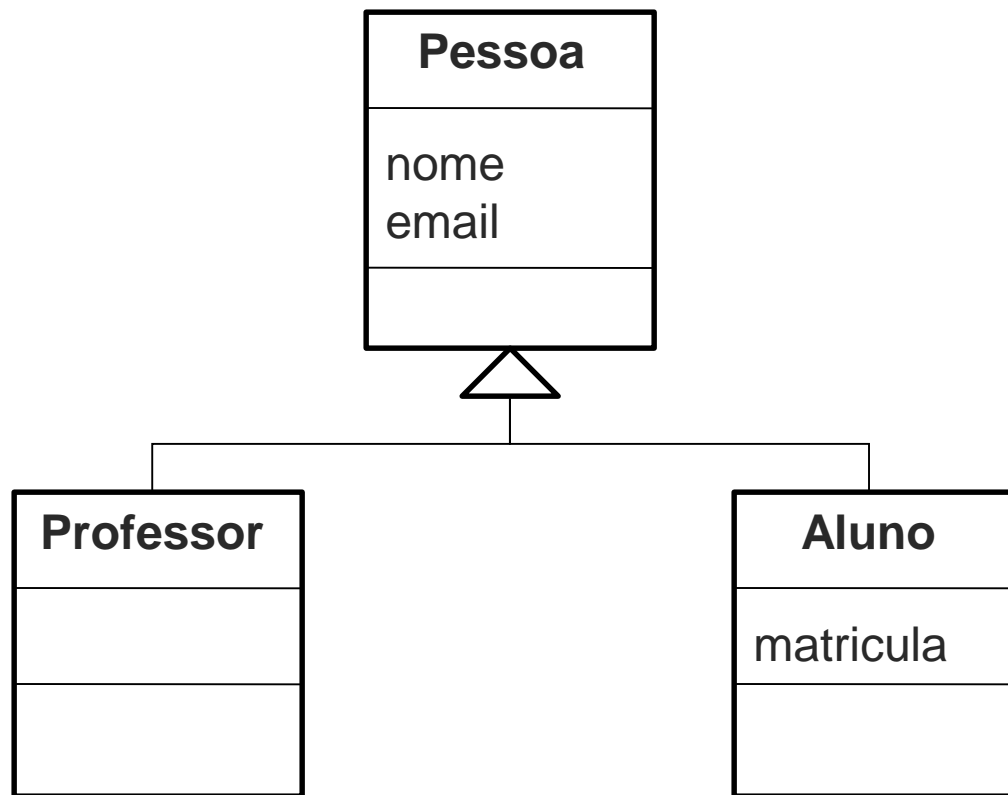


# Diagrama de Classes

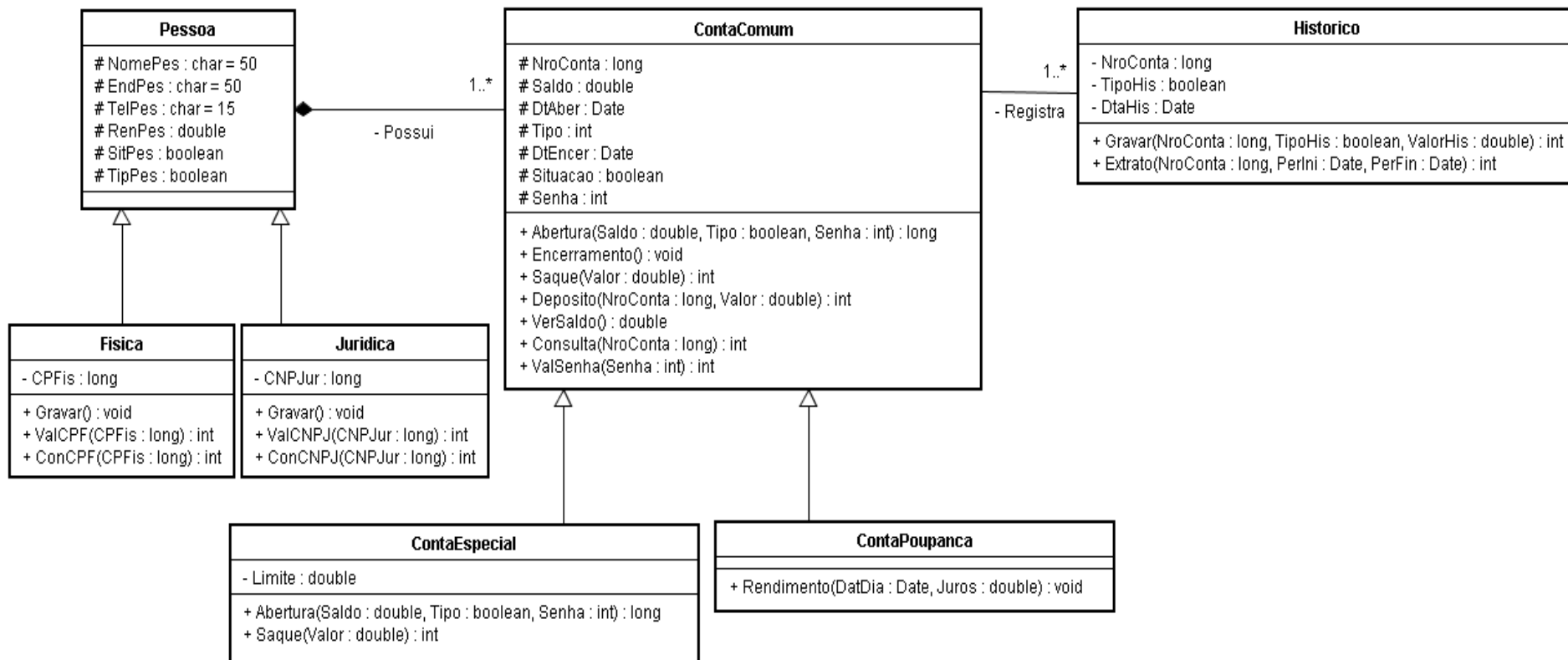
Eduardo Figueiredo

<http://www.dcc.ufmg.br/~figueiredo>

# [ Primeiro Diagrama de Classes ]



# Outro Diagrama de Classes



# Diagrama de Classes

- Serve de apoio para a maioria dos outros diagramas
- Define a estrutura das classes do sistema
  - Apresenta uma visão estática de como as classes estão organizadas
- Estabelece como as classes se relacionam

# [ Diagrama de Classes ]

- É o mais importante e o mais utilizado diagrama da UML
- Permite a visualização das classes que compõem o sistema
- Representa
  - Atributos e métodos de uma classe
  - Os relacionamentos entre classes

# [ Atributos ]

---

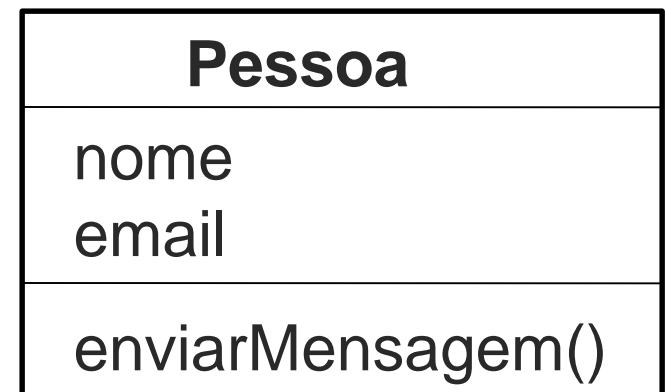
- Permite a identificação de cada objeto de uma classe
- Os valores dos atributos podem variar de instância para instância
- Atributos devem conter o tipo de dados a ser armazenado
  - Byte, boolean, int, double, char, String, etc.

# [ Métodos ]

- São apenas declarados neste diagrama
  - Diagrama de Classes não define a implementação
- Outros diagramas permitem modelar o comportamento interno dos métodos
  - Diagrama de Sequência
  - Diagrama de Atividades

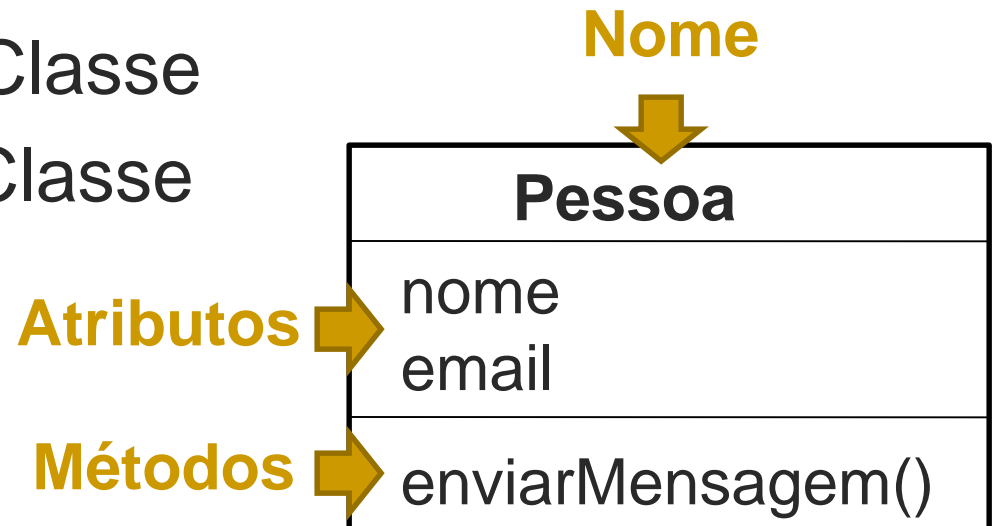
# [ Representação de uma Classe ]

- Uma classe é representada por um retângulo com três divisões:
  - Nome da Classe
  - Atributos da Classe
  - Métodos da Classe



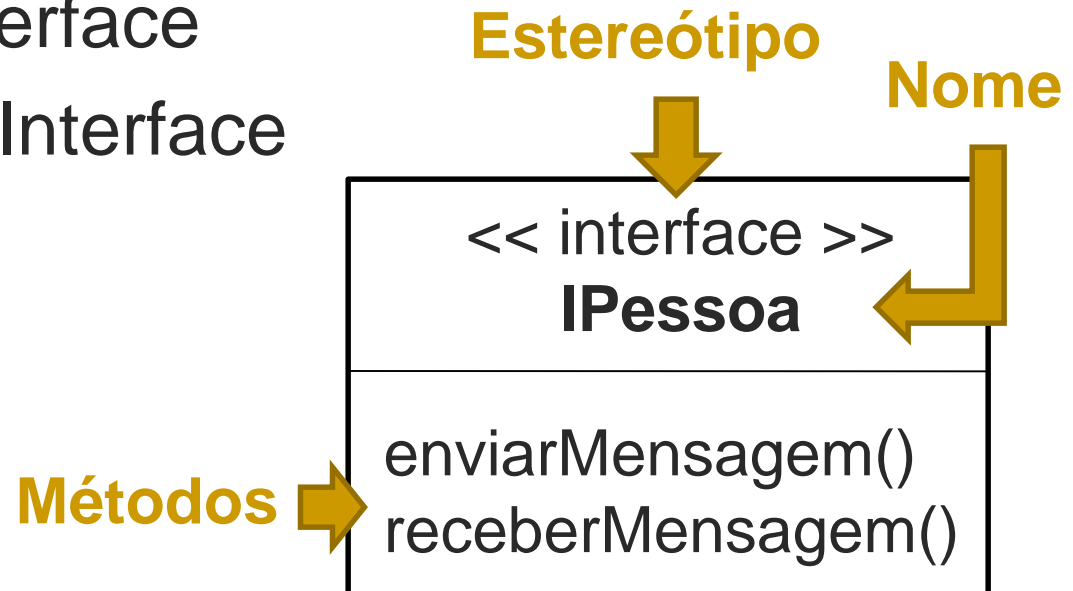
# [ Representação de uma Classe ]

- Uma classe é representada por um retângulo com três divisões:
  - Nome da Classe
  - Atributos da Classe
  - Métodos da Classe



# Representação de uma Interface

- Uma interface é semelhante a uma classe, mas não tem atributos
- Uma interface possui
  - Nome da Interface
  - Métodos da Interface
- Estereótipo
  - interface



# [ Tipos de Visibilidade ]

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

# [ Tipos de Visibilidade ]

## ■ Pública (+)

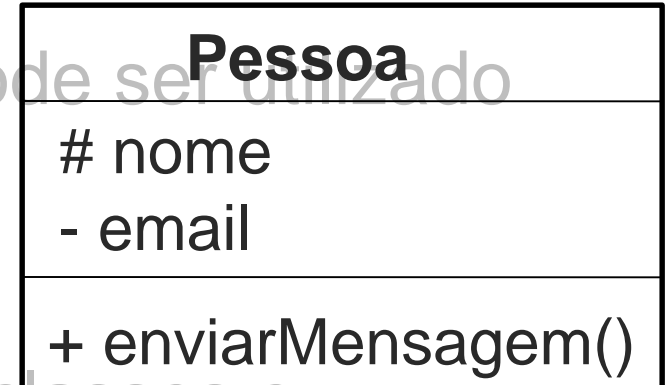
- O atributo ou método pode ser utilizado por qualquer classe

## ■ Protegida (#)

- Somente a classe, sub-classes e classes amigas (ex. pacote) têm acesso

## ■ Privada (-)

- Somente a própria classe terá acesso



# [ Comunicação entre Objetos (I) ]

- Conceitualmente, objetos se comunicam através da troca de mensagens
- Mensagens definem:
  - O nome do serviço requisitado
  - A informação necessária para a execução do serviço
  - O nome do requisitante.

# [ Comunicação entre Objetos (II) ]

- Na prática, mensagens são geralmente implementadas como chamadas de métodos
  - Nome = o nome do método
  - Informação = a lista de parâmetros
  - Requisitante = o método/objeto que realizou a chamada

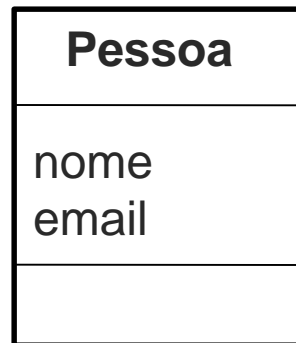
# [ Notas ]

---

- Informativos

- Têm função de comentários em classes, métodos ou atributos
- Informa restrição de funcionalidade
- Indicar condições para os relacionamentos, etc.

# [ Notas ]



Nome é obrigatório para toda instancia desta classe

# [ Bibliografia ]

- G. Booch, J. Rumbaugh, I. Jacobson. **UML, Guia do Usuário**. 2ª Ed., Editora Campus, 2005.
  - Capítulos 4, 8 e 9
- M. Fowler. **UML Essencial**, 2a Edição. Bookmann, 2000.
  - Capítulos 4 e 6