



# Modelagem do sistema com a análise orientada a objetos





# Análise e desenvolvimento de software orientado a objetos

Bloco 1

Iolanda Cláudia Sanches Catarino





# Evolução da análise e desenvolvimento de software

- Qual é o melhor modelo de Processo de Software?
  - Modelos de processo prescritivo.
  - Modelos de processo especializados.
  - Processo unificado.
  - Desenvolvimento ágil.

Figura 1 - Modelos de processo de software



Fonte: Iurri Motov/ iStock.com.

# Evolução da análise e desenvolvimento de software

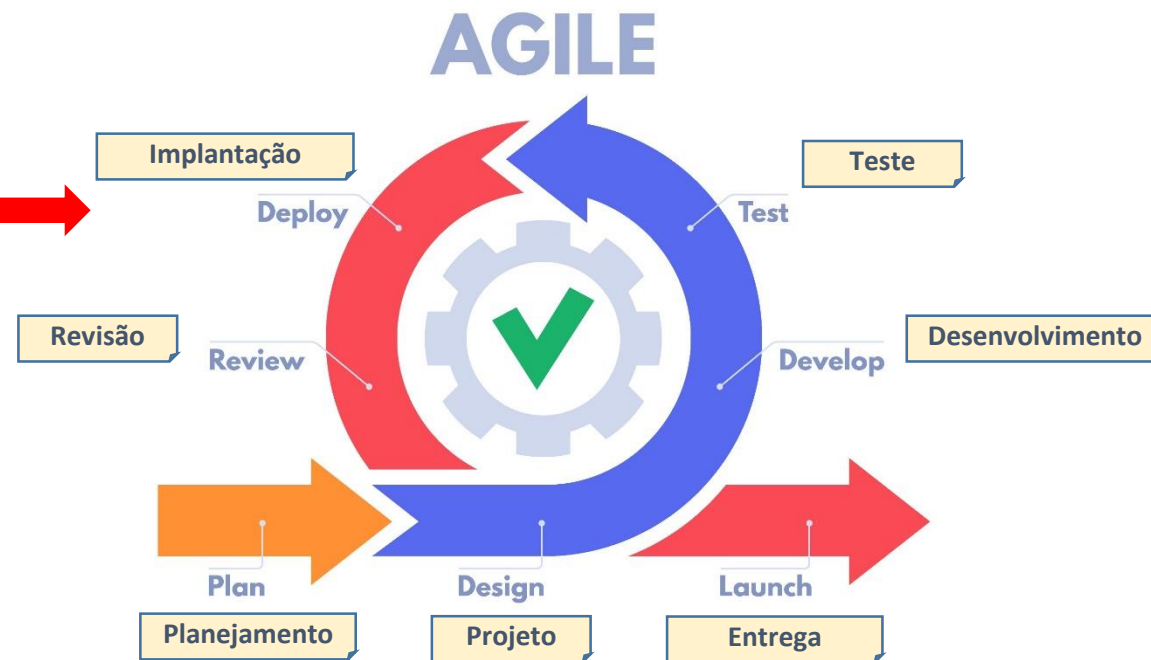
- Camadas da engenharia de software:

Figura 2 – Camadas da Engenharia de Software



Fonte: Pressman e Maxim (2016, p.16).

Figura 3 – Fases de um processo de desenvolvimento de software

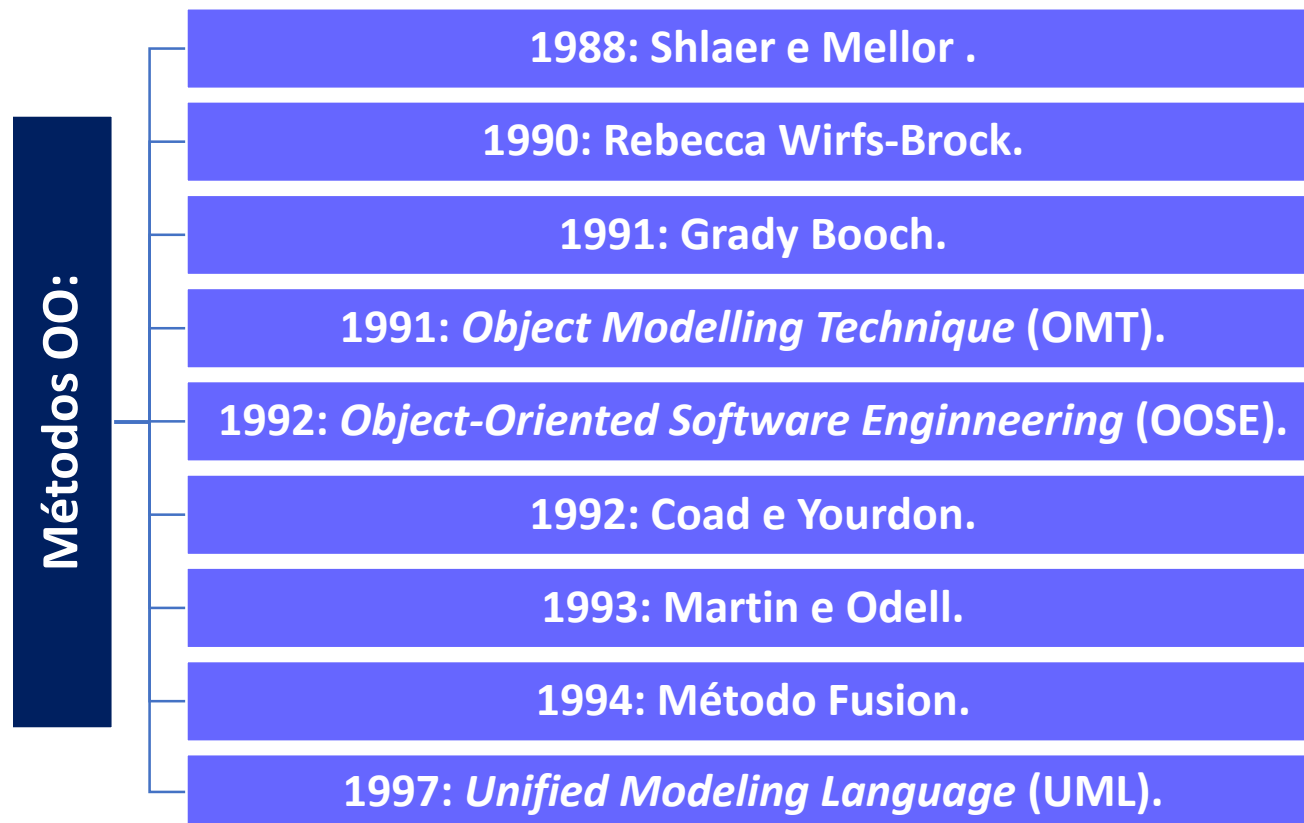


Fonte: Tetiana Lazunova/iStock.com

# Evolução da análise e desenvolvimento de software

- Principais Métodos Orientados a Objetos (OO):

Figura 4 – Métodos Orientados a Objetos



Fonte: elaborada pela autora.



# Análise e desenvolvimento de software orientado a objetos

Bloco 2

Iolanda Cláudia Sanches Catarino



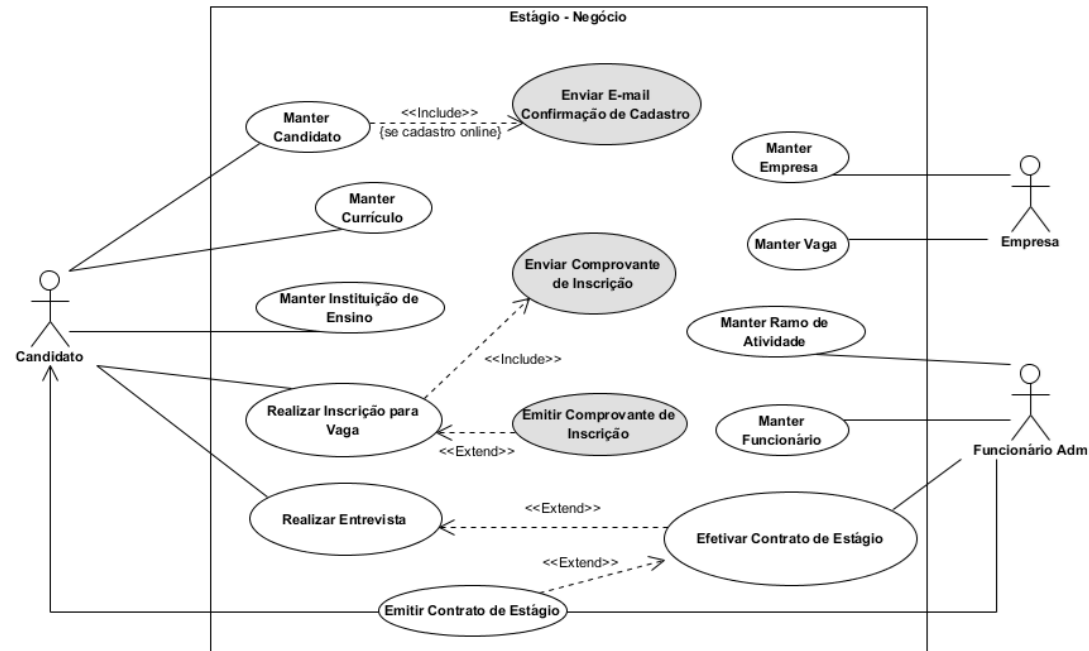
# Técnicas de modelagem orientada a objetos

- **Modelo:** é uma descrição simplificada da realidade, apresentado a partir de uma perspectiva específica e criado para proporcionar melhor compreensão do sistema.
- Cada **modelo** pode ser expresso em diferentes níveis de precisão.
- Um **modelo** é constituído por um conjunto de diagramas (desenhos) consistentes entre si, acompanhados de descrições textuais dos elementos que aparecem nos vários diagramas.

# Técnicas de modelagem orientada a objetos

- Diagrama de *Use Cases*:

Figura 5 – Exemplo de diagrama de *Use Cases*



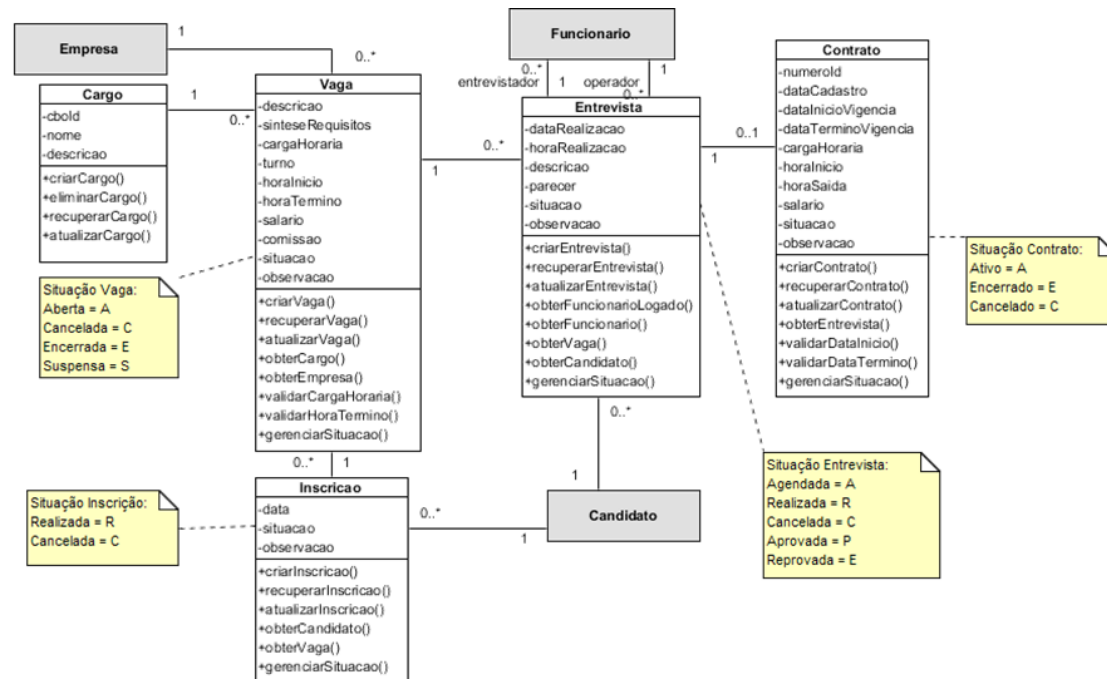
Fonte: elaborada pela autora.



# Técnicas de modelagem orientada a objetos

- Diagrama de classes:

Figura 6 - Exemplo de Diagrama de *Use Cases*



Fonte: elaborada pela autora.



# Análise e desenvolvimento de software orientado a objetos

Bloco 3

Iolanda Cláudia Sanches Catarino



# Conceitos da orientação a objetos

- **Objeto:** pode ser definido como qualquer coisa concreta ou abstrata do mundo real, com características e comportamentos próprios, em uma única estrutura, sendo possível identificá-lo.

**Figura 7 - Objetos concretos**



Fonte: thitivong/iStock.com

**Figura 8 - Objetos abstratos**



Fonte: Florin1605/iStock.com

**Figura 9 - Objetos concretos**



Fonte: kasto80/iStock.com



# Conceitos da orientação a objetos

- **Abstração:** consiste na concentração dos aspectos importantes e relevantes dos objetos, considerando o contexto analisado e o domínio do sistema.

**Figura 10 - Abstração – Meios de transporte**



Fonte: phaisam2517/iStock.com

**Figura 11- Abstração – Jogadores ou esportes**



Fonte: Eugene\_Onischenko/iStock.com



# Conceitos da orientação a objetos

- **Classe:** representa um grupo de objetos do mundo real que possuem tipos de características e de comportamento em comum.

**Figura 12 – Classe de objetos – Alunos**



Fonte: monkeybusinessimages/iStock.com

**Figura 13 – Classe de objetos – Funcionários**

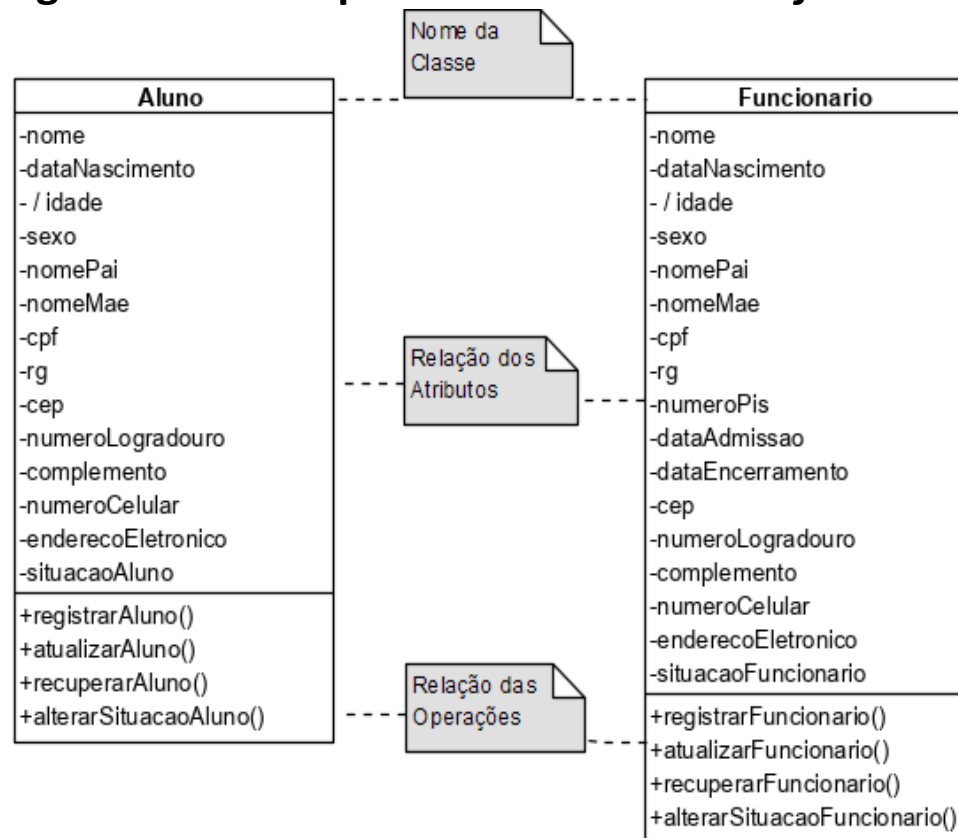


Fonte: nd3000/iStock.com

# Conceitos da orientação a objetos

- Atributos, operações, métodos, eventos, mensagens e estados.

Figura 14 – Exemplo de classes *alunos* e *funcionários*



Fonte: elaborada pela autora.

# Conceitos da orientação a objetos

- **Encapsulamento:** representa o ato de reunir em uma estrutura, chamada classe, os atributos e operações dos objetos, permitindo que um objeto proteja a integridade de suas partes.

**Figura 15 – Exemplo de encapsulamento – Funções do Laptop**

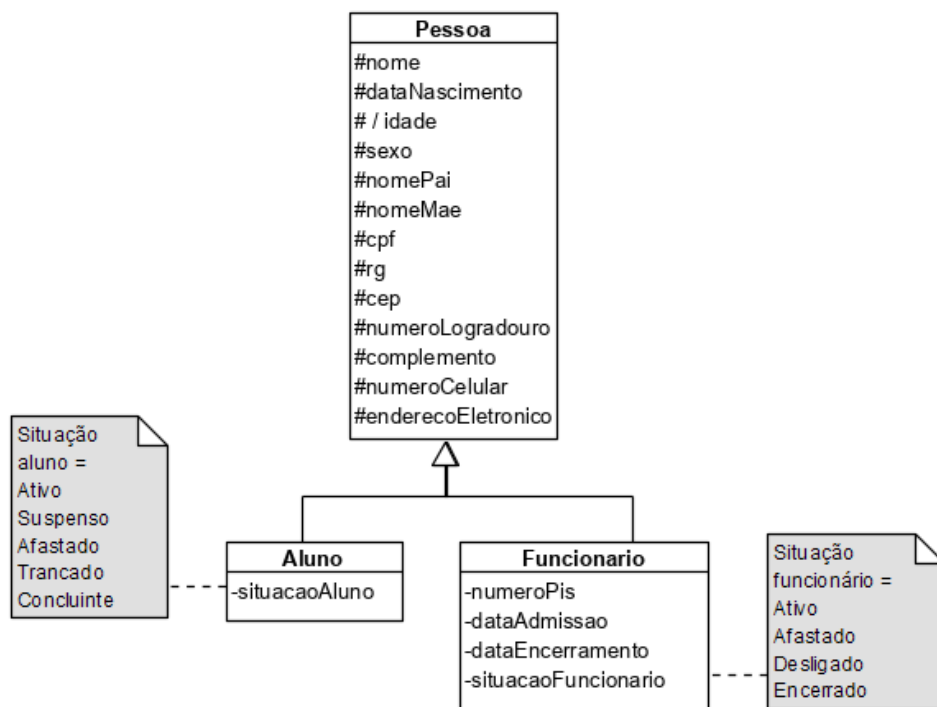


Fonte: aphaspirit/iStock.com

# Conceitos da orientação a objetos

- **Generalização:** também denominado de **herança**, representa a propriedade pela qual uma classe pode herdar atributos e operações de uma classe que generaliza as características e comportamentos comuns de um grupo de objetos.

**Figura 16 – Exemplo de generalização**



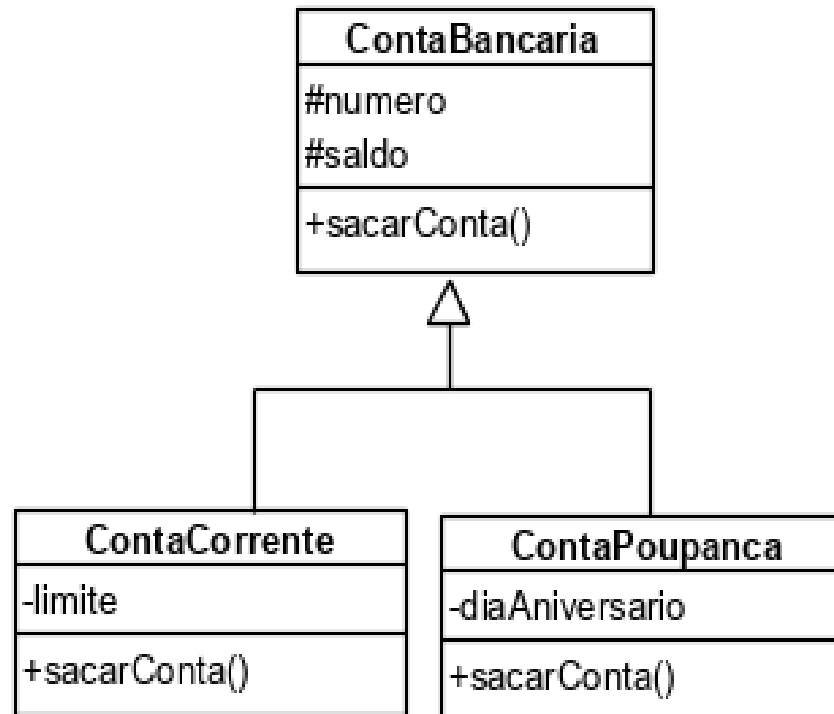
Fonte: elaborada pela autora.



# Conceitos da orientação a objetos

- **Polimorfismo:** significa que a mesma operação pode atuar de diversas formas em classes distintas. Uma operação polimórfica possui o mesmo nome em classes distintas, mas em cada classe o método implementado é diferente.

Figura 17 – Exemplo de polimorfismo



Fonte: elaborada pela autora.



# Análise e desenvolvimento de software orientado a objetos

Bloco 4

Iolanda Cláudia Sanches Catarino



# Atividade – Modelagem da análise orientada a objetos

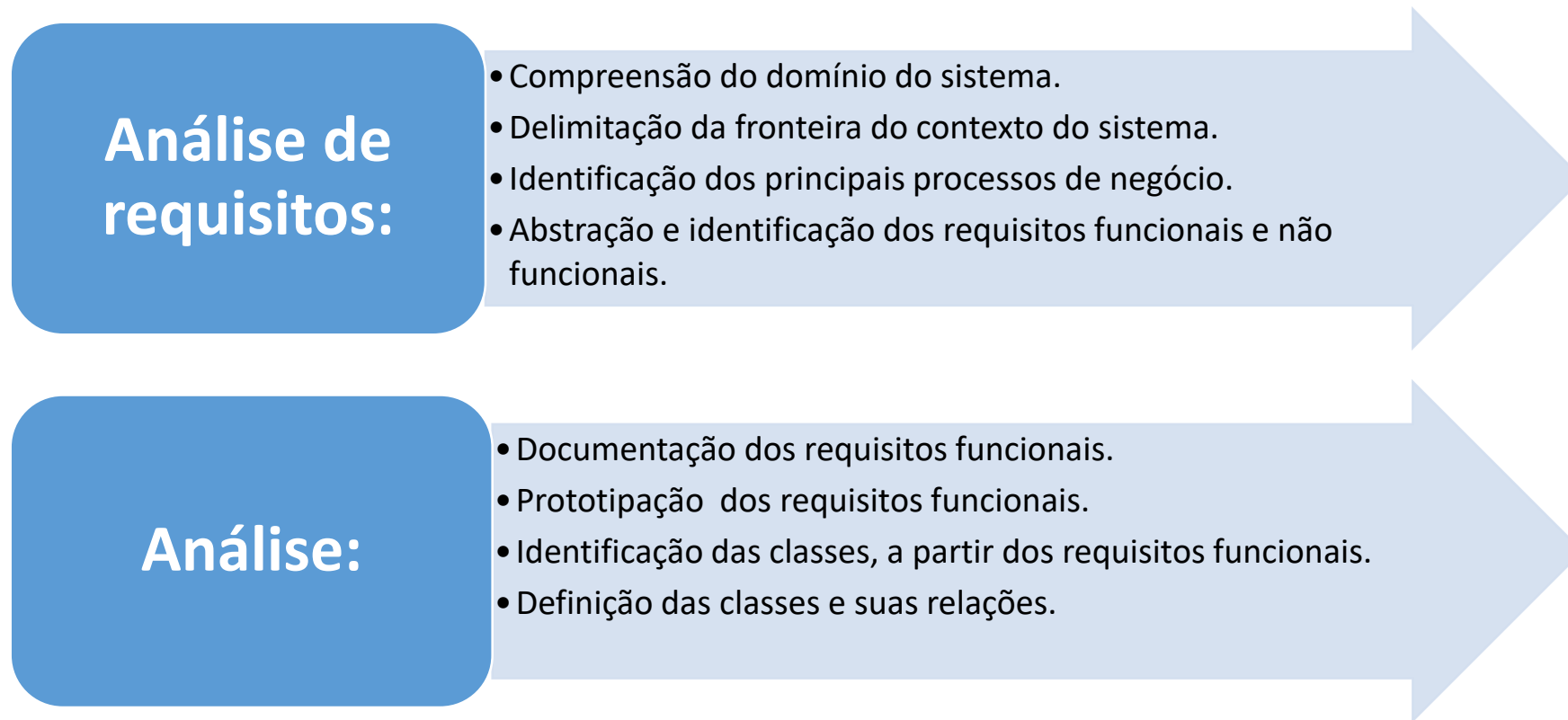
## Controle de eventos científicos:

O sistema deve controlar a submissão e avaliação de trabalhos para eventos científicos. Um autor pode realizar muitas submissões, a partir do envio de seu trabalho, respeitando o deadline do evento. Existem três tipos válidos de submissão de trabalhos: artigos curtos ou longos, cursos ou palestras. Um autor ou avaliador deve se cadastrar no sistema, criando seu login e senha. Uma submissão pode ser elaborada por mais de um autor, totalizando cinco autores, no máximo, com a indicação de um autor responsável pela submissão. Toda submissão é avaliada por uma comissão de três avaliadores, considerando a atribuição de uma nota para diferentes quesitos de qualidade do trabalho. É de responsabilidade do coordenador do evento notificar os autores sobre a aceitação ou não de suas submissões no evento.

# Atividade – Modelagem da análise orientada a objetos

- Identificação dos requisitos funcionais e classes:

**Figura 18 – Especificação dos requisitos funcionais e classes**



Fonte: elaborada pela autora.



# Identificação dos requisitos funcionais

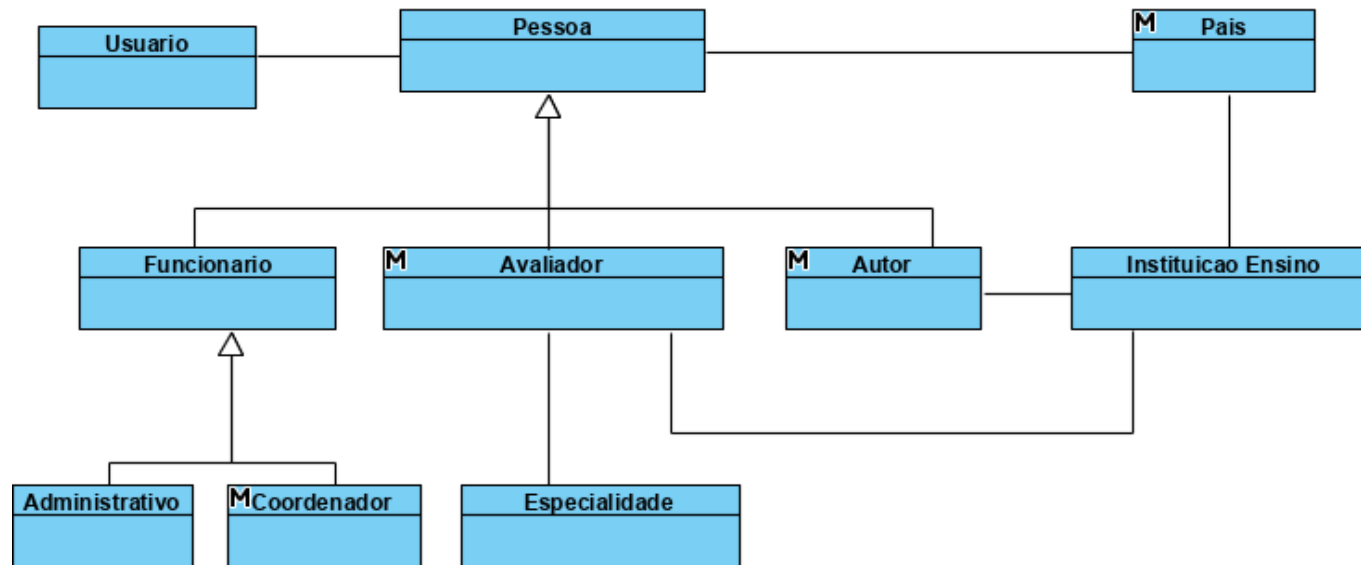
Figura 19 - Listagem dos requisitos funcionais

Nº Requisito	Nome	Descrição
RF01	Manter Evento	O sistema deve prover um cadastro de eventos (simpósio, congresso, workshop etc), que realiza a inclusão, alteração e exclusão.
RF02	Manter Tema de Trabalho	O sistema deve prover um cadastro de tema de trabalhos que realiza a inclusão, alteração e exclusão.
RF03	Manter Tipo de Evento	O sistema deve prover um cadastro de tipos de evento que realiza a inclusão, alteração e exclusão.
RF04	Manter Autor	O sistema deve prover um cadastro de autor com seus dados pessoais e um resumo de seu currículo.
RF05	Manter Instituição de Ensino	O sistema deve prover um cadastro de instituição de ensino que os autores estão vinculados.
RF06	Manter Avaliador	O sistema deve prover um cadastro de avaliador com seus dados pessoais.
RF07	Manter Especialidade de Avaliador	O sistema deve prover um cadastro de especialidade dos avaliadores.
RF08	Manter País	O sistema deve prover um cadastro de países para vincular os autores e avaliadores.
RF09	Manter Requisito da Avaliação	O sistema deve prover um cadastro de requisitos para avaliar um trabalho.
RF10	Manter Usuário	O sistema deve prover um cadastro de usuário com login e senha.
RF11	Logar Sistema	O sistema deve prover a funcionalidade de acessar o sistema.
RF12	Submeter Trabalho	O sistema deve prover a funcionalidade para submissão de trabalhos (artigo, curso ou palestra) para um evento.
RF13	Avaliar Trabalho	O sistema deve prover a funcionalidade para avaliação de um trabalho submetido.
RF14	Registrar Comentário de Trabalho	O sistema deve prover a funcionalidade para registrar comentários para um trabalho avaliado.
RF15	Aprovar Trabalho	O sistema deve prover a funcionalidade para aprovação dos trabalhos submetidos.
RF16	Notificar Autor de Trabalho	O sistema deve prover a funcionalidade para notificar os autores dos trabalhos aprovados ou não.

Fonte: elaborada pela autora.

# Identificação das classes – Pacote pessoa

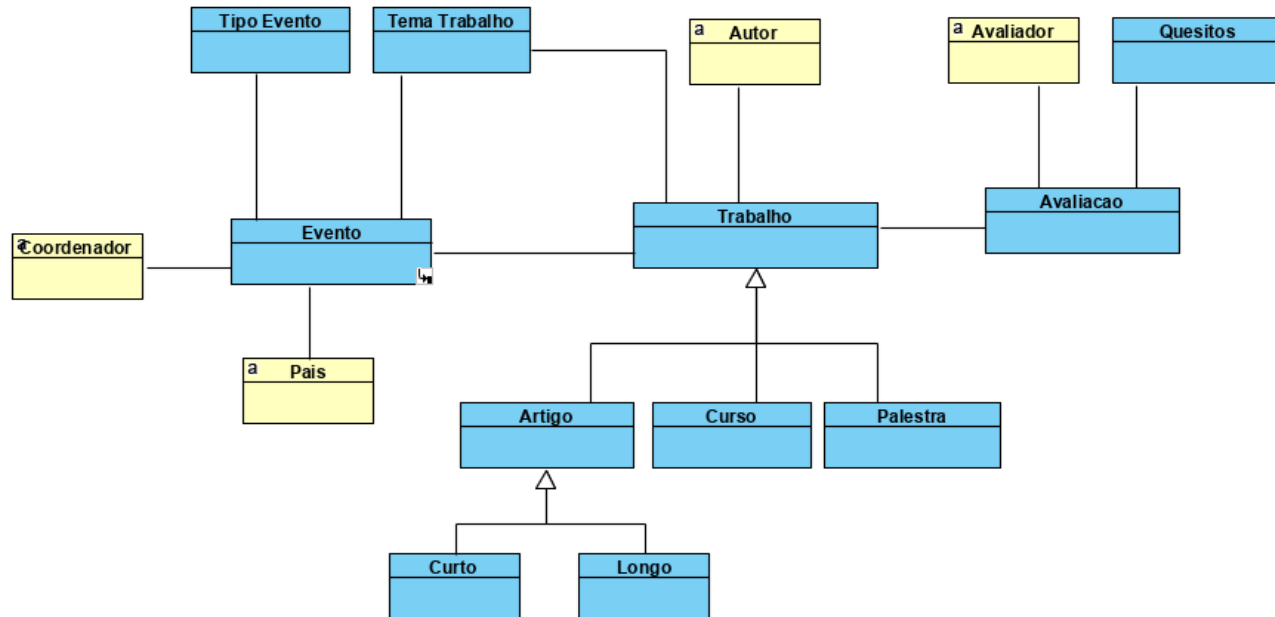
Figura 20 - Classes do pacote pessoa



Fonte: elaborada pela autora.

# Identificação das classes – Pacote negócio

Figura 21 – Classes do pacote negócio



Fonte: elaborada pela autora.



# Dica da Professora

Bloco 5

Iolanda Cláudia Sanches Catarino





## Dica da professora

- **Conhecendo algumas ferramentas CASE de modelagem orientada a objetos:**
  - **Visual Paradigm Community Edition ou Online.**  
Disponível em: <https://www.visual-paradigm.com/>.  
Acesso em: 2 out. 2020.
  - **Astah.** Disponível em: <https://astah.net/>. Acesso em: 2 out. 2020.
  - **Modelio SA.** Disponível em:  
<https://www.modeliosoft.com/en/products/modelio-sa-system-architects.html>. Acesso em: 2 out. 2020.
  - **Enterprise Architect.** Disponível em:  
<https://sparxsystems.com/>. Acesso em: 2 out. 2020.

## Referências

BEZERRA, E. **Princípios de análise e projeto de sistemas com UML**. 3. ed. Rio de Janeiro: Elsevier, 2014.

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **UML: guia do usuário**. 2. ed. Rio de Janeiro: Campus, 2006.

GUEDES, G. T. A. **UML: uma abordagem prática**. 3. ed. São Paulo: Novatec, 2018.

PRESSMAN, R.; MAXIM, B. **Engenharia de software: uma abordagem profissional**. 8. ed. Porto Alegre: AMGH, 2016.

RUMBAUGH, J. *et al.* **Modelagem e projetos baseados em objetos**. Rio de Janeiro: Campus, 1997.



Bons estudos!

