# Introdução da aula

#### **Qual é o foco da aula?**

Nesta aula você irá aplicar os conhecimentos adquiridos ao longo das aulas para solucionar as situações-problema apresentadas.

#### **Objetivos gerais de aprendizagem**

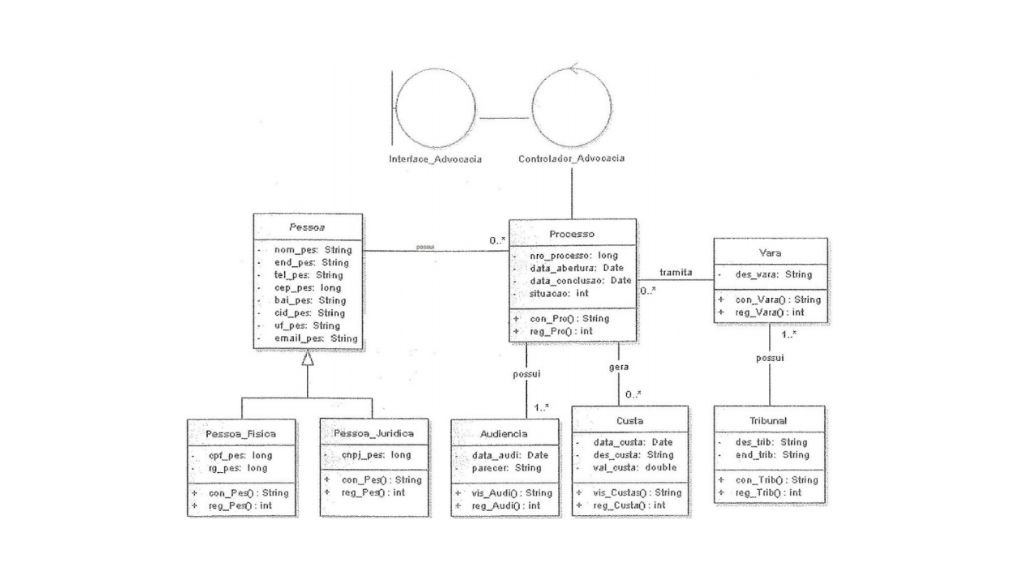
Ao longo desta aula, você irá:

* Analisar as situações-problema apresentadas;
* Formular possíveis soluções levando em conta as aulas previamente estudadas;
* Comparar suas respostas com as soluções apresentadas.

# Diagramas UML

Você é analista de sistemas e está trabalhando no diagrama de classes para sistema de controle de advocacia Silva e Souza, por isso sabe que o diagrama de classe está no ranking dos diagramas UML como o mais utilizado. Há muitos elementos nesse diagrama além das classes, tais como estereótipos, vários tipos de relacionamentos e multiplicidade.

Analise os relacionamentos do diagrama de classe apresentado e identifique de quais tipos são.



# Diagramas UML- resolução

Ao analisarmos um diagrama de classe conseguimos, por meio dos relacionamentos e multiplicidades, compreender muitas características do modelo, como é o caso deste diagrama. As informações fornecidas pelos tipos de associações e multiplicidades permitem sua compreensão, mesmo sem ler qualquer especificação textual a respeito do caso de uso do qual ele deriva.

Neste caso vemos que a classe Processo é a classe principal.

As classes Pessoa\_Física e Pessoa\_Jurídica são especializações da classe pessoa. Os demais relacionamentos são de associação.

A multiplicidade indicada nas associações significam: 0 – nenhum elemento (possui ou gera) e \* - indica vários elementos, caso omisso (não indicado) indica um (1) elemento apenas.

Desta forma, Processo tem (está associado) apenas a uma (1) pessoa e Pessoa possui 0 (zero) ou vários (\*) Processos.

Processo possui pelo menos uma (1) audiência ou várias (\*) e pode não gerar custas (0) ou várias (\*). É certo que tanto uma audiência ou uma custa só existe se existir Processo, mas no caso não podemos concluir isto sem conhecer as regras de negócio do sistema; é uma conclusão pragmática que deve ser evitada. Assim, transformar a associação em um relacionamento de composição desconhecendo a regra de negócio pode gerar um erro de projeto.

 Veja o atributo situação da classe Processo: neste sistema, esse atributo pode representar Processo extinto, e não ser excluído o objeto que o representa do sistema; um processo pode ser extinto ou arquivado, mas continuam a existir suas audiências e custas.

Um Processo tramita em uma Vara e uma Vara está associada a nenhum (0) Processo ou a vários (\*).

Um tribunal possui pelo menos uma (1) Vara ou várias (\*).

# Videoaula: paradigma orientado a objetos

Meu vídeo não funciona

#### **Referências**

BEEDLE, M. et al. Manifesto Ágil. [S.l.], 2001. Disponível em: http://agilemanifesto.org/. Acesso em: 10 nov. 2020. Boston: Addison-Wesley, 1999.

CMMI INSTITUTE. Página inicial. 2020. Disponível em: https://cmmiinstitute.com/. Acesso em: 10 nov. 2020.

D’ÁVILA, M. Sucessos e falhas em projeto de TI. Blog do Márcio, 2010. Disponível em: http:// blog.mhavila.com.br/2010/06/17/sucessos-e-falhas-em-projetos-de-ti/. Acesso em: 10 nov. 2020.

DEITEL, H. M. Java: como programar. Tradução: Edson Furmankiewicz. 8. ed. São Paulo: Pearson, 2010.

DIJKSTRA, E. W. Go To Statement Considered Harmful. (Reprinted from) Communications of the ACM, v. 11, n. 3, mar. 1968, p. 147-148. Association for Computing Machinery Inc., 1968. Disponível em: https://web.archive.org/web/20070703050443/http://www.acm.org/classics/oct95/. Acesso em: 10 nov. 2020.

FÉLIX, R. Programação orientada a objetos. São Paulo: Pearson Education do Brasil, 2016.

FOWLER, M. UML Essencial: um breve guia para a linguagem de modelagem de objetos. Tradução: João Tortello. 3. ed. Porto Alegre, Bookman, 2005.

GOLDBERG, A. Smalltalk-80: the language and its implementation. Xerox Corporation, 1983.

GUEDES, G. T. A. UML 2: uma abordagem prática. 2. ed. São Paulo: Novatec Editora, 2011.

JACOBSON, I.; BOOCH, G.; RUMBAUGH, J. El Proceso Unificado de Desarrollo de Software. 1. ed. rev. Madrid: Pearsoan Educacion S.A., 2000.

JACOBSON, I.; BOOCH, G.; RUMBAUGH, K. The unified software development process.

KAY, A. The Early History of Smalltalk. Gagne.homedns.org. ACM 1993 (HOPL-II/4/93/MA), US. Disponível em: http://gagne.homedns.org/~tgagne/contrib/EarlyHistoryST.html. Acesso em: 10 nov. 2020.

KOTLER, P.; KELLER, K.L. Administração de Marketing. Tradução: Sonia M. Yamamoto. 15. ed. São Paulo: Pearson Education do Brasil, 2018.

KRUCHTEN, P. B. The 4+1 View Model of architecture. IEEE Software, v. 12, n. 6, p. 42-50, nov. 1995.

LARMAN, C. Utilizando UML e padrões. 3. ed. São Paulo: Bookman, 2007.

OLIVEIRA, L. Faça 10 tipos de bolos caseiros com apenas uma massa! Parte 1. 1 vídeo (25 min. 41 seg.). Canal Chef Léo Oliveira. Disponível em: https://www.youtube.com/watch?v=COxdVrTvbM0. Acesso em: 2 dez. 2019

OLIVEIRA, L. Faça 10 tipos de bolos caseiros com apenas uma massa! Parte 2. 1 vídeo (31 min. 27 seg.). Canal Chef Léo Oliveira. Disponível em: https://www.youtube.com/watch?v=- jh2Ogn8\_\_zc. Acesso em: 2 dez. 2019.

PRESSMAN, R. S. Engenharia de software [recurso eletrônico]: uma abordagem profissional. Tradução: Ariovaldo Griesi. Rev. técnica: Reginaldo Akakaki, Julio Akakai, Renato Manzan de Andrade. 7. ed. Dados eletrônicos. Porto Algre: AMGH, 2011.

RAMOS, D. C. A. Conceitos de Orientação a Objetos: Curso. Programa de TIC – Tecnologia Informação e Comunicação do CCUEC - Unicamp, Campinas-SP. 10 mar. 2017. Disponível em: http://ftp.unicamp.br/pub/apoio/treinamentos/linguagens/POO.pdf. Acesso em: 10 nov. 2020.

REZENDE, D. A. Engenharia de Software e sistemas de informações. 2. ed. Rio de Janeiro: Brasport, 2002.

SEBESTA, R. W. Conceitos de linguagens de programação. 5. ed. São Paulo: Bookman, 2006.

SINTES, T. Aprenda Programação Orientada a Objetos em 21 dias. Tradução: João Eduardo N. Tortello. São Paulo: Pearson Education do Brasil, 2002. Disponível em: https://plataforma. bvirtual.com.br/Leitor/Publicacao/8/pdf. Acesso em: 3 dez. 2019.

SOFTEX. MpsBr. Melhoria do Processo de Software Brasileiro. Disponível em: https://softex.br/ mpsbr/. Acesso em: 12 dez. 2019.

SOMMERVILLE, I. Engenharia de software. 8. ed. São Paulo: Pearson - Addison Wesley, 2007. Disponível em: https://plataforma.bvirtual.com.br/Leitor/Publicacao/276/pdf. Acesso em: 3 dez. 2019

TUCKER, A. B.; NOONAN, R. E. Linguagens de programação: princípios e paradigmas. 2. ed. Porto Alegre: AMGH, 2010.