

Análise e Modelagem de Sistemas

Paradigma Orientado a Objetos

Prof^a. Dr^a. Iolanda Cláudia Sanches Catarino

1

- Unidade de Ensino: 4.
- Competência da Unidade: Conhecer as características, princípios e conceitos do Paradigma Orientado a Objetos, além do Processo Unificado e os principais Métodos Ágeis.
- Resumo: Estudo dos princípios e conceitos do Paradigma Orientado a Objetos, bem como, as características do Processo Unificado e os principais Métodos Ágeis.
- Palavras-chave: Paradigma Orientado a Objetos, Processo Unificado, Métodos Ágeis – Scrum, XP, FDD, MSF e DSDM.
- Título da Teleaula: Paradigma Orientado a Objetos.
- Teleaula nº: 04

2

Conceitos

Introdução ao Paradigma Orientado a Objetos

3

Introdução ao Paradigma Orientado a Objetos



Fonte: <https://www.publicdomainpictures.net/pictures/270000/nahied/car-dealerhttp-car-car-showroom.jpg>



Fonte: <https://www.gettyimages.com/image/2746734449/5c746c2ab08f7701b-y46g4y608b10P4X1/2558x302x>

- Segundo Bezerra (2007), pode-se dizer que o termo “paradigma de orientação a objetos” é uma forma de abordar um problema, visualizando um sistema de software como uma coleção de agentes interconectados chamados **objetos**, sendo cada objeto responsável por realizar tarefas específicas.

4

Introdução ao Paradigma Orientado a Objetos

- A Programação Orientada a Objetos (POO) teve início na década de 70 com a linguagem SIMULA, parte da linguagem Smalltalk, mas ganhou grande visibilidade na década de 80.
- Os métodos de modelagem orientados a objetos surgiram no final da década de 80, sendo que na década de 90 uma grande diversidade de autores lançou seus métodos, praticamente com propostas semelhantes.

5

Introdução ao Paradigma Orientado a Objetos

- Programação Estruturada vs Programação Orientada a Objetos:



6

Introdução ao Paradigma Orientado a Objetos

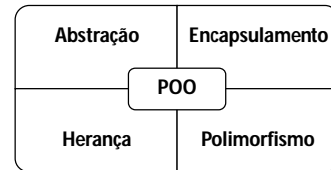
• Características do POO:

- Qualquer coisa é um objeto.
- Objetos realizam tarefas através da requisição de serviços a outros objetos.
- Cada objeto pertence a uma determinada classe.
- Uma classe agrupa objetos similares.
- A classe é um repositório para comportamento associado ao objeto.
- Classes são organizadas em hierarquias.

7

Introdução ao Paradigma Orientado a Objetos

• Princípios do POO:



8

Introdução ao Paradigma Orientado a Objetos

- **Abstração:** Para Rumbaugh (1994, p. 599), abstração representa a “habilidade mental que permite aos seres humanos visualizarem os problemas do mundo real com vários graus de detalhe, dependendo do contexto corrente do problema”.
- **Herança:** Para Rumbaugh (1994, p. 4), o conceito de herança é “o compartilhamento de atributos e operações entre classes com base em um relacionamento hierárquico. Uma classe pode ser definida de forma abrangente e depois refinada em sucessivas subclasses mais definidas”.

9

Introdução ao Paradigma Orientado a Objetos

- **Encapsulamento:** Para Rumbaugh (1994, p. 10), encapsulamento é “também chamado de ocultamento de informações, consiste na separação dos aspectos externos de um objeto, acessíveis por outros objetos, dos detalhes internos da implementação daquele objeto, que ficam ocultos dos demais objetos”.
- **Polimorfismo:** Na definição de Rumbaugh (1994, p. 4), polimorfismo “significa que a mesma operação pode atuar de modos diversos em classes diferentes; a mesma operação pode se aplicar a muitas classes diferentes”.

10

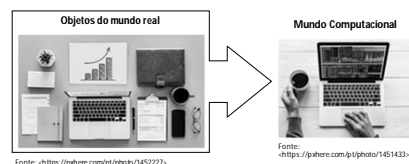
Conceitos

Conceitos Básicos da Orientação a Objetos

11

Conceitos da Orientação a Objetos

- **Orientação a Objetos:** É uma maneira de pensar nos problemas utilizando modelos organizados a partir de conceitos do mundo real.



12

Conceitos da Orientação a Objetos

- **Objeto:** Para Booch; Jacobson e Rumbaugh (2006, p. 456), um objeto é “uma entidade com uma fronteira bem-definida e uma identidade que encapsula o estado e comportamento”.
- Os objetos são descritos por seus **atributos e operações**.
 - **Atributo:** é uma característica particular possuída por todos os objetos.
 - **Operação:** é uma ação que um objeto executa ou está sujeito, ou seja, é uma ordem que faz um objeto reagir.

13

Conceitos da Orientação a Objetos

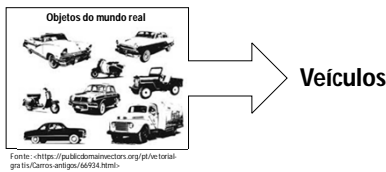
- **Objeto:** Um **objeto** pode ser definido como qualquer coisa concreta ou abstrata com existência no mundo real, com características e comportamento próprio, sendo possível identificá-lo como único.



14

Conceitos da Orientação a Objetos

- **Abstração:** Consiste na concentração dos aspectos essenciais e relevantes de um objeto, inerentes ao contexto e ao domínio do sistema.



15

Conceitos da Orientação a Objetos

- **Classe:** Para Rumbaugh (1994, p. 32), uma classe “descreve um grupo de objetos com propriedades semelhantes (atributos), o mesmo comportamento (operações), os mesmos relacionamentos com outros objetos e a mesma semântica”.
- **Classe:** Representa um grupo de objetos do mundo real que possui tipos de características e de comportamento em comum.

16

Conceitos da Orientação a Objetos

- **Exemplo de Classe:**

Ambiente Empresarial – Funcionários:



Fonte: <<https://gettyimages.com/pl/photos/7573331>>

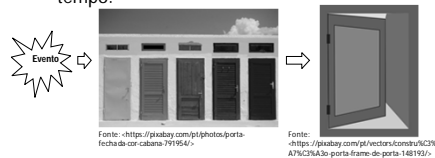
```

Funcionario
- matricula
- nomeCompleto
- dataNascimento
+ / classe
- sexo
- cpf
- rg
- cep
- endereco
- numeroLogradouro
- bairro
- cidade
- estado
- telefoneResidencial
- celular
- enderecoEletronico
+ criar(Funcionario)
+ recuperar(Funcionario)
+ validar(Funcionario)
+ validarDataNascimento()
+ validarCpf()
+ validarCpf()
+ validarExistenciaCpf()
+ validarEnderecoEletronico()
  
```

17

Conceitos da Orientação a Objetos

- **Evento:** é o acontecimento que faz o objeto mudar de estado.
- **Estado:** representa a abstração de uma forma de apresentação dos objetos de uma classe em um determinado instante de tempo.



18

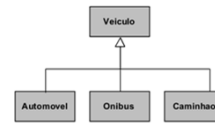
Conceitos da Orientação a Objetos

- **Evento:** Para Rumbaugh (1994, p. 114-115), um evento é “um estímulo individual de um objeto para outro; é algo que acontece em certo momento. Um evento é uma transmissão ou informação unidirecional de um objeto para outro”.
- **Estado:** Para Booch; Jacobson e Rumbaugh (2006, p. 290), um estado é “uma condição ou situação na vida de um objeto durante a qual o objeto satisfaz alguma condição, realiza alguma atividade ou aguarda um evento. Um objeto permanece em um estado por uma quantidade finita de tempo”.

19

Conceitos da Orientação a Objetos

- **Generalização:** Representa a propriedade pela qual uma classe pode herdar características e comportamento de outra, para obter o reaproveitamento dos atributos e operações.



Fonte: Elaborada pela Autora.

20

Conceitos

Fundamentos do Processo Unificado

21

Fundamentos do Processo Unificado

- Ciclo de Vida Básico de um Processo de Desenvolvimento de Software:



Fonte: Elaborada pela Autora.

22

Fundamentos do Processo Unificado

- Evolução dos processos de desenvolvimento de software:
 - a) Modelo Sequencial Linear;
 - b) Modelo de Prototipação;
 - c) Modelo Espiral.
- O **Processo Unificado** foi criado para apoiar o desenvolvimento orientado a objetos com a UML, fornecendo uma forma sistemática de especificar sistemas de softwares para diferentes domínios e tamanhos de projetos.

23

Fundamentos do Processo Unificado

- **Processo Unificado:**
 - Evolução do *Rational Objectory Process* (ROP).
 - O Processo Unificado é dirigido por **Casos de Uso**; é centrado em arquitetura; e é iterativo e incremental.
 - Com a evolução da *Rational* e à aquisição de outras empresas, surgiu o *Rational Unified Process* (RUP), que é um subconjunto do Processo Unificado.

24

Fundamentos do Processo Unificado

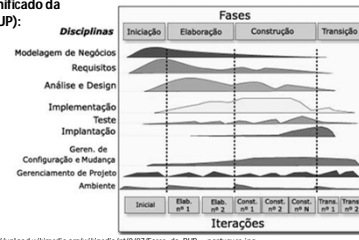


Fonte: BOOCH, RUMBAUGH e JACOBSON (2006).

25

Fundamentos do Processo Unificado

Processo Unificado da Rational (RUP):



Fonte: <https://upload.wikimedia.org/wikipedia/pt/0/07/Fases_do_RUP_-_portugues.jpg>

26

Resolução da SP

Exercício: Abstração de Objetos

27

Interação

Envie suas dúvidas!

28

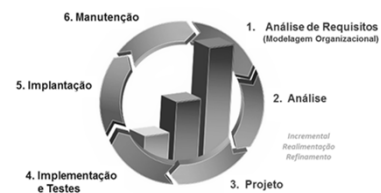
Conceitos

Métodos Orientados a Objetos

29

Métodos Orientados a Objetos

- Processo Básico de Desenvolvimento de Software – Ciclo de Vida:



Fonte: Elaborada pela Autora.

30

Métodos Orientados a Objetos

- **Método:** proporciona os detalhes de “como fazer” para construir o software. Envolve um amplo conjunto de atividades que incluem: planejamento, análise de requisitos, análise e projeto, implementação e testes.
- Para Sommerville (2011), um método é uma abordagem estruturada para o desenvolvimento de software, facilitando a produção de software com qualidade e uma boa relação custo-benefício.

31

Métodos Orientados a Objetos

- **Métodos de Análise Estruturada:**
 - DeMarco – 1978.
 - Jackson (JSD) – 1983.
 - Gane – 1983.
- **Métodos Orientados a Funções:**
 - Martin – década de 90.
 - Yourdon – 1992.

32

Métodos Orientados a Objetos

- **Métodos Orientados a Objetos (199X):**
 - James Rumbaugh (OMT - *Object Modeling Technique*).
 - Ivar Jacobson (OOSE – *Object-Oriented Software Engineering*).
 - Grady Booch.
 - James Martin.
 - Shlaer & Mellor.
 - Coad & Yourdon
 - Rebeca Wirfs-Brock (Projeto).
 - Fusion.
 - *Unified Modeling Language (UML)*: fusão dos métodos de Booch, Jacobson e Rumbaugh.

33

Conceitos

Introdução à *Unified Modeling Language 2.0 (UML)*

34

Linguagem de Modelagem Unificada - UML

- ***Unified Modeling Language (UML)* - Linguagem de Modelagem Unificada:**
- A UML é uma linguagem para visualização, especificação, construção e documentação de artefatos que façam uso de sistemas complexos de software (Booch, 2006).
- É uma linguagem visual utilizada para modelar sistemas computacionais por meio do paradigma de Orientação a Objetos (OO) (Guedes, 2005).

35

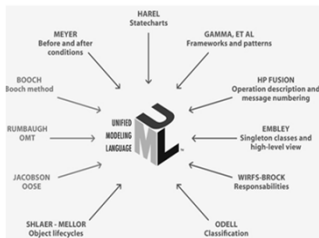
Linguagem de Modelagem Unificada - UML

- A UML fornece representação gráfica (técnicas de modelagem) para os elementos (objetos, classes, atributos, operações etc) do paradigma OO.
- É uma linguagem de modelagem para a elaboração da estrutura de projetos de software (BOOCH, 2006).
- A UML não é uma metodologia. Não diz quem deve fazer o quê, quando e como.
- A UML apoia o desenvolvimento **incremental**.

36

Linguagem de Modelagem Unificada - UML

Influências da UML:



Fonte: <https://arquivo.devmedi.a.com.br/arquivos/Renato_Groffo/Modelagem_UML/Modelagem_UML1.jpg>

37

Linguagem de Modelagem Unificada - UML

- Modelos podem evoluir com a inclusão de novos detalhes.
- A UML privilegia a descrição de um sistema segundo **três perspectivas**:
 - Dados (estrutural)**: Diagrama de Classes.
 - Operações (funcional)**: Diagrama de *Use Cases*.
 - Eventos (temporal)**: Diagrama de Atividades, Diagrama de Sequência e Diagrama de Máquina de Estados.

38

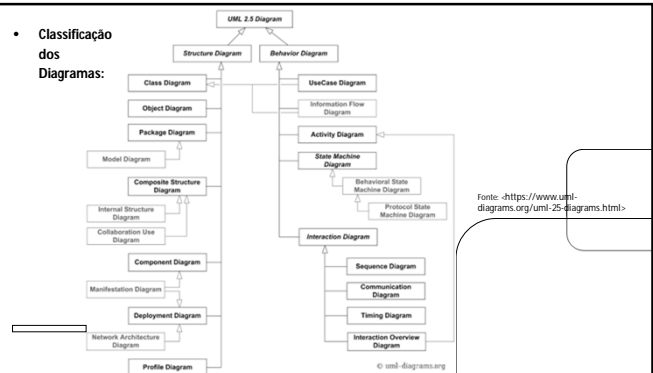
Linguagem de Modelagem Unificada - UML

- Os diagramas da UML 2.0 (versão – 2.5.1 em dez, 2017) dividem-se em:

Estruturais	Comportamentais
Diagrama de Classes	Diagrama de Casos de Uso
Diagrama de Objetos	Documentação de Casos de Uso
Diagrama de Estruturas Compostas	Diagrama de Atividade
Diagrama de Pacotes	Diagrama de Máquina de Estados
Diagrama de Componentes	Diagrama de Sequência
Diagrama de Implantação	Diagrama de Comunicação
Diagrama de Perfil	Diagrama de Interação Geral
	Diagrama de Tempo

39

- Classificação dos Diagramas:



40

Linguagem de Modelagem Unificada - UML

- Diagramas da UML 2.5**, disponível em: <<https://www.uml-diagrams.org/>>.
- Exemplos dos Diagramas da UML**, disponível em: <<https://www.uml-diagrams.org/index-examples.html>>.
- Guia de referência completo da UML, disponível em: <<https://www.uml.org/>>.

41

Conceitos

Principais Métodos Ágeis

42

Principais Métodos Ágeis

- **Métodos Ágeis:** tem como objetivo acelerar o desenvolvimento do software, visando a melhoria contínua dos processos.
- **Benefícios:**
 - Aumento da comunicação e interação da equipe.
 - Organização diária para o alcance da meta definida.
 - Diminuir falhas na elaboração no projeto.
 - Respostas rápidas às mudanças.
 - Aumento significativo da produtividade.

43

Principais Métodos Ágeis

- **Principais Métodos Ágeis:**
 - Scrum.
 - eXtreme Programming (XP).
 - Feature Driven Development (FDD).
 - Microsoft Solutions Framework (MSF).
 - Dynamic System Development Model (DSDM).

44

Principais Métodos Ágeis



- **Scrum:**
 - Método ágil mais usado atualmente. Aplica-se não só ao desenvolvimento de softwares como a qualquer ambiente de trabalho. Focado na gestão do projeto e tem como base o planejamento iterativo e incremental.
 - “Os princípios do **Scrum** são coerentes com o manifesto ágil e são usados para orientar as atividades de desenvolvimento dentro de um processo que incorpora as seguintes atividades metodológicas: requisitos, análise, projeto, evolução e entrega” (PRESSMAN & MAXIM, 2016, p. 78).

45

Principais Métodos Ágeis



- **Melhores práticas:**
 - Indivíduos e interação mais do que processos e ferramentas.
 - Software em funcionamento mais do que documentação.
 - Colaboração com o cliente mais do que contratos e negociações.
 - Respostas a mudanças mais do que planejamento.

46

Principais Métodos Ágeis



- **Papéis:**
 - **Product Owner:** é o CEO do projeto, quem faz a ponte entre cliente e Scrum Team. É o responsável pelos resultados do projeto e por gerar valor para o cliente.
 - **Scrum Master:** é o coach do Scrum Team, a pessoa responsável por orientar os profissionais em relação às melhores práticas Scrum, remover barreiras e motivar.
 - **Scrum Team:** são os profissionais que efetivamente executam o desenvolvimento do projeto.

47

Principais Métodos Ágeis



- **Elementos:**
 - **Sprints:** representa um tempo definido dentro do qual um conjunto de atividades deve ser executado, geralmente duram de 2 a 4 semanas.
 - **Product Backlog:** funcionalidades a serem implementadas no projeto, mantidas em uma lista.
 - **Daily Scrum:** Diariamente, em uma Sprint, a equipe faz uma breve reunião de no máximo 15 minutos com todos os participantes em pé.

48

Principais Métodos Ágeis

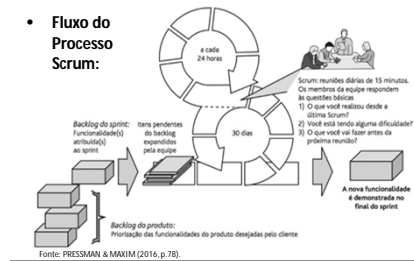


- **Sprint Review Meeting:** Ao final de um Sprint, a equipe apresenta as funcionalidades implementadas em uma Sprint Review Meeting onde o time mostra o que foi alcançado neste Sprint.
- **Burn Down Chart:** O Burndown é um simples gráfico, com dois eixos X e Y, baseado nas atividades que não ultrapassem um dia de trabalho. O eixo X indica o número de tarefas existentes no Sprint e o eixo Y os dias que representam o tamanho do Sprint.

49

Principais Métodos Ágeis

- **Fluxo do Processo Scrum:**



50

Principais Métodos Ágeis

- **eXtreme Programming (XP):**
 - Criado em meados da década de 90.
 - É a abordagem mais amplamente utilizada para desenvolvimento de software ágil.
 - A equipe XP garante a integração e a sinergia necessárias para um bom desempenho.
 - Focado no desenvolvimento de softwares que tem três pilares como base: agilidade no desenvolvimento da solução, economia de recursos e qualidade do produto final.

51

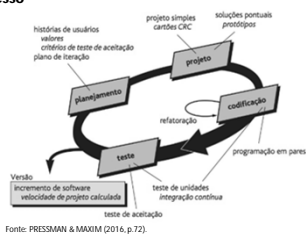
Principais Métodos Ágeis

- **Satisfação do Cliente durante todo o processo de desenvolvimento:**
 - Cliente sempre à disposição.
 - Reuniões de planejamento (*planning game*).
 - Reuniões diárias, de 15 minutos, para alinhamento (*stand up meeting*).
 - Integração contínua dos módulos desenvolvidos.
 - Mudanças incrementais.
 - Entregas frequentes ao cliente (*small releases*).
 - Design simples e funcional.
 - Testes de aceitação.
 - Refatoração ou melhoria contínua.

52

Principais Métodos Ágeis

- **O Processo XP:**



53

Principais Métodos Ágeis

- **Feature Driven Development (FDD):**

- Criado em Cingapura, entre 1997 e 1999, o FDD é um método ágil que reúne as melhores práticas de outros métodos, como o de Coad.
- **Premissa básica:** focada em funcionalidades, o que permite à equipe do projeto realizar um planejamento incremental por fases com integração contínua das funcionalidades.
- Aplica controle de qualidade e gerenciamento de configurações em todas as fases do projeto.
- Planejamento incremental com testes de software.

54

Principais Métodos Ágeis

- **Microsoft Solutions Framework (MSF):**
 - Destinado ao desenvolvimento de soluções tecnológicas **por equipes reduzidas**.
 - Focado na diminuição de riscos para o negócio e no aumento da qualidade do produto final.
 - O MSF está mais focado na gestão do projeto do que no desenvolvimento da solução em si.
- **Premissas:** alinhamento da tecnologia desenvolvida aos objetivos de negócio do cliente; escopo bem estruturado e detalhado; desenvolvimento iterativo; gerenciamento de riscos; e agilidade na resposta a mudanças.

55

Principais Métodos Ágeis

- **Dynamic System Development Model (DSDM):**
 - É um dos métodos ágeis mais antigos empregados não só no desenvolvimento de projetos como no meio tecnológico.
 - Ele é destinado ao desenvolvimento de projetos com orçamento fixo e prazos curtos.
- **Melhores Práticas:** desenvolvimento incremental e iterativo; colaboração entre cliente e equipe; e integração de funcionalidades.

56

Resolução da SP

Exercício: Identificação de Classes

57

Descrição – Agência de Estágios

Deseja-se desenvolver um sistema para uma agência de estágios que atua no ramo de seleção e contratação de candidatos para as empresas conveniadas. A agência precisa manter os cadastros de: candidato, empresa, instituição de ensino, entrevista e contrato.

O candidato é a pessoa que se cadastra na agência a procura de uma vaga de estágio. Um candidato pode fazer o seu cadastro na agência desde que seja estudante. Ele pode participar de várias entrevistas. Um candidato é descrito por: CPF, nome, endereço completo, idade, telefone, sexo, filiação, escolaridade e instituição de ensino atual. O candidato pode fazer o seu cadastro on-line e cadastrar o seu currículo. Ao realizar o seu cadastro on-line, o candidato receberá uma confirmação do seu cadastro e uma senha provisória. O candidato pode se inscrever, via Web, para uma vaga ofertada pela agência. Ao se inscrever em uma vaga, o candidato poderá emitir um comprovante de inscrição, sendo que o candidato receberá uma confirmação da inscrição efetuada, via e-mail.

58

Descrição – Agência de Estágios

A empresa é a pessoa jurídica que se cadastra na agência, ofertando vagas de estágio para estudantes. Uma empresa pode ofertar várias vagas de estágio. Uma empresa é descrita por: CNPJ, nome fantasia, razão social, inscrição estadual, ramo de atividade, endereço completo, telefone, fax, e-mail e contato. A vaga é descrita por uma descrição, cargo, quantidade, requisitos, salário, horário e período.

A agência cadastra todas as instituições de ensino médio e superior do município e da região. Uma instituição de ensino é cadastrada por: CNPJ, nome fantasia, razão social, endereço completo, telefone, fax, e-mail e contato.

A cada entrevista realizada com um candidato deve-se registrar: a data de realização, dados da empresa, dados da vaga, descrição da entrevista, nome do funcionário que realizou a entrevista, observação e situação (agendada, cancelada, realizada, aprovado ou reprovado). Para cada vaga pode-se realizar várias entrevistas. Algumas entrevistas geram a contratação (contrato), envolvendo os dados: n° do contrato, candidato, empresa, cargo, data de início, data de término, carga horária semanal, horário e salário.

59

Exercício: Identificação das Classes

- A partir da descrição do Estudo de Caso:
 - a) Identifique as classes de objetos e liste seus atributos.

60

Interação

Envie suas dúvidas!

61

Referências e Bibliografia

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

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **The unified modeling language: user guide**. Massachusetts: Longman, 2000.

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **UML: guia do usuário**. 2ed. Rio de Janeiro: Elsevier, 2006.

GUEDES, Gilleanes T. A.. **UML 2: uma abordagem prática**. 3ed. São Paulo: Novatec, 2018.

KIRCHOFF, Egon. **BPMN em exemplos: aprenda como modelar processos de negócio**. Kirchhoff, 2015.

LAUDON, K. C.; LAUDON, J. Price. **Gerenciamento de sistemas de informação**. 3 ed. LTC: Rio de Janeiro, 2001.

LAUDON, Kenneth C.; LAUDON, Jane Price. **Sistemas de informação gerenciais**. 11 ed. São Paulo: Pearson Prentice Hall, 2014.

62

Referências e Bibliografia

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

RAINER, R. K.; CEGIELSKI, C. G. **Introdução a sistemas de informação: apoiando e transformando negócios na era da mobilidade**. 5ed. Rio de Janeiro: Elsevier, 2016.

RUMBAUGH, James et al.. **Modelagem e projetos baseados em objetos**. Rio de Janeiro: Campus, 1994.

SOMMERVILLE, Ian. **Engenharia de software**. 9ed. São Paulo: Addison-Wesley, 2011.

VALLE, R.; OLIVEIRA, S. B. (Org.). **Análise e modelagem de processos de negócio: foco na notação BPMN (Business Process Modeling Notation)**. São Paulo: Atlas, 2013.

63



64