

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/325115973>

Princípios de Análise e Projeto de Sistemas com UML

Book · May 2018

CITATIONS

17

READS

6,816

1 author:



[Eduardo Bezerra](#)

Centro Federal de Educação Tecnológica Celso Suckow da Fonseca (CEFET/RJ)

44 PUBLICATIONS 56 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Data Preprocessing [View project](#)

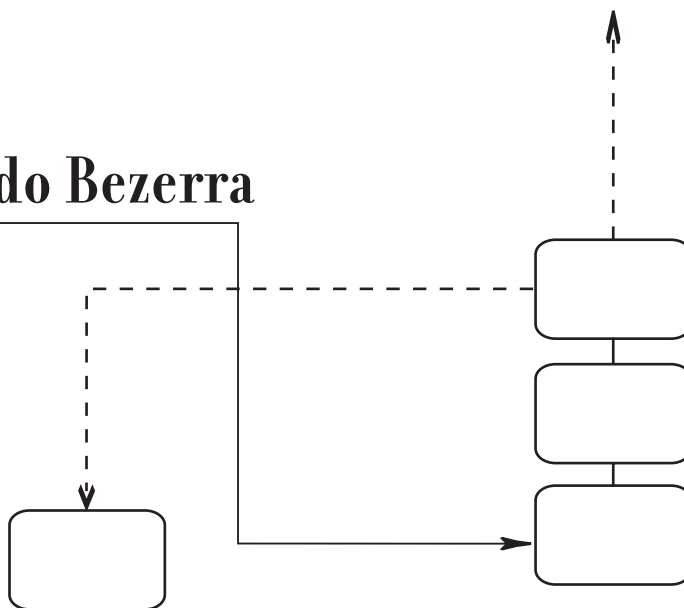


Applied Deep Learning [View project](#)

PRINCÍPIOS DE ANÁLISE E PROJETO DE SISTEMAS COM UML



Eduardo Bezerra



PRINCÍPIOS DE ANÁLISE E PROJETO DE SISTEMAS COM UML

Consultoria Editorial

Lorenzo Ridolfi
Gerente Sênior Accenture

Sérgio Colcher
*Professor do Departamento
de Informática da PUC-Rio*

3ª edição totalmente revista e atualizada



© 2015, Elsevier Editora Ltda.

Todos os direitos reservados e protegidos pela Lei nº 9.610, de 19/02/1998.

Nenhuma parte deste livro, sem autorização prévia por escrito da editora, poderá ser reproduzida ou transmitida, sejam quais forem os meios empregados: eletrônicos, mecânicos, fotográficos, gravação ou quaisquer outros.

Copidesque: Gabriel Pereira

Editoração Eletrônica: Mojo Design

Elsevier Editora Ltda.

Conhecimento sem Fronteiras

Rua Sete de Setembro, 111 – 16º andar

20050-006 – Rio de Janeiro – RJ

Rua Quintana, 753 – 8º andar

04569-011 – Brooklin – São Paulo – SP

Serviço de Atendimento ao Cliente

0800 026 53 40

atendimento1@elsevier.com

ISBN: 978-85-352-2626-3

ISBN (versão digital): 978-85-352-2627-0

Nota: Muito zelo e técnica foram empregados na edição desta obra. No entanto, podem ocorrer erros de digitação, impressão ou dúvida conceitual. Em qualquer das hipóteses, solicitamos a comunicação à nossa Central de Atendimento, para que possamos esclarecer ou encaminhar a questão.

Nem a editora nem o autor assumem qualquer responsabilidade por eventuais danos ou perdas a pessoas ou bens, originados do uso desta publicação.

CIP-BRASIL. CATALOGAÇÃO-NA-FONTE
SINDICATO NACIONAL DOS EDITORES DE LIVROS, RJ

B469p
3. ed.

Bezerra, Eduardo, 1972-

Princípios de análise e projeto de sistemas com UML / Eduardo Bezerra.

- [3. ed.] - Rio de Janeiro : Elsevier, 2015.

416 p. : il. ; 24 cm.

ISBN 978-85-352-2626-3

1. Métodos orientados a objetos (Computação). 2. UML (Computação).
3. Análise de sistemas. 4. Projeto de sistemas. I. Título.

14-18048

CDD: 005.117 CDU: 004.414.2

Agradecimentos

J á se passaram 12 anos desde o lançamento da primeira edição deste livro. Durante todo esse tempo, diversas pessoas me ajudaram a esclarecer meu entendimento sobre os assuntos de que trato neste livro. A todas elas, devo meus sinceros agradecimentos. Começo por agradecer aos diversos leitores das duas primeiras edições, que contribuíram com críticas e sugestões para o melhoramento da mesma. Agradeço também a meus alunos, nas diversas instituições de ensino pelas quais passei. Certamente a tarefa de professor é uma das melhores maneiras de aprender. Devo agradecimentos também a todos os meus colegas professores com os quais troquei ideias e ensinamentos sobre o problema da modelagem de sistemas de software: Ronaldo Goldschmidt, Carmem de Queiróz, Jorge Soares, Ismael Humberto, Leandro Chernicharo, Ricardo Choren, dentre outros. Obrigado também à equipe editorial da Elsevier, por toda a paciência e profissionalismo durante o tempo em que trabalhamos na produção desta edição. Finalmente, e não menos importante, agradeço a toda a minha família, pelo carinho e incentivo. Em especial, agradeço a meus irmãos, José, Edmar, Emanuel e Helton, à minha querida esposa Aline, ao meu filho, Felipe, e ao Janu.

Sumário

Prefácio.....	XIII
1 Visão geral.....	1
1.1 Modelagem de sistemas de software.....	2
1.2 O paradigma da orientação a objetos.....	4
1.2.1 Classes e objetos.....	7
1.2.2 Operação, mensagem e estado.....	7
1.2.3 O papel da abstração na orientação a objetos.....	8
1.3 Evolução histórica da modelagem de sistemas.....	13
1.4 A Linguagem de Modelagem Unificada (UML).....	15
1.4.1 Visões de um sistema.....	16
1.4.2 Diagramas da UML.....	17
2 O processo de desenvolvimento de software.....	21
2.1 Atividades típicas de um processo de desenvolvimento.....	22
2.1.1 Levantamento de requisitos.....	22
2.1.2 Análise.....	26
2.1.3 Projeto (desenho).....	29
2.1.4 Implementação.....	30
2.1.5 Testes.....	30
2.1.6 Implantação.....	31
2.2 O componente humano (participantes do processo).....	32
2.2.1 Gerentes de projeto.....	32
2.2.2 Analistas.....	32
2.2.3 Projetistas.....	34
2.2.4 Arquitetos de software.....	34
2.2.5 Programadores.....	34
2.2.6 Especialistas do domínio.....	35
2.2.7 Avaliadores de qualidade.....	36
2.3 Modelos de ciclo de vida.....	36
2.3.1 O modelo de ciclo de vida em cascata.....	36
2.3.2 O modelo de ciclo de vida iterativo e incremental.....	38

2.4	Utilização da UML no processo iterativo e incremental	42
2.5	Prototipagem	42
2.6	Ferramentas CASE	43
3	Mecanismos gerais	47
3.1	Estereótipos	47
3.2	Notas explicativas	48
3.3	Etiquetas valoradas (<i>tagged values</i>)	49
3.4	Restrições	50
3.5	Pacotes	50
3.6	OCL	52
4	Modelagem de casos de uso	53
4.1	Modelo de casos de uso	54
4.1.1	Casos de uso	54
4.1.2	Atores	60
4.1.3	Relacionamentos	61
4.2	Diagrama de casos de uso	69
4.3	Identificação dos elementos do MCU	73
4.3.1	Identificação de atores	74
4.3.2	Identificação de casos de uso	74
4.4	Construção do modelo de casos de uso	77
4.4.1	Construção do diagrama de casos de uso	78
4.4.2	Documentação dos atores	80
4.4.3	Documentação dos casos de uso	80
4.5	Documentação suplementar ao MCU	84
4.5.1	Regras do negócio	84
4.5.2	Requisitos de desempenho	86
4.5.3	Requisitos de interface gráfica	87
4.6	O MCU em um processo de desenvolvimento iterativo	87
4.6.1	O MCU nas atividades de análise e projeto	88
4.6.2	O MCU e outras atividades do desenvolvimento	90
4.7	Estudo de caso	91
4.7.1	Descrição da situação	91
4.7.2	Regras do negócio	92
4.7.3	Documentação do MCU	93
5	Modelagem de classes de análise	109
5.1	Estágios do modelo de classes	111
5.2	Diagrama de classes	112
5.2.1	Classes	112

5.2.2	Associações	113
5.2.3	Generalizações e especializações	129
5.3	Diagrama de objetos	136
5.4	Técnicas para identificação de classes	138
5.4.1	Análise textual de Abbott	139
5.4.2	Análise dos casos de uso	140
5.4.3	Técnicas baseadas em responsabilidades	145
5.4.4	Padrões de análise	156
5.4.5	Outras técnicas de identificação	160
5.4.6	Discussão	160
5.5	Construção do modelo de classes	163
5.5.1	Definição de propriedades	163
5.5.2	Definição de associações	165
5.5.3	Organização da documentação	166
5.6	Modelo de classes no processo de desenvolvimento	168
5.7	Estudo de caso	169
5.7.1	Análise do caso de uso: Fornecer Grade de Disponibilidades	170
5.7.2	Análise do caso de uso: Realizar Inscrição	171
5.7.3	Análise do caso de uso: Lançar Avaliações	174
5.7.4	Análise das regras do negócio	175
5.7.5	Documentação das responsabilidades	177
5.7.6	Glossário de conceitos	178
6	Passagem da análise para o projeto	183
6.1	Detalhamento dos aspectos dinâmicos	185
6.2	Refinamento dos aspectos estáticos e estruturais	185
6.3	Projeto da arquitetura	186
6.4	Persistência de objetos	187
6.5	Projeto de interface gráfica com o usuário	188
6.6	Projeto de algoritmos	188
6.7	Padrões de software	189
7	Modelagem de interações	191
7.1	Elementos da modelagem de interações	193
7.1.1	Mensagens	195
7.1.2	Atores	199
7.1.3	Objetos	199
7.1.4	Classes	200
7.1.5	Coleções de objetos	201
7.2	Diagrama de sequência	203
7.2.1	Linhas de vida	204

7.2.2 Mensagens.....	205
7.2.3 Ocorrências de execução.....	206
7.2.4 Criação e destruição de objetos.....	206
7.3 Diagrama de comunicação.....	208
7.4 Modularização de interações.....	210
7.4.1 Quadros.....	211
7.4.2 Diagrama de visão geral da interação.....	216
7.5 Construção do modelo de interações.....	217
7.5.1 Responsabilidades e mensagens.....	217
7.5.2 Coesão e acoplamento.....	218
7.5.3 Encapsulamento.....	221
7.5.4 Procedimento de construção do modelo de interações.....	226
7.6 Modelo de interações em um processo iterativo.....	231
7.7 Estudo de caso.....	234
7.7.1 Operações de sistema.....	234
7.7.2 Observações gerais.....	234
7.7.3 Modelos de interações.....	235
7.7.4 Visão geral das interações em um caso de uso.....	250

8 Modelagem de classes de projeto.....255

8.1 Reúso: padrões, frameworks, bibliotecas, componentes.....	256
8.2 Especificação de atributos.....	260
8.2.1 Notação da UML para atributos.....	260
8.3 Especificação de operações.....	263
8.3.1 Notação da UML para operações.....	266
8.3.2 Dicas práticas.....	265
8.3.3 Projeto por contrato.....	265
8.3.4 Operações de criação e destruição de objetos.....	268
8.3.5 Seletores e modificadores.....	268
8.3.6 Outras operações típicas.....	269
8.4 Especificação de associações.....	270
8.4.1 O conceito de dependência.....	270
8.4.2 Transformação de associações em dependências.....	272
8.4.3 Navegabilidade de associações.....	272
8.4.4 Implementação de associações.....	274
8.5 Herança.....	279
8.5.1 Tipos de herança.....	279
8.5.2 Classes abstratas.....	280
8.5.3 Operações polimórficas.....	282
8.5.4 Interfaces.....	284
8.5.5 Acoplamentos concreto e abstrato.....	288

8.5.6	Reúso por delegação.....	290
8.5.7	Classificação dinâmica.....	291
8.6	Padrões de projeto.....	293
8.6.1	Composite.....	294
8.6.2	Observer.....	295
8.6.3	Strategy.....	296
8.6.4	Factory Method.....	297
8.6.5	Mediator.....	299
8.6.6	Façade.....	299
8.7	Modelo de classes de projeto em um processo iterativo.....	300
8.8	Estudo de caso.....	302
9	Modelagem de estados.....	311
9.1	Diagrama de transição de estado.....	312
9.1.1	Estados.....	312
9.1.2	Transições.....	314
9.1.3	Eventos.....	314
9.1.4	Condição de guarda.....	316
9.1.5	Ações.....	316
9.1.6	Atividades.....	317
9.1.7	Ponto de junção.....	317
9.1.8	Cláusulas entry, exit e do.....	318
9.1.9	Transições internas.....	320
9.1.10	Exemplo.....	321
9.1.11	Estados aninhados.....	321
9.1.12	Estados concorrentes.....	322
9.2	Identificação dos elementos de um diagrama de estados.....	323
9.3	Construção de diagramas de transições de estados.....	324
9.4	Modelagem de estados no processo de desenvolvimento.....	326
9.5	Estudo de caso.....	327
10	Modelagem de atividades.....	331
10.1	Diagrama de atividade.....	331
10.1.1	Fluxo de controle sequencial.....	332
10.1.2	Fluxo de controle paralelo.....	333
10.1.2.1	Raias de natação.....	333
10.2	Diagrama de atividade no processo de desenvolvimento iterativo.....	334
10.2.1	Modelagem dos processos do negócio.....	335
10.2.2	Modelagem da lógica de um caso de uso.....	335
10.2.3	Modelagem da lógica de uma operação complexa.....	335
10.3	Estudo de caso.....	335

11	Arquitetura do sistema	339
11.1	Arquitetura lógica	340
11.1.1	Conceito de camada de software	342
11.1.2	Camadas típicas de um sistema de informação	344
11.1.3	O padrão MVC e sua relação com a arquitetura lógica	352
11.2	Arquitetura física	355
11.2.1	Alocação de camadas lógicas aos nós de processamento	356
11.2.2	Alocação de componentes aos nós de processamento	359
11.2.3	Padrões e tecnologias para distribuição de objetos	362
11.3	Projeto da arquitetura no processo de desenvolvimento	363
12	Mapeamento de objetos para o modelo relacional	365
12.1	Projeto de banco de dados	366
12.1.1	Conceitos do modelo de dados relacional	367
12.1.2	Mapeamento de objetos para o modelo relacional	369
12.1.3	Classes e seus atributos	370
12.1.4	Associações	371
12.1.5	Agregações e composições	374
12.1.6	Associações reflexivas	374
12.1.7	Associações ternárias	375
12.1.8	Classes associativas	375
12.1.9	Generalização	376
12.2	Construção da camada de persistência	379
12.2.1	Acesso direto ao banco de dados	381
12.2.2	Uso de um SGBDOO ou de um SGBDOR	381
12.2.3	Padrão DAO	383
12.2.4	Frameworks ORM	385
	Referências	391
	Índice	395

Prefácio

S seja bem-vindo à terceira edição de *Princípios de Análise e Projeto de Sistemas com UML*. Este livro é uma introdução aos conceitos fundamentais necessários para se realizar a análise e o projeto de sistemas de software orientados a objetos com o uso da Linguagem de Modelagem Unificada (UML). Desde o lançamento da 1ª edição desta obra, já existiam bons livros disponíveis aqui no Brasil discutindo a modelagem de sistemas orientados a objetos com UML. No entanto, uma razão que me levou a escrever esta obra foi o fato de alguns desses livros darem uma ênfase maior à descrição da UML em si.

De fato, a UML define uma notação padrão que pode ser utilizada por desenvolvedores de software orientado a objetos. Sem dúvida o domínio dessa notação é importante para qualquer desenvolvedor que queira aproveitar todas as capacidades que a UML fornece. Mas, igualmente importante, em especial para iniciantes no desenvolvimento de software, é o entendimento de como aplicar a notação da UML na modelagem. É esse enfoque que procurei dar neste livro. Em vista disso, esta obra *não* fornece uma referência completa sobre a notação definida pela UML. Em vez disso, ela descreve uma parte dessa notação e também como realizar a análise e o projeto de sistemas orientados a objetos através de parte da notação mais utilizada.

Durante todo o livro, exemplos são utilizados para demonstrar a aplicação da UML em situações práticas de modelagem. Ao fim de cada capítulo

são fornecidos exercícios para testar o conteúdo apreendido pelo leitor. Além disso, um estudo de caso, o Sistema de Controle Acadêmico (SCA), é desenvolvido para os principais tópicos abordados com o objetivo de exemplificar a aplicação dos procedimentos e dicas de modelagem que são apresentadas em cada capítulo.

Público-alvo

Este livro é destinado a estudantes de cursos técnicos, de graduação ou pós-graduação em informática, computação, sistemas de informação ou engenharia de software que devem cursar uma ou mais disciplinas de análise e projeto orientados a objetos. Esta obra também pode ser utilizada como guia por estudantes no desenvolvimento de seus projetos finais de curso. Profissionais que desenvolvem sistemas segundo outros paradigmas (que não o orientado a objetos) também podem encontrar neste livro uma boa iniciação aos conceitos da orientação a objetos e da sua aplicação à modelagem de sistemas de software. Em todos os casos, o livro pode servir como uma fonte de referência e dicas práticas sobre a aplicação da UML e de outras técnicas no desenvolvimento de um sistema de software orientado a objetos.

O conhecimento de alguma linguagem de programação orientada a objetos (p. ex., Java, C#, C++ etc.) é desejável (mas não obrigatório) para o bom entendimento dos assuntos tratados neste livro. Mais especificamente, este livro fornece diversos exemplos de trechos de código-fonte em linguagem Java. Entretanto, esses exemplos devem ser facilmente entendidos por profissionais familiarizados com outras linguagens orientadas a objetos.

Organização dos capítulos

O Capítulo 1 apresenta uma breve introdução à utilização do paradigma da orientação a objetos e da UML. O objetivo deste capítulo é fornecer uma visão geral sobre a análise e o projeto de sistemas de software sob o ponto de vista de orientação a objetos. Os principais conceitos do paradigma da orientação a objetos são introduzidos neste capítulo.

O Capítulo 2 descreve as principais atividades constituintes de um processo de desenvolvimento de software. Também descrevemos os principais profissionais envolvidos nesse processo, juntamente com suas respectivas atribuições. O processo de desenvolvimento em cascata é apresentado com o objetivo de motivar o surgimento do processo incremental e evolutivo. Em seguida, este último é também descrito e apresentado como a forma atual de se desenvolver sistemas orientados a objetos. Na maioria dos capítulos seguintes são feitas alusões à utilização da UML em um processo de desenvolvimento incremental e evolutivo.

O Capítulo 3, o menor deste livro, é apenas uma apresentação dos mecanismos de uso geral da UML. Essa apresentação se faz necessária em virtude de esses mecanismos serem utilizáveis em diversos diagramas da UML. Nos capítulos posteriores, fazemos uso e estendemos os conceitos introdutórios apresentados neste capítulo.

No Capítulo 4, apresentamos o modelo de casos de uso e os diversos elementos do diagrama de casos de uso da UML. Além disso, são fornecidas diversas dicas práticas que podem ser utilizadas na construção desse modelo. Esse capítulo também enfatiza o modelo de casos de uso como um ponto central de um processo de desenvolvimento que utilize a UML como linguagem de modelagem.

O Capítulo 5 descreve a construção do modelo de classes de análise de um sistema de software orientado a objetos (SSOO). Os principais elementos de notação definidos pela UML para a construção do diagrama de classes são descritos. Também é apresentado o conceito de responsabilidade de um objeto. Descrevemos, além disso, diversas técnicas úteis na identificação das classes iniciais de um SSOO, como a análise textual de Abbot, a análise de casos de uso e o uso de padrões de análise. Nesta 3ª edição, adiciono a este capítulo uma pequena introdução aos padrões táticos do DDD (*Domain Driven Design*) no contexto de identificação de classes do domínio.

O Capítulo 6 serve como uma apresentação do conteúdo dos capítulos que o seguem. A partir desse capítulo, a descrição das atividades de projeto começa a tomar o lugar da descrição das atividades de análise.

A modelagem de interações entre objetos em um SSOO é discutida no Capítulo 7. Nesse capítulo, apresento a ideia de que as construções do modelo de classe e do modelo de interações são interdependentes: a construção de um modelo fornece informações para a construção do outro e vice-versa. Seguindo a filosofia das edições anteriores, não me preocupei em apresentar todos os elementos de notação, mas apenas os que, na minha visão, são os mais importantes e relevantes em situações práticas de modelagem. Nessa 3ª edição, estendi o conteúdo desse capítulo com a descrição de boas práticas e princípios de projeto relevantes para a construção correta do modelo de interações.

O Capítulo 8 retoma a discussão sobre o modelo de classes, agora com um enfoque nas características de modelagem referentes à fase de projeto. Conceitos fundamentais ao projeto de um SSOO são apresentados: classe abstrata, interface, polimorfismo, tipos de acoplamento, projeto por contrato etc. Na 2ª edição, apresentei uma pequena introdução a um assunto um tanto avançado, mas cada vez mais sedimentado no desenvolvimento de um SSOO: padrões de projeto. Essa descrição sobre padrões de projeto continua nesse capítulo, mas é aprofundada em outras partes do livro.

O Capítulo 9 descreve a sintaxe, a semântica e a construção dos diagramas de transições de estados.

O Capítulo 10 finaliza a apresentação dos diagramas da UML relacionados à parte comportamental do sistema. Esse capítulo descreve os diagramas de atividades.

O Capítulo 11 faz uma introdução aos conceitos relacionados à arquitetura de um sistema de SSOO. Termos como subsistema, componente e camada são descritos. Outros diagramas da UML são apresentados: o de componentes, o de pacotes e o de implantação. Nesta 3ª edição, esse capítulo foi estendido com um maior detalhamento acerca das camadas tipicamente encontradas na arquitetura em um sistema de informação.

Finalmente, o Capítulo 12 descreve alternativas de representação de objetos em um mecanismo de armazenamento persistente como um sistema de gerência de bancos de dados relacional. É feita também uma introdução a questões relacionadas à implementação de uma camada de persistência em um SSOO.

Recursos na web

Como informação suplementar à contida neste livro, é fornecido um site na própria editora Elsevier. Acesse a página da Editora (www.elsevier.com.br). Nesse endereço, o leitor pode obter informações e material relacionado ao livro. Entre os recursos que podem ser encontrados no site, estão os seguintes:

- *Soluções de alguns dos exercícios propostos no livro.* O leitor pode encontrar diversos exercícios resolvidos no material disponibilizado no site da editora.
- *Complementos ao estudo de caso apresentado no livro.* O estudo de caso que desenvolvo no livro é denominado *Sistema de Controle Acadêmico* (SCA). No final de alguns capítulos, forneço diversos exemplos de modelagem no contexto do SCA. Um problema que surge é como continuar e complementar esses exemplos. Uma solução que adoto a partir dessa segunda edição é utilizar a Internet como fornecedora de novos materiais acerca deste estudo de caso.
- *Outras fontes de informação.* O material disponível no site da editora contém também endereços para outras fontes interessantes sobre modelagem de sistemas de software orientados a objetos. Seguindo a natureza dinâmica da Internet, o conteúdo do site será modificado de tempos em tempos. O leitor também pode utilizar esse site para entrar em contato comigo, com o objetivo de trocar ideias sobre o livro.

Convite ao leitor

Finalmente, convido o leitor a prosseguir pelo restante desta obra. Espero que as informações contidas neste livro o ajudem de alguma forma, e que a leitura seja a mais agradável possível. Tentei dar o meu melhor para produzir um texto cuja leitura seja aprazível e didática. Entretanto, pelo fato de a produção de um livro ser uma tarefa bastante complexa, tenho consciência de que erros e inconsistências ainda se escondem por entre as linhas que o compõem. Para os que quiserem entrar em contato comigo para trocar ideias e fornecer críticas e sugestões, fiquem à vontade para enviar uma mensagem.

Eduardo Bezerra

Rio de Janeiro

ebezerra@cefet-rj.br

18 de agosto de 2014

