

**Vitor Amadeu Souza**

Introdução a

**UML**

Com exemplos no Java

Volume II

© 2023 by Cerne Tecnologia e Treinamento Ltda.

© 2023 by Vitor Amadeu Souza

Nenhuma parte desta publicação poderá ser reproduzida sem autorização prévia e escrita de **Cerne Tecnologia e Treinamento Ltda.** Este livro publica nomes comerciais e marcas registradas de produtos pertencentes a diversas companhias. O editor utiliza as marcas somente para fins editoriais e em benefício dos proprietários das marcas, sem nenhuma intenção de atingir seus direitos.

**Janeiro de 2024**

**Direitos reservados por:**

Cerne Tecnologia e Treinamento Ltda

*Produção: Cerne Tecnologia e Treinamento*

*E-mail da Empresa: cerne@cerne-tec.com.br*

*Home Page: www.cerne-tec.com.br.com.br*

*Atendimento ao Consumidor: sac@cerne-tec.com.br*

*Contato com o Autor: vitor@cerne-tec.com.br*



**FEITO NO BRASIL**

***“Recompensou-me o Senhor conforme a minha justiça,  
retribuiu-me conforme a pureza das minhas mãos.”***

**SI 18:20**

## Kits Didáticos e Gravadores da Cerne Tecnologia

A Cerne tecnologia têm uma linha completa de aprendizado para os microcontroladores da família PIC, 8051, Holtek, dsPIC, ARM, STM etc. Veja os detalhes de um kit na figura abaixo:



### Kit Cerne PIC12F

- Microcontrolador PIC12F675
- Botões
- Leds
- Gravação ICSP
- E muito mais!

Uma linha completa de componentes para o desenvolvimento de seus projetos eletrônicos como displays, PICs, botões, leds, cristais dentre outros. Visite a nossa página na Internet, no endereço [www.cerne-tec.com.br](http://www.cerne-tec.com.br) e conheça melhor nossos serviços e produtos.



# Sumário

<b>Introdução.....</b>	<b>6</b>
<b>Capítulo I – Diagrama de Classe.....</b>	<b>7</b>
1. UML.....	7
2. Visual Paradigm.....	12
3. Componentes.....	15
4. Exemplo 1: Classe Automóvel.....	19
5. Exemplo 2: Classe Automóvel com Herança.....	23
6. Exemplo 3: Abstração.....	27
7. Exemplo 4: Agregação.....	30
8. Exemplo 5: Composição.....	34
9. Exemplo 6: Associação 1x1.....	37
10. Exemplo 7: Associação 1xN.....	39
11. Exemplo 8: Associação NxN.....	42

# Introdução

A proposta desta literatura é apresentar a modelagem de software usando a UML através de exemplos práticos, que envolvam modelagem e codificação no qual a linguagem Java foi utilizada como referência.

Em cada parte desta série de livros um diagrama é abordado, no qual o tema está no sumário da obra. Em todos os casos, exemplos práticos escritos em Java procuram envolver modelagem com programação, facilitando desta forma a compreensão deste assunto.

É importante que o leitor tenha conhecimentos de programação em Java, de modo a implementar as modelagens que serão apresentadas ao longo da obra.

Recomenda-se a leitura prévia ou conhecimentos dos assuntos abordados na obra *Antologia de projetos em Java No ambiente IntelliJ IDEA* do mesmo autor e editora.

# Capítulo I

## Diagrama de Classe

### 1. UML

A UML que significa Linguagem de Modelagem Unificada ou em inglês *Unified Modeling Language*, é uma linguagem padrão para visualização, especificação, construção e documentação de sistemas de software. Ela fornece um conjunto de notações gráficas para criar modelos visuais de sistemas, permitindo que os desenvolvedores comuniquem efetivamente as ideias e os conceitos relacionados ao design e à implementação de um software.

Ela não foi inventada por uma única pessoa, mas sim foi desenvolvida por um consórcio de profissionais de software no final da década de 1990. O desenvolvimento da UML foi liderado por Grady Booch, Ivar Jacobson e James Rumbaugh, conhecidos como "Os Três Amigos". Cada um desses especialistas trouxe suas próprias metodologias de modelagem (Booch, OMT e OOSE, respectivamente) para o processo de criação da UML.

Em 1997, a UML foi adotada como um padrão pela Object Management Group (OMG), uma organização sem fins lucrativos que define padrões abertos para o desenvolvimento e a gestão de sistemas distribuídos. Desde então, a UML tem sido mantida e evoluída pela OMG, com contribuições da comunidade de desenvolvimento de software. A UML tornou-se um padrão amplamente aceito na indústria de software para modelagem de sistemas complexos e é utilizada por muitos profissionais e organizações em todo o mundo.

A UML foi desenvolvida para unificar várias abordagens de modelagem utilizadas na indústria de software. Ela oferece um conjunto de diagramas que abrangem diferentes aspectos do desenvolvimento de software, permitindo que as equipes de desenvolvimento capturem e comuniquem informações sobre a estrutura, o comportamento, as interações e os processos de um sistema.

A UML possui vários tipos de diagramas. No momento da escrita deste livro a versão disponível é a 2.5.1 e conta com os seguintes tipos de diagramas:

**Diagrama de Casos de Uso:**

- Descreve interações entre um sistema e seus atores (usuários ou outros sistemas).
- Identifica casos de uso (funcionalidades) e como os atores interagem com o sistema.

**Diagrama de Classes:**

- Representa a estrutura estática de um sistema, identificando classes, atributos, métodos e os relacionamentos entre as classes.

**Diagrama de Sequência:**

- Mostra a interação entre objetos ao longo do tempo.
- Destaca a ordem das mensagens trocadas entre os objetos.

**Diagrama de Atividades:**

- Representa o fluxo de controle ou o fluxo de trabalho de um processo ou atividade.
- Usa diferentes formas para representar ações, decisões, forks, e merges.

**Diagrama de Estados:**

- Descreve os diferentes estados que um objeto ou sistema pode ter e as transições entre esses estados.
- Útil para modelar comportamentos reativos.

**Diagrama de Componentes:**

- Ilustra a estrutura física de um sistema, mostrando os componentes e suas interações.
- Indica as dependências entre os componentes.

**Diagrama de Implantação:**

- Descreve a disposição física de componentes em hardware.
- Mostra a topologia do ambiente de implantação.

**Diagrama de Objetos:**

- Representa instâncias de classes e os relacionamentos entre elas em um momento específico.

**Diagrama de Pacotes:**

- Organiza elementos em grupos lógicos (pacotes) para facilitar a gestão e compreensão do sistema.

**Diagrama de Comunicação:**

- Mostra interações entre objetos em termos de mensagens sequenciais.

**Diagrama de Visão Geral de Interação:**

- Oferece uma visão geral de como vários diagramas de interação se relacionam.