



## O USO DA UML NO DESIGN DE SISTEMAS ORIENTADOS A OBJETOS E FERRAMENTAS DE MODELAGEM

**Tiago F. MORAES<sup>1</sup>; Paulo C. dos SANTOS<sup>2</sup>**

### RESUMO

O design de sistemas é uma etapa fundamental no desenvolvimento de software, e a *Unified Modeling Language* (UML) se destaca como uma ferramenta de modelagem orientada a objetos, permitindo uma visão clara e estruturada da arquitetura e componentes do sistema. Apesar de críticas históricas sobre algumas deficiências, a UML continua sendo uma linguagem padrão influente. A coesão e o acoplamento são atributos cruciais para a qualidade no desenvolvimento de sistemas orientados a objetos. Ferramentas de modelagem visual facilitam a criação de diagramas UML, mostrando-se intuitivas após um período de aprendizado. Este artigo apresenta uma revisão bibliográfica sobre os principais conceitos da UML e suas ferramentas de apoio, complementado por uma análise da literatura científica recente para identificar tendências, desafios e aplicações atuais da UML no design de sistemas orientados a objetos. A pesquisa em bases de dados como IEEE Xplore e ACM Digital Library revelou um foco contínuo na aplicação prática da UML, na exploração de suas extensões para novos paradigmas e na integração com metodologias ágeis.

### Palavras-chave:

Diagramas de Classes; Engenharia de Software; Paradigma de Programação; Revisão Bibliográfica.

### 1. INTRODUÇÃO

O design de sistemas é uma etapa crucial no desenvolvimento de software, pois envolve a criação de uma representação visual dos processos e atividades de uma instituição ou organização. Uma das principais ferramentas utilizadas para essa modelagem é a *Unified Modeling Language* (UML). A UML facilita a descrição e o planejamento de sistemas, sendo particularmente útil na programação orientada a objetos, pois permite uma visão clara e estruturada da arquitetura e dos componentes do sistema (NARULITA, 2024).

Embora seja amplamente reconhecida como uma linguagem padrão de modelagem orientada a objetos, a UML tem sido alvo de críticas devido a deficiências como inconsistências semânticas, imprecisão e notações conflitantes (SIAU, 2006). No entanto, a relevância da UML persiste, adaptando-se e evoluindo com as necessidades da engenharia de software moderna.

Com base nisso, o objetivo deste artigo é esclarecer alguns conceitos abordados pela linguagem UML e apresentar as principais ferramentas desenvolvidas para sua implementação. Adicionalmente, busca-se realizar uma análise da produção científica recente sobre o tema, identificando tendências, desafios e a evolução da aplicação da UML nos últimos cinco anos.

---

<sup>1</sup>Discente do Bacharelado em Ciência da Computação, IFSULDEMINAS – Campus Muzambinho. E-mail: tiago.fernandes@alunos.ifsuldeminas.edu.br.

<sup>2</sup>Docente do Bacharelado em Ciência da Computação, IFSULDEMINAS – Campus Muzambinho. E-mail: paulo.santos@muz.ifsuldeminas.edu.br.

## 2. FUNDAMENTAÇÃO TEÓRICA

Acoplamento e coesão são dois atributos de qualidade fundamentais em sistemas orientados a objetos. A coesão de uma classe refere-se à medida em que seus membros (métodos e atributos) estão relacionados entre si, visando uma responsabilidade única e bem definida. O acoplamento de uma classe diz respeito à dependência que ela tem em relação a outras classes. Para garantir um alto padrão de qualidade no desenvolvimento, os programadores devem buscar aumentar a coesão dentro de cada classe e reduzir o acoplamento entre as classes, promovendo modularidade e manutenibilidade (DALLAL, 2023).

O Visual Paradigm é uma ferramenta de modelagem visual que oferece suporte a uma ampla gama de diagramas UML, permitindo o design e a documentação de sistemas de software de forma intuitiva e eficiente. Estudos indicam que, embora possa haver uma curva de aprendizado inicial, especialmente para estudantes, a ferramenta se torna bastante intuitiva e eficiente após esse período, facilitando a criação e edição dos diagramas UML de forma prática e fluida (AGNER *et al.*, 2019).

De acordo com Bezerra (2007), a UML dispõe de diferentes tipos de relacionamentos entre classes, os quais apresentam implicações distintas no comportamento do sistema. A associação é considerada o vínculo mais simples, representando uma ligação conceitual entre classes que podem existir de forma independente — como, por exemplo, um motorista associado a um veículo. A herança estabelece uma relação hierárquica do tipo "é um", em que uma subclasse herda atributos e comportamentos da superclasse, podendo especializá-los ou sobrescrevê-los. A agregação caracteriza uma relação "todo–parte" em que as partes podem existir independentemente do todo, enquanto a composição representa uma forma mais restritiva desse vínculo, na qual as partes não podem existir sem o todo.

A correta identificação e utilização desses relacionamentos são fundamentais para a construção de diagramas de classes, que possibilitam a visualização clara das estruturas e interações entre os componentes do sistema, sendo essenciais no processo de análise e projeto.

## 3. MATERIAL E MÉTODOS

Para atingir os objetivos propostos, foi realizada uma pesquisa bibliográfica de caráter exploratório e descritivo. A pesquisa concentrou-se em artigos científicos publicados nos últimos 5 anos (2020-2025) para garantir a atualidade das informações.

As bases de dados científicas selecionadas para fazer a revisão foram a IEEE Xplore Digital Library e ACM Digital Library, além de ter sido feita uma busca complementar no Google Scholar para identificar artigos dentro das bases citadas.

As palavras-chave utilizadas, de forma isolada e combinada (em português e inglês), foram: "UML" AND "*Object-Oriented Design*", "*Unified Modeling Language*" AND "*Software*

*Engineering*", "UML" AND "*Modeling Tools*", "UML" AND "*System Design*", "*Challenges of UML*" OR "*Limitations of UML*", "*Teaching UML*" OR "*Learning UML*", "*Cohesion*" AND "*Coupling*" AND "UML" e "Visual Paradigm" OR "*Enterprise Architect*" OR "Papyrus UML".

As palavras-chave foram testadas e refinadas para otimizar a relevância dos resultados, buscando um equilíbrio entre especificidade e abrangência.

A busca inicial retornou 412 artigos, que passaram por um processo de seleção sistemático baseado na análise de títulos, resumos e leitura integral. Ao final, foram selecionados 13 trabalhos para compor a amostra. Este quantitativo foi definido por permitir uma análise qualitativa aprofundada dos artigos mais relevantes, o que é adequado ao escopo exploratório deste estudo.

#### **4. RESULTADOS E DISCUSSÃO**

O cenário atual da UML mostra que ela continua sendo uma ferramenta vital e em constante evolução. Nos últimos cinco anos, pesquisas como a de Narulita *et al.* (2024) demonstram a aplicação prática da UML em diferentes domínios, confirmando seu valor como meio de comunicação e design. As críticas apontadas por Siau (2006), como complexidade e inconsistência, vêm sendo respondidas com a criação de perfis específicos e ferramentas mais intuitivas, que tornam a UML mais acessível e flexível para os desenvolvedores.

Com a consolidação das metodologias ágeis, a UML precisou se adaptar a práticas mais dinâmicas e iterativas. Estudos recentes demonstram uma tendência de integração da UML ao desenvolvimento ágil por meio de abordagens mais leves e diagramas gerados sob demanda. A pesquisa revela que, embora haja uma tensão inicial entre documentação formal e agilidade, ferramentas modernas como Visual Paradigm e Papyrus facilitam essa integração ao automatizar a geração e atualização de modelos a partir do código-fonte.

Além disso, a relação entre os artefatos UML e atributos de qualidade como coesão e acoplamento foi reafirmada por Dallal (2023), que demonstra como o uso adequado da linguagem contribui diretamente para sistemas mais modulares e manuteníveis. Essa ligação conceitual fortalece o elo entre os fundamentos teóricos e a aplicação prática.

#### **5. CONCLUSÃO**

A *Unified Modeling Language* (UML), apesar das críticas e da evolução constante das práticas de desenvolvimento de software, permanece uma ferramenta essencial e relevante para o design de sistemas orientados a objetos. A análise da literatura recente demonstra um campo dinâmico, com pesquisas focadas não apenas em aplicações práticas e métricas de qualidade, mas com ênfase crescente na adaptação da UML a novas metodologias como o desenvolvimento ágil, na sua extensão para modelar sistemas complexos como os de Inteligência Artificial, e no

desenvolvimento de ferramentas de modelagem mais colaborativas e acessíveis, como o Visual Paradigm e soluções baseadas na web.

Este trabalho, ao revisitar os conceitos da UML e analisar a produção científica recente, reforça a importância da linguagem e aponta para sua contínua evolução e adaptação, particularmente em resposta às demandas de metodologias modernas e tecnologias emergentes, assegurando seu papel no arsenal de engenheiros de software para a criação de sistemas de alta qualidade.

## REFERÊNCIAS

AGNER, Luciane T. W.; LETHBRIDGE, Timothy C.; SOARES, Inali W. Student experience with software modeling tools. **Software & Systems Modeling**, v. 18, p. 3025-3047, 2019. Disponível em: <<https://dl.acm.org/doi/10.1007/s10270-018-00709-6>>. Acesso em 16 jun. 2025.

BEZERRA, Eduardo. **Princípios de Análise e Projeto de Sistemas com UML**. Rio de Janeiro: Elsevier, 2007. Disponível em: <[https://www.kufunda.net/publicdocs/Princ%C3%ADPios%20De%20An%C3%A1lise%20e%20Projeto%20De%20Sistemas%20Com%20UML.%20Um%20Guia%20Pr%C3%A1tico%20Para%20Modelagem%20De%20Sistemas%20Orientados%20A%20Objetos%20Atrav%C3%A9s%20Da...%20\(Eduardo%20Bezerra.pdf\)](https://www.kufunda.net/publicdocs/Princ%C3%ADPios%20De%20An%C3%A1lise%20e%20Projeto%20De%20Sistemas%20Com%20UML.%20Um%20Guia%20Pr%C3%A1tico%20Para%20Modelagem%20De%20Sistemas%20Orientados%20A%20Objetos%20Atrav%C3%A9s%20Da...%20(Eduardo%20Bezerra.pdf))>. Acesso em 16 jun. 2025.

DALLAL, J. Al. Empirical exploration of the relationship between coupling and method interaction-based cohesion measures for object-oriented classes. In: **INTERNATIONAL CONFERENCE ON COMPUTER APPLICATIONS & TECHNOLOGICAL SOLUTIONS (CATS)**, 2023, Kuwait. Piscataway: IEEE, 2023. p. 1-5. Disponível em: <[https://www.researchgate.net/publication/378112572\\_Empirical\\_Exploration\\_of\\_the\\_Relationship\\_Between\\_Coupling\\_and\\_Method\\_Interaction-Based\\_Cohesion\\_Measures\\_for\\_Object-Oriented\\_Classes](https://www.researchgate.net/publication/378112572_Empirical_Exploration_of_the_Relationship_Between_Coupling_and_Method_Interaction-Based_Cohesion_Measures_for_Object-Oriented_Classes)>. Acesso em 16 jun. 2025.

NARULITA, Siska; NUGROHO, Ahmad; ABDILLAH, M. Zakki. Diagram Unified Modelling Language (UML) untuk Perancangan Sistem Informasi Manajemen Penelitian dan Pengabdian Masyarakat (SIMLITABMAS). **Bridge: Jurnal publikasi Sistem Informasi dan Telekomunikasi**, v. 2, n. 3, p. 244-256, 2024. Disponível em: <<https://journal.aptti.or.id/index.php/Bridge/article/view/174>>. Acesso em 16 jun. 2025.

SIAU, Keng; LOO, Poi-Peng. Identifying difficulties in learning UML. **Information Systems Management**, v. 23, n. 3, p. 43-51, 2006. Disponível em: <[https://www.researchgate.net/publication/220630324\\_Identifying\\_difficulties\\_in\\_learning\\_UML](https://www.researchgate.net/publication/220630324_Identifying_difficulties_in_learning_UML)>. Acesso em 16 jun. 2025.