

O USO DA *UNIFIED MODELING LANGUAGE* (UML) NA MODELAGEM DE SOFTWARE

Data de aceite: 01/08/2023

Henderson Matsuura Sanches

Centro Universitário UDF
Orcid: 0000-0003-2354-3393

RESUMO: Em seus primórdios, a *Unified Modeling Language* (UML) surgiu com o objetivo de juntar as diversas linguagens de modelagem da época, tornando-se, desta forma, a linguagem universal de modelagem de sistemas, atualmente a UML encontra-se na versão 2.5.1 conforme informado pela *Object Management Group* (OMG), o objetivo deste trabalho objetivo apresentar a modelagem de software utilizando a UML, apresentando alguns conceitos importantes dos diagramas como: Diagrama de Caso de Uso (DCU), Diagrama de Classe (DC), Diagrama de Atividade (DA) e do Diagrama de Sequência (DS) e apresentar alguns trabalhos utilizando a UML sendo a metodologia sendo a pesquisa bibliográfica realizada nas bases de pesquisa sendo; CAPES, Biblioteca Digital Brasileira de Teses e Dissertações (BDTD), *Google Acadêmico*, *SciELO*, *OpenLib (SOL)* a biblioteca digital mantida pela Sociedade Brasileira de Computação (**SBC**) e concluir-se que a UML continua sendo importante na

modelagem de *software*.

PALAVRAS-CHAVE: UML, Sistema de Informação, Especificação, Diagramas, Modelagem de Software.

ABSTRACT: In its early days, the Unified Modeling Language (UML) emerged with the aim of joining the various modeling languages of the time, thus becoming the universal language for systems modeling, currently the UML is in the current version 2.5 .1 as informed by the Object Management Group (OMG), the objective of this work is to present software modeling using the UML, presenting some important concepts of the diagrams such as: Use Case Diagram (UCD), Class Diagram (DC), Activity Diagram (DA) and Sequence Diagram (DS) and present some works using the UML being the methodology being the bibliographic research carried out in the research bases being; CAPES, Brazilian Digital Library of Theses and Dissertations (BDTD), Google Scholar, SciELO, OpenLib (SOL) the digital library maintained by the Brazilian Computing Society (SBC) and it is concluded that the UML continues to be important in software modeling.

KEYWORDS: UML, Information System, Specification, Diagrams, Software Modeling.

1 | INTRODUÇÃO

Em seus primórdios, a *Unified Modeling Language* (UML) surgiu com o objetivo de juntar as diversas linguagens de modelagem da época, tornando-se, desta forma, a linguagem universal de modelagem de sistemas. Desde sua criação, e até a sua versão atual 2.5.1 conforme informado no *site da Object Management Group* (OMG) organização responsável pela UML.

A UML tornou-se a linguagem líder para a modelagem de sistemas orientados a objetos e é largamente utilizada pelas empresas de *software* e pelas academias ao ensinar modelagem. A UML não oferece uma “receita de bolo” para essa escolha. Pode até parecer estranho, mas a linguagem também não diz quando devem ser criados os diagramas que compõem o modelo. O importante é que quando especificados eles sejam consistentes e bem construídos (VIEIRA, 2003). Porém um diagrama que é consenso entre inúmeros autores é o Diagrama de Caso de Uso (DCU) sendo o principal diagrama para a criação dos demais assim para toda a documentação (SANCHES, 2023).

1.1 Objetivo Geral

O presente trabalho tem como objetivo apresentar a modelagem de *software* utilizando a UML.

2 | PROCESSO DE MODELAGEM DE SOFTWARE

Kitchenham e Pfleeger (2008, p. 82) definem a modelagem de sistemas da seguinte forma: “Os modelos são usados durante o processo de engenharia de requisitos para ajudar a extrair os requisitos do sistema; durante o processo de projeto, são usados para descrever o sistema para os engenheiros que o implementam; e, após isso, são usados para documentar a estrutura e a operação do sistema.”. Afirmam, pois, que a importância da modelagem não está concentrada somente na atividade de planejamento do processo de *software* e sim, distribuída por todo o processo.

“O mundo moderno não poderia existir sem o *software*. Infraestruturas e serviços nacionais são controlados por sistemas computacionais, e a maioria dos produtos elétricos inclui um computador e um *software* que o controla.” (SOMMERVILLE, 2011, p. 2), com estas palavras Ian Sommerville dá início ao seu livro sobre Engenharia de *Software*. Se para Sommerville, em 2011, a sentença supracitada era uma verdade, agora, há quase uma década da publicação da obra, é possível se atestar a grande dependência do mundo aos produtos de *software*.

Cada vez mais são necessários produtos de *software* mais rápidos, mais complexos e melhores para suprir as necessidades da sociedade em geral. O processo de

modelagem de *software* conta com algumas etapas entre planejamento, desenvolvimento e implementação (BOOCH; RUMBAUGH; JACOBSON, 2006), “A modelagem é uma parte central de todas as atividades que levam à implantação de um bom *software*” (BOOCH; RUMBAUGH; JACOBSON, 2006, p. 32), deste trecho pode-se notar uma grande valorização da modelagem, pela possibilidade de, por meio dela, esboçar como é ou como se deseja que o sistema seja, orientar o desenvolvimento, documentar as decisões tomadas, entre outros. Nesse contexto, em 1997, surgiu a UML, com o intuito de unificar as modelagens presentes na época e se tornou, desta forma, uma linguagem universal no quesito de modelagem de *software*, sendo a linguagem mais utilizada em modelagem atualmente (SOMMERVILLE, 2011).

Desde então a UML passou a ser incluída no ensino de engenharia de *software*, presente na vasta maioria da literatura que aborda o tema e também nas grades acadêmicas dos cursos que possuem relação com o desenvolvimento de *software*, como é o caso dos cursos de Bacharelado em Ciências da Computação (CC), Engenharia de *Software* (ES), Engenharia de Computação (EC), Sistema de Informação (SI), Tecnólogos como: Análise e Desenvolvimento de *Software* (ADS), entre outros. Porém, pela grande quantidade de diagramas existentes na linguagem em questão, é prática comum que haja um maior enfoque, dentro do ambiente acadêmico, em alguns diagramas que se imagina serem os mais condizentes com o ambiente profissional, sendo os diagramas de casos de uso, de classe, de sequência e de atividade.

2.1 *Unified Modeling Language* (UML)

A UML é definida da seguinte maneira no manual oficial da linguagem: “A UML, Linguagem Unificada de Modelagem, é uma linguagem gráfica para visualização, especificação, construção e documentação de artefatos de sistemas complexos de software. A UML proporciona uma forma-padrão para a preparação de planos de arquitetura de projetos de sistemas, incluindo aspectos conceituais, tais como processos de negócios e funções do sistema, além de itens concretos como as classes escritas em determinada linguagem de programação, esquemas de bancos de dados e componentes de software reutilizáveis.” (BOOCH; RUMBAUGH; JACOBSON, 2006, p. 7).

2.2 Diagrama de Caso de Uso (DCU)

Os diagramas de casos de uso têm como objetivo a visualização, especificação e documentação do comportamento de um dos elementos do sistema em uma perspectiva externa, de usuário, para tal mostra conjuntos de casos, atores e os relacionamentos entre eles (BOOCH; RUMBAUGH; JACOBSON, 2006). Como exemplo de um diagrama de caso de uso, a Figura 1 mostra dois atores sendo “usuário” e “professor” com uma generalização.

O DCU apresenta a funcionalidade na visão do usuário.

Todas as Figuras foram criadas no aplicativo *Drawio* por ser gratuito e já conter a modelagem da UML 2.5 e todas de autoria do autor.

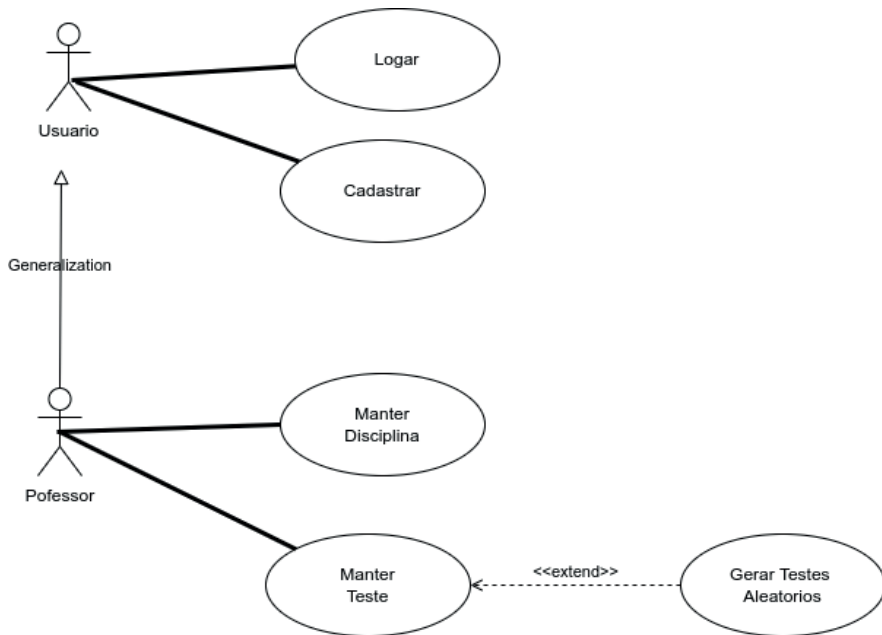


Figura 1: Diagrama de Caso de Uso com dois Atores com uma Generalização (Fonte: Autor).

Pode-se observar na Figura 1 dois atores sendo o “usuário” e o “professor” sendo que o professor está generalizando (herdando) as funcionalidades do usuário e onde o professor tem 3 (três) casos de uso sendo um “*extend*” sendo uma regra de negócio do sistema. Todos os casos de uso devem iniciar-se com verbo no infinitivo, as associações devem ser abaixo ou próximo do braço do ator e jamais puxar casos de uso da cabeça ou do pé do ator.

No DCU tem a notação “sistema”, o sistema é usado para definir o objetivo do caso de uso e é desenhado como um retângulo, este fica a critério do analista de requisitos ao modelar, porém é obrigatório quando for apresentado a interação do ator com o sistema exemplo prático é o usuário interagindo com o caixa eletrônico, neste caso deve-se colocar o retângulo para fazer apresentar o limite do sistema.

2.3 Diagrama de Classe (DC)

Os diagramas de classes têm como objetivo a visualização estática das classes do sistema, com seus atributos e métodos e a relação entre as classes existentes (GUEDES, 2018). A Figura 2 apresenta o diagrama de classe da Figura 1.

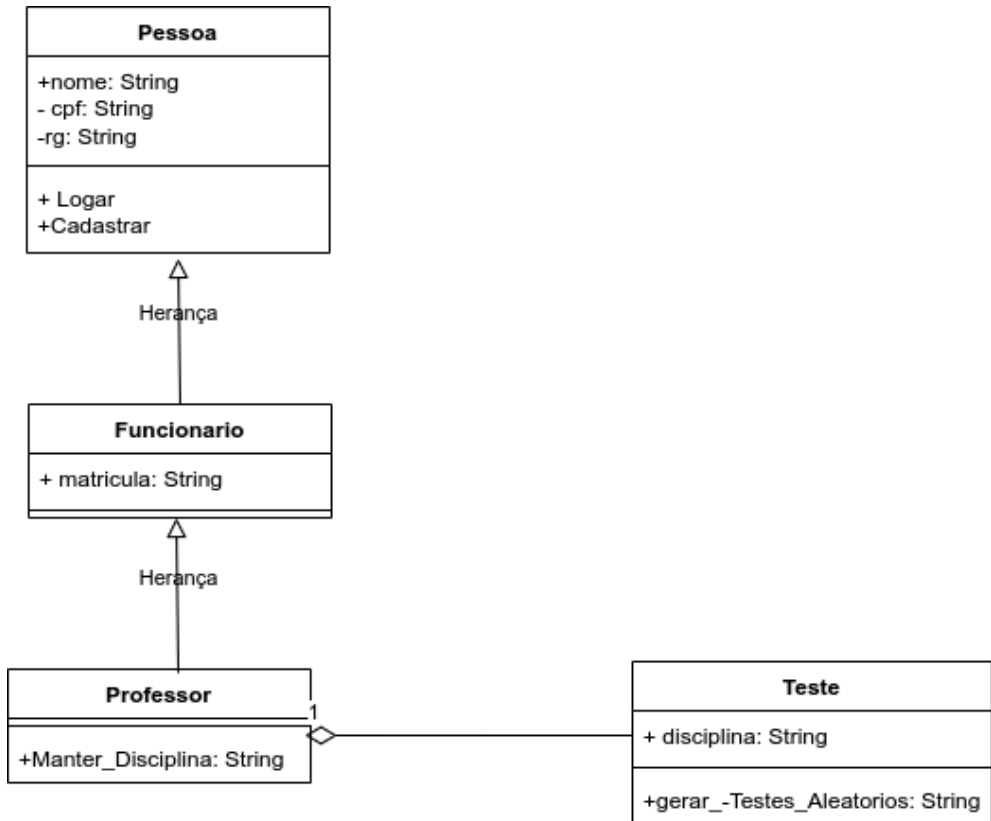


Figura 2: Diagrama de Classe da Figura 1 (Fonte: Autor).

Pode-se observar na Figura 2 que o usuário agora virou “classe pessoa” com seus atributos, pois todo usuário na vida real é uma pessoa, e no DC é algo concreto real que existe na vida real, se temos como colocar atributo então é classe, lembrando toda classe tem atributo, porém nem todas as classes precisam ter método. Tem-se também a “classe funcionário” com o atributo “matricula” o qual herda da classe pessoa e em seguida a “classe professor” que herda da classe funcionário o qual tem o método “manter_Disciplina” e por fim a “classe teste” com o atributo “disciplina” e o método “gerar_Testes_Aleatórios” sendo uma agregação da classe professor.

No DC os nomes devem sempre iniciar com letra minúscula e é recomendado colocar o que ela é como: String, Real, Inteiro, Caracter, Float, Boleana, entre outros. Lembrando que o DC pode gerar código do programa.

2.4 Diagrama de Atividade (DA)

Os diagramas de atividades têm por objetivo exibir uma sequência de condições que coordenam a forma com que o sistema se comporta. Este diagrama é o que possui maior

proximidade com o nível de algoritmo em si (GUEDES, 2018). Exemplificando este tipo de diagrama, a Figura 3 mostra as atividades das etapas.

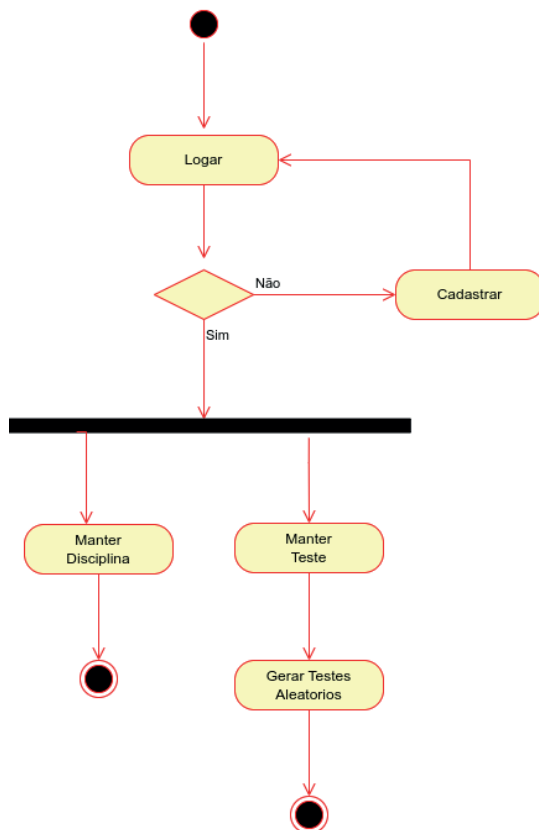


Figura 3: Diagrama de Atividade da Figura 1 (Fonte: Autor).

Pode-se observar na Figura 3 que o usuário inicia logando tendo uma condição “sim” e “não” o qual se ele tiver cadastrado o mesmo acessa o sistema caso contrário deve realizar o cadastro para efetuar o acesso, acessando temos um *fork* onde o professor pode escolher umas das duas opções e caso realize ele finaliza.

2.5 Diagrama de Sequência (DS)

Os diagramas de sequências têm por objetivo descrever, no tempo, a ordem das mensagens. Trazendo no topo os objetos que fazem parte da interação e as mensagens, vão sendo representadas em ordem crescente de tempo, de cima para baixo. Esses diagramas se diferem pela presença da “linha de vida”, que indica o intervalo de tempo que um objeto existiu (BOOCH; RUMBAUGH; JACOBSON, 2006). A Figura 4 apresenta do diagrama de sequência da Figura 1.

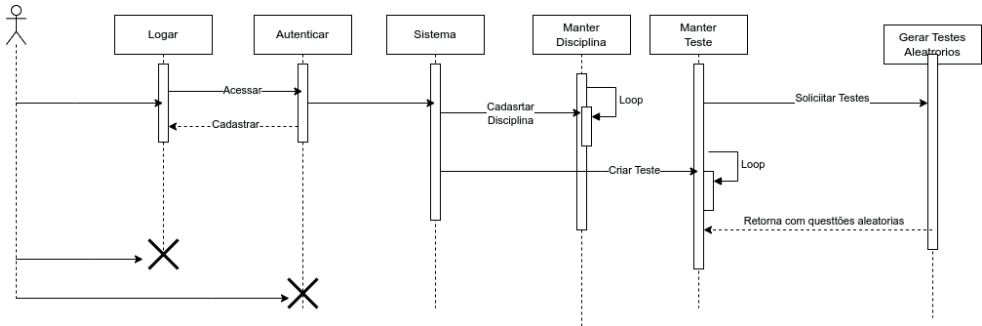


Figura 4: Diagrama de Sequência da Figura 1 (Fonte: Autor).

Pode-se observar na Figura 4 que o usuário ao logar será autenticado, caso não seja retorna uma mensagem para cadastrar, tendo cadastro o usuário terá acesso ao sistema e poderá realizar as suas funções conforme é mostrando na Figura acima.

3 I METODOLOGIA

Foi realizada uma pesquisa bibliográfica nas bases de dados sendo; CAPES, Biblioteca Digital Brasileira de Teses e Dissertações (BDTD), *Google Acadêmico*, *SciELO*, *OpenLib* (SOL) a biblioteca digital mantida pela Sociedade Brasileira de Computação (SBC), o intuito é demonstrar a importância da UML na modelagem de *software*, na busca de sobre o tema, bem como publicações e artigos científicos em torno do assunto como pode ser visto na Tabela 1 trabalhos voltados a UML.

Autor	Trabalho
Bento, 2020	UML: Um estudo sobre o uso em empresas de desenvolvimento de software em São Carlos - SP e região
Margarida, 2018	Avaliação da aplicabilidade da UML como uma ADL de software
Ferreira, 2009	Modelagem de domínios temporais de planejamento com UML.P
Bertran, 2009	Avaliação da qualidade de software com base em modelos uml
Vieira, 2003	Uso da UML na Especificação do Sistema de Informação da Área de Infra-Estrutura da UFRGS
Costa, 2001	A aplicação da linguagem de modelagem unificada (UML) para o suporte ao projeto de sistemas computacionais dentro de um modelo de referência

Tabela 1: Revisão da Literatura de Trabalhos Voltados a Modelagem com UML.

CONCLUSÃO

Com este trabalho foi possível realizar a revisão bibliográfica sobre diversos conceitos na modelagem de *software* utilizando a UML e revendo alguns conceitos importantes como a fundamentação, e pode-se ver que a UML continua sendo utilizado para modelar *software* antes de serem codificados para amenizar os erros no desenvolvimento.

REFERENCIAS

BENTO, Luiz Henrique Ten Caten UML: Um estudo sobre o uso em empresas de desenvolvimento de software em São Carlos - SP e região / Luiz Henrique Ten Caten Bento. – São Carlos – SP, 2020. 60 p.; 29,7 cm.

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. UML: guia do usuário. Elsevier, 2006. ISBN 9788535217841. Disponível em: <<https://books.google.com.br/books?id=ddWqxcDKGF8C>> Acessado em 12/04/2023.

GUEDES, G. T. UML 2-Uma abordagem prática. [S.l.]: Novatec Editora, 2018.

KITCHENHAM, B. A.; PFLEEGER, S. L. Personal opinion surveys. In: Guide to Advanced Empirical Software Engineering. London: Springer London, 2008. p. 63–92. ISBN 978-1-84800-044-5. Disponível em: <https://doi.org/10.1007/978-1-84800-044-5_3> Acessado 11/03/2023.

MARGARIDA Marina de Oliveira: **AVALIAÇÃO DA APLICABILIDADE DA UML COMO UMA ADL DE SOFTWARE**; UNIVERSIDADE FEDERAL DE OURO PRETO, INSTITUTO DE CIENCIAS EXATAS E APLICADAS DEPARTAMENTO DE COMPUTAÇÃO E SISTEMAS, 2018.

SANCHES, Henderson Matsuura: ENGENHARIA DE REQUISITOS DESENVOLVIMENTO DE SOFTWARE, Engenharia elétrica e de computação: docência, pesquisa e inovação tecnológica / Organizadora Lilian Coelho de Freitas. – Ponta Grossa - PR: Atena, 2023.

SOMMERVILLE, I. Engenharia de software. PEARSON BRASIL, 2011. ISBN

9788579361081. Disponível em: <<https://books.google.com.br/books?id=H4u5ygAACAAJ>> Acessado em 12/04/2023.

OMG (Object Management Group,) Disponível em <<https://www.omg.org/spec/UML/>> Acessado em 12/05/2023.

VIEIRA, Ricardo Uso da UML na Especificação do Sistema de Informação da Área de Infra-Estrutura da UFRGS / Ricardo Vieira. _ Porto Alegre: Curso de Especialização em WEB e Sistemas de Informação, 2003.