

# O ENSINO E APRENDIZAGEM DE UML: UM MAPEAMENTO SISTEMÁTICO

## UML TEACHING AND LEARNING: A SYSTEMATIC REVIEW

Patrícia Maria Barbosa Ferreira\*

Andressa Bezerra Ferreira\*\*

### RESUMO

Modelagem de *software* é considerado um dos temas mais importantes no ensino da engenharia de *software*. Atualmente, a *Unified Modeling Language* (UML) é a linguagem de modelagem de *software* mais difundida e utilizada na indústria da engenharia de *software*. Embora a UML seja constantemente aprimorada e estudada, muitos trabalhos mostram que há dificuldade no ensino-aprendizagem do tema, devido à complexidade de seus conceitos e às dificuldades cognitivas dos discentes com a abstração. Ainda, os estudantes enfrentam dificuldades para compreender a semântica e a sintaxe dos modelos, bem como estruturar as informações nesses modelos. Além disso, há dificuldades para os docentes em encontrar diferentes estratégias pedagógicas, com o objetivo de ensinar a modelagem. Diante desse cenário, foi conduzido um Mapeamento Sistemático da Literatura para coletar os estudos primários existentes de janeiro de 2018 a março de 2023 que apresentam as dificuldades percebidas pelos estudantes ao criar diagramas UML e as estratégias de ensino utilizadas pelos professores para apoiar o processo de ensino e aprendizagem de modelagem de *software* com a UML. Selecionamos duas bibliotecas digitais para realizar essa busca, são elas: *Google scholar* e *ACM Digital Library*. Foram considerados 1.500 estudos para a condução do mapeamento sistemático, após a leitura completa dos artigos, utilizamos apenas 44 estudos para a extração de dados para este mapeamento. Ao analisarmos os 44 estudos, identificamos e classificamos os mesmos em três categorias, são elas: dificuldades, ferramentas e metodologias de ensino. Os resultados mostraram que cerca de 48% dos estudos apresentam metodologias que podem auxiliar no processo de ensino e aprendizagem de modelagem de *software* utilizando diagramas UML, em torno de 32% dos estudos relatam as dificuldades enfrentadas pelos estudantes ao criar diagramas UML e 20% dos estudos apresentam ferramentas que apoiam o processo de ensino-aprendizagem de modelagem com a UML.

**Palavras-chave:** Mapeamento Sistemático da Literatura. Ensino e Aprendizagem. *Unified Modeling Language*. Modelagem de *Software*.

---

\* Graduada em Bacharelado em Ciência da Computação no Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE), Aracati, Ceará, Brasil. E-mail: patricia.maria.ferreira06@aluno.ifce.edu.br

\*\* Mestre em Ciência da Computação, Docente do Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE), Aracati, Ceará, Brasil. E-mail: andressa.ferreira@ifce.edu.br

## ABSTRACT

Software modeling is considered one of the most important subjects in software engineering education. Currently, the Unified Modeling Language (UML) is the most widespread and used software modeling language in the software engineering industry. Although the UML is constantly improved and studied, many works show that there are difficulties in teaching and learning the subject, due to the complexity of its concepts and the students' cognitive difficulties with abstraction. Still, students face difficulties in understanding the semantics and syntax of the models, as well as structuring the information in these models. In addition, there are difficulties for teachers to find different pedagogical strategies, with the objective of teaching modeling. Given this scenario, a Systematic Literature Review was conducted to collect existing primary studies from January 2018 to March 2023 that present the difficulties perceived by students when creating UML diagrams and the teaching strategies used by teachers to support the teaching process and learning software modeling with the UML. We selected two digital libraries to carry out this search, they are: Google Scholar and ACM Digital Library. 1,500 studies were considered for conducting the systematic review, after reading the articles in full, we used only 44 studies to extract data for this review. When analyzing the 44 studies, we identified and classified them into three categories, namely: difficulties, tools and teaching methodologies. The results showed that about 48% of the studies present methodologies that can help in the teaching and learning process of software modeling using UML diagrams, around 32% of the studies report the difficulties faced by students when creating UML diagrams and 20% of studies present tools that support the teaching-learning process of modeling with the UML.

**Keywords:** Systematic Review of Literature. Teaching and learning. Unified modeling language. Software Modeling.

## 1 INTRODUÇÃO

Linguagem de Modelagem Unificada (UML) é uma linguagem gráfica de modelagem orientada a objetos para visualização, especificação, construção e documentação para desenvolver sistemas computacionais de propósito geral que pode ser aplicada a todos os *softwares* e que permite demonstrá-lo de forma padronizada (BOOCH; RUMBAUGH; JACOBSON, 2006). A UML baseia-se em diagramas de modelagem visual, e se propõe ser uma boa modelagem de sistemas orientada a objeto, pois é uma modelagem unificada e os envolvidos no desenvolvimento do sistema tem facilidade de interpretação no projeto.

A modelagem de *software* com modelos da UML é considerada um dos temas mais importantes no ensino da engenharia de *software* (ES), sobretudo por se tratar de uma disciplina obrigatória nos cursos de computação e representar um padrão de uso internacionalmente adotado pela indústria da ES (GUEDES, 2018). Muitos estudos mostram que o ensino-aprendizagem do

tema é um processo difícil devido à complexidade de seus conceitos (SILVA et al., 2020). Além disso, em geral, muitos estudantes enfrentam dificuldades em entender a sintaxe e a semântica dos modelos (MA, 2017), estruturar as informações nos modelos e aplicar os relacionamentos generalização/especialização (BOLLOJU; LEUNG, 2006). Essas dificuldades podem estar diretamente relacionadas como a disciplina é ensinada (AL-TAHAT, 2014). Além disso, docentes enfrentam dificuldades em encontrar diferentes estratégias pedagógicas com o objetivo de ensinar a modelagem (SILVA et al., 2020).

A UML oferece um catálogo de diferentes requisitos e artefatos de *design* que podem ser usados para modelar os requisitos e projeto de um sistema sob diferentes pontos de vista e perspectivas (EL-ATTAR et al., 2015). Portanto, aprender UML se torna uma necessidade para profissionais e estudantes de Engenharia de *Software* (SIAU; LOO, 2006).

Apesar de ser um padrão para a modelagem orientada a objetos, a UML tem sido criticada por possuir uma série de dificuldades de uso, como inconsistências semânticas, inadequação das notações e ambiguidades dos diagramas e dos constructos (SIAU; LOO, 2006). Essas questões são os principais obstáculos ao utilizar os diagramas da UML. Por este motivo, foi decidido realizar um levantamento das principais dificuldades relacionadas ao uso destes diagramas. Por isso, optou-se por realizar um Mapeamento Sistemático da Literatura (MSL).

Um Mapeamento Sistemático da Literatura (MSL) é um método de categorizar e resumir as informações existentes sobre uma questão de pesquisa de maneira imparcial. Este tipo de estudo ajuda a identificar lacunas na pesquisa atual, a fim de sugerir áreas para futuras investigações (KITCHENHAM; CHARTERS, 2007).

Ao contrário das revisões informais da literatura, onde o pesquisador não segue um processo definido para sua condução, um Mapeamento Sistemático é executado de maneira formal obedecendo a um protocolo pré-estabelecido. Em comparação com revisões informais de literatura, os mapeamentos sistemáticos requerem maior rigor na sua realização. Em compensação, seu resultado tende a ser mais confiável, visto que eles fazem uso de uma metodologia rigorosa e passível de auditoria e repetição (KITCHENHAM; CHARTERS, 2007).

O objetivo desta pesquisa é identificar, por meio de um mapeamento sistemático da literatura, as dificuldades enfrentadas por estudantes no aprendizado de diagramas UML. E também identificar as ferramentas e metodologias que os professores utilizam no ensino para minimizar essas dificuldades.

Examinamos mais de 1.500 artigos de pesquisa coletados em duas bibliotecas digitais e aplicamos um rigoroso processo de seleção, resultando em 44 artigos selecionados. Em seguida, extraímos as dificuldades evidenciadas nos artigos, bem como as ferramentas e metodologias que auxiliam no ensino de diagramas UML.

Este trabalho está organizado da seguinte forma: a Seção 2 apresenta o Referencial Teórico, na Seção 3 temos o Objetivo deste trabalho, na Seção 4 são apresentados os Trabalhos Relacionados; já na seção 5, apresenta-se o Protocolo do Mapeamento Sistemático, na seção 6 temos os Resultados do Mapeamento e por fim, na seção 7, consta a Conclusão.

## 2 REFERENCIAL TEÓRICO

Nesta seção, comentamos, brevemente, alguns aspectos teóricos específicos que condicionaram o mapeamento sistemático da literatura e foram levados em consideração na análise dos trabalhos encontrados.

### 2.1 Linguagem de Modelagem Unificada (UML)

A importância da modelagem, segundo (BOOCH; RUMBAUGH; JACOBSON, 2006), está diretamente relacionada ao sucesso da empresa de software e à garantia na entrega de um produto final de qualidade aos usuários. Para auxiliar a modelagem dos sistemas utiliza-se a Linguagem de Modelagem Unificada (do inglês, *Unified Modeling Language* - UML).

A UML foi criada por Grady Booch, Ivar Jacobson e James Rumbaugh em 1994 para especificar, documentar, construir e visualizar a arquitetura do *software*. Em 1997 foi adotada pela OMG, instituição que regulariza e aprova padrões abertos para aplicações orientadas a objetos, como linguagem padrão. A UML surgiu com a unificação das notações criadas por esses três autores que desenvolveram metodologias próprias para modelagem de sistemas Orientado a Objetos (OO).

A modelagem além de auxiliar na idealização do *software* por meio de notações gráficas, também colabora na comunicação daquilo que o *software* precisa fazer para resolver para os usuários. Essas notações são chamadas de diagramas e permitem compreender desde a estrutura até o comportamento do *software* a ser desenvolvido (AZEVEDO; SOUZA; SOUZA, 2021).

É importante destacar que os diagramas são divididos em: (i) estruturais e (ii) comportamentais. Os diagramas estruturais contribuem com a visualização lógica e estática do *software*. Dentre os diagramas estruturais é importante destacar: diagrama de classe, componentes, implantação, objetos, pacote, estrutura composta e de perfil. Por outro lado, os diagramas comportamentais viabilizam a compreensão das funcionalidades a serem desenvolvidas. Os diagramas comportamentais estão classificados em diagramas de caso de uso, atividades, sequência, máquina de estados, comunicação, visão geral da interação e de tempo (GUEDES, 2018).

### 2.2 Mapeamento Sistemático da Literatura (MSL)

O Mapeamento sistemático da literatura é um estudo que auxilia a investigação de pesquisas relevantes sobre um tópico contribuindo na estruturação de uma área de pesquisa. Assim, um MSL é um estudo secundário<sup>1</sup> que tem como objetivo identificar e classificar a pesquisa relacionada a um tópico amplo de pesquisa (KITCHENHAM; CHARTERS, 2007).

Os MSLs são realizados com metodologias formalizadas em um protocolo contendo procedimentos com aspectos de confiabilidade, rigor, auditoria e imparcialidade para a identificação, agrupamento e análise de estudos. Ainda, indicam os passos para a divulgação das decisões da

---

<sup>1</sup> Um estudo que revisa todos os estudos primários relacionados a uma questão de pesquisa específica, contribuindo para sintetizar evidências sobre a questão de pesquisa.

pesquisa e os resultados obtidos, possibilitando a reexecução e reprodução do estudo. O processo de MSL envolve três fases principais: Planejamento, Condução do Mapeamento e Publicação dos Resultados (KITCHENHAM; CHARTERS, 2007). Essas fases, bem como suas atividades, são conduzidas de modo iterativo. A Figura 1 mostra as fases e atividades do processo de MSL, as quais são detalhadas a seguir.

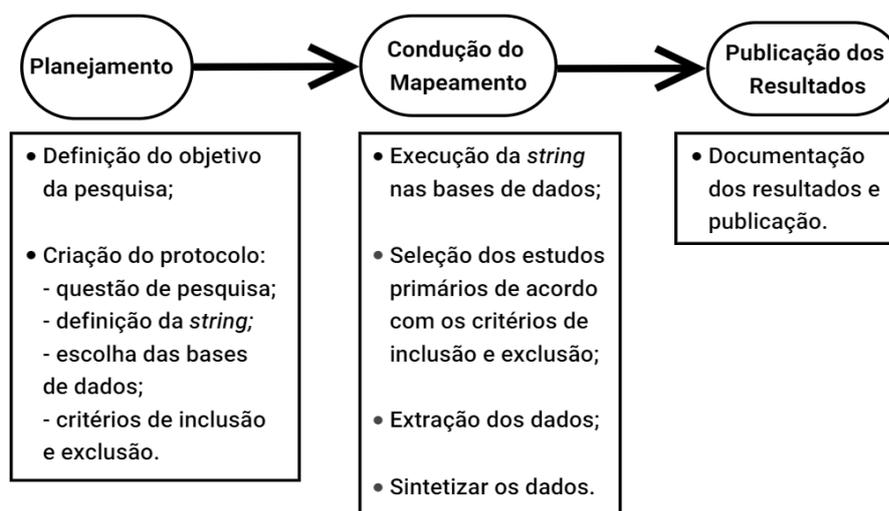


Figura 1 – Fases e Atividades do Processo de MSL.

A fase de planejamento tem como objetivo identificar a real necessidade, ou seja, a motivação para a execução de um MSL. Após identificar a necessidade da realização de um mapeamento sistemático, é definido o protocolo da revisão que é o elemento essencial para executar a revisão. O protocolo especifica as questões de pesquisa, definição da *string* de busca, escolha das bases de dados e os critérios para inclusão e exclusão dos estudos.

Após a validação do protocolo, inicia-se a fase de condução do MSL. Durante essa fase, os estudos primários são identificados. Uma vez identificados, os estudos precisam ser selecionados por meio da aplicação de critérios de seleção (critérios de inclusão e de exclusão). Os critérios de seleção devem especificar as principais características e/ou conteúdos que os estudos devem ter para serem incluídos ou excluídos. Após a atividade de seleção, os dados contidos nos estudos incluídos devem ser extraídos e sintetizados. Formulários de extração de dados são utilizados para coletar os dados que sejam necessários para responder as questões de pesquisa da revisão e para facilitar posteriormente as análises e síntese dos resultados.

A última fase do processo de MSL é a escrita e divulgação dos resultados. Esses resultados podem ser divulgados por meio de relatórios técnicos, artigos de periódicos ou conferências, em capítulos de livros ou como uma seção de um trabalho de conclusão de curso.

Vale destacar que todas as informações produzidas nas etapas de planejamento e condução devem ser adequadamente registradas para permitir a posterior publicação dos resultados do mapeamento. Para tal, é imprescindível usar ferramentas como planilhas eletrônicas e ferramentas de software para o gerenciamento de referências para registrar tais informações. Uma opção

ainda mais interessante é usar uma ferramenta de *software* de apoio à realização de MSL, tal como a ferramenta *Start*.

### 2.3 Ferramenta *StArt*

A ferramenta *StArt*<sup>2</sup> foi desenvolvida pelo Laboratório de Pesquisa em Engenharia de Software (LaPES) da Universidade Federal de São Carlos (UFSCar) para dar suporte ao processo de Revisão Sistemática da Literatura. A ferramenta implementa o modelo proposto por (KITCHENHAM; CHARTERS, 2007), que é dividido em três etapas: planejamento, condução e publicação dos resultados, como pode ser visto na Figura 1. O usuário da ferramenta define o protocolo do mapeamento, informando palavras-chave, formato dos dados a serem coletados e os critérios para seleção dos trabalhos acadêmicos e fontes de pesquisa. Em seguida, o usuário informa os termos para que seja realizada a busca nas bases de dados, e anexa os arquivos resultantes da exportação dos resultados. Em seguida, é possível executar as tarefas de seleção dos artigos e de extração dos seus dados relevantes. Por fim, é possível registrar, na própria ferramenta, um resumo do mapeamento em formato de texto livre (HERNANDES et al., 2010).

Além das funcionalidades mencionadas, a *StArt* permite que o usuário gere relatórios que auxiliam na extração de informações armazenadas na ferramenta durante a execução do processo de mapeamento sistemático. Exemplos desses relatórios são: geração de todas as informações pertinentes ao mapeamento sistemático, correspondendo ao empacotamento do mapeamento sistemático; resumos de todos os estudos importados por meio do arquivo *BibTex* para que o usuário possa lê-los no formato impresso; conjunto das informações extraídas pelo usuário de cada um dos estudos aceitos na etapa de Extração de Informações; arquivo no formato *BibTex* contendo todos os estudos pertencentes ao mapeamento sistemático possibilitando que eles sejam importados em um gerenciador de referências.

Essa ferramenta foi escolhida, pois permitiu executar o mapeamento sistemático em conformidade com o processo definido na literatura, contribuindo para a realização do processo corretamente, foi de grande valia utilizá-la, pois a mesma detectou trabalhos duplicados na seleção, ajuda na organização dos estudos selecionados e, é bastante simples e intuitiva de ser utilizada. A ferramenta foi concebida para uso em *desktop* com sistema *Windows*, é disponibilizada de forma gratuita, através de uma licença que não explicita limitações de uso.

## 3 OBJETIVO DA PESQUISA

Este trabalho tem por objetivo geral realizar um Mapeamento Sistemático da Literatura (MSL) capaz de sumarizar as dificuldades inerentes ao processo de ensino-aprendizagem na modelagem de sistemas com UML, mediante análise de publicações científicas com o propósito de externar essas dificuldades.

---

<sup>2</sup> [http://lapes.dc.ufscar.br/tools/start\\_tool](http://lapes.dc.ufscar.br/tools/start_tool)

### 3.1 Objetivos Específicos

Para atingir o objetivo principal deste trabalho, são definidos os seguintes objetivos específicos:

- Identificar as dificuldades enfrentadas pelos estudantes enquanto aprendem a modelagem de sistemas com os diagramas da UML.
- Identificar as estratégias de ensino, metodologias e ferramentas relatadas na literatura, que são utilizadas pelos professores, para apoiar o ensino e aprendizagem de modelagem de *software* com os diagramas da UML

## 4 TRABALHOS RELACIONADOS

Buscando trabalhos cujo enfoque está relacionado a esta pesquisa, ou seja, um mapeamento sistemático da literatura no contexto de ensino e aprendizagem de UML, esta seção tem por finalidade apresentá-los.

(SHEN; TAN; SIAU, 2018) Em um esforço para entender a UML no contexto da educação em Engenharia de *Software*, realizou um estudo empírico sobre a aprendizagem da UML. Especificamente, utilizou uma técnica cognitiva (mapeamento conceitual<sup>3</sup>) para explorar os desafios dos alunos em aprender elementos de notação UML. Em seguida, analisaram os desafios de aprendizagem percebidos pelo aluno através da lente da representação esquemática e do raciocínio, identificaram estratégias instrucionais para facilitar os processos perceptivos e conceituais envolvidos nos desafios de aprendizagem da UML e ofereceram *insights* sobre como aprimorar o ensino de UML. Os participantes deste estudo foram alunos do último ano e de pós-graduação que concluíram um curso de sistema de *design* orientado a objetos em uma grande universidade pública dos Estados Unidos, ao todo foram 32 alunos participantes do estudo.

(REUTER et al., 2020) Realizaram uma revisão sistemática da literatura de outubro a dezembro de 2018, para obter uma visão geral do trabalho existente sobre os problemas dos alunos na modelagem de *software* com a UML. Escolheram os seguintes mecanismos de busca: *ACM*, *ERIC*, *IEEE Explore*, *Sagepub*, *Scopus*, *Wiley*, *Scencedirect*, *Springer*, e *Web of Knowledge*. Para as consultas utilizaram o termo de busca: [(*difficulties OR problems*) AND (*teaching OR learning*) AND (*modeling software OR modelling software OR software design OR software modeling OR software architecture OR software modelling*) AND UML]. Ao todo, identificaram 288 artigos. Após o processo de filtragem, identificaram 12 artigos completos como altamente relevantes para o estudo. Focaram no aspecto capaz de identificar dificuldades nos processos reais de modelagem dos alunos.

(SILVA; STEINMACHER; CONTE, 2023) Este estudo visa investigar e classificar as evidências empíricas existentes sobre as dificuldades que os engenheiros de *software* enfrentam

<sup>3</sup> O mapeamento conceitual é uma representação gráfica que facilita a visualização de informações de acordo com os princípios da arquitetura da informação, hierarquizando ideias e conceitos segundo sua ordem de importância (TROCHIM, 1989).

ao aprender a projetar *software* usando diagramas UML e categorizar as tecnologias que estão sendo usadas para auxiliar os engenheiros no processo de modelagem. Método: Realizaram um estudo de mapeamento sistemático para coletar os estudos existentes de 1998 a 2018 que apresentam as dificuldades de criar diagramas UML e as tecnologias propostas para minimizar tais dificuldades, selecionaram quatro bibliotecas digitais para realizar essa busca, são elas: *Scopus*, *Engineering Village*, *Web of Science* e *ACM Digital Library*. Utilizaram uma abordagem de codificação inspirada nos procedimentos de codificação aberta e axial da teoria fundamentada para categorizar as dificuldades relatadas pelos estudos selecionados. Resultados: Ao analisarem 181 estudos, identificaram evidências empíricas de dificuldades enfrentadas por engenheiros de *software* ao criar diagramas UML. Também identificaram e classificaram as dificuldades de acordo com o diagrama. Os resultados mostram que cerca de 68% dos estudos apresentam tecnologias que podem auxiliar os engenheiros de *software* na modelagem utilizando diagramas UML. Conclusão: Os resultados são úteis para pesquisadores, instrutores e profissionais de *software* que desejam investigar ou implementar ferramentas para dar suporte a engenheiros durante o projeto de modelo UML.

O Quadro 1, apresenta um comparativo entre os trabalhos relacionados, além de um comparativo com este trabalho.

Quadro 1 - Comparativo dos Trabalhos Relacionados.

| Trabalhos          | Período de busca           | Técnica                              | Qtd. de trabalhos relevantes | Objetivo da pesquisa  | Quantidade de bases de dados | Ferramenta de apoio ao processo |
|--------------------|----------------------------|--------------------------------------|------------------------------|---|------------------------------|---------------------------------|
| Shen et. al 2018   | Não contém                 | Mapeamento conceitual                | Não contém                   | Desafios dos alunos em aprender elementos de notação UML.   | Não contém                   | Não contém                      |
| Reuter et. al 2020 | outubro a dezembro de 2018 | Revisão sistemática da literatura    | 12                           | Dificuldades nos processos reais de modelagem dos alunos.   | 09                           | Não contém                      |
| Silva et. al 2023  | 1998 a 2018                | Mapeamento sistemático da literatura | 181                          | Dificuldades que os engenheiros de <i>software</i> enfrentam ao aprender a projetar <i>software</i> usando diagramas UML. | 04                           | Não contém                      |
| Este trabalho      | 2018 a março de 2023       | Mapeamento sistemático da literatura | 44                           | Dificuldades que podem interferir no processo de ensino e aprendizagem da modelagem de <i>software</i> com a UML.         | 02                           | Start                           |

Fonte: Elaborada pelos autores.

## 5 PROTOCOLO DO MAPEAMENTO SISTEMÁTICO DA LITERATURA

Mapeamento sistemático (em inglês, *systematic review*, SR, também conhecido como revisão de escopo) é uma metodologia que tem como objetivo identificar e categorizar as pesquisas disponíveis sobre uma área de interesse, sendo considerado um estudo secundário (KITCHENHAM; CHARTERS, 2007). É de grande contribuição, pois prover uma visão geral e resumida sobre um tópico, se torna importante à medida que a pesquisa na área amadurece e

surge uma grande quantidade de estudos e resultados disponíveis, sendo a abordagem temática um método interessante de análise (PETERSEN; VAKKALANKA; KUZNIARZ, 2015).

Para executar um mapeamento sistemático ou uma revisão sistemática da literatura, o primeiro passo a ser dado é definir um protocolo de pesquisa que deve ser claramente relatado antes da execução da revisão da literatura, (DERMEVAL; COELHO; BITTENCOURT, 2020).

(KITCHENHAM; CHARTERS, 2007) especificam os elementos que são itens obrigatórios do protocolo de um mapeamento sistemático de literatura, são eles: as questões de pesquisa que devem ser respondidas, as estratégias que devem ser adotadas para realizar a busca e a seleção dos estudos, bases de pesquisa, critérios de inclusão e exclusão, procedimentos para seleção dos estudos primários, extração dos dados, análise dos dados extraídos, sumarização dos resultados encontrados e, por fim, a escrita do relatório reportando todo o processo de execução e os resultados encontrados no mapeamento.

Para esta pesquisa, optou-se pela condução do Mapeamento Sistemático da Literatura (MSL) para coletar informações sobre ensino e aprendizagem de diagramas UML, considerando as diretrizes fornecidas por (KITCHENHAM; CHARTERS, 2007), que sugere três fases: (i) Planejamento; (ii) Condução; (iii) Análise, ver Figura 1.

Nas próximas seções, está descrito detalhadamente o protocolo para a condução do mapeamento sistemático da literatura deste trabalho.

## 5.1 Questão De Pesquisa

A questão de pesquisa principal do mapeamento sistemático foi:

- QP: Quais as dificuldades constatadas na literatura sobre o processo de ensino e aprendizagem de modelagem de *software* com a UML?

## 5.2 Definição Da *String* De Busca

O mecanismo de busca disponível na maioria das bibliotecas digitais é baseado em expressões de busca textual. Assim, a definição da *string* de consulta é essencial para a eficácia da etapa de busca e para o mapeamento sistemático como um todo, (STEINMACHER et al., 2015). A *string* de consulta utilizada na busca automática foi construída com base em três termos extraídos da questão geral de pesquisa: (1) dificuldades, (2) ensino e aprendizagem; e (3) UML.

Para a construção da *string* de busca foram seguidos quatro passos: (i) identificação dos termos (*keywords*) relevantes à QP; (ii) identificação dos sinônimos desses termos em inglês; (iii) utilização do operador booleano “OR” entre os sinônimos; e (iv) utilização do operador booleano “AND” para concatenar os principais termos.

A busca foi realizada por trabalhos que tratassem do tema no período dos últimos 5 anos, a partir de janeiro de 2018 a março de 2023.

A *string* de consulta final usada neste mapeamento é mostrada a seguir. Os operadores lógicos AND e OR são aplicados para conectar os termos principais e os alternativos ou relacio-

nados (BRERETON et al., 2007), respectivamente. Assim, o argumento de busca utilizado foi, veja Figura 2.

```
((“dificuldade” OR “desafio” OR “difficulty” OR “challenge”)
AND (“ensinar” OR “aprender” OR “aprendizagem” OR “ensino”
OR “learning” OR “teach” OR “teaching” OR “learn”)
AND (“unified modeling language” OR “uml” OR “linguagem de
modelagem unificada” OR “modelagem de software” OR
“ferramenta de modelagem” OR “software modeling” OR
“modeling tool”))
```

Figura 2 – *String* de busca utilizada no mapeamento sistemático.

Em seguida foram definidas as fontes de busca, nas quais a *string* construída foi aplicada para execução da busca automática. Para tal, baseou-se nas diretrizes fornecidas por (BRERETON et al., 2007) e na mentoria da professora orientadora.

### 5.3 Bases De Dados De Pesquisa

As bases de dados foram escolhidas devido à disponibilidade de acesso e que, na maioria dos casos, as mesmas dispusessem o texto completo para leitura. Por permitirem o uso de expressões lógicas para as pesquisas ou mecanismo similar para localizar as publicações através do título e resumo (*abstract*) das publicações, as bases escolhidas foram as seguintes, apresentadas no Quadro 2.

Quadro 2 - Bases de dados utilizadas.

| Fonte                      | Link  |
|----------------------------|---|
| Google Scholar (acadêmico) | <a href="https://scholar.google.com.br">https://scholar.google.com.br</a> |
| ACM Digital Library        | <a href="https://dl.acm.org/">https://dl.acm.org/</a>                     |

Elegemos a base *Google Scholar* (acadêmico), porque é uma ferramenta que oferece uma maneira simples e ampla de buscar por pesquisas científicas. Ele indexa trabalhos - artigos, teses, livros, dentre outros - de diversas fontes, como bibliotecas digitais, repositórios acadêmicos, sociedades profissionais, entre outros. Os resultados são retornados por ordem de relevância, considerando aspectos como o texto completo, meio em que foi publicado e última citação. Por estas características, o *Google Scholar* foi escolhido como uma ferramenta de busca deste trabalho.

A *ACM Digital Library* (DL) foi escolhida por ser uma coleção abrangente de artigos de texto completo e registros bibliográficos nos campos de computação e tecnologia da informação. O banco de dados de texto completo inclui a coleção completa das publicações da ACM, incluindo periódicos, anais de conferências, revistas, boletins informativos e títulos multimídia. Por estas características, a *ACM Digital Library* foi selecionada como base de dados para este trabalho.

## 5.4 Critérios Para Seleção Dos Trabalhos

Após a criação e aplicação da *string* nas fontes de busca, o resultado da pesquisa deve passar por um processo de seleção. Este processo é conduzido, inicialmente, por alguns critérios a serem elaborados pelos autores. Eles conferem segurança ao processo de filtragem, que consiste na inclusão ou exclusão, dos estudos primários que realmente possam contribuir para a elucidação da questão de pesquisa.

No contexto deste mapeamento sistemático a seleção seguiu os seguintes critérios de inclusão (CI):

CI-1: Trabalhos discutindo sobre dificuldades/facilidades/lições aprendidas sobre ensinar/aprender/utilizar diagramas da UML;

CI-2: Trabalhos que apresentam alguma metodologia, ferramenta, técnica ou abordagem que viabilize o ensino/aprendizagem de diagramas da UML;

CI-3: Trabalhos que discutam aspectos relacionados ao ensino/aprendizagem de diagramas da UML.

Trabalhos que atenderam a pelo menos um dos seguintes critérios de exclusão (CE) foram excluídos:

CE-1: Trabalhos duplicados (por exemplo, um artigo com um estudo publicado em diferentes lugares ou em diferentes datas). Neste caso, será considerado apenas a versão mais completa e mais recente;

CE-2: Trabalhos publicados antes de 2018;

CE-3: Estudos fora do contexto desta pesquisa;

CE-4: O idioma do trabalho que não seja português ou inglês;

CE-5: O texto completo do trabalho não está disponível para leitura e *download*;

CE-6: Revisões e mapeamentos sistemáticos da literatura.

## 6 RESULTADOS

Nas subseções seguintes serão apresentados os resultados deste Mapeamento Sistemático da Literatura.

### 6.1 Resultado Da Busca

A busca por estudos sobre ensino e aprendizagem de UML foi feita em duas bases de dados *Google Scholar* e *ACM Digital Library*, utilizando a *string* de pesquisa definida na subseção 5.2, Figura 2. A busca foi realizada no dia 13/03/2023 em ambas as bases, sendo aplicados os seguintes filtros: últimos 5 anos (2018 a 2023) e a linguagem dos trabalhos nos idiomas português e inglês.

Na Tabela 1, temos um demonstrativo da quantidade de estudos encontrados e que serão utilizados para este mapeamento sistemático. Somando a quantidade de trabalhos encontrados

em cada máquina de busca temos um total de 9.596 artigos encontrados após a execução da *string* nas máquinas de busca.

Tabela 1 - Quantidade de estudos encontrados.

| Bases de dados      | Quantidade de estudos |              |                       |
|---------------------|-----------------------|--------------|-----------------------|
|                     | Encontrados           | Considerados | Potenciais relevantes |
| Google Scholar      | 7.390                 | 1.000        | 141                   |
| ACM Digital Library | 2.206                 | 500          | 65                    |
| <b>Total</b>        | <b>9.596</b>          | <b>1.500</b> | <b>206</b>            |

Fonte: Elaborada pelos autores.

A base de dados do *Google Scholar* retornou um total de 7.390 trabalhos ordenados por relevância. Iniciamos a análise dos trabalhos retornados um a um, levando em consideração apenas o título e uma pequena descrição do trabalho, ambos apresentam palavras em negrito que são semelhantes a *string* de busca, mediante essas informações foram salvos os trabalhos que consideramos importantes para nossa pesquisa. Após serem analisadas uma determinada quantidade de páginas, percebeu-se que ao decorrer da análise não estavam sendo mais encontrados trabalhos relevantes para esta pesquisa, por este motivo a busca por estudos candidatos limitou-se a 50 páginas, cada página contém 20 trabalhos, mediante este fato um total de 1.000 trabalhos foram considerados, para assim iniciarmos nosso mapeamento e aplicarmos os critérios de inclusão e exclusão.

A biblioteca digital ACM, reportou 2.206 trabalhos como resultado. Apesar do grande volume de trabalhos retornados, foram considerados apenas os primeiros 500 trabalhos, pois essa biblioteca apresenta os trabalhos por relevância mediante ao termo de busca definido, e ao longo da análise dos trabalhos percebeu-se que não estavam sendo mais encontrados trabalhos potencialmente relevantes. Foi configurado para que cada página apresentasse 50 resultados, sendo assim foram analisadas 10 páginas, dando um total de 500 trabalhos salvos para serem objetos de estudo para este mapeamento sistemático.

A partir dos 1.500 artigos considerados é que se deu início ao processo de mapeamento. No primeiro filtro, os artigos foram selecionados com base em seu título e resumo, mediante aplicação dos critérios de inclusão e exclusão descritos na subseção anterior. Ficando assim, 141 artigos do Google Acadêmico e 65 artigos da *ACM Digital Library*, totalizando 206 artigos potencialmente relevantes.

Assim sendo, com os 206 artigos potencialmente relevantes selecionados, ver Figura 3, damos início ao uso da ferramenta Start<sup>4</sup> para apoio ao processo. Foram armazenadas o conjunto de referências bibliográficas dos artigos selecionados e importadas para a ferramenta.

<sup>4</sup> Ferramenta de apoio ao planejamento e execução de revisões sistemáticas.

Ao importar as referências a ferramenta faz uma verificação de possíveis artigos duplicados e ao fazer esta inspeção foram encontrados 26 artigos duplicados.



Figura 3 – Fase de seleção dos artigos.

Uma vez que as duas primeiras informações (título e resumo) não poderiam ser suficientes para identificar se o estudo é realmente relevante para esta pesquisa, foi realizado um segundo filtro que consistiu na leitura da introdução e conclusão dos estudos selecionados no primeiro filtro, sendo 115 artigos rejeitados e 65 artigos aceitos para leitura completa do texto.

Após a leitura completa do texto dos 65 artigos, detectamos trabalhos que não eram relevantes para a nossa pesquisa, os mesmos foram rejeitados, uma quantidade de 21 artigos foram excluídos após este terceiro filtro, ficando apenas 44 artigos aceitos para darmos início a extração das informações pertinentes para este trabalho, ver Figura 4.

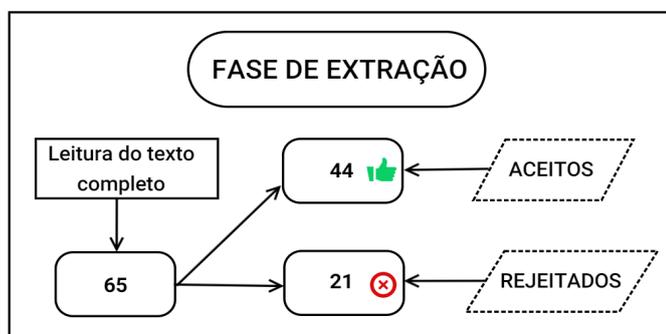


Figura 4 – Fase de extração.

Finalizados os procedimentos de seleção dos artigos e extração de informações, temos 44 artigos aceitos para extração de dados pertinentes para a condução deste mapeamento, correspondendo a 21% dos artigos selecionados considerados potencialmente relevantes. A lista completa dos 44 estudos primários finais está disponível na Tabela 2 do Apêndice. Foram detectados pela ferramenta *Start* 26 artigos duplicados, correspondendo a 13% dentre os artigos considerados potencialmente relevantes, os mesmos foram contabilizados apenas 1 vez, e por fim um total de 136 artigos foram rejeitados, correspondendo a 66% dos artigos potencialmente relevantes, ver Figura 5.

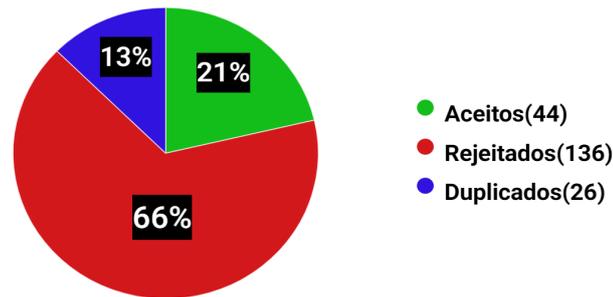


Figura 5 – Quantitativo de artigos para condução do Mapeamento.

## 6.2 Tendências De Publicação

A dimensão correspondente a distribuição dos estudos por ano será logo discutida, pois é um resultado direto do levantamento de estudos primários e básico para um mapeamento sistemático. A Figura 6 mostra a quantidade de estudos por ano de publicação, consideramos apenas os 44 estudos selecionados.

Podemos observar que os estudos selecionados foram publicados entre 2018 e 2023. De uma perspectiva temporal (Figura 6), observa-se que o ensino-aprendizagem de diagramas UML e as dificuldades e tecnologias<sup>5</sup> que suportam esse processo foi uma área que diminuiu a quantidade de publicações, um fator que pode ter contribuído para isso, foi o fato de termos uma pandemia no ano de 2020, isso pode ter acarretado na diminuição do interesse de se publicar sobre tal assunto. Este resultado evidencia o interesse científico e a necessidade de pesquisas sobre as dificuldades e tecnologias que apoiam o ensino da UML.

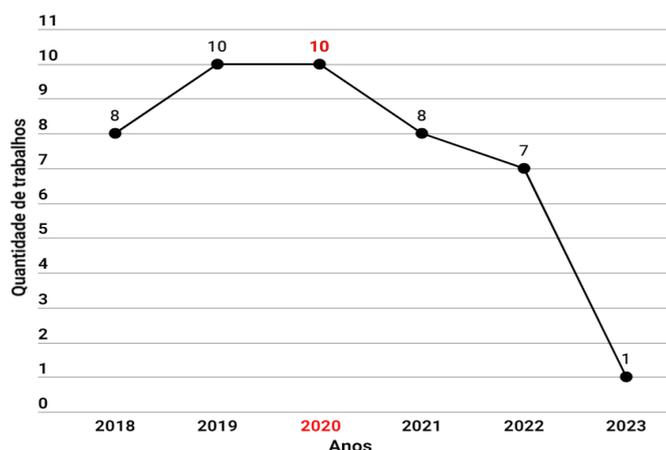


Figura 6 – Quantidade de trabalhos publicados por ano.

## 6.3 Frequência De Estudos Por Tipo/Meio De Publicação

Os estudos analisados foram classificados para avaliar sua distribuição por tipo de publicação (ou seja, conferência, revista, simpósio ou artigo do *workshop*), como mostra a Figura

<sup>5</sup> O termo “tecnologia” é utilizado como generalização para procedimentos, ferramentas, técnicas, metodologias e outras propostas realizadas na área de Engenharia de *Software* (SANTOS et al., 2012).

7. A maioria dos artigos analisados foram publicados em conferências um total de 13 (30%) trabalhos, seguido de trabalhos publicados em *journal/revista* 12 (27%) artigos, em terceiro vem artigos de simpósio com 11 (25%) trabalhos e, por fim, trabalhos de *workshops*, biblioteca digital, repositório e encontro/evento que somam 18% das publicações.

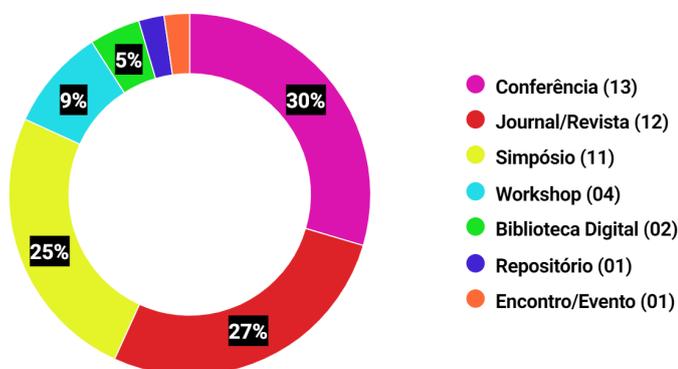


Figura 7 – Tipo/meio de publicação.

Por ter sido classificado como majoritário o meio de publicação (conferência), listamos a quantidade de trabalhos e suas respectivas conferências:

Temos 4 trabalhos publicados na *International Conference on Model Driven Engineering Languages and Systems (MODELS)*, Conferência Internacional sobre Linguagens e Sistemas de Engenharia Orientados a Modelos, é a principal conferência para software baseado em modelos e engenharia de sistemas, cobrindo todos os aspectos da modelagem, desde linguagens e métodos até ferramentas e aplicativos, desde 1998.

Contamos com 3 trabalhos publicados na *International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*, Conferência Internacional sobre Engenharia de *Software*: Educação e Treinamento em Engenharia de *Software* que busca contribuições originais que abranjam todas as dimensões do aprendizado e ensino de tópicos de engenharia de *software*.

Possuímos 3 trabalhos publicados na *IEEE Global Engineering Education Conference (EDUCON)*, Conferência Global de Educação em Engenharia que fornece um fórum para colaboração acadêmica, de pesquisa e industrial no ensino global de engenharia.

Dispomos de um trabalho publicado na *International Conference on Distance Education and Learning (ICDEL)*, Conferência Internacional sobre Educação e Aprendizagem a Distância, um trabalho publicado na *ECSEE: European Conference of Software Engineering Education*, Conferência Europeia de Educação em Engenharia de *Software*, e um trabalho publicado na *ICPC: International Conference on Program Comprehension*, Conferência Internacional sobre Compreensão de Programa.

#### 6.4 Idioma Predominante Nos Trabalhos

O idioma prevaiente nos artigos foi o Inglês com 32 (73%) estudos, por este ser adotado pela grande maioria das conferências e periódicos internacionais relacionados ao tema

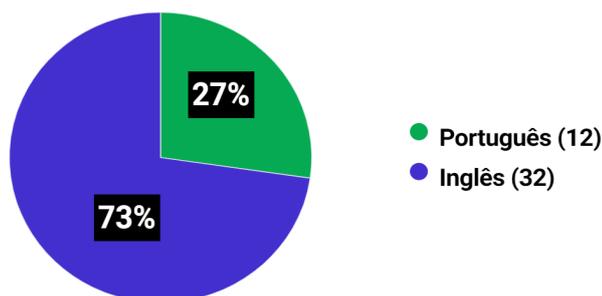


Figura 8 – Idioma dos artigos.

da pesquisa. Mas, temos também trabalhos escritos em português que abordam a temática no qual estamos pesquisando, foram selecionados um total de 12 estudos escritos em português.

### 6.5 Classificação Dos Trabalhos Por Categorias

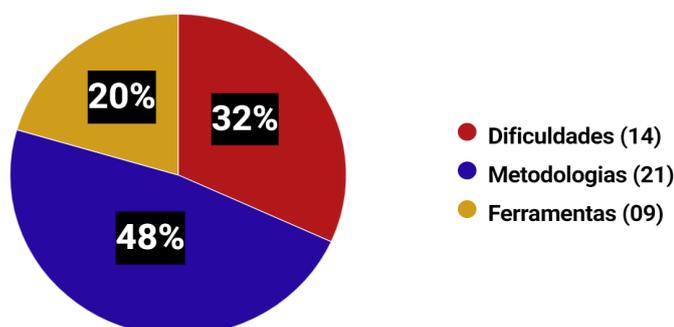


Figura 9 – Categorias dos trabalhos.

Analisando todos os trabalhos aceitos de uma forma geral, percebemos que poderíamos agrupá-los e classificá-los por categorias, sendo assim, constatamos um total de 21 (48%) trabalhos que se enquadram na categoria de metodologias de ensino-aprendizagem, 14 (32%) trabalhos na categoria dificuldades e por último a categoria ferramentas com 09 (20%) trabalhos, ver Figura 9.

### 6.6 Dificuldades Mencionadas Nos Trabalhos

Nesta subseção, mencionamos as dificuldades mais citadas nos trabalhos da categoria dificuldades por tipos de diagramas, ver Figura 10.

Com relação às dificuldades relacionadas aos diagramas, percebemos que os diagramas de classes, diagramas de sequência e diagramas de casos de uso tenham sido os diagramas com mais dificuldades relatadas, ver Figura 11. (BOBERIC-KRSTICEV; TESENDIC, 2013) relataram que esses três diagramas são os mais frequentemente usados para a modelagem de sistemas. Isso pode ocorrer porque esses diagramas representam as três visualizações UML mais comuns (MA, 2017): visualização funcional (diagrama de casos de uso), visão estrutural (diagrama de classes) e visão comportamental (diagrama de sequência).

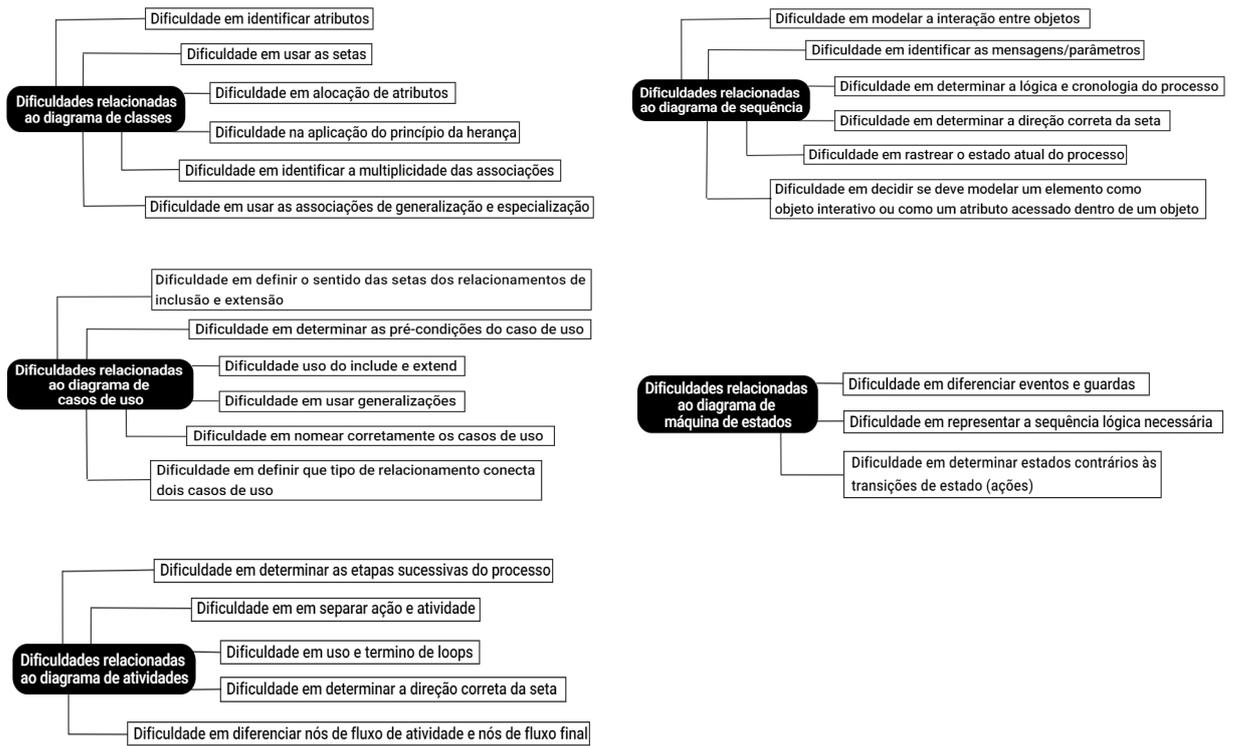


Figura 10 – Dificuldades relacionadas aos diagramas.

Sobre os diagramas de comunicação e objeto, não foram mencionadas dificuldades específicas para os mesmos. Por este motivo, não são relatados resultados para os mesmos.

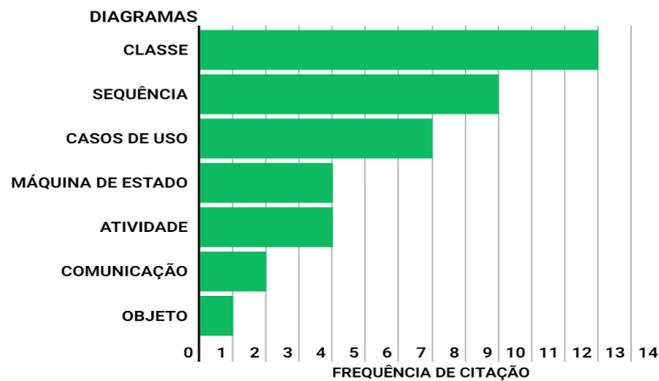


Figura 11 – Frequência de citação dos diagramas.

## 6.7 Ferramentas Mencionadas Nos Trabalhos

Essas foram as ferramentas encontradas nos trabalhos para apoio ao ensino-aprendizagem de modelagem de sistemas com a UML.

| Ano  | Ferramenta   | Descrição   |
|------|--|---|
| 2018 | UML Diagram Learning (UDL)   | Verificação e detecção automatizada de erros de diagramas de casos de uso e diagramas de classes.   |
| 2018 | Ferramenta para o Apoio ao Ensino-Aprendizagem do Paradigma Orientado a Objetos                    | O Avaliador de Logs gera um alerta para cada aluno, sempre que o total de intervenções sobre um elemento do Diagrama de Classes é menor que o total de intervenções que o professor fez, ou ainda igual ou maior que o dobro do total de intervenções que o professor fez.          |
| 2018 | Class Diagram Recommender System (CDRS)  | Ferramenta Educacional para análise e comparação de diagrama de classe criado por um aluno com uma coleção de diagramas de classe válidos armazenados em um repositório.  |
| 2018 | CD2Sys: Um Framework para a Aprendizagem Experiencial de Modelagem de Sistemas com UML             | Framework de geração de sistemas a partir de modelos de diagramas de classes UML.   |
| 2018 | Ferramenta UML-Code Consistency Checker Tool (UCCCT)   | Detectar inconsistências verticais e horizontais entre diagrama de classe, diagrama de caso de uso e diagrama de sequência usando regras de consistência.   |
| 2019 | Ferramenta de avaliação eletrônica formativa para construir diagramas de classe UML (e-assessment) | Avaliar automaticamente as soluções dos alunos e identificar o padrão de projeto que eles aplicaram. Esta informação é então usada para aconselhar os alunos sobre sua escolha de padrão de projeto e dar mais feedback centrado em uma solução exemplar.                           |
| 2020 | Ferramenta DUDE (Deep UML DEtector)  | Ferramenta que utiliza o aprendizado profundo para detectar visualmente os elementos do diagrama de classe UML e compará-los com uma determinada solução.   |
| 2021 | Ferramenta Blind Modeling system (B-Model)   | Ferramenta viabilizada por meio de uma linguagem chamada Blind Modeling Language (BML) para padronizar a especificação de requisitos a fim de facilitar a geração automática do diagrama de caso de uso correspondente.   |
| 2022 | Sistema AutoER   | Sistema para geração e avaliação automática de diagramas de projeto de banco de dados UML. Os alunos interagem diretamente com o texto da pergunta, e o sistema gera continuamente uma representação visual de sua resposta, além de fornecer feedback imediato a qualquer momento. |

Figura 12 – Ferramentas mencionadas nos artigos.

## 6.8 Metodologias Mencionadas Nos Trabalhos

As metodologias de ensino-aprendizagem de UML mencionadas nos trabalhos foram classificadas em três tipos, são elas: Engenharia Reversa, Utilização de projetos *FLOSS* e Metodologias Ativas de Aprendizagem 18 (85%) estudos, sendo essa a mais utilizada, ver Figura 13.

(LIU et al., 2021) utilizou a técnica de Engenharia Reversa, que consiste no estudo a partir da resolução de questões. Com a análise completa da questão e suas alternativas, ainda que o estudante erre, é possível que ele compreenda o que errou e o que precisa saber para não errar. É aí que está o segredo, já que o foco do estudo passa a ser mais objetivo com o treinamento para acertar a questão. O modo de processo é prática->teoria->re-prática, os alunos

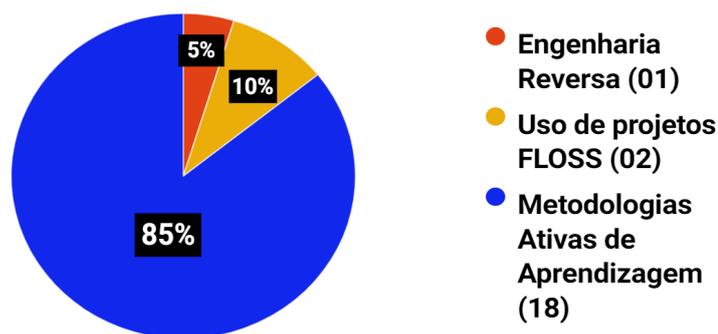


Figura 13 – Metodologias mencionadas nos artigos.

têm uma compreensão intuitiva do nível teórico, e a combinação de teoria e prática é mais fácil de entender e dominar conhecimento teórico.

(BRITO et al., 2019) e (SILVA et al., 2020) citam o uso de projetos FLOSS<sup>6</sup> no processo de ensino e aprendizagem de atividades de modelagem com diagramas UML.

Projetos de *Software Livre/Libre/Open Source (FLOSS)* podem ser uma solução viável para introduzir práticas realistas no processo de ensino e aprendizagem de Engenharia de *Software*. Esta abordagem é considerada como uma estratégia de aprendizagem experiencial e uma oportunidade para proporcionar aos alunos uma experiência vívida, conectada com possíveis experiências futuras. Como este é um ambiente autêntico onde o *software* real está sendo produzido, o corpo docente geralmente pode cobrir a maioria das áreas de conhecimento da Engenharia de *Software* (BRITO et al., 2019).

As metodologias ou abordagens de aprendizagem ativa, por sua vez, têm obtido notório destaque na área da educação, estando cada vez mais presentes na pauta de discussão da área, de forma a serem pesquisadas as suas possibilidades e as limitações da implementação, nos diferentes contextos e níveis de ensino (BERSSANETTE et al., 2021).

Um aspecto relevante sobre a aprendizagem ativa concerne à intencionalidade da aprendizagem, pois a ação de aprender se torna mais atrativa quando o assunto é de interesse do estudante, sendo pertinente ao contexto em que se situa. Tais aspectos possibilitam o aumento da flexibilidade cognitiva, da capacidade de realizar tarefas diversas e de adaptação a situações inesperadas (BACICH; MORAN, 2018).

Levando em consideração que a implementação da metodologia ativa envolve quatro esferas (instituição de ensino, docente, discente e o processo de ensino-aprendizagem), observou-se nos estudos analisados alguns métodos e/ou modelos de metodologias ativas de aprendizagem que são utilizadas nas práticas pedagógicas. Dentre os quais destacam-se: *Gamificação*; Aprendizagem Colaborativa; Aprendizagem entre Pares; Aprendizagem Baseada em Problema e Aprendizagem Interativa. Dentre elas, a *Gamificação* foi a mais utilizada nos estudos com 44%, ver Figura 14.

A *gamificação* na educação é um dos principais métodos de aprendizagem ativa utilizados hoje (BACICH; MORAN, 2018). Trata-se, essencialmente, de trazer elementos comuns a

<sup>6</sup> Free/Libre and Open Source *Software* (*Software* gratuito/livre e de código aberto)

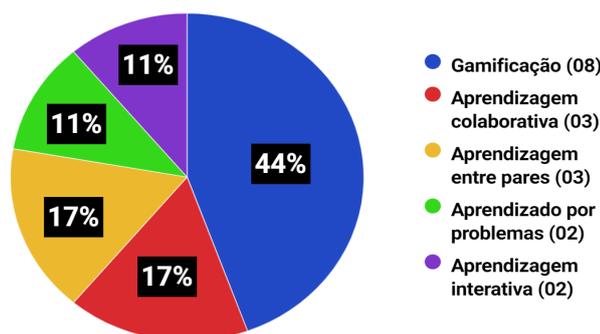


Figura 14 – Tipos de Metodologias Ativas de Aprendizagem.

videogames (como desafios, regras e narrativas) para o ensino. Desse modo, é possível expor os alunos a problemas baseados em diferentes situações, disponibilizando recursos diferenciados para que possam resolvê-los, seja individualmente ou em grupo. É uma prática que estimula o ensino lúdico e o pensamento analítico, desenvolvendo habilidades antes inéditas na sala de aula.

Aprendizagem colaborativa é uma estratégia diferenciada de ensino que se baseia na interação e na participação ativa dos alunos no processo de construção do conhecimento. O objetivo é promover a troca de experiências, o cooperativismo e o engajamento dos estudantes, colocando-os como protagonistas. Dessa forma, todos são convocados a estudar em grupo em busca de um único objetivo, com a ajuda de um professor que conduzirá esse processo. Com essa abordagem, a escola deixa de apresentar respostas prontas para seus estudantes, permitindo que eles pensem e participem ativamente do processo de aprendizagem (ANGELO et al., 2023).

A aprendizagem entre pares (ou times), também conhecida como *peer instruction* ou *team based learning*, é uma metodologia ativa que incentiva o debate e a reflexão em conjunto. Para isso, a turma de alunos é dividida entre pares ou grupos com o objetivo de gerar a troca de ideias sobre o conteúdo estudado. Desse modo, o aprendizado é formado conjuntamente, o que incentiva o desenvolvimento do senso crítico e da capacidade de argumentação dos alunos (ANGELO et al., 2023).

Aprendizagem Baseada em Problemas (ABP). Trata-se de uma metodologia de ensino-aprendizagem em que a aquisição do conhecimento e o desenvolvimento das habilidades dos alunos estão interligados. Durante todo o seu processo, os alunos trabalham ativamente e colaborativamente na proposta de solução para algum problema indicado pelo professor. Assim, eles aprimoram suas análises e sínteses de informações e se comprometem com o aprendizado. Porém, é importante destacar que o objetivo principal dessa metodologia não é simplesmente resolver um problema, e sim que ela seja utilizada como base para identificar os temas de aprendizagem para o estudo dos alunos, de maneira individual ou em grupos. Nela, os alunos passam a assumir responsabilidade e confiança e precisam desenvolver a habilidade de dar e receber críticas orientadas pelo professor para a melhoria do desempenho (ANGELO et al., 2023).

Aprendizagem interativa diz respeito ao processo de ensino baseado na troca e interação constante entre professores e estudantes, proporcionando uma aprendizagem com níveis mais

elevados de motivação e engajamento. Nessa abordagem, os professores servem como um líder do processo de aprendizagem, atuando não só na transmissão de conhecimentos e informações aos estudantes como também motivando a sua dedicação aos estudos. Os estudantes, por sua vez, formam a parte que permanece ativa ao longo da jornada do ensino, não apenas consumindo o conteúdo transmitido pelos professores como também participando ativamente ao expor suas ideias, discussões e *feedbacks* (ANGELO et al., 2023).

## 7 CONCLUSÃO

Este trabalho objetivou investigar o processo de ensino-aprendizagem de modelagem de *software* com a UML, em estudos acadêmico-científicos entre os anos de 2018 a 2023. Para isto, foi realizado um mapeamento sistemático da literatura, definindo a questão de pesquisa, de modo a caracterizar o estado da arte das publicações disponíveis sobre o tema. A partir de um conjunto inicial de 1.500 artigos, um total de 44 estudos primários foram selecionados neste mapeamento.

Como resultado desse mapeamento sistemático, identificamos trabalhos que relatam as dificuldades, ferramentas e metodologias que auxiliam no processo de criação de diagramas da UML.

No que concerne as dificuldades percebidas pelos estudantes na criação de diagramas UML, os diagramas de classe, sequência e caso de uso foram os diagramas com mais dificuldades relatadas. Esses três diagramas são os mais usados, porque eles representam as três visões UML mais comuns: visão funcional (diagrama de caso de uso); visão estrutural (diagrama de classes) e visão comportamental (diagrama de sequência).

Sobre às ferramentas identificadas, observou-se que a maioria delas é construída para fins acadêmicos. Elas podem aumentar o desempenho, reduzindo a sobrecarga e facilitando o trabalho de professores e dos estudantes durante o processo de ensino e aprendizado dos diagramas da UML. Percebemos que as tecnologias têm sido constantemente modificadas para melhor suportar o projeto e o aprendizado de diagramas UML. Portanto, não existe uma tecnologia única que seja adequada para todas as circunstâncias e tipos de artefatos de diagramas UML. Depende do propósito dos estudantes e do tipo de artefato que é elaborado (por exemplo, diagrama de classes, diagrama de sequência, diagrama de caso de uso).

Com relação aos métodos de ensino identificados, estudos têm evidenciado que empregar novas estratégias de ensino nos cursos, por exemplo, estratégias ativas de aprendizagem, proporciona uma experiência educacional bem-sucedida. Os alunos adquirem as competências instrumentais e sistêmicas relacionadas à modelagem de *software*, bem como habilidades interpessoais, como trabalho colaborativo, comprometimento com o trabalho e outras.

O presente trabalho pode fornecer uma base para futuras pesquisas relacionadas ao ensino-aprendizagem de modelagem e servir como uma referência para professores e pesquisadores em diagramas UML. Além disso, entender as dificuldades pode ajudar os professores a entender como isso afeta a qualidade dos diagramas resultantes.

## REFERÊNCIAS

- AL-FEDAGHI, S. Classes in object-oriented modeling (uml): Further understanding and abstraction. **arXiv preprint arXiv:2106.00267**, 2021.
- AL-TAHAT, K. An innovative instructional method for teaching object-oriented modelling. **International Arab Journal of Information Technology (IAJIT)**, v. 11, n. 6, 2014.
- ALHAZMI, S.; THEVATHAYAN, C.; HAMILTON, M. Learning uml sequence diagrams with a new constructivist pedagogical tool: Sd4ed. In: **Proceedings of the 52nd ACM Technical Symposium on Computer Science Education**. New York, NY, USA: Association for Computing Machinery, 2021. (SIGCSE '21), p. 893–899. ISBN 9781450380621. Disponível em: <<https://doi.org/10.1145/3408877.3432521>>.
- ALI, N. M. et al. Id no. upm 008 topic: Uml diagram learning tool. **UNIVERSITY CARNIVAL on e-LEARNING (IUCEL) 2018**, p. 408, 2018.
- ALMADI, S. H.; ALI, N. M.; ADMODISASTRO, N. Id no. upm 019 topic: Class diagram recommender system (cdrs): An educational tool for uml class diagram. **UNIVERSITY CARNIVAL on e-LEARNING (IUCEL) 2018**, p. 434, 2018.
- ANGELO, D. F. dos S. et al. Metodologias ativas e sua implementação no processo de ensino-aprendizagem: uma revisão integrativa. **Editora Licuri**, p. 126–143, 2023.
- AZEVEDO, P. de; SOUZA, F. C.; SOUZA, A. B-model uma ferramenta para auxiliar estudantes com deficiência visual na modelagem de sistemas. **Revista Eletrônica de Iniciação Científica em Computação**, v. 19, n. 3, 2021.
- AZEVEDO, R. C. d. et al. Rp-uml: uma arquitetura pedagógica para apoiar ensino-aprendizagem de modelagem de sistemas. Universidade Federal do Amazonas, 2021.
- BACICH, L.; MORAN, J. **Metodologias ativas para uma educação inovadora: uma abordagem teórico-prática**. [S.l.]: Penso Editora, 2018.
- BAHIA, C.; GADELHA, B. Cd2sys: Um framework para a aprendizagem experiencial de modelagem de sistemas com uml. In: **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)**. [S.l.: s.n.], 2018. v. 29, n. 1, p. 408.
- BERSSANETTE, J. H. et al. Metodologias ativas de aprendizagem e a teoria da carga cognitiva para a construção de caminhos no ensino de programação de computadores. Universidade Tecnológica Federal do Paraná, 2021.
- BOBERIC-KRSTICEV, D.; TESENDIC, D. Experience in teaching ooad to various students. **Informatics in Education**, Vilnius University Institute of Data Science and Digital Technologies, v. 12, n. 1, p. 43–58, 2013.
- BOLLOJU, N. Teaching and learning domain modeling through collaboration patterns: A controlled experiment. **ACM Trans. Comput. Educ.**, Association for Computing Machinery, New York, NY, USA, v. 22, n. 3, jun 2022. Disponível em: <<https://doi.org/10.1145/3513139>>.
- BOLLOJU, N.; LEUNG, F. S. Assisting novice analysts in developing quality conceptual models with uml. **Communications of the ACM**, ACM New York, NY, USA, v. 49, n. 7, p. 108–112, 2006.

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **UML: guia do usuário**. [S.l.]: Elsevier Brasil, 2006.

BRERETON, P. et al. Lessons from applying the systematic literature review process within the software engineering domain. **Journal of systems and software**, Elsevier, v. 80, n. 4, p. 571–583, 2007.

BRITO, M. S. et al. Floss in software engineering education: Supporting the instructor in the quest for providing real experience for students. In: **Proceedings of the XXXIII Brazilian Symposium on Software Engineering**. New York, NY, USA: Association for Computing Machinery, 2019. (SBES '19), p. 234–243. ISBN 9781450376518. Disponível em: <<https://doi.org/10.1145/3350768.3353815>>.

BUCCHIARONE, A. et al. Papyrus for gamers, let's play modeling. In: **Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings**. New York, NY, USA: Association for Computing Machinery, 2020. (MODELS '20). ISBN 9781450381352. Disponível em: <<https://doi.org/10.1145/3417990.3422002>>.

BUCCHIARONE, A. et al. Gamifying model-based engineering: the papygame experience. **Software and Systems Modeling**, Springer, p. 1–21, 2023.

CHOURIO, P. et al. Most common errors in software modeling using uml. In: **Proceedings of the XXXIII Brazilian Symposium on Software Engineering**. New York, NY, USA: Association for Computing Machinery, 2019. (SBES '19), p. 244–253. ISBN 9781450376518. Disponível em: <<https://doi.org/10.1145/3350768.3353820>>.

CHREN, S. et al. Mistakes in uml diagrams: Analysis of student projects in a software engineering course. In: **Proceedings of the 41st International Conference on Software Engineering: Software Engineering Education and Training**. IEEE Press, 2019. (ICSE-SEET '19), p. 100–109. Disponível em: <<https://doi.org/10.1109/ICSE-SEET.2019.00019>>.

COSTA, Í. et al. Activities in space: Design e avaliação de um jogo sério para o ensino de modelagem de diagrama de atividades. **SBC-Proceedings of SBGames**, 2019.

COSTA, R.; CASTRO, A.; GADELHA, B. Aprendizagem de modelagem de sistemas com uml: Concepção de uma arquitetura pedagógica. In: **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**. [S.l.: s.n.], 2019. v. 8, n. 1, p. 881.

DERMEVAL, D.; COELHO, J. A. d. M.; BITTENCOURT, I. I. Mapeamento sistemático e revisão sistemática da literatura em informática na educação. **Metodologia de Pesquisa Científica em Informática na Educação: Abordagem Quantitativa**. Porto Alegre: SBC, 2020.

EL-ATTAR, M. et al. Extending the uml statecharts notation to model security aspects. **IEEE Transactions on Software Engineering**, IEEE, v. 41, n. 7, p. 661–690, 2015.

ERGIN, H. et al. A study of modeling perception in a first-time modeling class. In: **Proceedings of the 22nd International Conference on Model Driven Engineering Languages and Systems**. IEEE Press, 2021. (MODELS '19), p. 680–689. ISBN 9781728151250. Disponível em: <<https://doi.org/10.1109/MODELS-C.2019.00104>>.

FEICHAS, F. A. et al. Avaliação da percepção de uso de uma plataforma gamificada sob a perspectiva discente: uma abordagem no estudo da uml. Universidade Federal de Itajubá, 2022.

FELISBINO, C. M.; NETO, A. G. S. S.; BASTOS, L. C. Supporting to the teaching and learning process in object orientation during the construction of class diagrams. In: **Proceedings of the XXXII Brazilian Symposium on Software Engineering**. New York, NY, USA: Association for Computing Machinery, 2018. (SBES '18), p. 338–347. ISBN 9781450365031. Disponível em: <<https://doi.org/10.1145/3266237.3266258>>.

FLORES, P.; ALVAREZ, M.; TORRES, J. Identifying difficulties of software modeling through class diagrams: A long-term comparative analysis. **IEEE Access**, IEEE, v. 10, p. 28895–28910, 2022.

FOSS, S.; URAZOVA, T.; LAWRENCE, R. Learning uml database design and modeling with autoer. In: . New York, NY, USA: Association for Computing Machinery, 2022. (MODELS '22), p. 42–45. ISBN 9781450394673. Disponível em: <<https://doi.org/10.1145/3550356.3559091>>.

GARCÍA-HOLGADO, A.; VÁZQUEZ-INGELMO, A.; GARCÍA-PEÑALVO, F. J. C4 model in a software engineering subject to ease the comprehension of uml and the software development process. IEEE, 2020.

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

HERNANDES, E. et al. Avaliação da ferramenta start utilizando o modelo tam e o paradigma gqm. In: **Proceedings of 7th Experimental Software Engineering Latin American Workshop (ESELAW 2010)**. [S.l.: s.n.], 2010. p. 30.

HUBER, F.; HAGEL, G. Work-in-progress: Towards detection and syntactical analysis in uml class diagrams for software engineering education. In: IEEE. **2020 IEEE Global Engineering Education Conference (EDUCON)**. [S.l.], 2020. p. 3–7.

JUNIOR, A. J. S.; CAMPOS, I. C. de. Ensino de diagramas da uml para cegos. In: **20ª Mostra de Ensino, Pesquisa e Extensão do Campus Porto Alegre**. [S.l.: s.n.], 2019.

Júnior, E.; FARIAS, K. Modelgame: A quality model for gamified software modeling learning. In: **15th Brazilian Symposium on Software Components, Architectures, and Reuse**. New York, NY, USA: Association for Computing Machinery, 2021. (SBCARS '21), p. 100–109. ISBN 9781450384193. Disponível em: <<https://doi.org/10.1145/3483899.3483910>>.

KALIAPPAN, V.; ALI, N. M. Improving consistency of uml diagrams and its implementation using reverse engineering approach. **Bulletin of Electrical Engineering and Informatics**, v. 7, n. 4, p. 665–672, 2018.

KITCHENHAM, B.; CHARTERS, S. **Guidelines for performing systematic literature reviews in software engineering**. [S.l.]: UK, 2007.

KRUSCHE, S. et al. An interactive learning method to engage students in modeling. In: **Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Software Engineering Education and Training**. New York, NY, USA: Association for Computing Machinery, 2020. (ICSE-SEET '20), p. 12–22. ISBN 9781450371247. Disponível em: <<https://doi.org/10.1145/3377814.3381701>>.

LIU, H. et al. Application of reverse engineering in software engineering teaching: Combining case teaching with task driven teaching. In: . New York, NY, USA: Association for Computing Machinery, 2021. (ICDEL 2021), p. 256–261. ISBN 9781450390033. Disponível em: <<https://doi.org/10.1145/3474995.3475038>>.

LOPES, A.; STEINMACHER, I.; CONTE, T. Uml acceptance: Analyzing the students' perception of uml diagrams. In: **Proceedings of the XXXIII Brazilian Symposium on Software Engineering**. New York, NY, USA: Association for Computing Machinery, 2019. (SBES '19), p. 264–272. ISBN 9781450376518. Disponível em: <<https://doi.org/10.1145/3350768.3352575>>.

MA, Z. An approach to improve the quality of object-oriented models from novice modelers through project practice. **Frontiers of Computer Science**, Springer, v. 11, p. 485–498, 2017.

PETERSEN, K.; VAKKALANKA, S.; KUZNIARZ, L. Guidelines for conducting systematic mapping studies in software engineering: An update. **Information and software technology**, Elsevier, v. 64, p. 1–18, 2015.

PLANAS, E.; CABOT, J. How are uml class diagrams built in practice? a usability study of two uml tools: Magicdraw and papyrus. **Comput. Stand. Interfaces**, Elsevier Science Publishers B. V., NLD, v. 67, n. C, jan 2020. ISSN 0920-5489. Disponível em: <<https://doi.org/10.1016/j.csi.2019.103363>>.

REISCHMANN, T.; KUCHEN, H. A web-based e-assessment tool for design patterns in uml class diagrams. In: **Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing**. New York, NY, USA: Association for Computing Machinery, 2019. (SAC '19), p. 2435–2444. ISBN 9781450359337. Disponível em: <<https://doi.org/10.1145/3297280.3297520>>.

REUTER, R. et al. Insights in students' problems during uml modeling. In: IEEE. **2020 IEEE Global Engineering Education Conference (EDUCON)**. [S.l.], 2020. p. 592–600.

SANTOS, G. et al. Strategic alignment between academy and industry: a virtuous cycle to promote innovation in technology. In: IEEE. **2012 26th Brazilian Symposium on Software Engineering**. [S.l.], 2012. p. 196–200.

SEABRA, R. D.; FEICHAS, F. A. Pge-uml: Uma plataforma web gamificada para o estudo da unified modeling language. **RENOTE**, v. 20, n. 2, p. 249–258, 2022.

SHEN, Z.; TAN, S.; SIAU, K. Challenges in learning unified modeling language: From the perspective of diagrammatic representation and reasoning. Association for Information Systems (AIS), 2018.

SHMALLO, R.; SHROT, T. Constructive use of errors in teaching the uml class diagram in an is engineering course. **Journal of Information Systems Education**, v. 31, n. 4, p. 282, 2020.

SIAU, K.; LOO, P.-P. Identifying difficulties in learning uml. **Information Systems Management**, Taylor & Francis, v. 23, n. 3, p. 43–51, 2006.

SILVA, F. G. et al. Teaching uml models with floss projects: A study carried out during the period of social isolation imposed by the covid-19 pandemic. In: **Proceedings of the XXXIV Brazilian Symposium on Software Engineering**. New York, NY, USA: Association for Computing Machinery, 2020. (SBES '20), p. 483–492. ISBN 9781450387538. Disponível em: <<https://doi.org/10.1145/3422392.3422491>>.

SILVA, W. et al. What are the differences between group and individual modeling when learning uml? In: **Proceedings of the XXXII Brazilian Symposium on Software Engineering**. New York, NY, USA: Association for Computing Machinery, 2018. (SBES '18), p. 308–317. ISBN 9781450365031. Disponível em: <<https://doi.org/10.1145/3266237.3266255>>.

SILVA, W. et al. Towards an open repository for teaching software modeling applying active learning strategies. In: **Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Software Engineering Education and Training**. New York, NY, USA: Association for Computing Machinery, 2020. (ICSE-SEET '20), p. 162–172. ISBN 9781450371247. Disponível em: <<https://doi.org/10.1145/3377814.3381709>>.

SILVA, W.; STEINMACHER, I.; CONTE, T. Students' and instructors' perceptions of five different active learning strategies used to teach software modeling. **IEEE Access**, IEEE, v. 7, p. 184063–184077, 2019.

SILVA, W.; STEINMACHER, I.; CONTE, T. A systematic literature mapping on the difficulties faced by software engineers in learning uml diagrams. 01 2023.

SILVA, W. A. F. et al. Opensmals: um repositório aberto para auxiliar no ensino de modelagem de software empregando estratégias de aprendizagem ativa. Universidade Federal do Amazonas, 2020.

STEINMACHER, I. et al. A systematic literature review on the barriers faced by newcomers to open source software projects. **Information and Software Technology**, Elsevier, v. 59, p. 67–85, 2015.

STIKKOLORUM, D. R.; NETO, F. G. de O.; CHAUDRON, M. R. V. Evaluating didactic approaches used by teaching assistants for software analysis and design using uml. In: **Proceedings of the 3rd European Conference of Software Engineering Education**. New York, NY, USA: Association for Computing Machinery, 2018. (ECSEE'18), p. 122–131. ISBN 9781450363839. Disponível em: <<https://doi.org/10.1145/3209087.3209107>>.

THOMAS, P. J.; PATEL, D.; MAGANA, A. J. Characterizing student proficiency in software modeling in terms of functions, structures, and behaviors. **ACM Trans. Comput. Educ.**, Association for Computing Machinery, New York, NY, USA, v. 21, n. 3, jul 2021. Disponível em: <<https://doi.org/10.1145/3458039>>.

TROCHIM, W. M. An introduction to concept mapping for planning and evaluation. **Evaluation and program planning**, Elsevier, v. 12, n. 1, p. 1–16, 1989.

UNKELOS-SHPIGEL, N.; SHEIDIN, J.; KUPFER, M. Climb your way to the model: Teaching uml to software engineering students: Teaching case. In: SPRINGER. **Advanced Information Systems Engineering Workshops: CAiSE 2019 International Workshops, Rome, Italy, June 3-7, 2019, Proceedings 31**. [S.l.], 2019. p. 40–46.

VENDOME, C.; RAPOS, E. J.; DIGENNARO, N. How do i model my system? a qualitative study on the challenges that modelers experience. In: **Proceedings of the 30th IEEE/ACM International Conference on Program Comprehension**. New York, NY, USA: Association for Computing Machinery, 2022. (ICPC '22), p. 648–659. ISBN 9781450392983. Disponível em: <<https://doi.org/10.1145/3524610.3529160>>.

YIGITBAS, E. et al. Gamification-based uml learning environment in virtual reality. In: . New York, NY, USA: Association for Computing Machinery, 2022. (MODELS '22), p. 27–31. ISBN 9781450394673. Disponível em: <<https://doi.org/10.1145/3550356.3559088>>.

## APÊNDICE A

Tabela 2 - Lista de estudos primários para a condução do mapeamento sistemático.

| ID  | Referências   |
|-----|---|
| A01 | (KALIAPPAN; ALI, 2018)                                  |
| A02 | (BAHIA; GADELHA, 2018)                                  |
| A03 | (STIKKOLORUM; NETO; CHAUDRON, 2018)                     |
| A04 | (ALI et al., 2018)                                      |
| A05 | (ALMADI; ALI; ADMODISASTRO, 2018)                       |
| A06 | (FELISBINO; NETO; BASTOS, 2018)                         |
| A07 | (SHEN; TAN; SIAU, 2018)                                 |
| A08 | (SILVA et al., 2018)                                    |
| A09 | (CHREN et al., 2019)                                    |
| A10 | (SILVA; STEINMACHER; CONTE, 2019)                       |
| A11 | (JUNIOR; CAMPOS, 2019)                                  |
| A12 | (UNKELOS-SHPIGEL; SHEIDIN; KUPFER, 2019)                |
| A13 | (CHOURIO et al., 2019)                                  |
| A14 | (COSTA et al., 2019)                                    |
| A15 | (LOPES; STEINMACHER; CONTE, 2019)                       |
| A16 | (BRITO et al., 2019)                                    |
| A17 | (COSTA; CASTRO; GADELHA, 2019)                          |
| A18 | (REISCHMANN; KUCHEN, 2019)                              |
| A19 | (SHMALLO; SHROT, 2020)                                  |
| A20 | (KRUSCHE et al., 2020)                                  |
| A21 | (SILVA et al., 2020)                                    |
| A22 | (BUCCHIARONE et al., 2020)                              |
| A23 | (REUTER et al., 2020)                                   |
| A24 | (SILVA et al., 2020)                                    |
| A25 | (HUBER; HAGEL, 2020)                                    |
| A26 | (GARCÍA-HOLGADO; VÁZQUEZ-INGELMO; GARCÍA-PEÑALVO, 2020) |
| A27 | (SILVA et al., 2020)                                    |
| A28 | (PLANAS; CABOT, 2020)                                   |
| A29 | (ALHAZMI; THEVATHAYAN; HAMILTON, 2021)                  |
| A30 | (AZEVEDO et al., 2021)                                  |
| A31 | (THOMAS; PATEL; MAGANA, 2021)                           |
| A32 | (AZEVEDO; SOUZA; SOUZA, 2021)                           |
| A33 | (ERGIN et al., 2021)                                    |
| A34 | (Júnior; FARIAS, 2021)                                  |
| A35 | (LIU et al., 2021)                                      |
| A36 | (AL-FEDAGHI, 2021)                                      |
| A37 | (FLORES; ALVAREZ; TORRES, 2022)                         |
| A38 | (VENDOME; RAPOS; DIGENNARO, 2022)                       |
| A39 | (BOLLOJU, 2022)   |
| A40 | (FEICHAS et al., 2022)                                  |
| A41 | (SEABRA; FEICHAS, 2022)                                 |
| A42 | (FOSS; URAZOVA; LAWRENCE, 2022)                         |
| A43 | (YIGITBAS et al., 2022)                                 |
| A44 | (BUCCHIARONE et al., 2023)                              |

Fonte: Elaborada pelos autores.