



**UNIVERSIDADE FEDERAL DO MARANHÃO – UFMA**  
**CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA – CCET**  
**ENGENHARIA DA COMPUTAÇÃO**

**MIQUÉIAS PEREIRA COELHO**

**Ensino de manutenção de software:** desenvolvimento, aplicação e validação de uma abordagem teórica e prática com dinâmica simulada cliente-empresa

**SÃO LUÍS – MA**

**2025**

MIQUÉIAS PEREIRA COELHO

**Ensino de manutenção de software:** desenvolvimento, aplicação e validação de uma abordagem teórica e prática com dinâmica simulada cliente-empresa

Trabalho de Conclusão de Curso II, apresentado ao curso de Engenharia da Computação, como requisito para a obtenção do Título de Bacharel em Engenharia da Computação. Centro de Ciência Exatas e Tecnológicas da Universidade Federal do Maranhão.

Orientador: Prof. Dr. Davi Viana dos Santos.

SÃO LUÍS – MA

2025

Ficha gerada por meio do SIGAA/Biblioteca com dados fornecidos pelo(a) autor(a).  
Diretoria Integrada de Bibliotecas/UFMA

Coelho, Miquéias.

Ensino de manutenção de software: desenvolvimento, aplicação e validação de uma abordagem teórica e prática com dinâmica simulada cliente-empresa / Miquéias Coelho. - 2025.

56 p.

Orientador(a): Davi Viana dos Santos.

Curso de Engenharia da Computação, Universidade Federal do Maranhão, São Luís, 2025.

1. Manutenção de Software. 2. Ensino Teórico-prático. 3. Dinâmica Simulada Cliente-empresa. 4. Engenharia de Software. 5. Metodologias de Ensino. I. Viana dos Santos, Davi. II. Título.

## **AGRADECIMENTOS**

Sou imensamente grato ao meu Deus, pois, sem Ele, nada eu seria ou poderia fazer, mesmo em meio a dificuldades, problemas e tempestades internas e externas Ele sempre está comigo.

Agradeço imensamente à minha mãe Isa, por todo o apoio, cuidado, ombro amigo e auxílio em momentos difíceis durante toda minha vida, uma mãe que jamais mediu esforços para me ajudar com tudo que esteve ao seu alcance, te amo minha mãe.

Sou grato aos meus avós, Francisca e Natanael, pois, sempre cuidaram e cuidam de mim, amo vocês.

Agradeço ao meu querido e amado tio Isaías, por todas as palavras de incentivo e conselhos durante toda a minha vida.

Agradeço as minhas Uldinar e Elite, pelas orações, apoio e conselhos em muitos momentos de minha vida.

Agradeço ao meu orientador, professor Davi Viana, bem como ao Lucas e Bruno, que me auxiliaram, corrigiram e me ensinaram bastante durante toda a jornada de preparo deste trabalho.

## RESUMO

A manutenção de software (MS) está relacionada ao contínuo desenvolvimento das aplicações de software, prezando por manter sua funcionalidade e aplicabilidade com o passar do tempo. Desse modo, o ensino formal de MS, dentro do âmbito acadêmico, torna-se indispensável para formação de profissionais das mais variadas áreas da tecnologia. O presente trabalho tem como objetivo principal propor uma abordagem de ensino de manutenção de software que faça uso de teoria e prática simulada cliente-empresa, buscando compreender a relevância de tal abordagem no aprendizado dos alunos. Embora a MS seja fundamental para a formação de engenheiros da computação, na Universidade Federal do Maranhão (UFMA), o ensino dessa área, nos cursos Bacharelado Interdisciplinar em Ciência e Tecnologia (BICT) e Engenharia da Computação (ECP) é parte da ementa no plano curricular da disciplina de Engenharia de Software (ES). Essa limitação pode resultar em dificuldades profissionais para engenheiros da computação e desenvolvedores de software no mercado de trabalho. Desta forma, torna-se relevante propor metodologias de ensino de MS inovadoras e que colaboram positivamente para o ensino da área, o que pode resultar na formação de profissionais mais bem preparados para o mercado de trabalho. O presente trabalho adota a MS como área abrangente, englobando modificações efetuadas no software, aprimoramentos, extensões de suas funcionalidades e correção de defeitos, por meio da aplicação de uma metodologia de ensino de MS prático-teórica com simulação cliente-empresa. Observou-se, ao questionar os participantes da pesquisa, sobre o quanto julgavam ter adquirido conhecimentos relativos a MS, fazendo um paralelo com antes e após a intervenção, que 65,5% dos participantes afirmam ter passado de um nível 1 e 2, sendo 1 pouco conhecimento sobre a área e 2 muito conhecimento sobre a área, para os níveis 3 e 4, um avanço considerável e notório observado após a intervenção. De modo semelhante, quando questionados se após a metodologia aplicada, os participantes se autoavaliaram mais aptos a trabalhar com manutenção de software, 92,3% dos participantes afirmam estar no nível 3 e 4, sendo 3 bastante preparados para lidar com a área.

**Palavras-Chave:** Manutenção de Software; Ensino Teórico-Prático; Dinâmica Simulada Cliente-Empresa; Engenharia de Software; Metodologias de Ensino.

## ABSTRACT

Software maintenance (SM) is related to the continuous development of software applications, ensuring their functionality and applicability over time. Therefore, formal software maintenance education within the academic environment is essential for training professionals in a wide range of technology fields. This paper aims to propose a software maintenance teaching approach that utilizes theory and simulated client-company practice, seeking to understand the relevance of this approach to student learning. Although software maintenance is essential for the training of computer engineers, at the Federal University of Maranhão (UFMA), teaching this area in the Interdisciplinary Bachelor's Degree in Science and Technology (BICT) and Computer Engineering (ECP) programs is part of the Software Engineering (SE) curriculum. This limitation can result in professional difficulties for computer engineers and software developers in the job market. Therefore, it is important to propose innovative SM teaching methodologies that positively contribute to the field's education, potentially resulting in the training of professionals better prepared for the job market. This study adopts SM as a comprehensive area, encompassing software modifications, enhancements, functional extensions, and defect corrections, through the application of a practical-theoretical SM teaching methodology with client-company simulation. When survey participants were asked about their perceived knowledge of SM, comparing pre- and post-intervention, 65.5% of participants reported having advanced from levels 1 and 2 (one with little knowledge of the field and four with a great deal of knowledge) to levels 3 and 4, a considerable and notable improvement observed after the intervention. Similarly, when asked whether, after applying the methodology, participants considered themselves more capable of working with software maintenance, 92.3% of participants stated that they were at levels 3 and 4, with 4 being quite prepared to deal with the area.

**Keywords:** Software Maintenance; Theoretical-Practical Teaching; Client-Company Dynamics; Software Engineering; Teaching Methodologies.

## ÍNDICE DE FIGURAS

|                                                                                         |    |
|-----------------------------------------------------------------------------------------|----|
| Figura 2 - Caderno Virtual (Notion).....                                                | 24 |
| Figura 3 - Estudo de Caso Simulado: apresentação da empresa simulada.....               | 29 |
| Figura 4 - Estudo de Caso Simulado: contato com o cliente.....                          | 29 |
| Figura 5 - Estudo de Caso Simulado: tabela de manutenções.....                          | 30 |
| Figura 5 - Software Pizza Mais.....                                                     | 31 |
| Figura 7 - Gráfico de experiência com manutenção de software.....                       | 34 |
| Figura 8 - Gráfico de familiaridade com a ferramenta Git.....                           | 35 |
| Figura 9 - Gráfico de familiaridade com a ferramenta GitHub.....                        | 35 |
| Figura 10 - Gráfico de familiaridade com a linguagem de programação Python.....         | 36 |
| Figura 11 - Gráfico de familiaridade sobre banco de dados relacionais.....              | 37 |
| Figura 12 - Nível de conhecimento sobre manutenção de software antes do projeto.....    | 38 |
| Figura 13 - Nível de conhecimento sobre manutenção de software após o projeto.....      | 39 |
| Figura 14 - Gráfico de identificação dos tipos de manutenção de software.....           | 40 |
| Figura 15 - Relevância do documento Estudo de Caso Simulado A.....                      | 40 |
| Figura 16 - Relevância do documento Estudo de Caso Simulado B.....                      | 41 |
| Figura 17 - Relevância do documento Estudo de Caso Simulado C.....                      | 41 |
| Figura 18 - Motivação para estudo de MS após aplicação da metodologia.....              | 42 |
| Figura 19 - Nível de preparo para lidar com MS após a parte prática da metodologia..... | 43 |

## ÍNDICE DE TABELAS

|                                                                                                                                                    |    |
|----------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Tabela 1 - Desafios nas ferramentas tecnológicas utilizadas na metodologia.....                                                                    | 44 |
| Tabela 2 - Dificuldades informadas pelos participantes.....                                                                                        | 44 |
| Tabela 3 - Aprendizado e sugestão dos participantes.....                                                                                           | 45 |
| Tabela 4 - Síntese geral das semelhanças entre a metodologia simulada cliente-empresa e outras abordagens de ensino de manutenção de software..... | 48 |

## **LISTA DE ABREVIATURAS**

|       |                                                      |
|-------|------------------------------------------------------|
| ABI   | Aprendizagem Baseada em Investigação                 |
| BICT  | Bacharelado Interdisciplinar em Ciência e Tecnologia |
| ECP   | Engenharia da Computação                             |
| ES    | Engenharia de Software                               |
| GSL   | Gerenciamento de Sistemas Legados                    |
| JED   | Jogo Educacional Digital                             |
| MS    | Manutenção de Software                               |
| SimMS | Simulador de Manutenção de Software                  |
| TCLE  | Termo de Consentimento Livre Esclarecido             |
| UFMA  | Universidade Federal do Maranhão                     |

## SUMÁRIO

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |           |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| <b>1. INTRODUÇÃO.....</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | <b>11</b> |
| 1.1. Justificativa.....                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 12        |
| 1.2. Objetivos.....                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | 14        |
| 1.2.1. Objetivo geral.....                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 14        |
| 1.2.2. Objetivos específicos.....                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 14        |
| <b>2. REFERENCIAL TEÓRICO E TRABALHOS RELACIONADOS.....</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | <b>15</b> |
| 2.1. Manutenção de software.....                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 15        |
| 2.2. Tipos de manutenção de software: corretivas, adaptativas e perfectivas.....                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 16        |
| 2.3. Metodologias aplicadas ao ensino de manutenção de software.....                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 17        |
| 2.3.1. SimMS.....                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 17        |
| 2.3.2. Hogwarts da Manutenção de Software.....                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | 17        |
| 2.3.3. Do code smell ao código limpo.....                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 19        |
| 2.3.4. Aplicação da metodologia ativa Aprendizagem Baseada em Investigação (ABI).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 20        |
| <b>3. MÉTODO DE PESQUISA.....</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | <b>22</b> |
| 3.1. Planejamento.....                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 23        |
| 3.2. Revisão de Materiais.....                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | 25        |
| 3.3. Estudo de Caso Simulado.....                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 28        |
| A Figura 5, tem por finalidade exemplificar como foram tratadas cada uma das manutenções no presente trabalho. Os participantes tinham como objetivo observar as interações do cliente e dessa forma preencher corretamente as lacunas da tabela, de forma a responderem de maneira correta, o tipo de manutenção a ser realizada para cada uma das interações apresentadas. Desta maneira, cobrindo assim os 3 tipos de manutenções apresentadas de maneira teórica. A tabela foi disposta com espaços em branco e espaços preenchidos, de forma que os alunos pudessem observar exemplos de como preencher, e à partir de tais exemplos completar os demais espaços..... |           |
| 3.4. Software para ambiente simulado cliente-empresa.....                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 30        |
| 3.5. Aulas piloto.....                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 32        |
| 3.6. Execução da metodologia.....                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 32        |
| <b>4. RESULTADOS E DISCUSSÕES.....</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | <b>33</b> |
| 4.1. Caracterização inicial dos participantes.....                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | 33        |
| 4.2. Percepção pós-aplicação da metodologia.....                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 37        |
| 4.2.1. Ferramentas tecnológicas utilizadas na aplicação da metodologia.....                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 43        |
| 4.3. Dificuldades encontradas pelos participantes.....                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 44        |
| 4.4. Experiência dos participantes quanto à metodologia simulada cliente-empresa.....                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 45        |
| 4.6. Comparação entre a metodologia simulada cliente-empresa e demais metodologias abordadas.....                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 47        |
| <b>5. CONCLUSÃO.....</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | <b>50</b> |
| <b>6. REFERÊNCIAS.....</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | <b>52</b> |
| <b>APÊNDICE A - CADERNO VIRTUAL NOTION.....</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | <b>54</b> |
| <b>APÊNDICE B - ESTUDO DE CASO SIMULADO.....</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | <b>55</b> |

|                                                                  |           |
|------------------------------------------------------------------|-----------|
| <b>APÊNDICE C - SOFTWARE PARA APLICAÇÃO DAS MANUTENÇÕES.....</b> | <b>56</b> |
|------------------------------------------------------------------|-----------|

## 1. INTRODUÇÃO

As atividades sociais, econômicas e culturais humanas passaram por transformações ao longo do tempo, principalmente devido ao avanço tecnológico. Computadores, celulares, dispositivos inteligentes, entre outros aparelhos eletrônicos, tornaram-se parte do cotidiano dos seres humanos. De maneira geral, estes avanços tecnológicos fazem uso de processadores, memórias de armazenamento, bem como lógicas de tarefas, que são definidas mediante o uso de softwares, que de maneira constante precisam passar por correções e adaptações oriundas de mudanças no meio em que estão inseridos. Segundo Amorim (2015), software é qualquer programa de computador que tenha capacidade de comandar um sistema computacional.

Como citado por Bordin *et al* (2024), estão entre as recomendações de inclusão em projetos pedagógicos de graduação, o incentivo à investigação. Desta forma, a ES, como demais áreas, também possui questões investigativas em aberto, como afirmado pelo mesmo autor. “A Aprendizagem Baseada em Investigação é uma estratégia educacional na qual os alunos seguem métodos e práticas semelhantes aos dos profissionais cientistas para construir conhecimento [Pedaste et al. 2015].” (Bordin et al, 2024, p. 2). De posse dessa afirmação, uma metodologia de ensino ativa, que busque simular um ambiente próximo ao ambiente de trabalho torna-se uma abordagem viável para a área de ES e mais especificamente o ensino de MS.

Nesse contexto, o ensino de MS é uma ferramenta para a formação de profissionais capazes não somente de desenvolver, mas também de manter o funcionamento dos softwares. Saiedian (1999) afirma que é importante que a educação de Engenharia de Software (ES) abranja não só conhecimentos basilares e habilidades referentes à Ciência da Computação, mas também esteja preocupada com a definição, desenvolvimento e manutenção de software. De acordo com o mesmo autor, é dessa forma que os esforços de hoje contribuirão para a qualidade dos softwares produzidos pela próxima geração de engenheiros.

No entanto, embora a MS seja fundamental para a formação de engenheiros da computação, na UFMA, o ensino dessa área está incluído no plano curricular da disciplina de ES para os alunos dos cursos BICT e ECP. Porém, essa abordagem limitada devido ao pequeno tempo dedicado à área de MS dificulta um ensino apoiado em práticas simuladas de um contexto real do mercado de trabalho. Essa limitação pode resultar em dificuldades profissionais para engenheiros da computação e desenvolvedores de software no mercado de trabalho, “[...] falta de glamour da atividade de manutenção,[...] impossibilidade de

desenvolvimento de carreira em manutenção de software" (Paduelli, 2007, p.44). Tal julgamento de valor pode decorrer da não compreensão da área de MS e sua importância no ciclo de vida dos softwares.

### 1.1. Justificativa

Assim como aborda Paduelli (2007), muitos desenvolvedores encaram a área de MS sem *glamour* e sem possibilidade de crescimento profissional. No entanto, quando considerado o notório crescimento da presença de softwares no cotidiano dos seres humanos, aliado ao aumento da complexidade, escala e tamanho dos projetos de software, como aborda Magalhães *et al* (2020), é possível perceber a necessidade de manter os softwares em pleno funcionamento. Muitas das atividades humanas fazem uso e dependem diretamente de linhas e mais linhas de código, sejam estas atividades bancárias, gestão escolar de alunos, análise de amostras de componentes químicos, entre outras. Como afirma Magalhães *et al* (2020), as atividades de manutenção de software são indispensáveis para assegurar o correto funcionamento do software durante seu ciclo de vida. Essa concepção dialoga com o exposto por Sirqueira (2013), para o autor a MS ocorre durante todo o ciclo de vida do software, impactando no custo total do software, assim como em seu tempo de vida.

Atividades de manutenção são essenciais para manter o pleno funcionamento de um software durante seu ciclo de vida. Como exemplos tem-se a adição de novas funcionalidades no software com tarefas do tipo, adaptativa e evolutiva tais quais requisições de mudança (*change request*), ou então por meio de tarefas do tipo corretiva para correção de defeitos expostos pelos relatórios de erros (*bug report*). A fase de manutenção compõe a maior parte do ciclo de vida de um software. Segundo Erlikh (2000), esta fase despende cerca de 90% de todo o esforço do desenvolvimento de um software. (Magalhães *et al*, 2020, p.1).

A manutenção de um software é algo que ocorre durante todo o ciclo de vida de um software, e que gera um grande impacto tanto em relação ao custo total do software, bem como em seu tempo total de vida. A última década assistiu a um grande progresso das técnicas de desenvolvimento de software. Novos processos, linguagens e ferramentas foram propostas e adotadas. A manutenção de software, ao contrário, parece ter ficado para trás. (Sirqueira, 2013, p. 9).

Deste modo, a MS revela-se um alicerce para engenheiros da computação e desenvolvedores de software. Mostra-se uma área presente durante todo o ciclo de vida do

software, requerendo cerca de 90% de todo o esforço relacionado ao processo de construção do software, como afirma Magalhães *et al* (2020). Levando em consideração tais afirmações, surge a necessidade de compreender a área de MS, bem como maneiras de ensino efetivas da área, para contribuir para a formação de profissionais da área de tecnologia.

No ambiente da indústria de desenvolvimento de software, o autor percebe a lacuna existente entre o ambiente profissional e acadêmico, quanto à organização, e documentação dos softwares, dessa forma, é necessário adquirir conhecimentos que vão além da teoria a respeito da área, pois, parte do trabalho que é exigido dentro do ambiente profissional não consta em livros. Desta forma, uma metodologia que visa preencher este espaço existente, dentro da perspectiva do autor, tende a contribuir para a construção de conhecimento na área de MS, de forma a aproximar os alunos do ambiente profissional.

A adoção de metodologias de ensino que simulem o ambiente do mercado de trabalho contribuem para a formação de bons profissionais. Segundo Prikladnicki (1999), uma aula expositiva tende a possuir pouca eficiência no aprendizado, por estimular somente a audição.

Eventos simulados e atividades vivenciais permitem assimilar situações diferentes na prática. Sendo assim, a aprendizagem mais prática acaba sendo caracterizada pelo envolvimento de uma pessoa em uma determinada atividade, com resultados mais positivos, visto que os alunos acabam sendo responsáveis pela definição de rumos de uma situação proposta. (Prikladnicki, 1999, p. 3).

Neste contexto, a disciplina de ES da UFMA, ministrada aos alunos dos cursos de graduação BICT e ECP, precisa assegurar a transmissão do conhecimento da área de MS de forma que os estudantes estejam engajados nas aulas e vivenciem um cenário similar ao qual serão expostos no mercado de trabalho. Deste modo, observa-se importante aplicar métodos de ensino que façam uso de práticas e ambientes simulados com dinâmicas que representem relações cliente-empresa.

Por esses motivos, essa pesquisa é cientificamente relevante, pois busca fornecer dados qualitativos e quantitativos para avaliar e servir de base para aperfeiçoar o modelo de ensino de MS no contexto apresentado. Para tal, fará uso de uma abordagem prática e teórica, aplicada ao ensino de MS, contribuindo para avaliação do modo de ensino utilizado pela instituição formadora, UFMA, dentro do contexto informado. Além do caráter científico, objetiva contribuir socialmente para a formação de profissionais da área de tecnologia cada

vez mais capacitados para o mercado. Isso pode resultar em resolução de problemas de software com maior rapidez e fluidez dentro das empresas prestadoras de serviço, beneficiando tanto os usuários dos sistemas, como as empresas donas desses serviços.

## **1.2. Objetivos**

### **1.2.1. Objetivo geral**

Este trabalho tem por objetivo geral apoiar o ensino de manutenção de software. Para tal, fazendo uso de uma abordagem de ensino teórica e prática, com simulação do ambiente de mercado de trabalho aplicando a dinâmica simulada cliente-empresa. Desta maneira, visa contribuir para o ensino de manutenção de software, de forma que venha preparar os estudantes da área para os desafios do mercado de trabalho.

### **1.2.2. Objetivos específicos**

- Definir a simulação de ambiente de mercado de trabalho, utilizando a dinâmica simulada cliente-empresa, fazendo uso de software próprio desenvolvido para esse fim.
- Investigar, por meio da aplicação de questionários, como os alunos avaliam a metodologia citada no ponto anterior no seu aprendizado de manutenção de software, comparado com aulas expositivas tradicionais sobre o tema.
- Avaliar se os alunos consideraram tal metodologia positiva para seu aprendizado na área de manutenção de software.

## 2. REFERENCIAL TEÓRICO E TRABALHOS RELACIONADOS

### 2.1. Manutenção de software

A princípio, é essencial compreender o que é MS. Ao buscar na literatura é possível encontrar diferentes visões a respeito do conceito de MS. Segundo Lientz (1983), MS está relacionada ao contínuo desenvolvimento da aplicação (sistema). Para Bennett e Rajlich (2000), a manutenção de software refere-se a atividades de âmbito geral em um software, após a sua entrega, “[...]usaremos manutenção para nos referirmos a atividades gerais pós-entrega[...]” (Bennet; Rajlich, 2000, p.75, tradução nossa)<sup>1</sup>.

Além das definições mostradas, há definições normativas. Segundo a norma técnica NBR ISO/IEC 12207 (1998), MS é definida como o gerenciamento das modificações no software, com o objetivo de manter sua operação e garantir sua atualização. Para o *Institute of Electrical and Electronics Engineers* (IEEE) (1998 citado por Paduelli 2007), o termo MS caracteriza-se pela modificação de um produto de software já entregue ao cliente, com a finalidade de corrigir erros, melhorar o desempenho ou adaptar o software a mudanças no ambiente. Os autores Lientz e Swanson (1980) expandem a definição de MS, tratando a manutenção de software como uma área mais abrangente, englobando modificações efetuadas no software, aprimoramentos, extensões de suas funcionalidades e correção de defeitos. Sommerville (2011) define MS como processo de mudanças realizadas no software após a sua entrega. Ainda de acordo com o autor existem três tipos de MS:

1. Correção de defeitos. Erros de codificação são relativamente baratos para serem corrigidos; erros de projeto são mais caros, pois podem implicar reescrever vários componentes de programa. Erros de requisitos são os mais caros para se corrigir devido ao extenso reprojetado de sistema que pode ser necessário.
2. Adaptação ambiental. Esse tipo de manutenção é necessário quando algum aspecto do ambiente do sistema, como o hardware, a plataforma do sistema operacional ou outro software de apoio sofre uma mudança. O sistema de aplicação deve ser modificado para se adaptar a essas mudanças de ambiente.
3. Adição de funcionalidade. Esse tipo de manutenção é necessário quando os requisitos de sistema mudam em resposta às mudanças organizacionais ou de negócios. A escala de mudanças necessárias para o software é, frequentemente, muito maior do que para os outros tipos de manutenção. (Sommerville, 2011, p. 170).

---

<sup>1</sup> “[...]we shall use maintenance to refer to general post-delivery activities, and evolution to refer to a particular phase in the staged model[...]” (Bennet; Rajlich, 2000, p.75)

Diante do exposto, o presente trabalho adotará a definição proposta por Lientz e Swanson (1980), que dialoga de maneira direta com a definição apresentada por Sommerville (2011). Segundo Lientz e Swanson (1980 citado por Paduelli (2007)), a MS pode ser classificada em três categorias: "As ações ligadas à atividade de manutenção de software foram classificadas de acordo com a sua natureza em três categorias (Lientz & Swanson, 1980): *corretivas, adaptativas e perfectivas*." (Paduelli, 2007, p. 34). Essas três categorias serão a base para o desenvolvimento de uma metodologia de ensino de manutenção de software apresentada neste trabalho, teórica e prática, por meio de uma dinâmica simulada cliente-empresa, que objetiva colaborar no ensino de manutenção de software na disciplina de ES, ministrada aos alunos de BICT e ECP.

## **2.2. Tipos de manutenção de software: *corretivas, adaptativas e perfectivas***

Como abordado anteriormente, Lientz e Swanson (1980 citado por Paduelli 2007) classificaram a MS em três tipos: *corretivas, adaptativas e perfectivas*.

- Manutenções corretivas: possuem como objetivo corrigir falhas (*bugs*), sejam elas emergenciais ou não, Paduelli (2007). Por exemplo, quando se espera que o software realize o envio de determinados arquivos em formato *xml* (*Extensible Markup Language*), porém, devido a *bugs*, o envio dos arquivos ocorre contendo erro de caracteres ou em formato diferente do esperado.
- Manutenções adaptativas: de acordo com Paduelli (2007), a manutenção adaptativa refere-se à adequação do software ao ambiente. Lima *et al* (2010) aponta uma situação real de manutenção adaptativa, onde uma operadora de telefonia móvel deseja que os aparelhos celulares quando vendidos, ao iniciarem seja exibido sua logomarca. Essa adaptação exige a modificação no código fonte do dispositivo embarcado.

Essa adaptação é executada através da modificação do código que é posteriormente embarcado no dispositivo móvel. Esse processo pode ser realizado de várias formas dependendo da tecnologia utilizada pelo equipamento. Estas adaptações (novas versões de software) para celulares são lançadas frequentemente, destinadas principalmente a atender especificações requeridas por operadoras para determinados modelos. (Lima *et al*, 2010, p. 328).

- Manutenções perfectivas: acréscimo de novos recursos ao software, Paduelli (2007). "Como exemplo, pode-se citar o pedido do usuário por um novo relatório com

informações que até então não podiam ser obtidas do banco de dados.” (Paduelli, 2007, p. 35).

### **2.3. Metodologias aplicadas ao ensino de manutenção de software**

Este tópico do trabalho apresenta de maneira objetiva algumas metodologias de ensino de manutenção de software encontradas na literatura, que fundamentam a proposta desse projeto de pesquisa.

#### **2.3.1. SimMS**

Simão, Correa e Júnior (2014), trazem uma abordagem de ensino que faz uso de um jogo educacional digital (JED), intitulado SimMS, definido pelos autores como um simulador de manutenção de software, conforme descrito na norma IEEE 1219 de 1998. O objetivo do jogo é ser um material de apoio didático para o ensino de MS. Trata-se de um jogo *single player* (jogador único), que simula o ambiente de trabalho em uma empresa de manutenção de software. Nesse contexto, o objetivo do jogador é concluir as manutenções cadastradas seguindo o padrão normativo.

A metodologia de ensino de MS utilizando o JED SimMS foi avaliada, por meio de questionário, por doze alunos do curso de graduação em Ciência da Computação da Universidade Federal de Goiás (Regional Jataí), durante a disciplina de ES. Ainda de acordo com os autores:

Com relação à facilidade uso percebida [...], pode-se destacar que 100% dos participantes da avaliação concordaram totalmente que o acesso ao jogo SimMS é fácil. [...]grande maioria (92%) concordou que utilizar o SimMS é uma boa ideia.(Simão; Correa; Júnior, 2014, p. 7).

#### **2.3.2. Hogwarts da Manutenção de Software**

Ainda abordando exemplos encontrados na literatura que fazem uso de modos de ensino de MS para além de aulas tradicionais, Lima e Rabelo (2023) trazem uma abordagem de ensino baseada na trama fantástica e *best-seller* do mundo de Harry Potter da autora J. K. Rowling lançado em 1995. A proposta feita pelos autores, está fundamentada na gamificação. “A gamificação se constitui na utilização da mecânica dos games em cenários non games”. (Alves; Minho; Diniz, 2014, p. 76).

Foram utilizados elementos lúdicos na proposta de ensino dos autores, com a finalidade de promover uma experiência lúdica ao jogador. Os elementos lúdicos são os que se seguem:

- Missões: conjunto de objetivos a serem alcançados.
- Pontos: quantidade de pontos acumulados durante a aplicação da metodologia.
- Avatar: representação visual de personagens, ou um papel visual dentro do processo da gamificação.
- Ranking: placar de colocação das equipes.
- Níveis: missões com progressão de nível à medida que os alunos avançam.
- Narrativa: artifício de correlação entre a proposta gamificada e o ensino da disciplina.
- Progressão: mecanismo que ilustra o avanço de cada participante no curso das atividades.
- Recompensas: benefícios ao concluir missões.
- Feedback: resumo do desempenho do aluno ao final de cada missão.

Durante o período de maio a junho de 2023, sessenta alunos do Campus Russas da Universidade Federal do Ceará (UFC), participantes da disciplina de Manutenção de Software, componente curricular obrigatório para o curso de Engenharia de Software e optativa para o curso de Ciência da Computação, foram submetidos à abordagem gamificada proposta pelos autores. Desse total, vinte e nove pertenciam ao curso de Engenharia de Software e trinta e um ao curso de Ciência da Computação.

No decorrer da aplicação do método de ensino, os alunos foram divididos em casas e equipes, assim como no *best-seller* citado, e foram designadas duas missões para cada equipe: documentação de software e documentação de manutenção. Com objetivo de ampliar a imersão dos alunos na proposta de ensino gamificada, os autores desenvolveram plataformas online.

Para intensificar a imersão dos alunos no cenário fictício, os autores implementaram uma página web informativa sobre a proposta gamificada. Um sistema bancário, baseado na obra fictícia, também foi construído pelos autores da pesquisa, visando facilitar o feedback e futuras trocas de pontos (benefícios atribuídos aos alunos). (Lima; Rabelo, 2023, p. 4).

Segundo os autores foram obtidos os seguintes resultados:

Em relação à organização da proposta gamificada, 61% alunos apresentaram satisfeitos. Ao serem questionados sobre a satisfação na construção das atividades, 72% alunos concordaram que realizar as atividades da disciplina durante a gamificação foi muito interessante. (Lima; Rabelo, 2023, p. 5).

### 2.3.3. Do *code smell* ao código limpo

Nascimento *et al* (2025) mostram em seu artigo um relato de ensino de refatoração de *code smells*, que foi realizado com estudantes da disciplina de Manutenção de Software, disciplina optativa do 7º período dos cursos: Bacharelado em Engenharia de Software e Ciência da Computação.

O público-alvo foram 42 alunos da disciplina de Manutenção de Software da Universidade Federal do Ceará, Campus Quixadá. Para o desenvolvimento do projeto, foram definidas quatro etapas, como seguem descritas abaixo.

- **Etapa 1 - Execução do Treinamento:** Nesta etapa inicial, foram realizados treinamentos com os alunos, expondo os principais *code smells*, bem como técnicas de refatoração. Ainda na primeira etapa, os participantes receberam materiais de apoio, e documentos de suporte para auxiliar na prática da disciplina.

Também elaboramos um material de apoio explicativo, abordando os conceitos e tipos mais relevantes de *code smells*, além das técnicas mais utilizadas para refatoração. Complementarmente, preparamos documentos de suporte, como instruções ilustradas de instalação e uso das ferramentas, descrição geral do trabalho e um vídeo demonstrando o processo de entrega das refatorações via GitHub, para facilitar a análise manual posterior. (Nascimento *et al*, 2025, p. 3).

- **Etapa 2 - Seleção das Ferramentas e *code smells*:** Foram selecionadas as ferramentas de apoio para os estudantes. Segundo Nascimento *et al* (2025) os *code smells* que deveriam vir a ser refatorados pelos estudantes, foram escolhidos de acordo com as ferramentas adotadas.
- **Etapa 3 - Seleção dos projetos:** Neste passo foram selecionados e filtrados projetos em linguagem JAVA, de forma que fossem de fácil entendimento e contendo no mínimo 4 *code smell* diferentes.
- **Etapa 4 - Refatoração dos *code smells*:** Os estudantes deveriam refatorar os problemas propostos, foram disponibilizados canais de apoio via *Discord* para tirar dúvidas. As entregas foram realizadas via repositórios na plataforma *GitHub*.

Os estudantes foram orientados a aplicar a técnica de refatoração mais adequada a cada situação, com base nas recomendações mais comuns para o tipo de code smell identificado. Como material de consulta, foram indicados os sites, que apresentam diversos exemplos práticos de técnicas de refatoração reconhecidas na literatura. Ao final da atividade, os estudantes deveriam entregar uma apresentação em vídeo e um relatório contendo o link para o repositório com as branches criadas. No relatório, precisavam descrever as ocorrências detectadas no início e ao término da refatoração de cada tipo de code smell escolhido (por exemplo, após a refatoração completa de todas as instâncias de data class). (Nascimento *et al*, 2025, p. 5).

Relativo aos resultados, os participantes da metodologia indicaram, em sua maioria terem adquirido as seguintes habilidades: “[...] capacidade de análise e resolução de problemas. Além de também trabalho em equipe, pensamento crítico e gerência de mudanças. Essas habilidades são inerentes à prática de programação.” (Nascimento *et al*, 2025, p. 7). Das dificuldades enfrentadas durante o estudo, grande parte dos participantes citam que entender o código é a parte mais difícil, conforme Nascimento (2025).

#### 2.3.4. Aplicação da metodologia ativa Aprendizagem Baseada em Investigação (ABI)

De acordo com Bordin *et al* (2024), a abordagem ABI é baseada na investigação e foi utilizada durante o ensino da unidade de Manutenção de Software da disciplina de Engenharia de Software II. Sendo empregada para os alunos do curso de Bacharelado em Tecnologias da Informação e Comunicação da Universidade Federal de Santa Catarina (UFSC) no semestre de 2023/1.

A metodologia está alicerçada nas seguintes fases:

- **Orientação:** A professora da disciplina apresentou aos alunos, alguns problemas e oportunidades de pesquisa referente ao Gerenciamento de Sistemas Legados (GSL). Posteriormente ainda na fase de orientação o tópico citado GSL foi detalhado, bem como foi definida a pergunta de pesquisa: “Como as empresas de desenvolvimento de software gerenciam os sistemas legados mantidos por elas?” (Bordin *et al*, 2024, p. 3).
- **Investigação:** Esta etapa ocorreu em um intervalo de 3 semanas. Durante esse período foram realizados planejamento da pesquisa, definindo como público alvo empresas de software brasileiras, elaboração do instrumento de coleta, sendo adotado questionário, lista das empresas candidatas a participar do estudo, envio do questionário para as

empresas que aceitaram participar em conjunto com o termo Termo de Consentimento Livre Esclarecido (TCLE), por fim, os alunos foram orientados a compartilhar os resultados obtidos por meio de artigo, assim como receberam orientações, por parte da docente para analisar e representar os dados obtidos do questionário.

- **Conclusão:** Discussão dos resultados obtidos e apresentação da versão final do artigo.

Os resultados obtidos apontam que todos os grupos de estudantes participantes, consideraram benéfico a parte teórica aplicada a um contexto prático, tornando o conteúdo ministrado esclarecedor, conforme Bordin *et al* (2024).

Além da auto percepção dos estudantes, o estudo conclui que cerca de 90% das empresas participantes, precisam tratar manutenção de código legado, sendo os maiores desafios relatados, tempo dispendido e entendimento do código fonte.

Para além de um relato de experiência de ensino indissociado da pesquisa, este artigo também apresentou os principais resultados de uma pesquisa exploratória conduzida pelos estudantes com empresas desenvolvedoras de software sobre o gerenciamento de sistemas legados mantidos por elas. Os resultados mostram que 90% das empresas contactadas atuam na manutenção de algum sistema legado e os principais de desafios para mantê-los são o tempo dispendido e a dificuldade de entendimento do código. A maioria também afirmou que os sistemas são avaliados em algum momento. As principais decisões tomadas a partir dessa avaliação são modernizar / realizar a reengenharia do software ou substituí-lo. (Bordin *et al*, 2024, p. 10).

### 3. MÉTODO DE PESQUISA

Esta seção mostra o percurso metodológico da pesquisa, abordando desde a fase inicial, levantamento de referências e estudo da bibliografia, até fases finais, como a aplicação da abordagem de ensino de MS teórica e prática com dinâmica simulada cliente-empresa, e a coleta de dados e formação dos resultados, como mostra a Figura 1.

Figura 1 - Método de pesquisa



Fonte: autor, 2025.

A dinâmica proposta no presente trabalho consiste na utilização conjunta de uma abordagem teórica, buscando unir a fundamentação acadêmica com a dinâmica do mercado de trabalho. Na parte teórica, foram aplicadas aulas expositivas tradicionais, trazendo conteúdos como: definições de MS, segundo diversos autores, definição adotada durante a metodologia de ensino, quais os tipos de manutenção de software.

Além da parte teórica tradicional, também utiliza-se a prática simulada, que consiste na divisão do público-alvo em equipes que passam a ser denominadas “empresas”. Cada “empresa” recebe uma documentação fictícia representando solicitações de um “cliente” ilustrativo. Esses pedidos são demandas de manutenção de software: corretivas, adaptativas e

perfectivas, que devem ser implementadas em um software previamente desenvolvido para esse fim. O software criado, dentro do contexto simulado, faz a gestão do empreendimento do “cliente”, e carece de manutenções. Tais manutenções surgem de observações do “cliente”, bem como de modificações do ambiente cujo software está inserido.

Essa metodologia se diferencia das demais apresentadas por unir ambos modelos de ensino: aulas expositivas e aulas práticas, promovendo vivência de situações do mercado de trabalho, dentro de um contexto controlado de ensino de MS. Enquanto as aulas expositivas fornecem a base teórica necessária para a disciplina, a dinâmica simulada cliente-empresa objetiva a aplicação do conhecimento adquirido na etapa anterior. Dessa forma, incentiva a resolução de problemas de maneira semelhante aos do mercado de trabalho. A adoção da metodologia teórica prática simulada cliente-empresa, em comparação às demais metodologias citadas, ocorre devido a experiência do autor do trabalho na área de MS, pois, entre o ambiente acadêmico e profissional, existe uma lacuna entre os conhecimentos necessários para correta realização do trabalho. Devido isto, a metodologia proposta busca, de forma simulada, sanar essa lacuna, buscando trazer uma vivência simulada de ambiente de trabalho, de forma que os alunos, ao ingressarem na área profissional, sintam-se mais preparados para desempenhar suas funções na área de MS.

### **3.1. Planejamento**

Inicialmente, foi realizado o levantamento e estudo de referências bibliográficas com os objetivos de delimitar o tema, objetivos (geral e específicos) do trabalho, tipo de pesquisa a ser realizada, como também, fundamentar teoricamente o presente trabalho. De posse das referências teóricas, a pesquisa foi definida enquanto pesquisa-ação, pois, para alcançar seu objetivo, como explicado na introdução deste trabalho, envolve a intervenção direta no contexto educacional, com aplicação de aulas teóricas e práticas, fazendo uso de uma abordagem para ensino de MS que utiliza a dinâmica simulada cliente-empresa.

A etapa seguinte foi voltada para o desenvolvimento de aulas expositivas e práticas a respeito de manutenção de software, abordando conceitos expostos na fundamentação teórica deste documento. De maneira paralela, foram desenvolvidos recursos de apoio para a ministração das aulas a respeito de MS. O software de estudo simulado para implementação dos tipos de manutenções aqui citados de acordo com os autores adotados, caderno virtual

utilizando a plataforma *Notion*, como mostra a Figura 2. O caderno virtual tem como foco principal concentrar toda a parte teórica abordada a respeito de MS, bem como trazer dicas úteis para a parte prática da metodologia, contendo padrões de *commit*, informações para *download* das ferramentas necessárias para o projeto, dando foco tanto em um material de apoio teórico como prático.

Figura 2 - Caderno Virtual (Notion)



Fonte: autor, 2025.

A Figura 2, na parte esquerda, mostra como está a organização do caderno virtual, material de apoio para a implementação da pesquisa. Cada um dos capítulos aborda assuntos teóricos, de maneira objetiva, buscando ser um manual teórico e prático de manutenção de software. Essa ideia foi inspirada na abordagem adotada pelo professor universitário Sérgio Costa da UFMA, na disciplina de Compiladores no segundo semestre letivo do ano de 2022. Do lado direito da figura, é mostrada a versão atual do software que foi desenvolvido para simular um software que necessita de manutenções, para a dinâmica simulada cliente-empresa. O software foi desenvolvido utilizando linguagem de programação *Python*, por conta da sua utilização nos cursos de BICT e ECP, devido à sua sintaxe de alto nível e paradigma orientado a objetos. O software conta com base de dados desenvolvida em SQLite para gerenciamento e permanência de dados, além de ter sido adotada a arquitetura M.V.C (*model, view, controller*).

De posse desses recursos, foram definidos o público-alvo da pesquisa: alunos da UFMA, cursando a disciplina de ES ministrada aos cursos de BICT e ECP.

### 3.2. Revisão de Materiais

Foram realizadas revisões e correções nos recursos desenvolvidos, bem como nos materiais expositivos preparados, sendo eles:

- Slides para ministração das aulas teóricas em sala de aulas desenvolvidos usando a plataforma *Google* Apresentações;
- Caderno virtual desenvolvido na plataforma *Notion*;
- Estudo de Caso Simulado feito utilizando a plataforma *Google Docs*;
- Software para utilização em cenário simulado cliente empresa, com o objetivo de receber manutenções, desenvolvido em linguagem *Python*.

Tais revisões feitas nos materiais citados acima foram realizadas com o objetivo de garantir a robustez do conteúdo utilizado durante a teoria e prática da metodologia. Por meio desse passo procurou-se assegurar um material de apoio fundamentado teoricamente, que contemple, também, o objetivo de apoiar o público-alvo em eventuais dúvidas durante o desenvolvimento do projeto e na posteridade da sua graduação a respeito da área de manutenção de software.

As revisões ocorreram com base na observação crítica do referencial teórico apresentado no capítulo anterior, sendo conduzida uma comparação direta entre os conceitos abordados e os slides utilizados nas aulas, corrigindo erros de concordância no que diz respeito ao conteúdo abordado e autores escolhidos, tendo como foco minimizar da melhor forma possível erros teóricos e/ou verbais. Essa revisão contínua contribuiu para a consolidação de um material didático mais preciso e alinhado com os fundamentos da área de MS, reduzindo riscos de interpretações equivocadas e promovendo a consistência conceitual, o que julgou-se importante, sendo caráter de preocupação durante todo o desenvolvimento do trabalho, pois, a metodologia está apoiada em ensino teórico e prático, sendo assim, faz-se necessário informações concretas e que tenham caráter de ensino correto e adequado para o público-alvo da pesquisa.

Após a atualização dos slides, foram feitas correções necessárias no caderno virtual, desenvolvido na plataforma *Notion*. Esse material de apoio foi estruturado com base nas apresentações utilizadas em sala de aula, alinhadas ao referencial teórico adotado, em conjunto com conceitos referentes à abordagem prática, que fossem capazes de auxiliar em boas práticas de *commit*, definindo a certo grau uma padronização que poderia ou não ser

adotada pelos alunos durante a execução do projeto. O caderno virtual também contemplou informações a respeito do *download* e instalação das ferramentas necessárias para boa execução do projeto, buscando, dentro dos limites, abranger áreas de dúvida ou dificuldade dos alunos. Dessa forma, garantiu-se a coerência entre os dois materiais: slides e caderno virtual, assegurando a uniformidade das informações e evitando duplicações ou contradições nos conteúdos apresentados.

Além dos materiais teóricos escritos e disponibilizados para os participantes, o software utilizado na prática simulada cliente-empresa passou por revisões e testes de usabilidade. Esses testes foram realizados pelo próprio discente, não havendo uma padronização dos testes ou testes unitários, dessa maneira, os testes de usabilidade consistiram em executar o software e manualmente testar as funcionalidades. A etapa de teste permitiu a identificação de comportamentos inesperados no sistema. A partir disso, foram feitas correções no software, com o intuito de garantir que o ambiente simulado funcionasse adequadamente como recurso didático e contribuísse efetivamente para a aprendizagem prática dos conceitos de manutenção de software.

A aplicação da pesquisa foi realizada por meio da ministração de quatro horários de aula destinados ao público-alvo na disciplina de ES destinada aos cursos BICT e ECP no período de 2025.1 na UFMA. As aulas ocorreram de forma presencial e usaram a abordagem teórica e prática, com apoio dos recursos desenvolvidos e detalhados neste capítulo.

A parte teórica contou com apresentações em slides divididas em quatro apresentações:

- **Sobre o discente:** breve apresentação do pesquisador (discente), incluindo suas formações, qualificações e aspectos pessoais relevantes, buscando gerar uma identificação entre o público alvo e o pesquisador.
- **Apresentação da metodologia:** exposição do formato adotado para o desenvolvimento da atividade, explicando como os conteúdos relacionados à MS seriam abordados durante os encontros. Esta apresentação elucidou aos participantes como seria o decorrer do projeto e o funcionamento da dinâmica simulada cliente-empresa.
- **Manutenção de software:** apresentação dos fundamentos teóricos da área, com definição de manutenção de software, discussão da importância da temática e detalhamento dos três tipos de manutenção (corretiva, adaptativa e perfectiva),

baseando-se em autores selecionados. Também foram incluídos exemplos ilustrativos para cada tipo. Este slide teve como objetivo revisar e reforçar conceitos já abordados na disciplina, para que assim os alunos pudessem tirar dúvidas a respeito do tema já abordado.

- **O código base:** foi apresentado o sistema (*software*) desenvolvido na linguagem de programação *Python* para a prática simulada de manutenção de software. Foram explicados seu funcionamento geral, arquitetura, sendo adotada a arquitetura M.V.C (*model, view, controller*), linguagem de programação utilizada, forma de armazenamento dos dados (banco relacional *SQLite*) e aspectos relevantes para a compreensão do código.

Finalizadas as apresentações de slides e apresentação teórica da metodologia, por meio de aulas expositivas em formato presencial em laboratório de computação da UFMA, foi apresentado o documento intitulado Estudo de Caso Simulado, que descreve o cenário simulado, contendo informações sobre a “empresa” assim como as solicitações do “cliente”, ambos fictícios. Os participantes se dividiram em grupos de no máximo 4 integrantes, para darem início à parte prática da metodologia de ensino. Como abordado anteriormente, os alunos foram incentivados e desafiados a encontrarem no código disponibilizado, as manutenções solicitadas pelo cliente fictício.

Como etapa final, foi realizada a obtenção dos dados qualitativos e quantitativos da pesquisa, que se deram por meio de dois questionários aplicados de forma voluntária e anônima junto aos participantes. O primeiro questionário foi respondido antes da parte prática da metodologia simulada cliente-empresa e teve como foco a caracterização dos discentes, a fim de categorizar o público-alvo com relação aos seus conhecimentos a respeito de manutenções de software e prática no uso de ferramentas para realização do projeto (*GitHub, Python, SQLite, Git*, dentre outras). O segundo questionário, aplicado ao final das atividades, teve como objetivo avaliar a percepção dos alunos quanto à metodologia, sua efetividade no processo de aprendizagem e demais aspectos envolvidos na execução do projeto, para fundamentar os resultados do presente trabalho, levando métricas quanto ao aprendizado de MS e demais conhecimentos adquiridos durante a intervenção. É válido ressaltar que tanto o questionário inicial, quanto o questionário de avaliação, dispunham de TCLE assegurando que todos os participantes estavam de acordo com os termos de pesquisa.

### 3.3. Estudo de Caso Simulado

Como parte da prática pedagógica abordada no presente trabalho, foi desenvolvido um documento utilizando a plataforma *Google Docs* intitulado: Estudo de Caso Simulado. O propósito desse documento é disponibilizar um cenário contextual que simula um cenário real do mercado de trabalho.

Este documento foi elaborado tendo como base uma empresa fictícia, assim como um cliente fictício, que é usuário do software desenvolvido. O documento descreve a empresa, o software desenvolvido por ela, como também possui parte específica para apontar as interações do cliente, tais interações com o cliente são o ponto chave do documento, pois, é por meio dessas interações que os alunos irão se nortear para realizar as manutenções necessárias no software proposto.

A organização do documento foi realizada da seguinte forma:

- **Apresentação da empresa simulada:** contextualização do ambiente de trabalho, incluindo nome da empresa, área de atuação, perfil do sistema em uso e breve histórico da aplicação mantida, como mostrado na Figura 3.
- **Contato com o cliente:** a Figura 3 mostra a seção responsável por descrever o conjunto de solicitações e mensagens simuladas de um cliente fictício, descrevendo problemas, necessidades de atualização ou melhorias, com base nas quais os alunos deveriam interpretar e realizar as devidas manutenções no código-fonte.
- **Tabela de registro de manutenções:** seção destinada ao preenchimento pelos discentes, onde deveriam classificar cada intervenção como manutenção corretiva, adaptativa ou perfectiva, além de descrever a ação tomada para resolução do problema, como apontado na Figura 5.

Figura 3 - Estudo de Caso Simulado: apresentação da empresa simulada

## **Estudo de Caso Simulado: Análise e Soluções de Manutenções no Sistema “Pizza Mais”**

Pizza Mais: O Desafio Real de um Sistema em Evolução

### **Descrição Geral do Cenário Simulado**

A **Criando Sonhos LTDA**, empresa do ramo de tecnologia com mais de dois anos de atuação no mercado, desenvolveu e implantou recentemente um novo sistema de gerenciamento para o restaurante do **senhor Natanael**, um comerciante local proprietário da pizzeria **Pizza Mais**.

O sistema foi desenvolvido com o objetivo de facilitar a rotina do estabelecimento, permitindo o controle de pedidos, atendimento e organização interna. No entanto, após as primeiras semanas de uso, o cliente relatou algumas dificuldades e pontos de melhoria que afetam a experiência dos atendentes e a eficiência do serviço.

Buscando garantir a qualidade do produto e aprimorar o atendimento ao cliente, a Criando Sonhos lançou um **desafio voltado a novos talentos da área de desenvolvimento de software**. A proposta é encontrar profissionais com boa capacidade técnica e habilidades interpessoais, que sejam capazes de identificar e corrigir falhas, além de propor melhorias de forma clara, eficiente e empática.

No desafio, as equipes participantes assumirão o papel de **desenvolvedores da Criando Sonhos**, e deverão:

Fonte: autor, 2025.

Figura 4 - Estudo de Caso Simulado: contato com o cliente

### **3. Interações com o Cliente**

#### **Diálogos do Cliente – (Cadastro de Pedido) - 02/12/2024**

*Oi, pessoal! Seguinte, queria pedir uma ajuda sobre a tela de fazer pedidos...*

##### **1. Menu confuso na hora de cadastrar um novo pedido:**

*Quando a gente vai cadastrar um novo pedido, o menu que aparece está meio confuso, sabe? O pessoal aqui tem tido dificuldade de entender o que fazer, principalmente os novos. Será que dá pra dar uma organizada nele, deixar mais claro?*

##### **2. Confirmação de cadastro falhando:**

*E outra coisa: às vezes, mesmo digitando certinho “sim” pra confirmar o pedido, o sistema não aceita e não faz nada. A gente acaba tentando de novo e não entende por que não funcionou. Isso acontece de vez em quando, mas atrapalha.*

Fonte: autor, 2025.

Figura 5 - Estudo de Caso Simulado: tabela de manutenções

 **Tabela de Manutenções – Sistema "Pizza Mais" -  
Necessário preencher a tabela**

| Nº | Descrição do Problema ou Solicitação                                                           | Manutenção | Ação Esperada                                                                                     |
|----|------------------------------------------------------------------------------------------------|------------|---------------------------------------------------------------------------------------------------|
| 1  |                                                                                                |            | Melhorar o layout do menu principal para torná-lo mais claro, organizado e visualmente acessível. |
| 2  | Entrada para confirmar o cadastro de novo pedido não funciona em alguns casos.                 |            |                                                                                                   |
| 3  | Ao adicionar novo item ao pedido, a confirmação falha ocasionalmente.                          | Corretiva  | Revisar a lógica de confirmação de novo item e corrigir falhas de reconhecimento ou fluxo.        |
| 4  |                                                                                                |            |                                                                                                   |
| 5  | O sistema não apresenta nenhuma resposta na opção inicial: Cadastrar pedido (y-Sim, n-Nao).    |            | Adicionar validação e exibir mensagens informando o erro ao usuário.                              |
| 6  | Ao tentar atualizar o status de um pedido, nenhuma ação ocorre (sistema ignora o comando).     | Corretiva  |                                                                                                   |
| 7  | Cliente solicita a criação de uma nova tela para cadastrar novos itens no menu do restaurante. |            | Implementar uma view de cadastro de itens, integrando com os controllers e models já existentes.  |

Fonte: autor, 2025.

A Figura 5, tem por finalidade exemplificar como foram tratadas cada uma das manutenções no presente trabalho. Os participantes tinham como objetivo observar as interações do cliente e dessa forma preencher corretamente as lacunas da tabela, de forma a responderem de maneira correta, o tipo de manutenção a ser realizada para cada uma das interações apresentadas. Desta maneira, cobrindo assim os 3 tipos de manutenções apresentadas de maneira teórica. A tabela foi disposta com espaços em branco e espaços preenchidos, de forma que os alunos pudessem observar exemplos de como preencher, e à partir de tais exemplos completar os demais espaços.

### 3.4. Software para ambiente simulado cliente-empresa

Um dos materiais produzidos consiste no software para simulação do ambiente cliente-empresa. O software consiste em uma aplicação construída em linguagem de

programação *Python*, usando especificamente a versão 3.10 do *Python*. O objetivo desse software é ser o código fonte para que o público-alvo aplique as manutenções solicitadas pelo cliente fictício que foi criado no Estudo de Caso Simulado.

A arquitetura utilizada, devido à familiaridade do público alvo, em decorrência da disciplina de ES, foi M.V.C (*model, view, controller*), a permanência de dados consistiu em uso de banco de dados relacional *SQLite*. A respeito de bibliotecas externas foi necessário a instalação da *lib reportlab* para gerenciamento de documentos em formato PDF.

O script possui interface gráfica executada diretamente do terminal, como representado na figura Figura 6, trazendo uma interface que busca objetividade.

Figura 5 - Software Pizza Mais

```
Bem-vindo ao software Pizza Mais
-Criando Sonhos-
Estabelecimento: Pizza Ciclano
"Seus sonhos tem formato e borda"
-----

1 - Cadastrar
2 - Pesquisar
3 - Relatorio
4 - Inserir Itens Menu
5 - Encerrar
Digite: 1
-----Menu-----

[(1, 'calabresa', 35.0, 'pizza', 'calabresa, orégano, queijo, cebola'), (2, 'frango-com-catupiry', 40
.0, 'pizza', 'frango desfiado, queijo catupiry, azeitona'), (3, 'Mussarela', 45.38, 'Pizza', 'queijo
mussarela, oregano e cebola')]

Cadastrar pedido (y-Sim, n-Nao): n
Voltando ao Menu inicial

1 - Cadastrar
2 - Pesquisar
3 - Relatorio
4 - Inserir Itens Menu
5 - Encerrar
Digite: 2
-----Pesquisar Pedido-----
Unico-1
Todos-2
Atualizar Estado-3
Digite: █
```

Fonte: autor, 2025.

Sendo um software para a gestão de um pequeno empreendimento, neste caso um estabelecimento fictício, denominado Pizza Ciclano, criado exclusivamente para fins de simulação, conta com as seguintes funções: Cadastro de pedidos; Pesquisar um ou todos os pedidos; Relatório de faturamento do estabelecimento; Inserção de itens no menu.

As funcionalidades citadas acima foram construídas de forma a ser necessário aplicar manutenções dos tipos: corretiva, adaptativa ou perfectiva, para o correto funcionamento do software. As manutenções a serem efetuadas pelos alunos foram descritas no documento Estudo de Caso Simulado por meio das interações com os clientes.

### **3.5. Aulas piloto**

Antes da aplicação da metodologia em sala de aula com os discentes, foi realizada uma aula piloto, contando com a participação do orientador do presente trabalho, bem como dois participantes do programa de doutorado da UFMA. A aula consistiu em uma breve apresentação de todos os materiais dispostos, dando mais ênfase no código fonte proposto e no documento Estudo de Caso Simulado, para que dessa forma fossem realizadas revisões nos materiais, para só então realizar a aplicação da metodologia junto aos participantes da disciplina de ES.

### **3.6. Execução da metodologia**

A metodologia teórica e prática simulada foi aplicada em sala de aula, por meio de aulas exclusivamente presenciais em laboratório da UFMA. Estas aulas foram ministradas pelo autor do trabalho, com a supervisão do docente da área. As aulas práticas tiveram como foco revisar e reforçar conceitos a respeito de manutenção de software, conceitos abordados no capítulo de desenvolvimento. As aulas expositivas tradicionais contaram com o auxílio de apresentações em slides, abrindo espaço para que os alunos tirassem dúvidas, fizessem apontamentos, contribuições e compartilhassem sua experiência e conhecimento sobre a área de MS.

Posteriormente as aulas práticas, que ocuparam maior parte das duas semanas de aulas, também realizadas em laboratório da UFMA, tiveram como objetivo tirar dúvidas dos participantes, auxiliar em alguma das manutenções. Durante toda a metodologia, tanto aulas práticas quanto teóricas, foram acompanhadas pelo autor do trabalho sob supervisão do docente da disciplina, em conjunto com a participação de dois alunos de doutorado também da UFMA.

Na última aula da metodologia, os alunos fizeram apresentações a respeito das manutenções feitas, foram questionados com relação a dúvidas sobre o código, banco de dados relacional e quais as maiores dificuldades encontradas. As apresentações dos participantes se deram também de maneira presencial em laboratório, onde exibiram o código

fonte, desta feita com as manutenções efetuadas, mostrando como pensaram para resolver os problemas propostos.

#### **4. RESULTADOS E DISCUSSÕES**

Neste capítulo são apresentados os resultados obtidos à partir da aplicação da metodologia de ensino descrita no presente documento, que combina teoria e prática no ensino de manutenção de software, bem como discussões a respeito dos resultados obtidos. A parte prática foi baseada em uma simulação de ambiente cliente-empresa, na qual os alunos participantes assumiram o papel de uma empresa fictícia responsável por atender às demandas de um cliente fictício. A coleta de dados foi realizada por meio de dois questionários, um de caracterização e outro avaliativo, utilizando a plataforma *Google Forms*. O primeiro questionário foi aplicado antes da implementação da metodologia e teve como objetivo mapear o perfil e o nível de familiaridade dos estudantes com os conceitos e práticas da área de MS. O segundo questionário foi aplicado após a intervenção, visando avaliar a percepção dos alunos sobre a metodologia e seu impacto na aprendizagem a respeito do tema abordado.

##### **4.1. Caracterização inicial dos participantes**

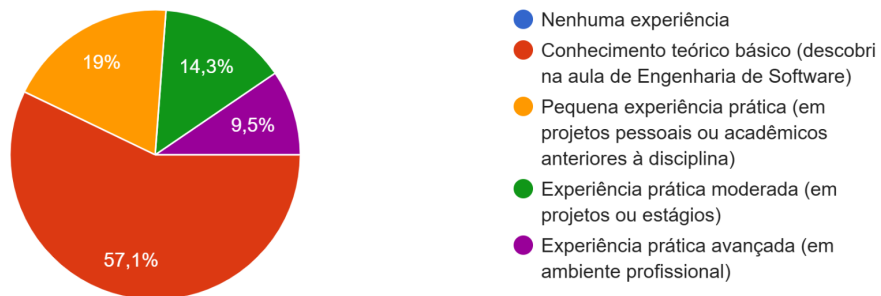
O primeiro questionário de caracterização foi aplicado no início do projeto e contou com a participação de 21 alunos dos cursos de BICT e ECP da UFMA. O objetivo desse questionário inicial foi compreender o perfil dos participantes em relação ao conhecimento prévio teórico e à experiência prática na área de manutenção de software.

Como mostra a Figura 7, a maioria dos estudantes (57,1%) possuíam conhecimento teórico básico a respeito de manutenção de software no início do projeto, adquirido nas aulas anteriores da disciplina de ES. Um total de 19% relatou ter uma pequena experiência prática, proveniente de projetos pessoais ou acadêmicos anteriores à disciplina. Outra parcela de 14,3% dos respondentes afirmaram possuir experiência moderada, geralmente adquirida por meio de projetos extracurriculares mais estruturados ou estágios supervisionados. Apenas 9,5% dos participantes declararam ter um nível avançado de conhecimento na área de MS.

Figura 7 - Gráfico de experiência com manutenção de software

1 - Qual seu nível de experiência com manutenção de software? (Marque uma opção)

21 respostas



Fonte: autor, 2025

Esses dados reforçam a percepção de que, apesar de os estudantes possuírem uma base conceitual inicial, há uma carência significativa de experiências práticas em contextos que simulem as demandas reais de MS. Esse cenário justifica a adoção de uma abordagem metodológica que integre teoria e prática, a fim de promover uma aprendizagem significativa e alinhada às exigências do mercado.

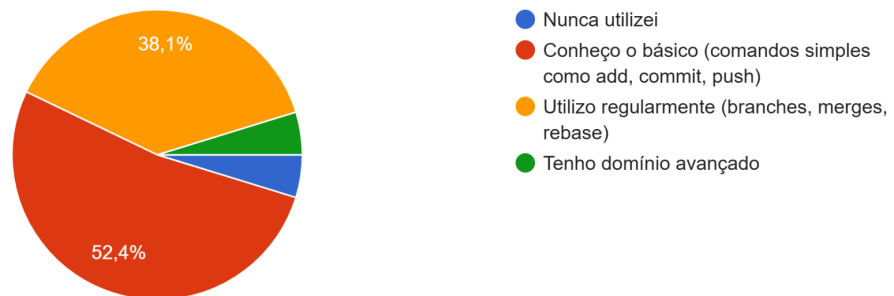
Além do conhecimento prévio sobre manutenção de software, o questionário inicial também investigou o nível de familiaridade dos alunos com ferramentas e tecnologias comumente associadas à prática da área, tais como o sistema de versionamento *Git*, a plataforma *GitHub*, a linguagem de programação *Python* e bancos de dados relacionais. Tecnologias que além de utilizadas na área de MS, foram úteis para aplicação da metodologia de prática simulada cliente-empresa.

Em relação à ferramenta *Git*, observou-se que grande parte dos participantes (52,4%) possuíam conhecimento básico, que consiste em comandos simples, tais como: *add*, *commit*, *push*, pode-se observar esse percentual por meio da Figura 8. Dessa forma é possível concluir que grande parte dos participantes não dispunham de aprofundamento nas funcionalidades mais avançadas da ferramenta, como resolução de conflitos, *branching* e *merge*, questão que fica mais nítida na seção de dificuldades encontradas pelos participantes da pesquisa.

Figura 8 - Gráfico de familiaridade com a ferramenta *Git*

2 - Qual o seu nível de familiaridade com a ferramenta Git?

21 respostas



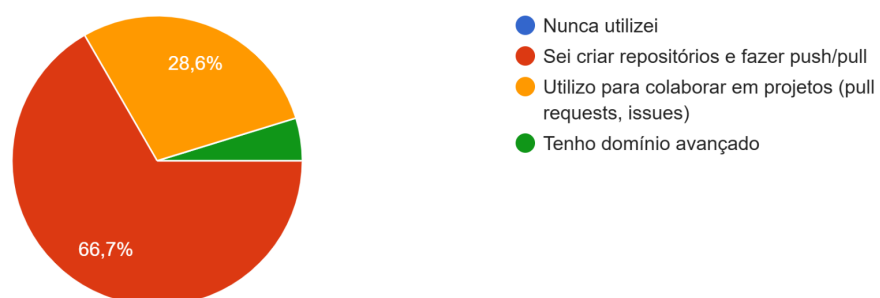
Fonte: autor, 2025.

Sobre o *GitHub*, o cenário foi semelhante à ferramenta citada anteriormente, como apontado na Figura 9. A maior parte dos respondentes (66,7%) declarou ter familiaridade em criar repositórios e realizar operações como *push/pull*. Do total de participantes, 28,6% afirmaram utilizar para colaborar em projetos. Por fim, apenas 4,7% afirmaram ter domínio avançado sobre a plataforma.

Figura 9 - Gráfico de familiaridade com a ferramenta *GitHub*

3 - Qual o seu nível de familiaridade com a ferramenta GitHub?

21 respostas



Fonte: autor, 2025.

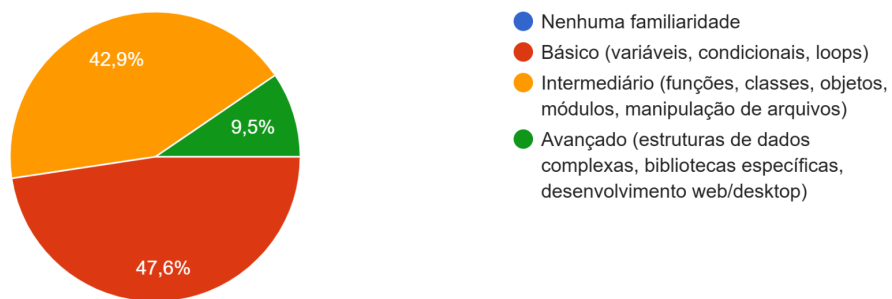
Em relação à linguagem de programação Python, utilizada para o desenvolvimento do projeto, houve uma divisão parcialmente equilibrada entre dois grandes grupos. O primeiro grupo, com 47,6% dos participantes afirmaram ter conhecimento básico a respeito da

linguagem de programação. Outra grande parte dos participantes da pesquisa, 42,9%, julgaram possuir conhecimento intermediário, conhecendo funções, módulos, classes, objetos e manipulação de arquivos. Os demais 9,5%, informaram que possuem conhecimento avançado a respeito da linguagem de programação Python (estruturas de dados complexas, bibliotecas específicas, desenvolvimento web/desktop), como mostrado na Figura 10.

Figura 10 - Gráfico de familiaridade com a linguagem de programação *Python*

4 - Qual o seu grau de familiaridade com a linguagem de programação Python?

21 respostas



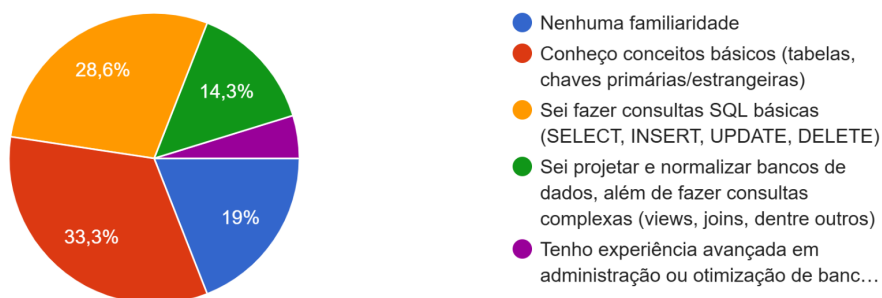
Fonte: autor, 2025.

Por fim, no que tange ao uso de bancos de dados relacionais, como *PostgreSQL*, *SQL Server*, *MySQL*, *SQLite* observa-se uma divisão mais equilibrada e igualitária entre os participantes. Uma porcentagem de 33,3% dos respondentes possui conhecimentos básicos (tabelas, chaves primárias/estrangeiras); 28,6% sabem realizar consultas SQL básicas (*SELECT*, *INSERT*, *UPDATE*, *DELETE*); 19% não possuíam qualquer familiaridade com banco de dados relacional; 14,3% sabem projetar e aplicar regras de normalização em banco de dados, além de serem capazes de realizar consultas complexas (views, joins, dentre outros). Por fim 4,8% possuíam experiência avançada com relação a banco de dados relacional, sendo capazes de aplicar otimizações, tais dados podem ser observados abaixo na Figura 11.

Figura 11 - Gráfico de familiaridade sobre banco de dados relacionais

5 - Qual o seu grau de familiaridade com banco de dados relacionais (ex.: SQLite, SQL Server, MySQL, PostgreSQL, etc.) ?

21 respostas



Fonte: autor, 2025.

Esses dados revelam que, embora os alunos apresentem certo contato inicial com as tecnologias investigadas, há uma lacuna prática significativa, especialmente no uso dessas ferramentas em ambientes simulados de trabalho colaborativo. Essa constatação, reforça a necessidade de práticas pedagógicas que integrem o uso real dessas tecnologias em situações-problema, mais próximas da realidade profissional.

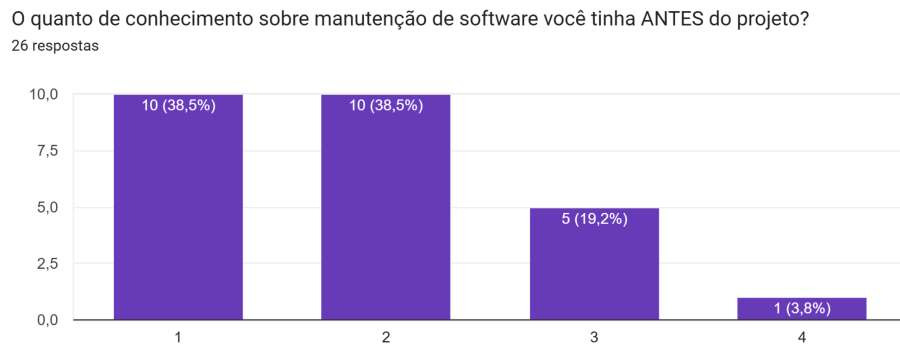
#### 4.2. Percepção pós-aplicação da metodologia

Após a realização da prática com abordagem teórico-prática simulada cliente-empresa, foi aplicado um segundo questionário com o objetivo de avaliar a percepção dos estudantes quanto à eficácia da metodologia adotada e seu impacto no processo de aprendizagem. O questionário foi respondido por 26 discentes dos cursos de Bacharelado Interdisciplinar em Ciência e Tecnologia e Engenharia da Computação, ambos da Universidade Federal do Maranhão. Em relação ao primeiro questionário, com 21 participantes, tivemos 5 participantes a mais nesse segundo momento.

Com base nas respostas obtidas, observou-se que a maioria dos participantes possuíam nenhum ou pouco conhecimento prévio na área de manutenção de software, conforme aferido por uma escala Likert de quatro pontos. Vale destacar que todas as escalas Likert utilizadas nesta pesquisa adotaram quatro níveis, evitando, propositalmente, a existência de um ponto de neutralidade nas respostas, o que favorece uma análise mais precisa da percepção dos

estudantes. Na Figura 12 é possível observar a percepção dos alunos a respeito do seu nível de conhecimento sobre manutenção de software antes da aplicação do presente projeto.

Figura 12 - Nível de conhecimento sobre manutenção de software antes do projeto



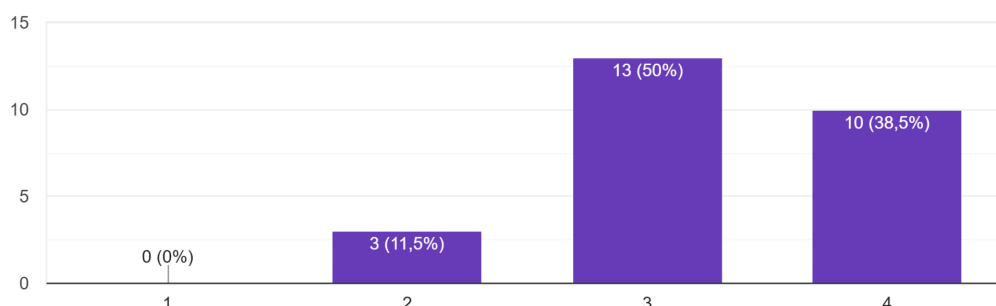
Fonte: autor, 2025.

Por meio da comparação entre o cenário inicial (Figura 12) e o cenário após a aplicação da metodologia teórico prática simulada cliente-empresa, é possível observar que no último cenário (Figura 13), 88,5% dos participantes afirmam estar em um nível de conhecimento 3 e 4, sendo 4 relativo a ter adquirido conhecimento sobre o tema. Dessa forma é possível fundamentar que a metodologia proposta contribuiu de maneira positiva para o aprendizado dos estudantes a respeito da área de MS. Observa-se também na Figura 13, que no contexto pós metodologia, nenhum participante (0%) julga ter ou estar em um nível de conhecimento 1 quanto a área de manutenção de software, sendo esse nível relativo a pouco ou nenhum conhecimento a respeito da área.

Figura 13 - Nível de conhecimento sobre manutenção de software após o projeto

O quanto de conhecimento sobre manutenção de software você adquiriu DURANTE o projeto?

26 respostas



Fonte: autor, 2025.

O crescimento acentuado no número de estudantes que declararam ter adquirido conhecimento durante a prática evidencia a relevância de uma abordagem de ensino que vá além das aulas puramente expositivas, incorporando atividades práticas e contextualizadas. Esses dados reforçam a ideia tratada na introdução, de que metodologias exclusivamente expositivas teóricas, possuem a tendência a apresentar baixa eficiência quanto ao processo de ensino-aprendizagem, conforme cita Prikladnicki (1999).

Outro ponto de investigação no questionário pós projeto, foi o julgamento dos alunos quanto à sua capacidade em identificar os tipos de manutenção de software durante a execução do projeto. Mais da metade dos participantes, 53,8% afirmaram que foram capazes de identificar as manutenções no software proposto, porém, com algum grau de dificuldade. Os outros 46,2% afirmam que foram capazes de identificar as manutenções com segurança. Sendo assim, observa-se que nenhuma parcela dos participantes não foi capaz de identificar as manutenções, o que reforça a ideia de que a metodologia colaborou positivamente para a construção do conhecimento a respeito dos tipos de manutenção de software e como identificá-los, como apontado na Figura 14.

Figura 14 - Gráfico de identificação dos tipos de manutenção de software

Você foi capaz de identificar os tipos de manutenção de software (corretiva, adaptativa, perfectiva) em todos os casos propostos?

26 respostas



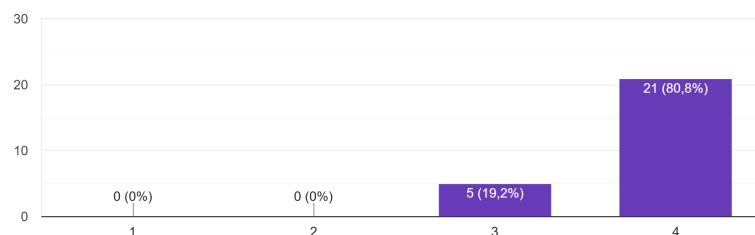
Fonte: autor, 2025.

Além disso, os alunos avaliaram a utilidade do documento Estudo de Caso Simulado, disponibilizado como material de apoio durante a realização da parte prática. De maneira geral, o documento foi considerado útil, como aponta a Figura 15, tanto em termos de conteúdo quanto de organização estrutural. A maioria dos participantes (80,8%) relatou que o material facilitou o processo de identificação das manutenções no código-fonte analisado. Quando questionados a respeito da organização do documento 61,5% dos participantes do estudo, indicam que a organização do documento contribuiu de forma acentuada para identificação das manutenções a serem realizadas no software simulado, como aponta a Figura 16.

Figura 15 - Relevância do documento Estudo de Caso Simulado A

O quanto você considerou útil o documento Estudo de Caso Simulado disponibilizado durante a realização da parte prática do projeto para a identificação das manutenções durante o projeto?

26 respostas

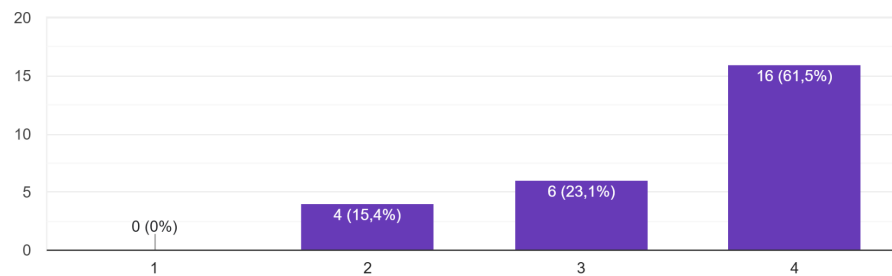


Fonte: autor, 2025.

Figura 16 - Relevância do documento Estudo de Caso Simulado B

Ainda sobre o Estudo de Caso Simulado. O quanto você julga que a organização do documento auxiliou a identificar as manutenções de software?

26 respostas



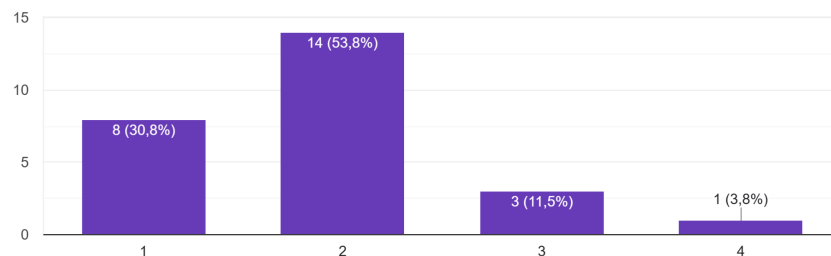
Fonte: autor, 2025.

Quando questionados sobre a dificuldade que teriam em realizar essa tarefa sem o documento de suporte de Estudo de Caso, apresentado no capítulo anterior, muitos indicaram que a ausência dele teria tornado o processo significativamente mais desafiador (53,8%), como observado na Figura 17. Esta afirmação reforça a importância de materiais de apoio elaborados em metodologias ativas de ensino.

Figura 17 - Relevância do documento Estudo de Caso Simulado C

Sem o documento citado na pergunta anterior, o quanto julga que seria possível identificar as manutenções presentes no código durante o projeto?

26 respostas

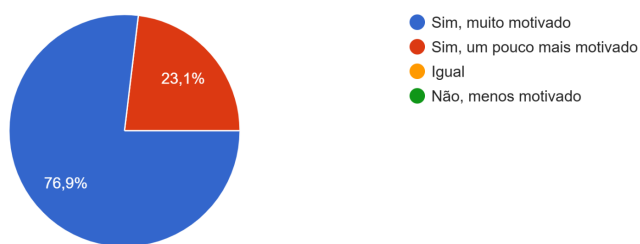


Fonte: autor, 2025.

Outro aspecto abordado foi a motivação dos estudantes em relação ao formato adotado. Quando comparado às aulas expositivas tradicionais, o modelo teórico-prático simulado cliente-empresa foi amplamente apontado como mais estimulante e envolvente, os percentuais podem ser observados com mais detalhes na Figura 18. Todos os participantes afirmam que se sentiram mais motivados a estudar a área de manutenção de software, sendo que 76,9% muito motivados e 23,1% um pouco mais motivados. Essas informações confirmam que experiências práticas dentro de um contexto simulado, são propensas a favorecer a motivação e engajamento dos participantes.

Figura 18 - Motivação para estudo de MS após aplicação da metodologia

Você se sentiu mais motivado a aprender/estudar com esse formato teórico e prático simulado cliente/empresa em comparação a aulas expositivas tradicionais?  
26 respostas



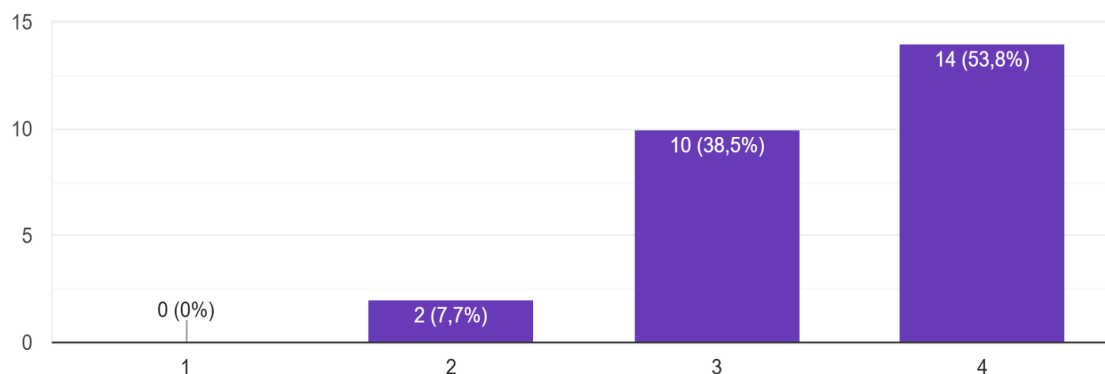
Fonte: autor, 2025.

Por fim, foi avaliado o nível de preparação percebido pelos estudantes para lidar com manutenção de software após a participação na metodologia de ensino proposta (prática simulada cliente-empresa). Os resultados apontam que a maior parte dos respondentes sente-se mais preparada para atuar na área (53,8%), como apontado na Figura 19. Esse sentimento de preparação não apenas valida a proposta metodológica, como também evidencia sua aderência às necessidades formativas dos futuros profissionais de tecnologia.

Figura 19 - Nível de preparo para lidar com MS após a parte prática da metodologia

Você se sente mais preparado para lidar com manutenção de software após essa prática?

26 respostas



Fonte: autor, 2025.

#### 4.2.1. Ferramentas tecnológicas utilizadas na aplicação da metodologia

Foi investigado se houve alguma ferramenta ou tecnologia que representou um desafio significativo durante a execução do projeto, dentre as tecnologias utilizadas ou outras tecnologias que os alunos tivessem utilizado durante a prática simulada. Embora a maioria dos participantes (46,6%) tenha relatado nenhuma dificuldade com relação a ferramentas utilizadas como apontado na Tabela 1, alguns discentes informaram dificuldades específicas, sobretudo com ferramentas de controle de versão (como o *Git* e *GitHub*) que tendo seus percentuais somados, apresentam um montante de 23%, uma parcela de 7,6% dos participantes informaram dificuldades com a linguagem de programação *Python*, 7,6% tiveram dificuldade no uso do banco de dados relacional *SQLite*. Por fim, 15,2% dos participantes tiveram dificuldades com outras ferramentas, tais como: ambiente para codificação, *streamlit*, *Copilot*, dentre outras.

Tabela 1 - Desafios nas ferramentas tecnológicas utilizadas na metodologia

| FERRAMENTA         | PERCENTUAL |
|--------------------|------------|
| Git                | 3,8%       |
| GitHub             | 19,2%      |
| Python             | 7,6%       |
| SQLite             | 7,6%       |
| Outros             | 15,2%      |
| Nenhuma Ferramenta | 46,6%      |

Fonte: autor, 2025.

#### 4.3. Dificuldades encontradas pelos participantes

Além das questões objetivas e de múltipla escolha, os participantes também tiveram espaço para relatar, de forma voluntária e anônima, as principais dificuldades enfrentadas durante a realização da atividade. As respostas revelaram uma diversidade de desafios, muitos dos quais refletem obstáculos comuns no ambiente profissional de desenvolvimento de software.

Entre as dificuldades citadas, destacam-se questões relacionadas à programação em si, à compreensão do código legado, especialmente devido à ausência de uma documentação adequada, e à integração entre lógica de controle (*controllers*) e banco de dados. Também foram mencionadas dificuldades técnicas específicas, como o uso do SQLite, a criação de novas *views* e o versionamento de código com *Git* e *GitHub*, incluindo conflitos frequentes durante os merges, essas informações estão dispostas na Tabela 2. Vale destacar que alguns participantes afirmaram não ter enfrentado dificuldades significativas, o que indica que, para parte dos discentes, os materiais de apoio, à condução da atividade e os recursos didáticos utilizados foram suficientes para suprir a necessidade desse ensino prático.

Tabela 2 - Dificuldades informadas pelos participantes

| Participante | Dificuldade  | Comentário                                                                |
|--------------|--------------|---------------------------------------------------------------------------|
| F10          | Código fonte | “Entender o quanto de modificação poderiam ser feitas no código original” |

|     |                                                              |                                                                                                                 |
|-----|--------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|
| F11 | Gestão de equipe                                             | “A maior foi a gestão de pessoas para realizar as tarefas dado análises e interpretações diferentes”            |
| F14 | Integração de <i>controllers</i> e banco de dados relacional | “Entender a lógica dos controllers com os elementos do banco de dados.”                                         |
| F17 | Versionamento de código ( <i>Git/GitHub</i> )                | “Foi o versionamento do código, pois durante a manutenção ocorriam conflitos frequentes ao realizar os merges.” |
| F18 | Criação de nova <i>view</i> (manutenção perfectiva)          | “A criação de uma nova view, ou seja, a criação do zero de uma nova janela de visualização para o usuário”      |

Fonte: autor, 2025.

#### 4.4. Experiência dos participantes quanto à metodologia simulada cliente-empresa

Como parte das perguntas argumentativas, os participantes tiveram espaço para expressar, de maneira opcional e anônima, sua experiência quanto à metodologia, bem como seu aprendizado. Abaixo seguem tais informações, dispostas na Tabela 3, quanto ao que aprenderam e sugestões e/ou críticas a respeito da metodologia.

Tabela 3 - Aprendizado e sugestão dos participantes

| Participante | Tipo<br>(Aprendizado Sugestão) | Comentário                                                                                                                                                         |
|--------------|--------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| F1           | Aprendizado                    | “Tudo no geral, desde a parte que envolve manutenção até a parte da programação em si.”                                                                            |
| F5           | Aprendizado                    | “Guia de boas práticas de versionamento de código. O material do notion foi muito útil e agregou muito. Agora os meus próximos commits serão todos organizados. Só |

|     |             |                                                                                                                                                                                                                                                                                                                                            |
|-----|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|     |             | tenho a agradecer por ter feito essa disciplina.”                                                                                                                                                                                                                                                                                          |
| F14 | Aprendizado | “Divisão de tarefas e identificação dos tipos de manutenção”                                                                                                                                                                                                                                                                               |
| F11 | Sugestão    | “Mudar a aplicação pra uma API, e desenvolver uma interface gráfica, para, dessa forma, os alunos ainda teriam um ambiente mais simplificado, mas entenderiam sobre integração back - front”                                                                                                                                               |
| F5  | Sugestão    | “A tarefa foi tão engajadora que a sensação era de que gostaria de fazer mais. Minha sugestão talvez seria a respeito de disponibilizar uma atividade extra aplicando outras partes de desenvolvimento de software (autenticação), deploy , conexão com API ou aplicação de microsserviços, mas claramente dentro da grade da disciplina.” |
| F3  | Sugestão    | “A metodologia foi boa. Colocar em prática o que foi estudado é importante e contribui para o aprendizado. Creio que usar o Github desktop é ótimo, ajuda bastante para o iniciante.”                                                                                                                                                      |

Fonte: autor, 2025.

Em síntese, os dados coletados nesta etapa apontam que a metodologia aplicada, ao simular situações reais de manutenção de software em um ambiente simulado cliente-empresa, proporcionou aos alunos participantes da intervenção uma experiência rica em aspectos técnicos, desafiadora e formadora de conhecimento.

#### **4.5. Propostas de evolução para a metodologia**

Com base no exposto na Tabela 3, é válido levantar propostas de melhorias para a presente metodologia. Como exposto por alguns participantes, a união entre um ambiente mais simplificado, no qual seja possível trabalhar e observar de maneira mais clara, a MS entre um ambiente de *front* e *backend*, torna-se uma proposta evolutiva possível de implementação, de forma que os alunos fossem capazes de colocar em prática atividades de manutenção ainda mais próximas de um ambiente real de desenvolvimento.

A aplicação da metodologia de maneira mais linear, sendo trabalhado por semana um novo tipo de manutenção, dando aos alunos um suporte quanto ao que deve ser implementado dentro do software. Esta proposta linear evolutiva pode fortalecer a metodologia aqui proposta quando colocada em uma perspectiva, o aumento gradual de nível de dificuldade quanto às manutenções propostas, dessa forma seriam trabalhados exemplos práticos com os alunos e posteriormente lançado o desafio de solucionarem manutenções de mesmo tipo com base nos exemplos expostos.

Por fim, a extensão da metodologia, de forma que fosse capaz de englobar mais áreas, como integração com APIs, *deploy* da aplicação, aliada a aplicação da mesma durante um período de tempo superior a duas semanas de aula, pode impactar de maneira positiva na percepção dos alunos quanto à metodologia e sua importância para a construção do conhecimento sobre a área de MS. Desta forma há espaço para refinamento, aperfeiçoamento e novas propostas, dentro da metodologia e do tema de MS, que são benéficos para contribuir positivamente para o fortalecimento e crescimento da metodologia simulada cliente-empresa.

#### **4.6. Comparação entre a metodologia simulada cliente-empresa e demais metodologias abordadas**

A metodologia teórico prática, com dinâmica simulada cliente-empresa, compartilha alguns aspectos com as demais metodologias abordadas no capítulo de desenvolvimento. Em relação à metodologia abordada por Simão, Correa e Júnior (2014), intitulada SimMS, pode-se observar os seguintes pontos de similaridade:

- Foco na área de MS
- Utilização de tecnologias para apoiar o ensino da área e adoção de um ensino prático, tais como: jogo *single player* e software simulado
- Aplicação em ambiente universitário

Alguns dos pontos abordados, são correlatos para as demais metodologias apontadas em relação à metodologia proposta no presente trabalho. Na Tabela 4 segue uma síntese entre os trabalhos abordados e a metodologia proposta.

Tabela 4 - Síntese geral das semelhanças entre a metodologia simulada cliente-empresa e outras abordagens de ensino de manutenção de software

| Aspecto analisado                      | Familiaridades                                                                                                                                                        |
|----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Integração entre teoria e prática      | Todas as metodologias analisadas unem conteúdos teóricos com atividades práticas, promovendo a aplicação do conhecimento em contextos contextualizados.               |
| Adoção de metodologias ativas          | As abordagens são fundamentadas em estratégias que estimulam a participação ativa do estudante, como, simulação, investigação e resolução de problemas.               |
| Foco na aprendizagem significativa     | As propostas buscam desenvolver a compreensão dos estudantes por meio de experiências práticas e contextualizadas, favorecendo a construção de conhecimento aplicado. |
| Uso de recursos tecnológicos           | Todas as metodologias fazem uso de tecnologias educacionais ou ferramentas digitais desenvolvidas ou adaptadas para apoiar as atividades didáticas.                   |
| Ênfase na formação para o mercado      | As metodologias priorizam competências demandadas pelo setor de tecnologia, como resolução de problemas, trabalho em equipe, comunicação e pensamento crítico.        |
| Aplicação em contexto universitário    | As práticas foram implementadas no ensino superior, em cursos da área de computação, evidenciando a viabilidade da aplicação em disciplinas curriculares.             |
| Atividades colaborativas               | A organização das atividades envolveu a atuação dos estudantes em grupos ou equipes, favorecendo a colaboração e a construção coletiva de soluções.                   |
| Avaliação da experiência dos discentes | Em todos os casos, a efetividade das metodologias foi avaliada com base na percepção dos estudantes, por meio de                                                      |

|  |                                                     |
|--|-----------------------------------------------------|
|  | questionários e relatos dos próprios participantes. |
|--|-----------------------------------------------------|

Fonte: autor, 2025.

Como aspecto de similaridade, nota-se que todas as metodologias visam promover o ensino de MS, em um contexto acadêmico, por meio de práticas ativas. Tais práticas possuem ênfase na formação profissional dos estudantes, de forma a prepará-los para situações do mercado de trabalho.

## 5. CONCLUSÃO

O objetivo principal do presente trabalho, consiste em apoiar o ensino de manutenção de software, utilizando uma metodologia de ensino que faça a união de aulas teóricas, bem como aulas práticas apoiadas pela dinâmica simulada cliente-empresa, com a intenção de apresentar aos estudantes desafios similares aos que podem ser enfrentados no mercado de trabalho na área de MS.

Para alcançar tal objetivo foi combinada uma abordagem de ensino que realiza a união de teoria e prática da área de manutenção de software, fazendo uso, para a parte prática, de um software desenvolvido exclusivamente para este fim. A metodologia aplicada neste trabalho buscou unir aulas expositivas com um cenário prático simulado. A avaliação dessa abordagem de ensino foi realizada a partir da aplicação de questionários de caracterização antes e após a intervenção, tendo como objetivo investigar a percepção dos alunos com relação ao aprendizado da manutenção de software.

Os resultados indicam que a metodologia aplicada contribuiu significativamente para o aumento do conhecimento dos alunos sobre MS, destacando-se a elevação da autopercepção de aprendizado e a capacidade dos discentes de identificar corretamente os tipos de manutenção trabalhados (corretiva, adaptativa e perfectiva). Além disso, a maioria dos estudantes avaliou positivamente o formato teórico-prático mediante a simulação cliente-empresa, apontando maior motivação e melhor preparo para atuar na área.

De modo geral pode-se observar que a metodologia desenvolvida e aplicada no presente trabalho, gerou resultados que contribuíram positivamente para o ensino da área de manutenção de software. Por meio dos percentuais obtidos através dos questionários de coleta de dados, foi possível evidenciar que grande parte dos participantes respondentes, afirmam que a metodologia empregada para essa etapa na disciplina de Engenharia de Software, colaborou de maneira significativamente positiva para o aprendizado da área de MS. Mediante os resultados, observa-se que ocorreu um salto significativo da percepção dos estudantes em relação ao seu conhecimento sobre manutenção de software antes e após a aplicação da metodologia, como pode ser observado no paralelo entre as Figuras 12 e 19.

O uso do documento Estudo de Caso Simulado e do software desenvolvido possibilitou uma vivência próxima da realidade profissional, promovendo não apenas a assimilação de conceitos técnicos, mas também o desenvolvimento de competências

comportamentais, como trabalho em equipe e gestão de conflitos. Das dificuldades relatadas pelos alunos, constam a dificuldade em entender o código fonte, dificuldade com versionamento de código e organização da equipe, tais dificuldades apontam certa similaridade com as barreiras encontradas no estudo realizado por Bordin *et al* (2024), o que fundamenta de forma mais segura o paralelo entre o presente trabalho e as demais metodologias apontadas como fundamentação teórica. Pode-se notar que esse tipo de barreira é comum dentro do ambiente profissional de manutenção de software.

Entre as limitações do trabalho, destaca-se o tempo relativamente curto dedicado à prática simulada, o que pode ter restringido a profundidade de algumas intervenções. Recomenda-se, para trabalhos futuros, a ampliação do tempo e a replicação da metodologia em diferentes turmas e contextos, bem como a inclusão de avaliações mais detalhadas do desempenho técnico, além de sugestões dos alunos de práticas de desenvolvimento mais abrangentes, abordando APIs, *deploy*, trazendo um cenário mais amplo dentro da perspectiva da disciplina. Além disso, foram sugeridas questões relacionadas à implementação de um ambiente gráfico, no qual os participantes teriam um ambiente mais simplificado para melhor compreensão da integração entre *front* e *backend*.

Em suma, a presente pesquisa demonstra que o ensino de manutenção de software pode ser enriquecido significativamente por metodologias que aproximam teoria e prática, fortalecendo a formação dos estudantes e contribuindo para sua preparação frente às exigências do mercado de trabalho. Desta forma a presente pesquisa reforçou a eficácia de metodologias ativas de ensino de MS, atingindo resultados similares aos demais estudos aqui abordados, contribuindo de maneira positiva para o aprendizado dos futuros profissionais da área de tecnologia.

## 6. REFERÊNCIAS

- ALVES, Lynn Rosalina Gama; MINHO, Marcelle Rose da Silva; DINIZ, Marcelo Vera Cruz. Gamificação: diálogos com a educação. In: FADEL, Luciane Maria et al. (Org.). Gamificação na educação. São Paulo: Pimenta Cultural, 2014. p. 74-97.
- AMORIM, DFB de. Softwares de sistemas e de aplicações livres: benefícios e limitações no uso dessas tecnologias nos negócios. **Revista Científica Semana Acadêmica**, v. 1, n. 69, p. 1-19, 2015.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR ISO/IEC 12207: Tecnologia de informação - Processos de ciclo de vida de software. Rio de Janeiro. 1998.
- BENNETT, Keith H.; RAJLICH, Václav T. Software maintenance and evolution: a roadmap. In: Proceedings of the Conference on the Future of Software Engineering. 2000. p. 73-87.
- BORDIN, Andréa Sabedra; GARCIA, Luiza Carolina Miranda; BOCALON, Bruna Reginato; GAROFALO, Ítalo Manzine Amaral Duarte. Indissociabilidade entre Ensino e Pesquisa em Manutenção de Software: Um Relato de Experiência. In: WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO (WEI), 32. , 2024, Brasília/DF. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2024 . p. 150-159. ISSN 2595-6175. DOI: <https://doi.org/10.5753/wei.2024.2510>.
- IEE PROC. *Resenha de: Software maintenance management*. LIENTZ, B. P.; SWANSON, E. B. Addison-Wesley, 1980. In: IEE PROC., v. 127, pt. E, n. 6, nov. 1980.
- LIENTZ, Bennet P. Issues in software maintenance. ACM Computing Surveys (CSUR), v. 15, n. 3, p. 271-278, 1983.
- LIMA, Israely; RABELO, Jacilane de Holanda. A Utilização da Gamificação como Ferramenta Motivadora no Ensino de Manutenção de Software. In: WORKSHOP DE VISUALIZAÇÃO, EVOLUÇÃO E MANUTENÇÃO DE SOFTWARE (VEM), 11. , 2023, Campo Grande/MS. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2023 . p. 11-15. DOI: <https://doi.org/10.5753/vem.2023.235570>.
- LIMA, Francisco F. P. et al. Manutenção Adaptativa de Software Embarcado para Telefones Celulares Apoiado por Ferramentas de Automação . In: SIMPÓSIO BRASILEIRO DE QUALIDADE DE SOFTWARE (SBQS), 9. , 2010, Belém. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2010 . p. 327-334. DOI: <https://doi.org/10.5753/sbqs.2010.15441>.
- MAGALHÃES, N. M.; ARAÚJO, M. A.; DAVID, J. M.; PIOLI, L.; DANTAS, M. A. Suporte às atividades de manutenção de software em bases de dados abertas e distribuídas. In: SIMPÓSIO EM SISTEMAS COMPUTACIONAIS DE ALTO DESEMPENHO (WSCAD 2020), XXI, 2020. Anais [...].
- NASCIMENTO, João Teixeira do; SOUSA, Caio Rian R. de; LIMA, Lara Gabrielly; CARVALHO FILHO, Edivar C.; BORGES, Guilherme P.; RABELO, Jacilane H.; BEZERRA, Carla Ilane Moreira. Do Code Smell ao Código Limpo: Uma Experiência Prática no Ensino de Refatoração para Manutenção de Software. In: WORKSHOP SOBRE

EDUCAÇÃO EM COMPUTAÇÃO (WEI), 33. , 2025, Maceió/AL. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2025 . p. 701-711. ISSN 2595-6175. DOI: <https://doi.org/10.5753/wei.2025.8463>.

PADUELLI, Mateus Maida. Manutenção de Software: problemas típicos e diretrizes para uma disciplina específica. 2007. Dissertação (Mestre em Ciências de Computação e Matemática Computacional) — Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2007.

PRIKLADNICKI, Rafael et al. Ensino de engenharia de software: desafios, estratégias de ensino e lições aprendidas. FEES-Fórum de Educação em Engenharia de Software, p. 1-8, 2009.

SAIEDIAN, Hossein. Software engineering education and training for the next millennium. **Journal of Systems and Software**, v. 49, n. 2-3, p. 113-115, 1999.

SIMÃO, Diógenes Dias; CORREA, Dyeimys Franco; JÚNIOR, Paulo Afonso Parreira. Simms-um jogo educacional de apoio ao ensino de manutenção de software. In: Maceió, Brasil: II Workshop on Software Visualization, Evolution and Maintenance. 2014.

SIRQUEIRA, Tássio Ferenzini Martins. Melhoria no processo de manutenção de software: um estudo de fatores de influência. 2013. Trabalho de conclusão de curso (Sistemas de Informação) - Centro de Ensino Superior, Juiz de Fora, 2013.

SOMMERVILLE, Ian. Engenharia de Software / Ian Sommerville ; tradução Ivan Bosnic e Kalinka G. de O. Gonçalves ; revisão técnica Kechi Hirama. — 9. ed. — São Paulo : Pearson Prentice Hall, 2011. Título original: Software engineering.

## **APÊNDICE A - CADERNO VIRTUAL NOTION**

O link aqui exibido direciona para o material didático produzido pelos autores para apoiar a metodologia proposta no trabalho, Caderno Virtual Notion.

<https://sleepy-bolt-bee.notion.site/Manuten-o-de-Software-Uma-abordagem-te-rica-e-pr-tica-151674186cac8073bcecff137ef65151?pvs=74>

## **APÊNDICE B - ESTUDO DE CASO SIMULADO**

O link aqui exibido direciona para o material didático produzido pelos autores para apoiar a metodologia proposta no trabalho, Estudo de Caso Simulado.

<https://docs.google.com/document/d/1ko1jYclh1JraTPVI6uLXApfpHNh2PedjnyXawAxyvYQ/edit?usp=sharing>

## **APÊNDICE C - SOFTWARE PARA APLICAÇÃO DAS MANUTENÇÕES**

O link aqui exibido direciona para o material didático produzido pelos autores para apoiar a metodologia proposta no trabalho, software para aplicação das manutenções.

<https://github.com/miqueiaspcoelho/CriandoSonhosLTDA>