



**UNIVERSIDADE FEDERAL DO CEARÁ  
CAMPUS RUSSAS  
CURSO DE GRADUAÇÃO EM ENGENHARIA DE SOFTWARE**

**ANTÔNIA NAELLY FREIRE DE LIMA**

**IMPLEMENTAÇÃO DE PRÁTICAS DE INSPEÇÃO CONTÍNUA: UM ESTUDO DE CASO  
NA DISCIPLINA DE MANUTENÇÃO DE SOFTWARE**

**RUSSAS**

**2024**

ANTÔNIA NAELLY FREIRE DE LIMA

**IMPLEMENTAÇÃO DE PRÁTICAS DE INSPEÇÃO CONTÍNUA: UM ESTUDO DE CASO NA  
DISCIPLINA DE MANUTENÇÃO DE SOFTWARE**

Trabalho de Conclusão de Curso apresentado ao  
Curso de Graduação em Engenharia de Software  
do Campus Russas da Universidade Federal do  
Ceará, como requisito parcial à obtenção do  
grau de bacharel em Engenharia de Software.

Orientadora: Profa. Dra. Jacilane de Holanda Rabelo.

RUSSAS

2024

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Sistema de Bibliotecas

Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

---

F933i Freire de Lima, Antônia Naelly.

Implantação de práticas de inspeção contínua : Um estudo de caso na disciplina de manutenção de software / Antônia Naelly Freire de Lima. – 2025.

81 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Russas, Curso de Ciência da Computação, Russas, 2025.

Orientação: Profa. Dra. Jacilane de Holanda Rabelo.

1. qualidade. 2. inspeção contínua. 3. Sonar Cloud;. 4. refatoração. I. Título.

CDD 005

---

**ANTÔNIA NAELLY FREIRE DE LIMA**

**IMPLEMENTAÇÃO DE PRÁTICAS DE INSPEÇÃO CONTÍNUA: UM ESTUDO DE CASO NA  
DISCIPLINA DE MANUTENÇÃO DE SOFTWARE**

Trabalho de Conclusão de Curso apresentado ao  
Curso de Graduação em Engenharia de Software  
do Campus Russas da Universidade Federal do  
Ceará, como requisito parcial à obtenção do  
grau de bacharel em Engenharia de Software.

Aprovada em: 19/09/2024.

**BANCA EXAMINADORA**

---

Profa. Dra. Jacilane de Holanda  
Rabelo (Orientadora)  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Alexandre Matos Arruda  
Universidade Federal do Ceará - UFC

---

Bacharel Israely Lima de Oliveira  
Universidade Federal do Ceará - UFC

Dedico este trabalho aos meus pais, que sob muito sol, fizeram-me chegar até aqui, na sombra.

## **AGRADECIMENTOS**

Primeiramente, quero agradecer a toda minha família, por acreditar na minha capacidade e me apoiar incondicionalmente em todos os momentos da minha graduação. Minha mãe, a mulher mais guerreira e corajosa que conheço, e a quem eu dedico essa vitória em minha vida profissional.

Também quero agradecer a Profa. Dra. Jacilane de Holanda Rabelo, pela orientação e apoio durante a elaboração do presente trabalho. E aos professores que compõem minha banca por aceitarem fazer parte da avaliação deste projeto.

Por fim, também agradeço a todos os meus amigos que participaram diretamente ou indiretamente dessa trajetória, em especial a minha amiga Rebecca Lelis que me deu força e apoio para que esse trabalho fosse entregue.

“Conhecimento não é aquilo que você sabe, mas  
o que você faz com aquilo que você sabe.”  
(Aldous Huxley)

## RESUMO

A inspeção contínua para garantir a entrega de *software* de qualidade é uma prática essencial que deve ser incorporada e dominada por estudantes da área de desenvolvimento de *software*. Porém, pesquisas indicam que o ensino e aprendizado dessas técnicas enfrentam desafios significativos e são subestimados na maioria das instituições de ensino superior. No mercado de trabalho, a inspeção contínua tornou-se uma prática fundamental no desenvolvimento de *software*, contribuindo para a melhoria da qualidade e eficiência do produto final. A pesquisa do presente trabalho focou na implantação da ferramenta de inspeção contínua, Sonar Cloud, durante um projeto de refatoração de código-fonte. Com isso, o experimento foi realizada em um ambiente de sala de aula com alunos da disciplina de Manutenção de *Software* da Universidade Federal do Ceará - Campus de Russas, no semestre de 2023.2. Com o intuito de avaliar como a introdução da inspeção contínua afeta o processo de manutenção dos projetos dos alunos, ajudando-os a minimizar os efeitos das mudanças e a evitar a introdução de novos problemas no *software*. Os resultados demonstraram que a adoção da inspeção contínua, por meio do Sonar Cloud, auxiliou significativamente os estudantes na identificação de maus hábitos de codificação, na detecção de *bugs* já existentes no código e na prevenção de novos erros durante a refatoração dos seus projetos. Além disso, mostraram que o trabalho em equipe foi altamente valorizado pelos alunos, que reconheceram a importância das práticas de inspeção para garantir a entrega de *software* de alta qualidade. Esta pesquisa evidencia a importância de integrar técnicas e ferramentas amplamente utilizadas no mercado de trabalho ao ambiente acadêmico, promovendo um aprendizado mais alinhado com as práticas profissionais e preparando melhor os estudantes para os desafios da indústria de software.

**Palavras-chave:** qualidade; inspeção contínua; Sonar Cloud; refatoração.

## ABSTRACT

Continuous inspection to ensure the delivery of quality software is an essential practice that should be incorporated and mastered by students in the software development field. However, research indicates that the teaching and learning of these techniques face significant challenges and are often underestimated in most higher education institutions. In the workforce, continuous inspection has become a fundamental practice in software development, contributing to the improvement of the final product's quality and efficiency. This research focused on the implementation of the continuous inspection tool, Sonar Cloud, during a source code refactoring project. The experiment was conducted in a classroom setting with students from the Software Maintenance course at the Federal University of Ceará - Campus Russas, in the 2023.2 semester. The aim was to assess how the introduction of continuous inspection affects the maintenance process of students' projects, helping them minimize the impact of changes and avoid introducing new problems into the software. The results demonstrated that the adoption of continuous inspection, through Sonar Cloud, significantly helped students identify poor coding habits, detect existing bugs in the code, and prevent new errors during the refactoring of their projects. Moreover, they showed that teamwork was highly valued by the students, who recognized the importance of inspection practices in ensuring the delivery of high-quality software. This research highlights the importance of integrating techniques and tools widely used in the industry into the academic environment, promoting a learning experience more aligned with professional practices and better preparing students for the challenges of the software industry.

**Keywords:** quality; continuous inspection; Sonar Cloud; refactoring.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Classificação dos tipos de manutenção . . . . .	17
Figura 2 – Inspeção em diferentes artefatos . . . . .	20
Figura 3 – Processo de Inspeção Contínua . . . . .	21
Figura 4 – Procedimentos metodológicos . . . . .	25
Figura 5 – Etapas de implementação das práticas de inspeção na disciplina de Manutenção de <i>Software</i> . . . . .	33
Figura 6 – Fluxo da proposta educacional . . . . .	34
Figura 7 – Fluxo do trabalho de refatoração . . . . .	35
Figura 8 – Passos de configuração do SonarCloud . . . . .	35
Figura 9 – Tela de análise . . . . .	36
Figura 10 – Tela de análise . . . . .	36
Figura 11 – Perfil das pessoas respondentes . . . . .	39
Figura 12 – Porcentagem dos alunos com avanço do conhecimento em Inspeção Contínua	40
Figura 13 – Porcentagem dos alunos com avanço no conhecimento em termos como <i>bug</i> , <i>vulnerability</i> , <i>code smell</i> e <i>security hotspot rules</i> . . . . .	41
Figura 14 – Conhecimento sobre alguma ferramenta para inspeção/análise de <i>software</i> .	42
Figura 15 – Utilização de ferramentas análise de <i>software</i> nos estudos e\ou trabalho .	43
Figura 16 – Importância do estudo do processo de inspeção na universidade . . . . .	43
Figura 17 – Dados sobre como as ferramentas de análise de código pode ajudar no processo de refatoração de um <i>software</i> . . . . .	44
Figura 18 – Dados coletados apartir do Inventário de Motivação Intrínseca (IMI) - Subescala Interesse/Prazer . . . . .	45
Figura 19 – Dados coletados a partir do Inventário de Motivação Intrínseca (IMI) - Subescala Pressão/Tensão . . . . .	46
Figura 20 – Dados coletados apartir do Inventário de Motivação Intrínseca (IMI) - Subescala Percepção de Escolha . . . . .	47
Figura 21 – Dados coletados apartir do Inventário de Motivação Intrínseca (IMI) - Subescala Percepção de Competência . . . . .	47
Figura 22 – Dados coletados apartir do Inventário de Motivação Intrínseca (IMI) - Subescala Valor/Utilidade . . . . .	48

Figura 23 – Dados do Inventário de Motivação de Materiais Instrucionais (IMMS) -	
Dimensão Atenção . . . . .	49
Figura 24 – Dados do Inventário de Motivação de Materiais Instrucionais (IMMS) -	
Dimensão Relevância . . . . .	49
Figura 25 – Dados do Inventário de Motivação de Materiais Instrucionais (IMMS) -	
Dimensão Satisfação . . . . .	50
Figura 26 – Respostas dos alunos sobre sua experiência com a ferramenta <i>Sonar Cloud</i> .	52
Figura 27 – Respostas dos alunos sobre sua experiência com o processo de Inspeção	
Continua e Ferramentas de Análise de Código . . . . .	54

## **LISTA DE TABELAS**

Tabela 1 – Comparativo entre os trabalhos apresentados e o experimento apresentado neste trabalho . . . . .	24
Tabela 2 – Mapeamento IMI . . . . .	31
Tabela 3 – Mapeamento IMMS - Dimensão Relevância . . . . .	31
Tabela 4 – Mapeamento IMMS - Dimensões Satisfação e Atenção . . . . .	32

## **LISTA DE ABREVIATURAS E SIGLAS**

ARCS	Attention, Relevance, Confidence, Satisfaction
CD	Continuous Delivery
CI	Continuous Integration
GF	Grau de Favorabilidade
IMI	Intrinsic Motivation Inventory
IMMS	Instructional Materials Motivation Survey
SIGAA	Sistema Integrado de Gestão de Atividades Acadêmicas

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO . . . . .</b>	<b>14</b>
<b>2</b>	<b>OBJETIVOS . . . . .</b>	<b>16</b>
<b>2.1</b>	<b>Objetivo Geral . . . . .</b>	<b>16</b>
<b>2.2</b>	<b>Objetivos Específicos . . . . .</b>	<b>16</b>
<b>3</b>	<b>FUNDAMENTAÇÃO TEÓRICA . . . . .</b>	<b>17</b>
<b>3.1</b>	<b>Manutenção de <i>Software</i> . . . . .</b>	<b>17</b>
<b>3.1.1</b>	<b><i>Refatoração</i> . . . . .</b>	<b>18</b>
<b>3.2</b>	<b>Inspeção de <i>Software</i> . . . . .</b>	<b>19</b>
<b>3.3</b>	<b>Inspeção Contínua . . . . .</b>	<b>20</b>
<b>3.3.1</b>	<b><i>Ferramentas para Inspeção Contínua</i> . . . . .</b>	<b>21</b>
<b>4</b>	<b>TRABALHOS RELACIONADOS . . . . .</b>	<b>23</b>
<b>5</b>	<b>METODOLOGIA . . . . .</b>	<b>25</b>
<b>5.1</b>	<b>Vivência de campo . . . . .</b>	<b>25</b>
<b>5.2</b>	<b>Revisão da literatura . . . . .</b>	<b>26</b>
<b>5.3</b>	<b>Refinamento do escopo da pesquisa e definição do público-alvo do estudo</b>	<b>26</b>
<b>5.4</b>	<b>Implantação do processo de inspeção contínua em sala de aula . . . . .</b>	<b>27</b>
<b>5.5</b>	<b>Estudo de caso . . . . .</b>	<b>28</b>
<b>5.6</b>	<b>Coleta de dados e avaliação dos resultados . . . . .</b>	<b>28</b>
<b>5.6.1</b>	<b><i>Avaliação pelo IMI</i> . . . . .</b>	<b>29</b>
<b>5.6.2</b>	<b><i>Avaliação pelo IMMS</i> . . . . .</b>	<b>30</b>
<b>6</b>	<b>APLICAÇÃO DO PROCESSO DE INSPEÇÃO CONTÍNUA . . . . .</b>	<b>33</b>
<b>6.1</b>	<b>Análise do cenário . . . . .</b>	<b>33</b>
<b>6.2</b>	<b>Delimitação do objetivo da aplicação . . . . .</b>	<b>33</b>
<b>6.3</b>	<b>Formulação da proposta educacional . . . . .</b>	<b>34</b>
<b>6.4</b>	<b>Fluxo de implementação das ferramentas . . . . .</b>	<b>35</b>
<b>7</b>	<b>ESTUDO DE CASO . . . . .</b>	<b>37</b>
<b>7.1</b>	<b>Definição do contexto e participantes . . . . .</b>	<b>37</b>
<b>7.2</b>	<b>Abordagens de coleta de dados . . . . .</b>	<b>37</b>
<b>7.3</b>	<b>Preparação do trabalho de refatoração . . . . .</b>	<b>37</b>
<b>7.4</b>	<b>Aplicação do processo . . . . .</b>	<b>38</b>

<b>8</b>	<b>RESULTADOS DA ANÁLISE DO PERFIL DAS PESSOAS RESPON-</b>	
	<b>DENTES . . . . .</b>	<b>39</b>
<b>9</b>	<b>RESULTADO DA ANÁLISE QUANTITATIVA DOS DADOS . . . . .</b>	<b>40</b>
<b>9.1</b>	<b>Conhecimentos após processo . . . . .</b>	<b>40</b>
<b>9.2</b>	<b>Questionário IMI . . . . .</b>	<b>45</b>
<b>9.3</b>	<b>Questionário IMMS . . . . .</b>	<b>48</b>
<b>9.4</b>	<b>Experiência com a Ferramenta Sonar Cloud . . . . .</b>	<b>51</b>
<b>9.5</b>	<b>Experiência com o processo de Inspeção Contínua e Ferramentas de Análise de código . . . . .</b>	<b>53</b>
<b>10</b>	<b>RESULTADO DA ANÁLISE QUALITATIVA DOS DADOS . . . . .</b>	<b>56</b>
<b>10.1</b>	<b>Conhecimentos prévios sobre inspeção contínua e ferramentas de análise</b>	<b>56</b>
<b>10.2</b>	<b>Como foi o trabalho em equipe durante o processo de refatoração? . . .</b>	<b>57</b>
<b>10.3</b>	<b>Se fosse utilizar novamente as práticas e ferramentas de análise de código, o que faria diferente? . . . . .</b>	<b>58</b>
<b>10.4</b>	<b>Qual sua opinião sobre o processo de Inspeção Contínua? . . . . .</b>	<b>60</b>
<b>10.5</b>	<b>O que você achou da ferramenta que foi apresentada? . . . . .</b>	<b>61</b>
<b>11</b>	<b>CONCLUSÕES E TRABALHOS FUTUROS . . . . .</b>	<b>63</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>65</b>
	<b>APÊNDICE A – QUESTIONÁRIO UTILIZADO ANTES DA IMPLAN-</b>	
	<b>TAÇÃO DA PESQUISA . . . . .</b>	<b>67</b>
	<b>APÊNDICE B – TERMO ÉTICO DO QUESTIONÁRIO 1 . . . . .</b>	<b>69</b>
	<b>APÊNDICE C – QUESTIONÁRIO UTILIZADO APÓS A IMPLAN-</b>	
	<b>TAÇÃO DA PESQUISA . . . . .</b>	<b>70</b>
	<b>APÊNDICE D – TERMO ÉTICO DO QUESTIONÁRIO 2 . . . . .</b>	<b>79</b>

## 1 INTRODUÇÃO

O processo de manutenção de um *software* representa um estágio crítico no ciclo de vida de qualquer sistema. Sua importância é inegável, não apenas para garantir a integridade, desempenho e confiabilidade contínuos do *software*, mas também para acompanhar as demandas e mudanças constantes do ambiente tecnológico em que os sistemas operam. Essa demanda incita a busca por abordagens inovadoras e eficazes, não apenas para corrigir falhas pontuais, mas para promover a evolução progressiva e estratégica dos sistemas (Pressman, 2011).

Neste contexto, este trabalho se concentra em melhorar e manter a qualidade e confiabilidade do *software* no processo de manutenção de projetos considerados finalizados. Ao incorporar práticas de inspeção de código na fase de refatoração durante a disciplina de Manutenção de *Software* na Universidade Federal do Ceará - Campus de Russas no semestre de 2023.2, este estudo busca não apenas corrigir problemas pontuais, mas antecipar e prevenir falhas potenciais em sistemas em operação. A inspeção de código é uma abordagem que enfatiza a detecção precoce de erros e aprimoramento constante do *software*, representa um paradigma promissor que transcende as técnicas tradicionais de correção de defeitos (Ackerman *et al.*, 1989).

A escolha desse tema é motivada pela necessidade premente de adotar estratégias que não apenas corrijam problemas, mas antecipem e previnam potenciais falhas em sistemas. Além disso, é fundamental introduzir no ambiente acadêmico ferramentas e técnicas usadas atualmente no mercado de trabalho para aprimorar o processo de refatoração e manutenção de *software*. Com isso, segundo Kalinowski e Spínola (2008) a revisão do *software* se destaca como uma resposta eficaz para identificar desvios e aprimorar a qualidade do código, o que pode contribuindo assim para elevar a confiabilidade e robustez dos sistemas desenvolvidos pelos alunos da Universidade Federal do Ceará - Campus de Russas.

Este estudo concentra-se na experiência prática de integrar a inspeção contínua como uma parte essencial do processo de manutenção em um ambiente acadêmico. Espera-se que este trabalho não apenas contribua para o avanço do conhecimento na área de manutenção de *software* dentro da Universidade Federal do Ceará - Campus de Russas, mas também inspire futuras iniciativas em direção a práticas mais eficazes e eficientes no contexto da engenharia de *software* no que diz respeito a qualidade de código. A ambição é não apenas melhorar o código dos sistemas acadêmicos, mas também promover uma mentalidade de prevenção e melhoria constante que seja levada para o ambiente profissional pós-formação dos estudantes.

O restante deste trabalho está organizado da seguinte forma: a seção 2 apresenta os objetivos gerais e específicos. A seção 3 descreve os principais conceitos para compreender esse estudo, a seção 4, consiste em trabalhos relacionados com o presente. Seguindo dessa forma, a seção 5, 6 e 7 descreve detalhadamente a forma como foi realizado o trabalho e sua implantação em sala de aula. Finalizando com duas seções, a seção 8 contendo os resultados obtidos com o presente trabalho e a última (9) a conclusão sobre o trabalho e possíveis melhorias identificadas durante a sua execução.

## 2 OBJETIVOS

### 2.1 Objetivo Geral

O presente trabalho tem como objetivo geral investigar o aprendizado do ensino prático do paradigma de inspeção contínua no ambiente acadêmico na disciplina de Manutenção de *Software* no processo de refatoração, da Universidade Federal do Ceará - Campus de Russas.

### 2.2 Objetivos Específicos

Este trabalho tem os seguintes objetivos específicos:

- Projetar um processo prático de inspeção contínua no ambiente acadêmico, estabelecendo etapas claras e métodos para integrar essa prática ao ciclo de manutenção de *software*. Isso incluindo material de estudo, ferramentas específicas e diretrizes para sua execução eficiente.
- Demonstrar como o paradigma de inspeção contínua pode ajudar os alunos a identificar seus maus hábitos de codificação e reduzir a introdução de novos problemas.
- Propor novas técnicas de refatoração de *software* usando ferramenta de inspeção de código estático.
- Fornecer recomendações sobre o processo e plataformas que os profissionais aplicam durante o processo de inspeção e desenvolvimento no mercado de trabalho atualmente.
- Promover a cultura de revisão entre os estudantes, incentivando a colaboração e a revisão mútua do código. Isso busca não apenas identificar problemas, mas também estimular a troca de conhecimento e boas práticas entre os alunos, fortalecendo as habilidades de revisão e análise crítica.

### 3 FUNDAMENTAÇÃO TEÓRICA

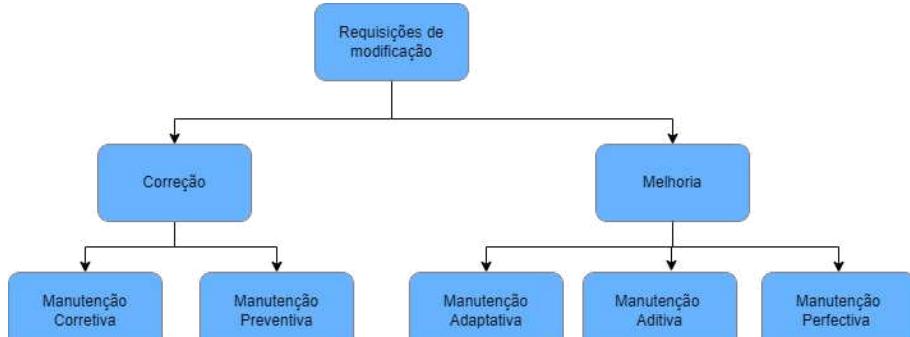
#### 3.1 Manutenção de *Software*

A manutenção de *software* é um processo definido como um conjunto de atividades realizadas após a entrega do sistema para garantir seu bom funcionamento e atender às necessidades em constante evolução dos usuários. Um processo que envolve a correção de falhas (erros observados), adaptação a novos ambientes, implantar melhorias de funções e desempenho do *software* e a prevenção de problemas futuros (ISO/IEC/IEEE, 2022).

Mudanças no *software* podem ocorrer durante ou após o desenvolvimento do sistema, seja para alcançar os requisitos dos usuários ou por questões tecnológicas e arquiteturas. Entretanto, neste trabalho será considerado que desenvolvimento é o processo de criação do *software* desde concepção inicial até sua entrada em operação e a manutenção é o processo de mudança em um sistema que já está em uso (Sommerville, 2011).

A requisição de modificação no *software* é classificada em correção ou melhoria, derivando dois tipos de manutenção para cada classificação. Como ilustrado na Figura 1, este trabalho adota a classificação dos tipos de manutenção sugerida pela ISO/IEC/IEEE (2022).

Figura 1 – Classificação dos tipos de manutenção



Fonte: Adaptado de (ISO/IEC/IEEE, 2022)

1. Manutenção Corretiva: tem como objetivo corrigir defeitos encontrados em um produto de *software* após a entrega, consolidando o reparo do *software* para alinhá-lo aos requisitos.
2. Manutenção Preventiva: manutenção que modifica o produto de *software* após a entrega visando corrigir falhas latentes no mesmo, antes que se tornem falhas operacionais.
3. Manutenção Adaptativa: alterações que visam adaptar o *software* a uma nova regra de negócio, ou seja, adaptar o sistema ao ambiente que ele está operando. Essas mudanças ajudam a manter o ritmo com o ambiente em mudança, por exemplo, uma atualização do

sistema operacional resulta em mudanças nos aplicativos.

4. Manutenção Aditiva: alteração de um *software* realizada após a entrega para incluir novas funcionalidades ou aprimorar a usabilidade do produto.
5. Manutenção Perfectiva: alterações que visam agregar melhorias para usuários, melhorias de informações para usuários e registro para melhorar o desempenho, a manutenibilidade ou outros atributos do *software*.

### **3.1.1 Refatoração**

Segundo Fowler (1999), a refatoração é uma alteração feita na estrutura do código-fonte para torná-lo mais fácil de ser entendido e menos custoso de ser modificado, sem alterar o seu comportamento externo. O objetivo é melhorar a legibilidade, a manutenibilidade e a extensibilidade do código, tornando-o mais eficiente e fácil de entender. No *software* em que o código é difícil de ser compreendido ou de ser modificado, têm-se basicamente duas opções: escrever um novo código ou aplicar a técnica da refatoração.

Quando um *software* passa por refatoração, o projeto existente é minuciosamente examinado quanto à presença de redundância, elementos de projeto não utilizados, algoritmos inefficientes ou desnecessários, estruturas de dados mal construídas ou inadequadas, ou qualquer outra falha de projeto que possa ser corrigida para resultar em um projeto aprimorado. O processo de refatoração envolve a aplicação de técnicas específicas, como extrair métodos, renomear variáveis, remover duplicação de código, entre outras, com o objetivo de promover um código mais limpo, organizado e sustentável, como apresentado por Fowler (1999). Por exemplo, em uma primeira etapa de um projeto, pode-se desenvolver um componente único que abrange várias funções, mesmo que essas funções tenham uma relação limitada entre si. Após a análise, pode-se optar por refatorar esse componente em partes menores e mais coesas, cada uma com uma responsabilidade bem definida. Um componente de gerenciamento de usuários, por exemplo pode ser dividido em três unidades distintas: autenticação, autorização e perfil de usuário. Esse processo de refatoração resulta em um *software* com alta coesão, facilitando integrações bem-sucedidas, testes eficazes e manutenção simplificada.

Embora o propósito da refatoração seja alterar o código de maneira a não impactar seu comportamento externo, é possível que ocorram efeitos colaterais involuntários dessas mudanças. Com objetivo de minimizar esses efeitos são utilizadas ferramentas de análise estáticas de *software* que auxiliam a analisar essas alterações automaticamente e geram relatório

sobre a estrutura do *software* (Sommerville, 2011).

### **3.2 Inspeção de Software**

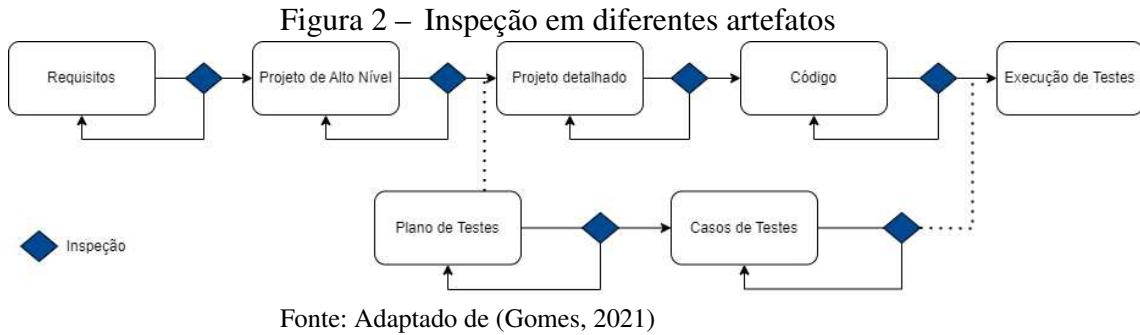
A inspeção é uma prática essencial no ciclo de vida do desenvolvimento de sistemas, cujo objetivo primário é identificar e corrigir possíveis defeitos e falhas no código antes que o produto final seja entregue aos usuários (Pressman, 2011). Além disso, essa técnica trata-se de um processo sistemático e estruturado que visa garantir a qualidade do *software* e de todos os artefatos gerados, contribuindo para a redução de custos e a melhoria da eficiência a longo prazo.

Uma das principais vantagens da inspeção é a sua capacidade de identificar falhas em um estágio inicial do desenvolvimento, o que resulta em economia de recursos e tempo (Fagan, 1999). A correção de defeitos identificados durante a revisão é significativamente menos dispendiosa do que quando detectados em fases avançadas do projeto ou após a entrega do *software* (Pressman, 2011).

Na IEEE... (2008), p 15 é apresentado a seguinte definição formal para inspeção de código:

*"A inspeção tem o objetivo de detectar e identificar anomalias em produtos de software. Uma inspeção é um exame sistemático por pares que realiza um ou mais dos seguintes procedimentos: verifica se o produto de software atende às suas especificações, verifica se o produto de software exibe atributos de qualidade especificados, verifica se o produto de software está em conformidade com regulamentos, padrões, diretrizes, planos e especificações aplicáveis e procedimentos..."*

Como apresentado por Gomes (2021), na inspeção de *software* é aplicada uma abordagem sistemática com objetivo de analisar um programa minuciosamente, o resultado final desta verificação mostra à pessoa responsável pelo artefato se o mesmo está em concordância com os padrões pré estabelecidos de qualidade e se o ciclo de vida do projeto pode seguir normalmente. Ademais, é possível observar na Figura 2, a inspeção em diferentes artefatos, como também o fluxo dessa análise durante o desenvolvimento do projeto. Após a conclusão de uma determinada etapa do processo, procede-se à revisão do artefato resultante, caso o mesmo seja aprovado na inspeção, proceder para a próxima etapa; caso contrário, retorna-se para ser refeito.



### 3.3 Inspeção Contínua

A inspeção contínua (ou "*Continuous Inspection*" em inglês) é uma prática no desenvolvimento de *software* que envolve a análise contínua do código-fonte para identificar possíveis problemas, como *bugs*, violações de padrões de codificação, vulnerabilidades de segurança e outras questões de qualidade segundo Ferreira *et al.* (2023). É uma técnica fundamental do processo de integração contínua (*Continuous Integration (CI)*) e entrega contínua (*Continuous Delivery (CD)*), que visa automatizar e facilitar a entrega do *software* de alta qualidade de forma rápida e confiável (Duvall *et al.*, 2007).

Além disso, também tornou-se uma parte importante da cultura *DevOps*, que promove a colaboração entre as equipes de desenvolvimento (Dev) e operações (Ops) para criar um ambiente de produção e entrega mais eficiente. Ademais, não existe um consenso claro sobre quando e onde exatamente o termo foi primeiramente utilizado, a expressão "inspeção contínua" pode ter diferentes origens e interpretações dependendo do contexto específico. No entanto, é possível rastrear suas origens até a popularização de práticas como integração e entrega contínua (*CI\CD*) nos últimos anos (Humble; Farley, 2010).

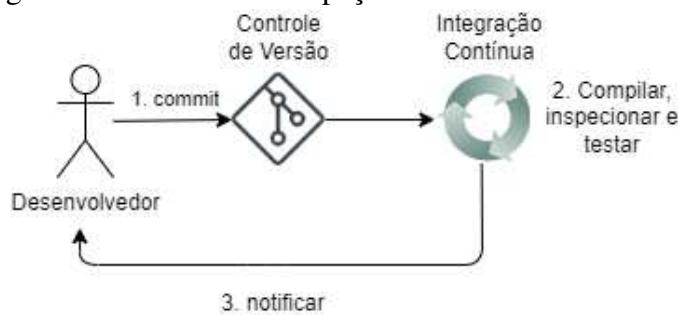
Diferentemente da "inspeção de *software*", a inspeção contínua se concentra especificamente na análise automatizada e regular do código-fonte durante o desenvolvimento ativo. É uma prática tipicamente realizada por ferramentas de análise estática de *software*, que examinam o código sem a necessidade de execução. Essas tecnologias podem identificar problemas como *bugs*, vulnerabilidades de segurança, complexidade excessiva, violações de padrões de codificação, entre outros (Gomes *et al.*, 2017).

A principal diferença entre o processo de inspeção de *software* e a inspeção contínua é o momento em que ocorrem e o nível de automação envolvida. A inspeção de *software* ao longo do ciclo de vida da aplicação, geralmente é uma atividade mais manual e formal, realizada em marcos específicos do projeto. A inspeção contínua, por outro lado, é automatizada e ocorre

em tempo real, à medida que o código é desenvolvido, tendo como objetivo a entrega de um *software* de acordo com os padrões de qualidade pré estabelecidos.

Na Figura 3 é ilustrado o fluxo de utilização das práticas da inspeção contínua no desenvolvimento de *software*.

Figura 3 – Processo de Inspeção Contínua



Fonte: Elaborado pelo autor (2023).

1. O processo inicia com um desenvolvedor submetendo as alterações em um sistema de controle de versão;
2. O servidor de integração contínua identifica alterações no repositório e dispara o gatilho para o processo de compilação, inspeção estática e teste desta alteração;
3. Ao finalizar a inspeção e testes, relatórios de qualidade e de análise são gerados nas ferramentas de revisão e de teste que foram utilizadas.

### 3.3.1 Ferramentas para Inspeção Contínua

Para a realização da inspeção de código seguindo a definição e objetivos da inspeção contínua, são utilizadas ferramentas de análise estática do código-fonte. Elas podem melhorar a velocidade e a precisão do desenvolvimento de *software* e são facilmente acessíveis em muitos ambientes de programação segundo Eilertsen (2020). Para isso é possível encontrar no ambiente comercial uma grande variedade de *softwares* que têm por objetivo a automatização do processo de revisão. Dentre essas, destacam-se ferramentas como o Visual Expert<sup>1</sup>, MiniTab<sup>2</sup>, Klocwork<sup>3</sup>, SonarQube<sup>4</sup> para soluções locais e o SonarCloud<sup>5</sup> para soluções em nuvem e o SQuORE<sup>6</sup>.

<sup>1</sup> <https://www.visual-expert.com/>

<sup>2</sup> <https://www.minitab.com/pt-br/>

<sup>3</sup> <https://www.perforce.com/products/klocwork>

<sup>4</sup> <https://www.sonarsource.com/products/sonarqube/downloads/>

<sup>5</sup> <https://www.sonarsource.com/products/sonarcloud/signup/>

<sup>6</sup> <https://www.vector.com/br/pt/produtos/products-a-z/software/square>

Existem diversas razões para utilizar esse tipo de análise em um projeto, e algumas delas incluem (Gomes, 2021):

- Facilita identificar trechos de código com defeitos, falhas de segurança e/ou problemas de estilo no projeto.
- As ferramentas de análise estática geralmente oferecem uma visão analítica mais objetiva do código, tornando mais simples para o desenvolvedor identificar uma falha cometida por desatenção, por exemplo.
- Os gerentes de projeto podem examinar os relatórios gerados pelas ferramentas e identificar padrões de falhas técnicas do projeto e perfis técnicos dos codificadores.
- Oferece suporte aos processos de melhoria contínua, onde a utilização dessas ferramentas impulsiona a padronização e otimização do código desenvolvido.

Ademais, esse tipo de ferramenta de *software* (ou *plugin*) tem o objetivo de medir diversos aspectos de qualidade do código, proporcionando uma análise abrangente de itens como: complexidade ciclomática, acoplamento entre módulos, coesão, conformidade com padrões de codificação, detecção de código duplicado, aderência às melhores práticas de segurança e detecção de vulnerabilidades conhecidas. Além disso, essas ferramentas também oferecem métricas quantitativas que permitem avaliar a legibilidade, a manutenibilidade e a escalabilidade do código-fonte, contribuindo assim para um desenvolvimento mais eficiente e sustentável do *software* (SonarSource, 2023).

#### **4 TRABALHOS RELACIONADOS**

Esta seção apresenta diferentes abordagens entre trabalhos que implementaram o processo de inspeção contínua seja no ambiente acadêmico ou comercial.

Lu Yao. Mao (2020), os autores apresentam o primeiro caso de estudo aplicando o paradigma de inspeção contínua em uma sala de aula para melhorar a qualidade da programação dos alunos. Em seu estudo, é feito um experimento controlado com 48 alunos do mesmo curso durante dois anos letivos para avaliar como o processo afeta a qualidade de programação dos estudantes. Seus resultados mostraram que a inspeção contínua pode ajudar os alunos a identificar seus hábitos incorretos de codificação e reduzir significativamente a densidade de problemas de qualidade de código introduzidos no código durante o processo de desenvolvimento. Ademais, para a realização da pesquisa, utilizaram como ferramenta de desenvolvimento colaborativo o GitLab integrando com o SonarQube, *software* de inspeção.

Gomes (2021), em seu estudo, exibe um projeto que tem como objetivo propor melhorias no processo de ensino-aprendizagem de programação e de qualidade dentro do ambiente acadêmico. No trabalho é introduzido uma ferramenta computacional de análise e registro de faltas de codificação. O SonarQube analisa o código desenvolvido pelos alunos para identificar trechos de código em desacordo com regras de qualidade e, ao finalizar as análises, gera um relatório contendo as violações encontradas, que por sua vez são persistidas em um *software* de repositório de violações o *Teacher Mate*. Em seguida, as regras violadas são analisadas pelo professor para identificar conceitos de programação incorretamente empregados pelos alunos. Com essa proposta os estudantes ficam cientes dos seus erros durante o desenvolvimento e pontos que precisam melhorar quando estão codificando.

Ademais, no experimento do Tomomitsu (2021), o autor evidencia um trabalho voltado para o uso de uma ferramenta de análise contínua SonarQube, para verificar a viabilidade na análise e manutenção de um sistema legado. Por meio das análises geradas pela ferramenta, o autor estabeleceu os pontos iniciais de análise de código, de acordo com a severidade, camada afetada e tipos de regras afetadas por *Bugs*, Vulnerabilidades e *Code Smells*. Com isso, foi possível reduzir o tempo gasto durante o processo de manutenção, como também determinar pontos que precisam ser melhorados em um projeto considerado finalizado. Ademais, verificar o resultado fornecido pela própria ferramenta SonarQube, que pode ser uma ferramenta importante para aprendizado de boas práticas de programação e monitoramento de qualidade de código.

Para mostrar com mais clareza os trabalhos correlatos, veja a Tabela 1 exposta abaixo

a qual traz um efeito comparativo entre os experimentos expostos.

Tabela 1 – Comparativo entre os trabalhos apresentados e o experimento apresentado neste trabalho

	Uso de ferramentas de inspeção	Foco na melhoria da qualidade do código	Refatoração no ambiente acadêmico	Ensino de técnicas de inspeção	Inspeção com ferramenta Web
Este trabalho	X	X	X	X	X
Lu Yao. Mao (2020)	X	X			
Gomes (2021)	X	X			
Tomomitsu (2021)	X	X			

Fonte: Elaborado pelo autor (2023).

Como é mostrado na Tabela 1 o experimento proposto neste trabalho se destaca por sua abordagem abrangente e focada na melhoria contínua da qualidade do código no processo de refatoração de um projeto já finalizado, por meio do uso de ferramentas de inspeção *web*, de fácil utilização e disponibilidade para os alunos. Ao comparar com trabalhos anteriores, como o estudo de Lu Yao. Mao (2020), observam-se similaridades no uso de ferramentas de inspeção e no foco na melhoria da qualidade do código.

Da mesma forma, o estudo de Gomes (2021) também compartilha a utilização de ferramentas de inspeção e o enfoque na melhoria da qualidade do código. Entretanto, o atual experimento vai além, incorporando não apenas a inspeção contínua, mas também a prática ativa de refatoração no ambiente acadêmico, proporcionando uma abordagem mais holística para o desenvolvimento de habilidades dos alunos.

Já o trabalho de Tomomitsu (2021) evidencia o uso de ferramentas de inspeção, porém sem o enfoque explícito no ensino dessas técnicas. Assim, enquanto os estudos anteriores abordaram aspectos importantes da inspeção de código com foco no desenvolvimento, o trabalho apresentado aqui se distingue ao integrar na refatoração as práticas e ferramentas de análise de código, conferindo uma abordagem mais completa, prática e simplificada para o desenvolvimento e aprimoramento do código em um ambiente acadêmico.

## 5 METODOLOGIA

Para atingir os objetivos estabelecidos, este trabalho seguirá algumas etapas cruciais para uma execução bem-sucedida. Nesta seção, será descrita e justificada a adoção desses processos, os quais foram divididos em seis etapas: (I) - Vivência de campo; (II) - Revisão da literatura; (III) - Refinamento do escopo da pesquisa e definição do público-alvo do estudo; (IV) - Implantação do processo de inspeção contínua em sala de aula; (V) - Estudo de caso e (VI) - Coleta de dados e avaliação dos resultados. A Figura 4 ilustra as etapas metodológicas adotadas neste trabalho.

Figura 4 – Procedimentos metodológicos



Fonte: Elaborado pelo autor (2023).

### 5.1 Vivência de campo

Primeiramente, além da formação acadêmica, o pesquisador do seguinte trabalho possui experiências profissionais como analista de sistemas e *DevOps*. Essas vivências proporcionaram uma visão prática e abrangente dos desafios enfrentados no desenvolvimento e manutenção de *software* em ambientes empresariais. Tal experiência reforça a relevância do presente estudo, que visa analisar e propor estratégias para o ensino efetivo da inspeção contínua nas instituições acadêmicas, a fim de preparar os futuros profissionais para as demandas do mercado de trabalho.

## 5.2 Revisão da literatura

Nessa etapa, foram conduzidas buscas em artigos e livros de estudos anteriores que abordaram temas como manutenção de *software*, técnicas de inspeção de código, práticas de integração e entrega contínua (*CI/CD*), ferramentas de revisão automatizada e metodologias de ensino acadêmicas. As buscas foram realizadas em bases de dados acadêmicas como *IEEE Xplore*, *ACM Digital Library* e *Scopus*, além de bibliotecas virtuais de universidades renomadas. Para cada um desses tópicos, identificaram-se artigos relevantes, os quais foram submetidos a uma análise crítica, interpretação e discussão. Com isso, foi possível identificar lacunas e áreas pouco exploradas no ensino de práticas de inspeção nas universidades, o que fundamentou o presente trabalho. Essa abordagem rigorosa na seleção de fontes e na análise crítica dos materiais encontrados fortaleceu a base teórica desta pesquisa.

## 5.3 Refinamento do escopo da pesquisa e definição do público-alvo do estudo

Esta etapa do trabalho teve como objetivo a seleção das ferramentas de automação para a implementação do processo de inspeção. Foi selecionado o SonarCloud<sup>1</sup> uma aplicação *Web* que não necessita instalação e possui integração com vários *softwares* de *CI/CD*, essa ferramenta é própria para a análise dos código e tem suporte para mais de 28 linguagens. Ademais, o GitHub<sup>2</sup> que possui uma plataforma de *CI/CD* o GitHub Action para a integração e hospedagem dos projetos dos alunos GitHub (2023). Na seção 6.4 será mostrado como foi feita a integração destas ferramentas nos projetos de refatoração da disciplina de manutenção de *software* no semestre 2023.2. Para a seleção das ferramentas que iriam ser aplicadas em sala de aula foram levados os seguintes critérios:

1. Facilidade de configuração e implantação em projetos;
2. Linguagem de programação que o *software* suporta;
3. Conexão com aplicações de desenvolvimento colaborativa como o GitHub;
4. Consolidação no mercado;
5. Conhecimento da aluna orientanda sobre as ferramentas;
6. Ferramenta que possui uma plataforma de integração contínua e entrega contínua (*CI/CD*) que permite automatizar pipeline de implantação;
7. Repositório remoto onde os trabalhos dos alunos estavam hospedados.

---

<sup>1</sup> <https://www.sonarsource.com/products/sonarcloud/signup/>

<sup>2</sup> <https://github.com/>

Além disso, foram selecionadas as fontes de estudo, dando preferência a estudos acadêmicos, e optou-se pelo componente curricular de Manutenção de *Software*. Este curso é oferecido durante o sexto período do curso de Engenharia de *Software* da Universidade Federal do Ceará - Campus de Russas, onde a docente responsável pela orientação deste estudo é a mesma que ministra a disciplina em questão. Ademais, durante a aplicação desse estudo no semestre 2023.2, 48 estavam matriculados nesta disciplina durante esse período. A escolha dessa disciplina considera argumentos importantes para a pesquisa, incluindo:

1. Pertinência ao tema: A manutenção de *software* está diretamente relacionada ao processo de inspeção, pois envolve atividades voltadas para a melhoria e correção de sistemas já existentes;
2. Abrangência dos conceitos: A disciplina aborda conceitos fundamentais para a compreensão e aplicação da inspeção de código, como técnicas de identificação e correção de falhas, boas práticas de programação, entre outros;
3. Relevância prática para os alunos: A pesquisa pode trazer benefícios diretos para os estudantes, ao promover uma prática que é altamente valorizada no mercado de trabalho.

#### **5.4 Implantação do processo de inspeção contínua em sala de aula**

O principal objetivo desta etapa é definir como seriam realizadas a implementação da inspeção contínua nos projetos de refatoração dos alunos da disciplina de Manutenção de *Software* no semestre de 2023.2 da Universidade Federal do Ceará - Campus de Russas. Este processo foi dividido em cinco etapas, detalhadas no Capítulo 6, a saber:

1. Análise do cenário: definição do escopo da implantação;
2. Delimitação do objetivo da aplicação: definição do objetivo geral da aplicação das técnicas de inspeção;
3. Implementação das ferramentas: detalhamento de como seria feita a implementação das ferramentas nos trabalhos dos alunos;
4. Formulação da proposta educacional: elaboração do planejamento de execução das atividades realizadas.

## 5.5 Estudo de caso

Objetivando investigar a importância do ensino do processo de inspeção nas universidades e propor novas técnicas de refatoração de *software* usando ferramenta de inspeção de código estático. O estudo de caso foi implementado em quatro etapas, estas serão mais detalhadas no Capítulo 7.

1. Definição do contexto e participantes: análise e detalhamento do contexto e do público;
2. Preparação do trabalho de refatoração: planejamento do trabalho de refatoração onde os alunos iriam utilizar as práticas de revisão;
3. Aplicação do processo de inspeção contínua: execução das técnicas de revisão na área de estudo;
4. Abordagens de coleta de dados: investigação em artefatos gerados pelos discentes e utilização dos questionários antes e depois da implantação e ensino das práticas de inspeção.

## 5.6 Coleta de dados e avaliação dos resultados

Inicialmente, antes de ministrar as práticas de inspeção e realizar a implementação das ferramentas de automação nos trabalhos de refatoração dos alunos da disciplina de Manutenção de *Software* no semestre de 2023.2, foi conduzida uma investigação sobre os conhecimentos prévios dos alunos a respeito dos conceitos abordados no presente trabalho. Tais investigações se deu por meio da aplicação de um questionário presente no ApêndiceA, que tem como base a escala *Likert* de cinco pontos, com cinco níveis de respostas indo de “Discordo totalmente” até “Concordo totalmente”. Ademais, para medir o nível de conhecimento dos alunos, foi aplicada a seguinte escala:

- Nenhum - nunca ouvi falar disso;
- Mínimo - ouço falar, mas nunca uso;
- Básico - tenho conhecimentos gerais, mas quase nunca os uso;
- Intermediário - tenho conhecimento médio e às vezes uso;
- Avançado - tenho um conhecimento profundo e costumo usar este;
- Especialista - sou um especialista no assunto e uso quase todos os dias.

Para o final da pesquisa, foi aplicado um segundo formulário, adicionado no ApêndiceC, contendo perguntas objetivas sobre o assunto e sobre a experiência dos alunos com a prática proposta neste trabalho, como também foram feitas algumas perguntas aplicadas no

primeiro questionário. Para este formulário como instrumento de investigação foi utilizado o Inventário da Motivação Intrínseca (Intrinsic Motivation Inventory (IMI)), usado para avaliação subjetiva da experiência vivida por sujeitos durante a realização de uma atividade a partir de diversos itens, segundo Theory (2015). Como também, o Inventário da Motivação de Materiais Instrucionais (Instructional Materials Motivation Survey (IMMS)), instrumento para avaliar a motivação de estudantes expostos a materiais ou estratégias educacionais (Júnior *et al.*, 2020).

### **5.6.1 Avaliação pelo IMI**

No presente trabalho, o Inventário da Motivação Intrínseca (IMI) foi avaliado em uma escala *Likert* variando de 1 a 7, onde 1 é "Discordo totalmente" e 7 é "Concordo totalmente". Os pesos das respostas dos alunos foram mapeados da seguinte forma:

- "Concordo totalmente"= 7
- "Concordo"= 6
- "Nem discordo, nem concordo"= 4
- "Discordo"= 2
- "Discordo totalmente"= 1

O instrumento IMI é composto por subescalas que avaliam interesse/prazer, competência percebida, esforço/importância, pressão/tensão, percepção da escolha e valor/utilidade e relacionamento durante a execução de uma determinada atividade. A subescala de interesse/prazer é considerada a principal medida de autorrelato da motivação intrínseca, avaliando diretamente esse aspecto. As subescalas de competência percebida e escolha percebida atuam como preditores positivos da motivação intrínseca, enquanto a subescala de pressão/tensão funciona como um preditor negativo. A subescala de esforço/importância desempenha um papel importante na motivação, e a subescala de valor/utilidade reflete como as pessoas internalizam e regulam suas ações com base no que consideram útil ou valioso para si mesmas. Por fim, a subescala de integração (relacionamento) está relacionada às interações interpessoais e à formação de amizades (Azevedo *et al.*, 2019).

Cada pergunta do questionário IMI foi mapeada e atribuída à subescala correspondente, como mostrado na Tabela 2, as perguntas foram mapeadas de acordo com o formulário 2. O cálculo do IMI é realizado a partir das respostas dos participantes a uma série de afirmações distribuídas entre as subescalas segundo Azevedo *et al.* (2019). Para calcular a média de cada subescala foi calculado a média aritmética, multiplicando a frequência de cada escala *Likert* pelo

peso da mesma, em seguida dividindo o número total de respostas dessa escala. Como pode ser observado na equação abaixo 5.1.

$$\text{media\_subescala} = \frac{\sum(\text{frequência\_escala} \times \text{peso\_escala})}{\text{qtde\_total\_resp\_escala}} \quad (5.1)$$

Com as médias das subescalas do IMI aferidas, a sua interpretação indica o nível de motivação intrínseca e o relacionamento do participante em relação à atividade específica. Pontuações mais altas em "Interesse/Prazer", por exemplo, indicam um maior nível de motivação intrínseca para a prática. Este processo é repetido para cada subescala, resultando em uma série de médias que representam os diferentes aspectos da motivação do participante.

Na análise das médias adquiridas de cada subescala, definiu-se uma graduação considerando os seguintes pontos de corte:  $\leq 3.0$  (não verdadeiro/não motivado),  $> 3.0$  e  $< 5.0$  (pouco verdadeiro/motivado) e  $\geq 5.0$  (muito verdadeiro/muito motivado). O ponto de corte foi aplicado de forma invertida para as questões com a marcação "inverso"(Azevedo *et al.*, 2019). Tais análises serão mostradas na seção 8, deste trabalho.

### **5.6.2 Avaliação pelo IMMS**

Para medir a motivação dos estudantes pelo Inventário da Motivação de Materiais Instrucionais (IMMS), as perguntas são alocados em quatro dimensões, nomeadas usando o Modelo Attention, Relevance, Confidence, Satisfaction (ARCS) segundo Oliveira (2024), que são descritas abaixo:

- Atenção (A): está relacionada à capacidade de despertar a curiosidade, o entusiasmo e o interesse dos alunos desde o início, incentivando uma participação ativa
- Relevância (R): refere-se à percepção dos alunos de que o elemento inovador introduzido no processo de aprendizado se conecta com suas próprias experiências, necessidades, objetivos e preferências, além do conteúdo apresentado.
- Confiança (C): diz respeito ao sentimento de controle pessoal e à expectativa de sucesso, permitindo ao aluno acreditar que conseguirá concluir o processo de aprendizagem proposto.
- Satisfação (S): Essa dimensão está relacionada à positividade com que os alunos enfrentam as experiências de aprendizagem.

O instrumento IMMS é conhecido por sua consistência interna e validade na avaliação

Tabela 2 – Mapeamento IMI

<b>Subescala IMI</b>	<b>Nº da pergunta</b>	<b>Descrição da Pergunta</b>
Interesse/Prazer	5	Enquanto eu estava fazendo a atividade de refatoração, eu estava pensando o quanto estava aproveitando.
	9	Achei que as atividades de Inspeção Contínua foi bem agradável.
	11	Fazer as atividades de refatoração foi divertida.
	12	Eu gostei muito de fazer as atividades de refatoração.
	14	Eu achei as atividades de refatoração muito chatas. (Inverso)
Pressão/Tensão	15	Eu achei as atividades de Inspeção Continua muito interessantes.
	10	Eu me senti tenso(a) durante as atividades de refatoração.
	6	Eu não me sentia nervoso(a) enquanto fazia a atividade de refatoração. (Inverso)
Percepção de Escolha	7	Eu senti que foi minha escolha fazer essa atividade de inspeção.
Percepção de Competência	8	Eu acho que sou muito bom (boa) nas atividades de refatoração.
	13	Eu estou satisfeito(a) com o meu desempenho nas atividades de refatoração.
	16	Depois de fazer as atividades de Inspeção Contínua por um tempo, me senti muito competente.
Valor/Utilidade	17	Eu achei o processo de Inspeção Continua muito importante para minha vida acadêmica e profissional.

Fonte: Elaborado pelo autor (2024).

das características motivacionais dos alunos em diferentes contextos de aprendizagem Júnior *et al.* (2020). A Tabela 3 ilustrada as perguntas e as respectivas dimensões usadas, o número da pergunta foi mapeada de acordo com o Formulário C. Apesar de quatro dimensões, no presente trabalho só foi aplicado três, sendo melhor demonstrada nas tabelas 3 e 4.

Tabela 3 – Mapeamento IMMS - Dimensão Relevância

<b>Dimensões IMMS</b>	<b>Nº da pergunta</b>	<b>Descrição da Pergunta</b>
Relevância	20	Havia histórias, figuras ou exemplos que me mostraram como esse processo de Inspeção poderia ser importante para algumas pessoas.
	19	O conteúdo e o estilo de escrita neste material dão a impressão de que vale a pena conhecê-lo.

Fonte: Elaborado pela autora (2024).

Tabela 4 – Mapeamento IMMS - Dimensões Satisfação e Atenção

<b>Dimensões IMMS</b>	<b>Nº da per- gunta</b>	<b>Descrição da Pergunta</b>
Satisfação	21	Esses materiais que envolvem o processo de inspeção são interessantes.
	23	Gostei tanto do Sonar Cloud que gostaria de saber mais sobre ele.
	24	Concluir o trabalho de refatoração nesta disciplina usando práticas de inspeção me deu um sentimento satisfatório de realização.
	22	Gostei muito de estudar essa atividade.
Atenção	18	Aprendi algumas coisas surpreendentes ou inesperadas durante o processo de Inspeção Contínua.

Fonte: Elaborado pela autora (2024).

Para a análise do IMMS, adotou-se o Grau de Favorabilidade (Grau de Favorabilidade (GF)), que corresponde à soma das respostas "concordo" e "concordo totalmente", para avaliar o julgamento positivo dos alunos em cada dimensão. Isso permite destacar as avaliações positivas em relação ao processo aplicado em sala de aula (Oliveira, 2024).

## 6 APLICAÇÃO DO PROCESSO DE INSPEÇÃO CONTÍNUA

Este capítulo expõe a implementação das práticas de inspeção na disciplina de Manutenção de *Software*.

Figura 5 – Etapas de implementação das práticas de inspeção na disciplina de Manutenção de *Software*



Fonte: Elaborado pelo autor (2023).

### 6.1 Análise do cenário

Para avaliar a eficácia da inspeção contínua no processo de manutenção de *software*, foi conduzido um caso de uso no âmbito de sala de aula durante a fase de refatoração do código dos alunos na disciplina de manutenção de *software* da Universidade Federal do Ceará - Campus de Russas, no semestre de 2023.2. A disciplina já aborda conceitos de refatoração e inspeção em seu plano de ensino, isso permitiu a implantação das práticas de inspeção contínua e as ferramentas que ela suporta.

### 6.2 Delimitação do objetivo da aplicação

Nesta fase foram expostos os objetivos a serem alcançados com a implantação das práticas de inspeção no processo de refatoração. O ensino de técnicas de revisão e ferramentas de automação e de análise estática de código podem ajudar os alunos a identificar seus maus hábitos de codificação e reduzir a introdução de novos problemas. Como também, deixar o processo

de manutenção mais otimizado e padronizado, já que um dos pilares da inspeção contínua é automatizar e facilitar a entrega do *software* de alta qualidade de forma rápida e confiável.

### 6.3 Formulação da proposta educacional

Nesta etapa de formulação da proposta educacional foi realizado o planejamento da execução do ensino na disciplina. Ademais, para a execução deste roteiro de ensino foi utilizado o cronograma e o plano de ensino já adotado pela professora responsável pela disciplina de manutenção, apenas foi adicionada a implantação das ferramentas de automação e análise estática de código no trabalho final de refatoração. Na Figura 6 é mostrado o fluxo em que foi implantado o processo na disciplina de manutenção de *software* na Universidade Federal do Ceará - Campus de Russas no semestre de 2023.2.

Figura 6 – Fluxo da proposta educacional



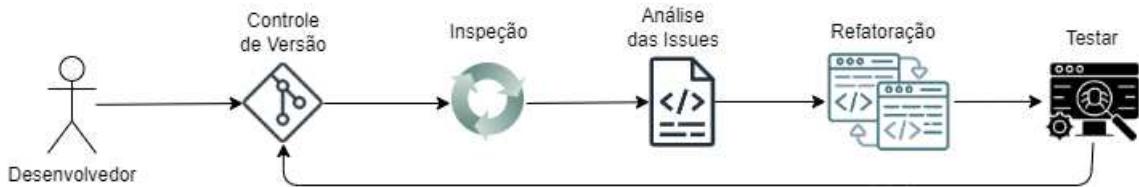
Fonte: Elaborado pelo autor (2023).

Para realizar o trabalho foi pedido as seguintes entregas:

1. Implantar o Sonar Cloud no projeto de refatoração;
2. Realizar a inspeção inicial do sistema;
3. Analisar as issues geradas pela revisão do Sonar Cloud;
4. Iniciar o processo de refatoração de acordo com as *issues* geradas;
5. Resolver os problemas principais apontados pela análise;
6. Testar as alterações;
7. Realizar o *commit* das *issues* resolvidas.

Para melhor ilustrar a Figura 7 mostra o fluxo a ser seguido pelos estudantes.

Figura 7 – Fluxo do trabalho de refatoração



Fonte: Elaborado pelo autor (2023).

#### 6.4 Fluxo de implementação das ferramentas

Nesta seção será mostrado como os alunos realizaram a implantação do Sonar Cloud nos seus projetos do GitHub.

Inicialmente, tiveram que fazer *login* no Sonar Cloud usando suas contas do GitHub, após isso cadastrar o código do repositório do trabalho de refatoração. Com os trabalhos cadastrados, deram início a implantação da ferramenta de análise no projeto. A ferramenta do SonarQube possui um passo a passo de configuração no GitHub e GitHub Action, na Figura 8 é exibida uma parte dos passos que os alunos tiveram que seguir. Entretanto, esse fluxo não foi totalmente seguido, pois, o Sonar Cloud possui a opção de inspeção automática com o GitHub para algumas linguagens de programação, sem necessidade desta configuração.

Figura 8 – Passos de configuração do SonarCloud



Fonte: (SonarSource, 2023)

Após finalizar a configuração, o Sonar Cloud inicia a inspeção onde é aberto uma *issue* sempre que uma seção do código quebra uma regra de codificação. Os problemas criados são divididos em (SonarSource, 2023):

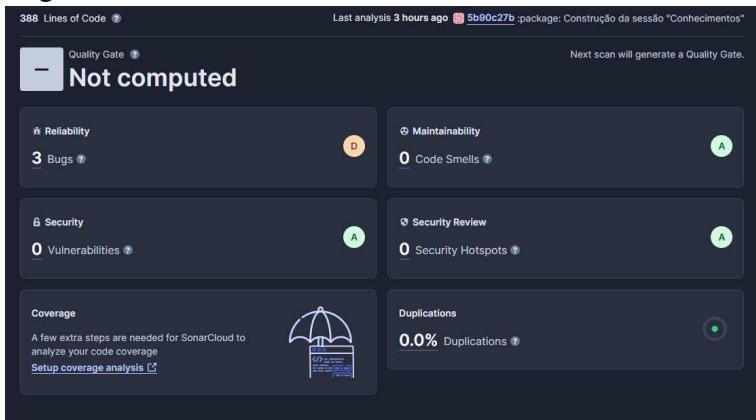
- *Security Hotspots*: trecho de código sensível à segurança que o desenvolvedor precisa

revisar;

- *Code Smells*: um problema de manutenção que torna o código confuso e difícil de manter;
- *Vulnerabilities*: um ponto do código que está aberto a ataques;
- *Bugs*: um erro de codificação que quebrará o código e precisa ser corrigido imediatamente.

Na Figura 9 é ilustrada uma tela exemplo do Sonar Cloud após uma inspeção.

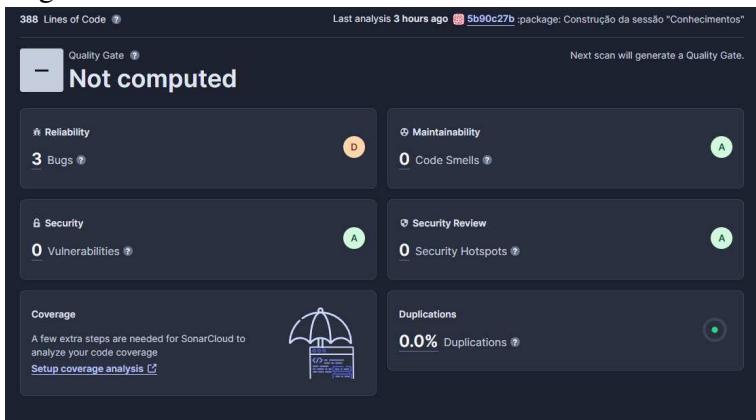
Figura 9 – Tela de análise



Fonte: (SonarSource, 2023)

Na Figura 10 é mostrado um exemplo de uma *issue* gerado pelo Sonar Cloud após a inspeção. No *link*<sup>1</sup> podemos observar algumas *issues* geradas pelo Sonar Cloud nos trabalhos de refatoração dos alunos do semestre de 2023.2 da disciplina de Manutenção de Software.

Figura 10 – Tela de análise



Fonte: (SonarSource, 2023)

<sup>1</sup> <https://drive.google.com/drive/folders/1PPGFJSnLCwwpHJcaGbLDVNRzZFEldhij?usp=sharing>

## 7 ESTUDO DE CASO

Esta seção mostra os passos realizados para aplicação da proposta idealizada no presente trabalho na disciplina de manutenção de *software* da Universidade Federal do Ceará - Campus de Russas no semestre de 2023.2.

### 7.1 Definição do contexto e participantes

O estudo de caso foi conduzido com os alunos dos cursos de Engenharia de *Software* e Ciência da Computação na disciplina de Manutenção de *Software* da Universidade Federal do Ceará - Campus de Russas, no semestre de 2023.2, foram 48 alunos participantes do estudo de caso. Além disso, tínhamos a professora responsável pela disciplina, a Profa. Dra. Jacilane de Holanda Rabelo e a aluna orientada. A professora e a orientando supervisionaram cada equipe do projeto durante o processo de refatoração e implantação do processo de inspeção nos seus trabalhos, como também análise da revisão dos códigos.

### 7.2 Abordagens de coleta de dados

Como foi abordado na Seção 5.6, para investigar os impactos e importância do ensino das práticas de inspeção no ambiente acadêmico, foram elaborados dois formulários para coleta de dados. O primeiro aplicado no início da disciplina de Manutenção de *Software* no semestre de 2023.2 e o outro aplicado depois da prática abordado no presente trabalho, no Apêndice A e C é mostrado os formulários implantados.

### 7.3 Preparação do trabalho de refatoração

Esta Sub Seção mostra os passos realizados para a preparação da proposta idealizada no presente trabalho no semestre de 2023.2.

No semestre de 2023.2 cada grupo escolheu um projeto que iria trabalhar durante a disciplina de manutenção de *software*. Com isso, foi solicitado aos alunos o cadastramento dos projetos, linguagem e link do *GitHub* que iriam utilizar a refatoração no fórum da turma do sistema Sistema Integrado de Gestão de Atividades Acadêmicas (SIGAA). Foram cadastrados 7 projetos, dentre os projetos escolhidas havia:

- Gerenciador de Pizzaria;

- CRUD de cadastro de usuário;
- Sistema de Aluguel de Fantasias;
- Sistema de venda de uma loja;
- Sistema de controle de estoque;
- Gerenciador de clínica médica;
- CRUD de sistema de gerenciamento de loja.

Com isso, a preparação do trabalho de refatoração no semestre de 2023.2, seguir a seguinte sequência:

1. Formação das equipes: à turma foi dividida em 9 equipes para a realização dos trabalhos da disciplina;
2. Cadastro do trabalho: os alunos cadastraram sua equipe e projeto utilizado no trabalho de refatoração no fórum da disciplina do sistema SIGAA;
3. Apresentação da proposta: foi apresentado a turma as práticas de inspeção, ferramentas utilizadas e como seriam cobrados em seus trabalhos;
4. Cadastro da tarefa: a professora responsável pela turma realizou o cadastro da entrega da tarefa no sistema SIGAA.

#### **7.4 Aplicação do processo**

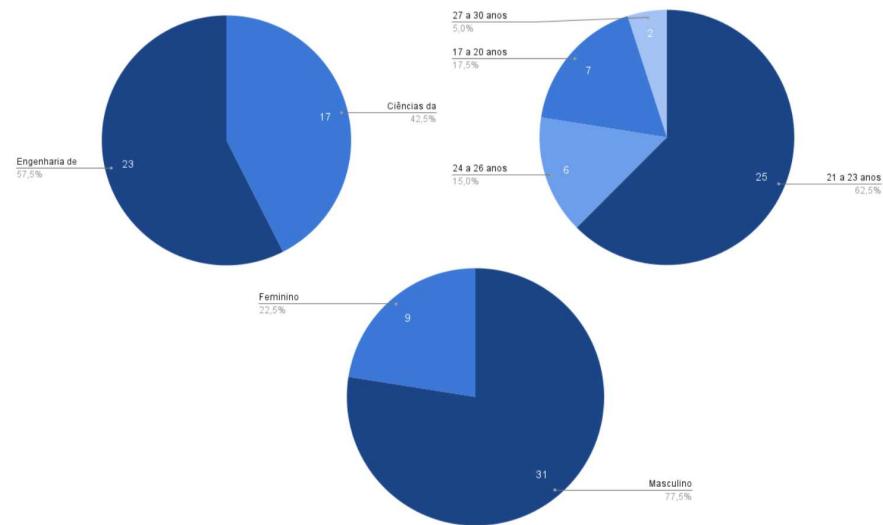
Inicialmente, antes da implementação do processo na disciplina, ocorreu uma aula sobre inspeção contínua, inspeção de *software* e ferramentas de inspeção, com foco no Sonar Cloud, utilizado no trabalho de refatoração dos alunos. A aula foi ministrada pela aluna orientada do presente trabalho.

A implantação das práticas de inspeção contínua no semestre de 2023.2 na disciplina de manutenção de *software* se iniciou em Novembro e permaneceu até o começo de Dezembro do semestre de 2023.2. Os estudantes ficaram responsáveis por implantar as ferramentas em seus trabalhos de refatoração, realizar a refatoração e documentar o que resolveram com o auxílio das ferramentas de inspeção. A entrega do trabalho teve a seguinte data: 24/11/2023, com suas apresentações do dia 24/11/2023 até 01/12/2023. Após as entregas, a docente que ministra a disciplina avaliou os trabalhos com os parâmetros que foram estabelecidos e atribuiu as pontuações correspondentes ao que foi entregue e apresentado pelos alunos.

## 8 RESULTADOS DA ANÁLISE DO PERFIL DAS PESSOAS RESPONDENTES

Os respondentes dos questionários são formados por acadêmicos da Universidade Federal do Ceará, Campus Russas, com o componente curricular Manutenção de *Software*. Ademais, a Figura 11 detalha o perfil das pessoas respondentes dos formulários. Embora 48 estudantes estivessem matriculados na disciplina, apenas 40 responderam aos questionários. Desses, 27 responderam ao primeiro e 36 ao segundo. Entre os respondentes 23 (57,5%) alunos cursam Engenharia de *Software* e 17 (42,5%) Ciências da Computação. Além disso, 31 (77,5%) são do gênero masculino e 9 (22,5%) feminino. A faixa etária dos acadêmicos é composta por 7 (17,5%) estudantes entre 17 a 20 anos, 6 (15,0%) entre 24 e 26 anos, 25 (62,5%) entre 21 a 23 anos e 2 (5,0%) respondentes estão entre 27 a 30 anos.

Figura 11 – Perfil das pessoas respondentes



Fonte: Elaborado pelo autor (2024)

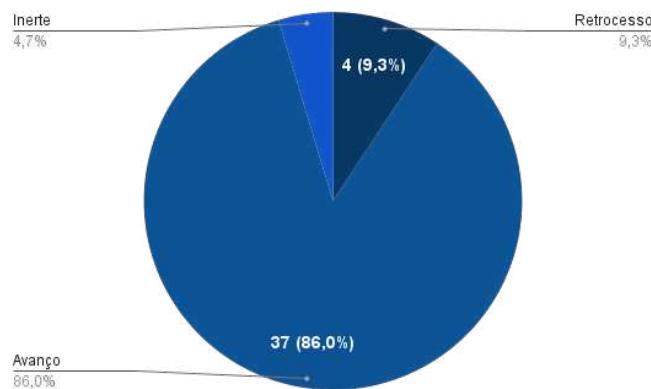
## 9 RESULTADO DA ANÁLISE QUANTITATIVA DOS DADOS

Nesta seção, é exposto à análise quantitativa dos dados coletados com os formulários aplicados na disciplina de Manutenção de *Software*, no semestre de 2023.2 da Universidade Federal do Ceará - Campus de Russas.

### 9.1 Conhecimentos após processo

Sobre os conhecimentos dos alunos respondentes sobre o processo de inspeção contínua a Figura 12 é ilustrado a porcentagem dos alunos com avanço no conhecimento em Inspeção Contínua. Foi realizado o comparativo dos dados coletados no primeiro e segundo formulário aplicado na turma da disciplina de Manutenção de *Software*, no semestre de 2023.2 da Universidade Federal do Ceará - Campus de Russas. Contudo, os resultados comparativos apresentados neste trabalho levam em consideração os alunos que responderam a ambos os formulários, aqueles que responderam apenas ao primeiro formulário foram classificados na categoria de retrocesso. Por outro lado, os alunos que responderam apenas ao segundo formulário foram classificados na categoria de avanço.

Figura 12 – Porcentagem dos alunos com avanço do conhecimento em Inspeção Contínua



Fonte: Elaborado pelo autor (2024)

O gráfico revela uma mudança significativa no nível de entendimento dos alunos. Após a prática de ensino, a análise indicou que 37 (86%) dos alunos demonstraram avanço em seu conhecimento sobre inspeção contínua. Esse aumento reflete uma compreensão mais aprofundada do conceito e suas aplicações.

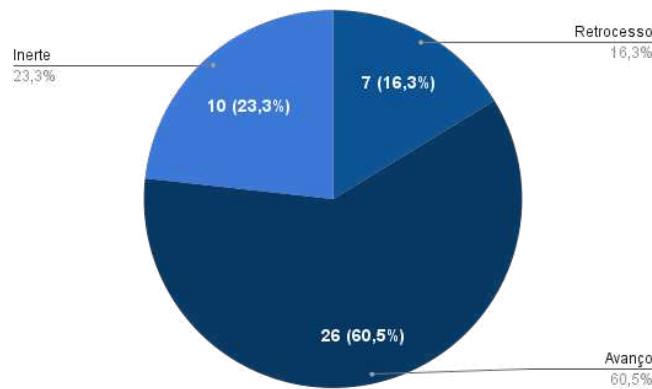
Por outro lado, uma parcela de 4 (9,3%) dos alunos mostrou um retrocesso em seu conhecimento durante a avaliação inicial, isso sugere uma possível falta de compreensão ou

confusão em relação aos conceitos discutidos antes da implementação da prática de ensino. Além disso, uma pequena parte 2 (4,7%) dos alunos apresentou inércia em seu conhecimento, indicando que não houve mudança significativa em sua compreensão do tema após a prática de ensino.

Posteriormente a implementação da prática de ensino, houve uma melhoria notável no conhecimento geral dos alunos em inspeção contínua, os dados coletados mostram que a maioria dos alunos experimentou um avanço em seu entendimento do assunto, evidenciando a eficácia da prática de ensino em melhorar os conhecimentos dos alunos. Esses resultados destacam a importância de métodos de ensino eficazes na promoção do aprendizado e na melhoria do conhecimento dos alunos, especialmente em disciplinas técnicas como a manutenção de *software*.

Na Figura 13 é mostrado o comparativo referentes aos conhecimentos das pessoas respondentes sobre alguns termos usados durante o processo de refatoração e desenvolvimento: *bug*, *vulnerability*, *code smell* e *security hotspot rules*. Os resultados indicaram que 26 (60,5%) dos alunos demonstraram avanço em seu conhecimento nestas expressões. Ademais, uma parcela de 7 (16,3%) dos alunos mostrou um retrocesso em seu conhecimento durante a avaliação. Esses dados podem indicar uma possível falta de familiaridade ou compreensão inadequada dos termos técnicos usados durante a aplicação do presente trabalho.

Figura 13 – Porcentagem dos alunos com avanço no conhecimento em termos como *bug*, *vulnerability*, *code smell* e *security hotspot rules*.



Fonte: Elaborado pelo autor (2024)

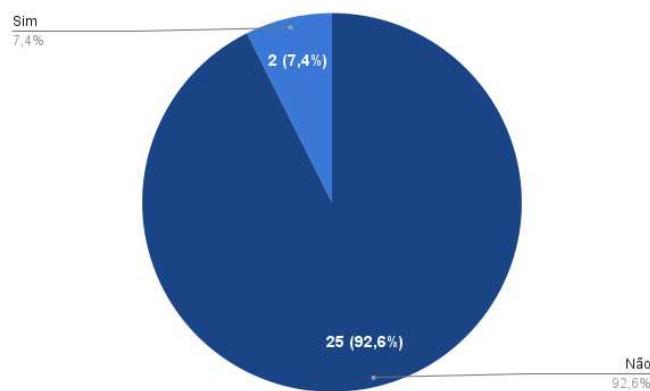
Além disso, uma proporção de 10 (23,3%) dos alunos apresentou inércia em seu conhecimento, indicando que não houve mudança significativa em sua compreensão destes termos após a prática de ensino. Na aplicação da prática, observou-se uma melhoria global no entendimento dos alunos em relação às expressões técnicas de *software* apresentadas, eviden-

ciando a eficácia da intervenção pedagógica em aprimorar o conhecimento dos alunos sobre termos usados no mercado de trabalho.

Esses resultados destacam a importância de estratégias de ensino direcionadas e eficazes na promoção do aprendizado efetivo de conceitos técnicos fundamentais e cruciais para o mercado acadêmico e profissional.

O gráfico apresentado na Figura 14 ilustra as respostas dos alunos à pergunta sobre o conhecimento de alguma ferramenta de inspeção/análise de *software* antes da aplicação do estudo. Dos participantes, 24 (92.6%) afirmaram não ter conhecimento de nenhuma ferramenta, enquanto 2 (7.4%) declararam estar familiarizados com alguma. Esses resultados indicam que a maioria dos alunos não possui o conhecimento necessário sobre ferramentas essenciais para a inspeção e análise de *software* durante o desenvolvimento ou refatoração.

Figura 14 – Conhecimento sobre alguma ferramenta para inspeção/análise de *software*



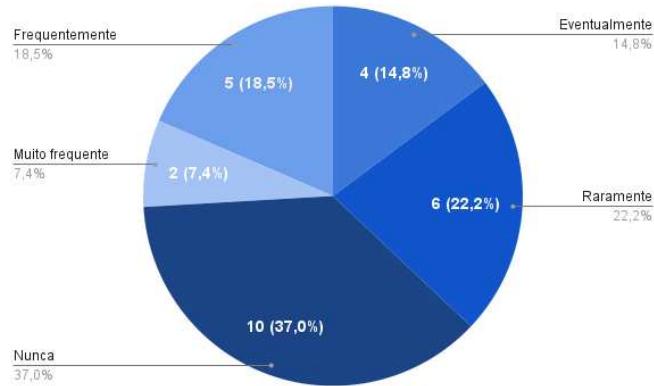
Fonte: Elaborado pelo autor (2023)

A relevância do conhecimento de ferramentas de inspeção/análise de *software* para o mercado de trabalho não pode ser subestimada. Como foi apresentado na Seção 3, essas ferramentas são frequentemente utilizadas em várias áreas da engenharia de *software*, incluindo garantia de qualidade, desenvolvimento de *software* e manutenção de sistemas. Portanto, melhorar o conhecimento dos alunos sobre essas ferramentas não só os prepara melhor para enfrentar os desafios técnicos no ambiente de trabalho, como também "enriquece" seus conhecimentos pessoais na área da computação, sendo um dos objetivos buscados no presente trabalho.

A Figura 15 apresenta dados relativos à utilização de ferramentas de análise de *software* nos estudos e/ou trabalho dos estudantes antes da aplicação da pesquisa. Dos participantes, 10 (37.0%) afirmaram nunca utilizar qualquer tipo de ferramenta de inspeção antes da prática, enquanto 6 (22.2%) a utilizam raramente. Além disso, 5 (18.5%) declararam utilizá-las com

frequência em seus estudos e/ou trabalho. Adicionalmente, 4 (14.8%) afirmaram utilizá-las eventualmente, e 2 (7.4%) as empregam com muita frequência.

Figura 15 – Utilização de ferramentas análise de *software* nos estudos e\ou trabalho

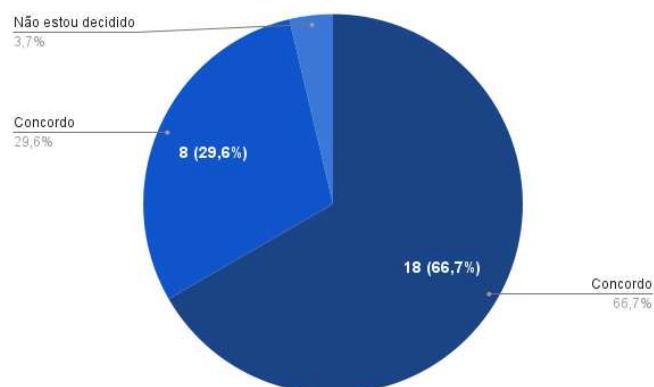


Fonte: Elaborado pelo autor (2023)

Os dados coletados sugerem que há uma necessidade significativa de aumentar a conscientização e a formação em ferramentas de análise de *software* entre os estudantes. Mais da metade dos participantes 59.2% (16) raramente ou nunca utilizam essas ferramentas, o que pode indicar lacunas no currículo educacional ou falta de recursos disponíveis para os alunos da área da computação.

Na Figura 16 é mostrada a resposta dos estudantes sobre a importância do ensino e estudo do processo de inspeção no ambiente acadêmico. Dentre os participantes, 18 (66.7%) afirmaram concordar totalmente com a afirmação sobre a relevância desse estudo, enquanto 8 (29.6%) concordaram e 1 (3.7%) demonstrou indecisão sobre o assunto.

Figura 16 – Importância do estudo do processo de inspeção na universidade



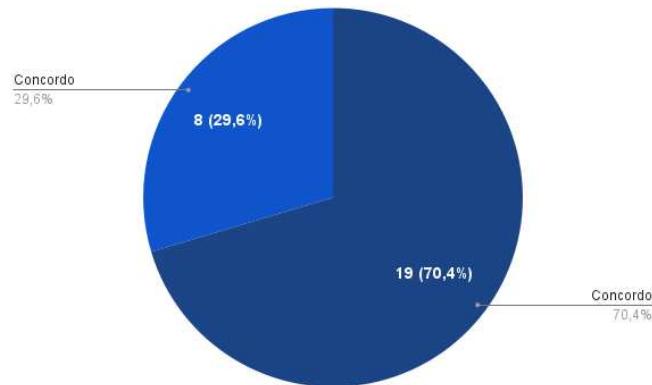
Fonte: Elaborado pelo autor (2023)

Observando os dados coletados, a maioria dos participantes 96,3% (26) concorda

ou concorda totalmente com a relevância desse tema no ambiente acadêmico, destacando a necessidade de integrá-lo no ensino das universidades. A concordância sobre a importância do estudo de inspeção reflete a disposição dos estudantes em adotar e promover uma cultura de revisão de código que os preparam para um mercado de trabalho cada vez mais exigente e focado na qualidade.

Por fim, foi afirmado aos alunos antes da aplicação da pesquisa se a utilização de ferramentas de análise de código pode contribuir para o processo de refatoração de um *software*. Os dados correspondentes estão disponíveis na Figura 17, onde dos 27 participantes, 19 (70.4%) afirmaram concordar integralmente com a afirmação, enquanto 8 (29.6%) apenas concordaram.

Figura 17 – Dados sobre como as ferramentas de análise de código pode ajudar no processo de refatoração de um *software*



Fonte: Elaborado pelo autor (2023)

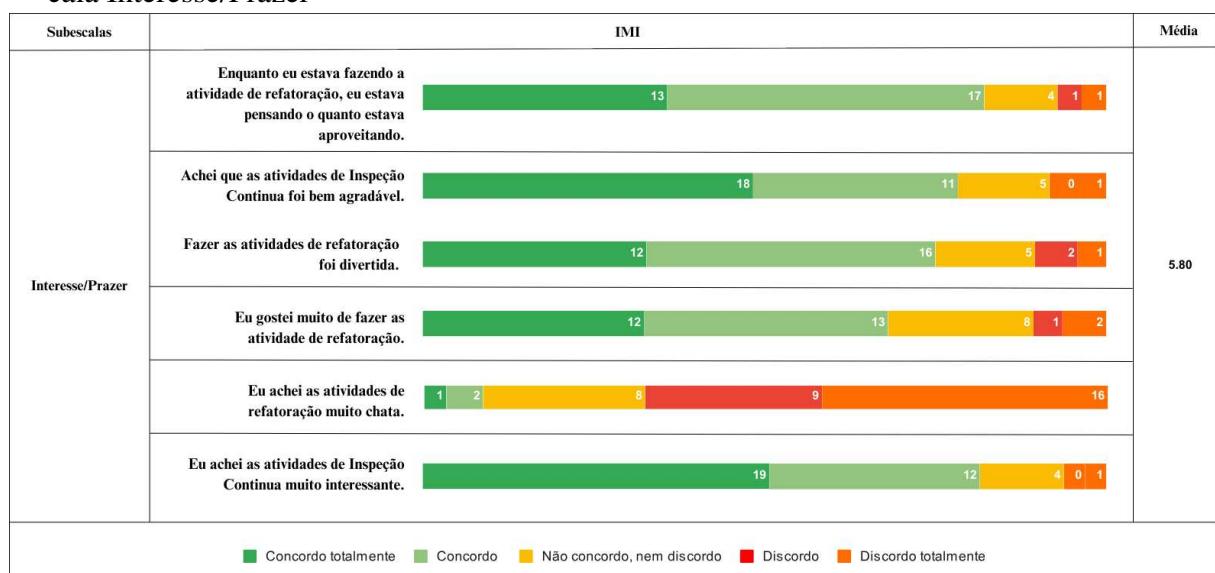
Os dados revelam uma percepção amplamente positiva entre os estudantes sobre a contribuição das ferramentas de análise de código para o processo de refatoração de *software*. Com 19 (70.4%) concordando integralmente e 8 (29.6%) concordando, nota-se reconhecimento da importância dessas ferramentas entre os estudantes participantes da pesquisa. A integração dessas práticas no currículo acadêmico, juntamente com treinamento e experiência prática como é o caso apresentado no presente trabalho, pode preparar melhor os estudantes para o mercado de trabalho. Além disso, esses resultados suportam os objetivos do trabalho, fornecendo recomendações valiosas sobre o uso de ferramentas de análise de código e promovendo uma cultura de revisão colaborativa entre os estudantes. A percepção positiva dos estudantes sugere que eles entendem os benefícios e estão dispostos a utilizar essas ferramentas em suas práticas de desenvolvimento após a implantação da prática realizada nesta pesquisa.

## 9.2 Questionário IMI

Obteve-se 36 respostas no segundo formulário aplicado na disciplina de Manutenção de *Software* da turma de 2023.2, houve uma divergência entre a quantidade de respondentes entre o primeiro e o segundo questionário. Como apresentado na Seção 5.6 foi aplicado o inventário IMI neste questionário, um instrumento que permite avaliar o grau de interesse/prazer dos estudantes durante a execução da prática apresentada no presente trabalho. Abaixo é ilustrado os resultados coletados a partir de cada afirmação dividido pelas subescalas do IMI e a sua média aritmética seguindo os pontos de corte definidos na 5.6.1. Para melhor visualização os dados foram divididos em gráficos de acordo com subescala.

Primeiramente, será ilustrada a subescala "**Interesse/Prazer**" no Gráfico 18.

Figura 18 – Dados coletados apartir do Inventário de Motivação Intrínseca (IMI) - Subescala Interesse/Prazer



Fonte: Elaborado pelo autor (2024)

A média elevada (5,80) indica que os participantes acharam o processo elaborado interessante e prazeroso. Este resultado sugere uma alta motivação intrínseca, onde os alunos se sentiram engajados e motivados durante a realização da atividade de aprendizado e implantação do processo de inspeção contínua em seus trabalhos de refatoração.

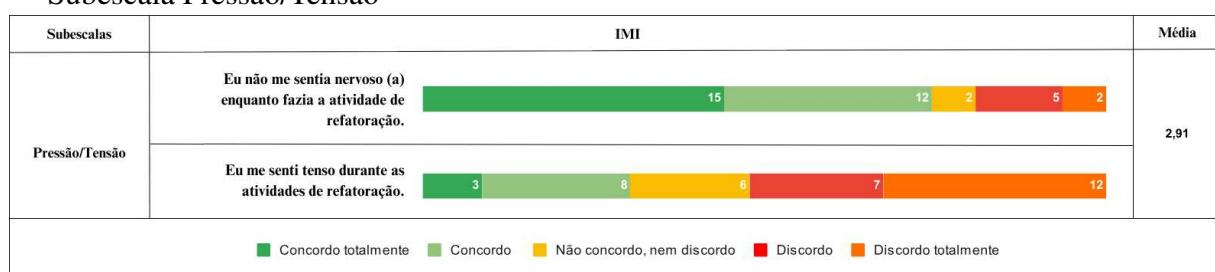
Na afirmação "Enquanto eu estava fazendo a atividade de refatoração, eu estava pensando o quanto estava aproveitando." a maioria dos estudantes (30) indicaram uma visão positiva do processo, com 13 deles concordando totalmente e 17 concordando. Apenas 4 estudantes foram neutros, enquanto um pequeno número (2) discordaram. Quanto à questão "Achei que as

atividades de Inspeção Continua foi bem agradável."18 alunos concordaram totalmente com a afirmação e 11 concordaram. Ademais, 5 estudantes ficaram neutros, 2 discordaram totalmente e nenhum discordou. Sobre a afirmação "Fazer as atividades de refatoração foi divertida.", 12 participantes concordam totalmente, 16 concordam e 5 foram neutros. Uma pequena parte (3) tiveram uma percepção negativa sobre a realização da atividade de inspeção, onde 2 discordaram e 1 discorda totalmente.

Questionados sobre se gostaram de fazer as atividades de refatoração, 25 alunos responderam que gostaram, 11 foram neutros ou não gostaram. Já sobre a afirmação "Eu achei as atividades de refatoração muito chatas", a maioria dos participantes (25) discordaram totalmente ou discordaram, 8 ficaram neutros e apenas 3 acharam a prática desinteressante. Em termos de interesse na afirmação "Eu achei as atividades de Inspeção Contínua muito interessantes.", a grande maioria dos alunos (31), achou a atividade de inspeção contínua muito interessante. Ademais, 4 ficaram neutros sobre seu interesse e apenas 1 considerou a atividade desinteressante.

Em relação às afirmações da subescala "**Pressão/Tensão**", ilustrado no Gráfico 19, observa-se uma média mais baixa (2,91), o que é positivo, pois indica que os alunos não se sentiram excessivamente pressionados ou tensos durante o processo. Um nível baixo de pressão é favorável para o sucesso da metodologia de ensino aplicado, pois os estudantes podem se concentrar melhor e aproveitar o processo de aprendizado.

Figura 19 – Dados coletados a partir do Inventário de Motivação Intrínseca (IMI) - Subescala Pressão/Tensão



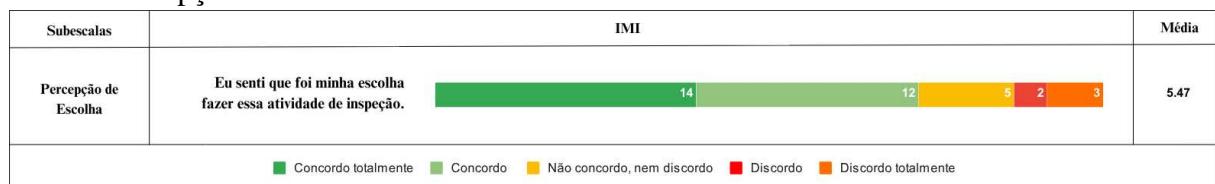
Fonte: Elaborado pelo autor (2024)

Sobre a percepção "Eu me senti tenso(a) durante as atividades de refatoração.", a maioria dos alunos (19) não se sentiram tensos durante a execução do processo apresentado no presente trabalho, embora um grupo menor (11) relataram tensão e 6 ficaram neutros. Quanto à questão sobre nervosismo durante a refatoração, 27 estudantes relatam que não se sentiram nervosos, com 15 concordando totalmente e 12 concordando. Enquanto, alguns alunos (7) demonstraram algum desconforto e 2 ficaram neutros. Com intuito de auxiliar o entendimento

da descrição acima foi gerado a seguinte Imagem 19.

No Gráfico 20 é ilustrado os dados coletados sobre a subescala "Percepção de Escolha".

Figura 20 – Dados coletados apartir do Inventário de Motivação Intrínseca (IMI) - Subescala Percepção de Escolha

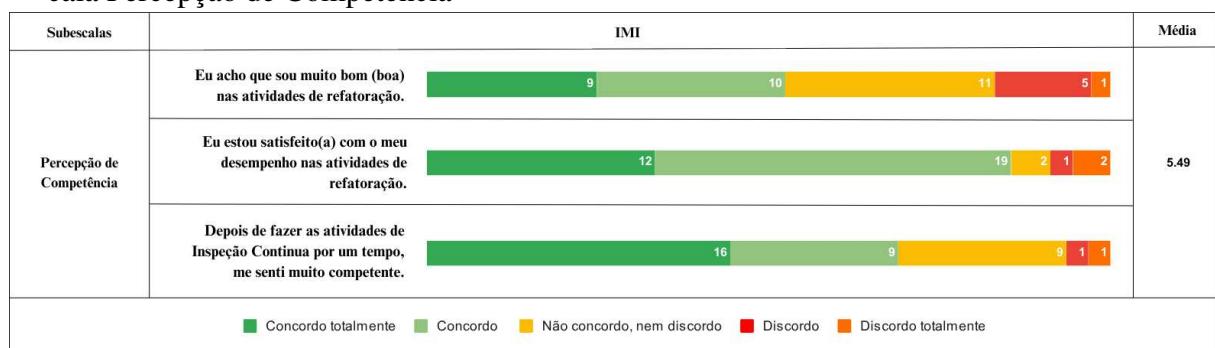


Fonte: Elaborado pelo autor (2024)

A média aritmética alta (5,47) sugere que os participantes sentiram um alto grau de autonomia e escolha durante o processo de refatoração usando práticas de inspeção. A sensação de escolha é um importante fator de motivação, contribuindo para o engajamento dos estudantes na atividade. Na afirmação "Eu senti que foi minha escolha fazer essa atividade de inspeção." sobre a percepção de autonomia dos participantes. 26 alunos sentiram que foi uma escolha pessoal participar das atividades de inspeção, com 14 concordando totalmente e 12 concordando. Enquanto uma minoria (5) não tiveram a mesma percepção de escolha e 5 alunos ficaram neutros na sua percepção.

Ademais, no Gráfico 21 é apresentado as afirmações sobre "Percepção de Competência" e sua respectiva média. Com média de 5,49, nota-se que os alunos se sentiram relativamente competentes e confiantes em suas habilidades ao realizarem a atividade de inspeção contínua em seus trabalhos. Esta percepção positiva de competência pode levar a uma maior participação e envolvimento dos estudantes nas atividades de inspeção.

Figura 21 – Dados coletados apartir do Inventário de Motivação Intrínseca (IMI) - Subescala Percepção de Competência

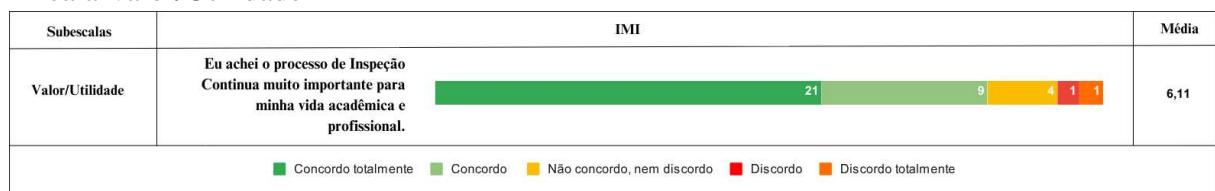


Fonte: Elaborado pelo autor (2024)

Na afirmação "Eu acho que sou muito bom (boa) nas atividades de refatoração." a maioria dos estudantes (19) indicaram uma visão positiva da sua competência durante a realização das atividade de refatoração propostas no presente trabalho, com 9 deles concordando totalmente e 10 concordando. 11 estudantes foram neutros, enquanto um pequeno número (6) discordaram totalmente ou discordaram. Quanto à questão "Eu estou satisfeito(a) com o meu desempenho nas atividades de refatoração." 12 alunos concordaram totalmente com a afirmação e 19 concordam. Ademais, 2 estudantes ficaram neutros, 3 discordaram totalmente ou discordaram. Sobre a afirmação "Depois de fazer as atividades de Inspeção Contínua por um tempo, me senti muito competente.", 16 participantes concordam totalmente, 9 concordam e 9 foram neutros. Uma pequena parte (2) discordaram ou discordaram totalmente.

Por fim, em relação às afirmações sobre a subescala "**Valor/Utilidade**" no Gráfico 22 é apresentado seus respectivos resultados.

Figura 22 – Dados coletados apartir do Inventário de Motivação Intrínseca (IMI) - Subescala Valor/Utilidade



Fonte: Elaborado pelo autor (2024)

A média mais alta (6,11), indica que os participantes veem um grande valor e utilidade no processo de inspeção contínua para suas vidas acadêmicas e profissionais. Reconhecer o valor prático das atividades é crucial para a motivação e o compromisso dos estudantes com o aprendizado. Na afirmação "Eu achei o processo de Inspeção Continua muito importante para minha vida acadêmica e profissional.", 30 estudantes concordaram totalmente ou concordaram com a relevância do processo de inspeção contínua para seu desenvolvimento acadêmico e profissional. Enquanto, apenas 2 participantes discordaram totalmente ou discordaram e 4 ficaram neutros.

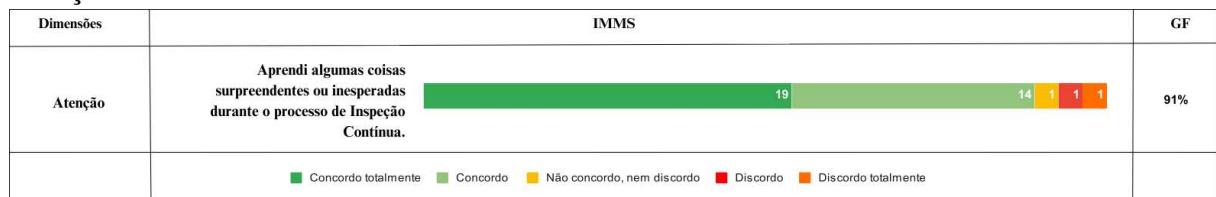
### 9.3 Questionário IMMS

Com o objetivo de investigar a motivação dos participantes da pesquisa em relação a estratégia educacional apresentada, foi aplicado o Inventário de Motivação de Materiais Instrucionais (IMMS). A análise dos dados do instrumento IMMS é baseada em 7 questões,

aplicadas no segundo formulário respondido pelos alunos da disciplina de Manutenção de Software do semestre 2023.2. Observar-se abaixo os dados coletados e o respectivo grau de favorabilidade (GF), a soma da quantidade de respondentes das escalas de “concordo” e “concordo totalmente” como apresentado na subseção 5.6.2.

Primeiramente, será ilustrada a dimensão "**Atenção**" no Gráfico 23. Avaliando tal dimensão, houve um alto grau de favorabilidade ( $75\% < GF < 98\%$ ) por parte dos estudantes, evidenciando que o material de ensino das práticas de inspeção estimularam a curiosidade, o entusiasmo, o interesse e a participação ativa dos alunos.

Figura 23 – Dados do Inventário de Motivação de Materiais Instrucionais (IMMS) - Dimensão Atenção

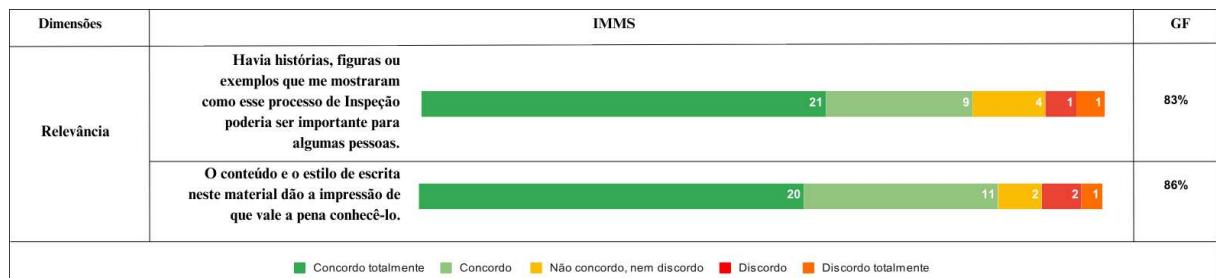


Fonte: Elaborado pelo autor (2024)

Na afirmação "Aprendi algumas coisas surpreendentes ou inesperadas durante o processo de Inspeção Contínua" foi totalmente aceita por 53% (19) dos participantes, com 38% (14) concordando. Apenas 6% (2) dos estudantes discordaram ou discordaram totalmente, e 3% (1) ficaram neutros, resultando em um grau de favorabilidade de 91% (33).

Ademais, o Gráfico 24 é apresentado os dados coletados relativos a dimensão "**Relevância**" do IMMS.

Figura 24 – Dados do Inventário de Motivação de Materiais Instrucionais (IMMS) - Dimensão Relevância



Fonte: Elaborado pelo autor (2024)

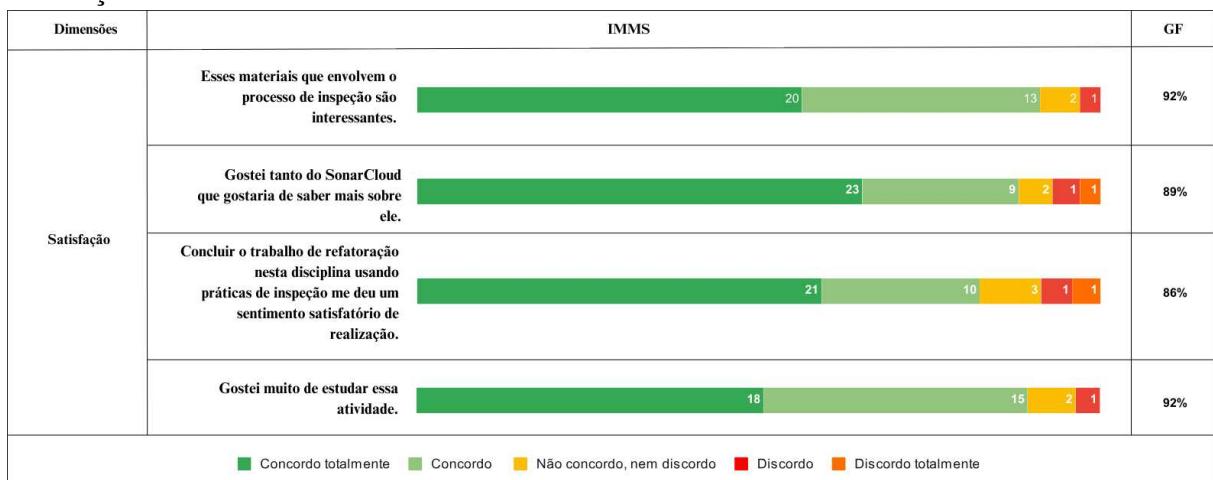
Na avaliação do critério **Relevância** houve alto grau de favorabilidade ( $81\% < GF < 100\%$ ) sugere que despertou na maioria dos estudantes da disciplina de Manutenção de Software do semestre de 2023.2, um sentimentos de conexão por meio das dinâmicas e o processo

de aprendizagem, alinhando o conteúdo apresentado com suas experiências, necessidades e preferências.

Na primeira afirmação "Havia histórias, figuras ou exemplos que me mostraram como esse processo de Inspeção poderia ser importante para algumas pessoas", obteve um grau de favorabilidade de 83% (30). Nesse caso, 58% (21) dos alunos concordaram totalmente, e 25% (9) concordaram, enquanto 6% (2) discordaram ou discordaram totalmente, e 11% (4) ficaram neutros. Já na questão "O conteúdo e o estilo de escrita neste material dão a impressão de que vale a pena conhecê-lo" foi totalmente aceita por 56% (20) dos participantes, com 30% (11) concordando. Apenas 6% (2) ficaram neutros, 6% (2) discordaram, e 2% (1) discordaram totalmente, resultando em um grau de favorabilidade de 86% (33).

Por fim, o Gráfico 25 é apresentado os dados coletados relativos à dimensão "**Satisfação**" do IMMS.

Figura 25 – Dados do Inventário de Motivação de Materiais Instrucionais (IMMS) - Dimensão Satisfação



Fonte: Elaborado pelo autor (2024)

Analizando os dados da dimensão **Satisfação**, observa-se um alto grau de favorabilidade ( $88\% < GF < 100\%$ ), evidenciando que o processo de inspeção contínua impactou positivamente os alunos, criando um sentimento de realização ao completar a atividade.

A primeira afirmação "Esses materiais que envolvem o processo de inspeção são interessantes" teve um grau de favorabilidade de 92% (33). Metade dos alunos, 56% (20) concordou totalmente, e 36% (13) concordaram, enquanto 3% (1) discordaram e nenhum discordou totalmente, e 5% (2) ficaram neutros. Na segunda questão "Gostei tanto do SonarCloud que gostaria de saber mais sobre ele" recebeu concordância total de 64% (23) dos alunos, e 25% (9) concordaram. Com apenas 6% (2) discordando ou discordando totalmente, e 5% (2)

neutros, o grau de favorabilidade de 89% (32) indica que o *Sonar Cloud* ferramenta utilizada durante a prática, despertou grande sentimento de satisfação entre os alunos.

Ademais, na pergunta "Concluir o trabalho de refatoração nesta disciplina usando práticas de inspeção me deu um sentimento satisfatório de realização", recebeu concordância total de 58% (21) dos participantes e concordância de 28% (10). Apenas 6% (2) dos estudantes discordaram da afirmação, enquanto 8% (3) permaneceram neutros. Com um grau de favorabilidade de 86%, esses resultados indicam que a grande maioria dos alunos se sentiu realizada ao usar práticas de inspeção, refletindo uma percepção positiva do processo proposto no presente trabalho.

Por fim, na afirmação "Gostei muito de estudar essa atividade", obteve um grau de favorabilidade ainda mais elevado, de 92%. Metade dos participantes, 50% (18) concordou totalmente com a afirmação, enquanto 42% (15) concordaram. Apenas 5% (2) ficaram neutros, 3% (1) dos alunos discordaram, e nenhum discordou totalmente.

Nas análises das dimensões de **Atenção, Relevância e Satisfação** do questionário IMMS, mostram que a implementação do processo de Inspeção Contínua na disciplina foi bem recebida pela maioria dos alunos participantes. Os altos graus de favorabilidade em várias afirmações indicam que os alunos acharam a experiência educativa valiosa, interessante e motivadora. Esses resultados são promissores para a continuidade e aprimoramento do processo de implantação de inspeção contínua na formação acadêmica e profissional dos estudantes.

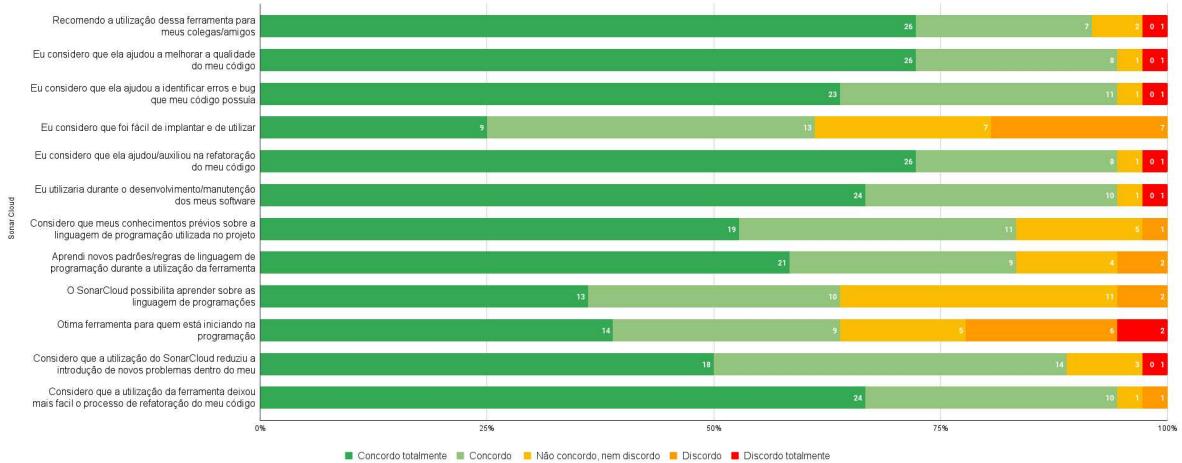
#### **9.4 Experiência com a Ferramenta Sonar Cloud**

Sobre a experiência dos estudantes com o Sonar Cloud na prática apresentada no presente trabalho, o Gráfico 26 mostra as respostas dos alunos. 36 alunos participaram da pesquisa no total.

Uma grande maioria dos estudantes (26) recomendaria a utilização do Sonar Cloud para outros colegas, indicando alta satisfação e confiança na ferramenta utilizada durante o processo. Apenas dois estudantes ficaram neutros, e nenhum discordou. Da mesma forma, 26 estudantes consideram que o Sonar Cloud ajudou a melhorar a qualidade do código, mostrando a eficácia da ferramenta na detecção e correção de problemas. Apenas um estudante ficou neutro, e nenhum discordou.

Vinte e três estudantes concordaram totalmente que o Sonar Cloud ajudou a identificar erros que não seriam facilmente encontrados em seus códigos, com mais onze concordando

Figura 26 – Respostas dos alunos sobre sua experiência com a ferramenta Sonar Cloud



Fonte: Elaborado pelo autor (2024)

parcialmente. Apenas um estudante ficou neutro, e nenhum discordou. No entanto, a facilidade de implantação e uso da ferramenta dividiu mais as opiniões: nove estudantes concordaram totalmente, e treze concordaram parcialmente, enquanto sete ficaram neutros e outros sete discordaram. Esses resultados indicam que, embora a maioria tenha achado a ferramenta acessível, há espaço para melhorar a experiência de usabilidade e também a explicação da utilização da ferramenta durante a aula de apresentação da prática.

Uma grande maioria (26) concorda totalmente que a ferramenta auxiliou na refatoração do código, com oito concordando parcialmente e apenas um neutro. Nenhum estudante discordou, destacando o papel crucial da ferramenta na melhoria contínua do *software*. Vinte e cinco estudantes afirmaram que utilizariam o Sonar Cloud em outros projetos, com mais dez concordando parcialmente. Nenhum estudante ficou neutro ou discordou, refletindo aceitação universal entre os alunos e a percepção de valor contínuo na utilização da ferramenta.

Dezenove estudantes consideraram que seus conhecimentos prévios sobre inspeção foram suficientes para utilizar o Sonar Cloud, com mais onze concordando parcialmente. Apenas cinco ficaram neutros e um discordou, indicando que a maioria se sentiu preparada, mas ainda há uma necessidade de preparação adicional para alguns. Tal dados reforça o que foi visto na afirmação sobre a facilidade de utilização do Sonar Cloud, onde é preciso melhorar a explicação da ferramenta durante a etapa de apresentação do processo. Ademais, vinte e um estudantes aprenderam novos padrões/regras de linguagem de programação com o Sonar Cloud, com mais nove concordando parcialmente. Apenas quatro ficaram neutros e dois discordaram, sublinhando o valor educacional da ferramenta.

Treze estudantes consideram que o Sonar Cloud possibilita aprender sobre as linguagens de programação utilizadas, com mais dez concordando parcialmente. Onze estudantes ficaram neutros e dois discordaram, indicando que a ferramenta pode oferecer mais *insights* específicos sobre as linguagens de programação, ponto a ser melhorado na metodologia de ensino. Quatorze estudantes acham que o Sonar Cloud é uma ótima ferramenta para iniciantes na área de desenvolvimento, com mais nove concordando parcialmente. No entanto, cinco ficaram neutros, seis discordaram e dois discordaram totalmente, sugerindo que a ferramenta pode ser melhor adaptada para iniciantes.

Dezoito estudantes concordaram totalmente que a utilização do Sonar Cloud reduziu o tempo de revisão de código, com mais quatorze concordando parcialmente. Apenas três ficaram neutros, e nenhum discordou, destacando a eficiência da ferramenta em acelerar o processo de revisão, um dos objetivos da utilização dessas ferramentas. Vinte e quatro estudantes consideram que o uso do Sonar Cloud tornou o processo de desenvolvimento mais eficiente, com mais dez concordando parcialmente. Apenas um estudante ficou neutro e um discordou, refletindo uma melhoria significativa na produtividade para esses alunos.

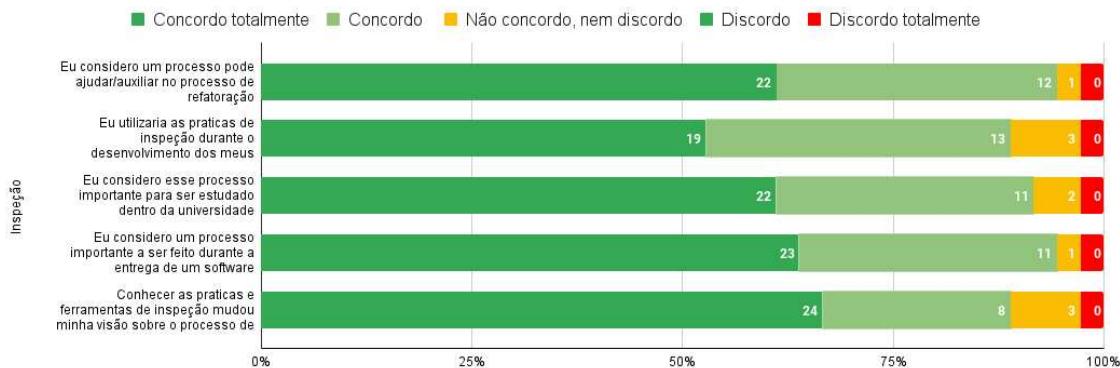
Os dados indicam que a implementação do Sonar Cloud na prática de Inspeção Contínua foi amplamente bem-sucedida e bem-recebida pelos estudantes. A maioria dos participantes reconheceu os benefícios da ferramenta em termos de melhoria da qualidade do código, identificação de erros, auxílio na refatoração e eficiência do processo de desenvolvimento. A alta taxa de recomendação e a intenção de utilizar a ferramenta em projetos futuros reforçam a percepção positiva e a aceitação do Sonar Cloud. Contudo, a análise aponta também para áreas de melhoria, especialmente na usabilidade para novos usuários e na oferta de mais *insights* específicos sobre as linguagens de programação utilizadas.

## **9.5 Experiência com o processo de Inspeção Contínua e Ferramentas de Análise de código**

O gráfico 27 ilustra os resultados do questionário em relação à experiência dos alunos sobre o processo de inspeção contínua e o uso de ferramenta de análise de código em geral.

Uma grande maioria dos estudantes, composta por 22 indivíduos, concorda totalmente que o processo de inspeção contínua pode ajudar na refatoração do código, com outros 12 concordando parcialmente. Apenas um estudante ficou neutro e nenhum discordou. Isso demonstra uma clara percepção dos benefícios dessas práticas na melhoria da qualidade do código e na diminuição na introdução de novos problemas no *software*.

Figura 27 – Respostas dos alunos sobre sua experiência com o processo de Inspeção Continua e Ferramentas de Análise de Código



Fonte: Elaborado pelo autor (2024)

Em relação à utilização das práticas de inspeção em outros projetos, 19 estudantes concordam totalmente e 13 concordam parcialmente que utilizariam essas práticas no desenvolvimento e manutenção de outros projetos e três estudantes ficaram neutros e nenhum discordou. Os dados indicam uma aceitação geral e a disposição de aplicar essas práticas em futuros trabalhos pelos alunos, um dos objetivos esperando durante a elaboração do presente trabalho, a promoção de uma cultura de revisão entre os estudantes.

A importância de aprender o processo de inspeção na universidade também foi destacada, com 22 estudantes concordando totalmente e 11 concordando parcialmente que é essencial ensinar essas práticas no ambiente acadêmico. Apenas dois estudantes ficaram neutros, e nenhum discordou, mostrando a relevância percebida da inspeção contínua como parte do currículo universitário.

Além disso, 23 estudantes concordam totalmente e 11 concordam parcialmente que o processo de inspeção é importante para ser realizado nas disciplinas de desenvolvimento de *software*. Apenas um estudante ficou neutro, o que reforça a necessidade de integrar essas práticas nas disciplinas voltadas para o desenvolvimento de *software*, reconhecendo sua contribuição significativa para a formação dos alunos.

Por fim, 24 estudantes concordam totalmente que conhecer as práticas e ferramentas de inspeção os ajudará no mercado de trabalho, com outros oito concordando parcialmente. Apenas três estudantes ficaram neutros, e nenhum discordou. A maioria dos estudantes vê essas práticas como habilidades valiosas para suas futuras carreiras, sublinhando a importância de ensinar essas competências para prepará-los melhor para o mercado de trabalho.

Os dados indicam que a implementação do processo de inspeção contínua e o uso de ferramentas de análise de código foram amplamente bem-recebidos pelos estudantes. A maioria

dos participantes reconheceu os benefícios do processo de inspeção na refatoração do código e está disposta a aplicar essas práticas em futuros projetos. Além disso, os estudantes percebem a importância de aprender essas práticas na universidade e de integrá-las nas disciplinas de desenvolvimento de *software*, reconhecendo sua relevância para o mercado de trabalho. Esses resultados reforçam a necessidade de incluir práticas de inspeção contínua no currículo acadêmico para preparar melhor os estudantes para suas carreiras profissionais.

## 10 RESULTADO DA ANÁLISE QUALITATIVA DOS DADOS

Nesta seção, é exposto à análise qualitativa dos dados coletados das perguntas abertas, acerca dos conhecimentos dos estudantes sobre inspeção contínua e ferramentas de análise antes e após as práticas feitas em sala de aula, do questionários respondidos pelos discentes matriculados na disciplina de Manutenção de *Software* do semestre 2023.2.

Com o propósito de preservar a confidencialidade dos alunos e realizar uma análise objetiva e desprovida de qualquer inclinação, as identidades individuais foram mantidas em sigilo e associadas a um identificador variável, indo de R01 a R36. Na sequência, serão descritas as respostas dos alunos, bem como alguns comentários acerca do que foi possível analisar a partir delas.

### 10.1 Conhecimentos prévios sobre inspeção contínua e ferramentas de análise

Em relação aos conhecimentos prévios sobre o processo de inspeção contínua os alunos responderam:

- R02: "É um processo constante de monitoramento para garantir a qualidade e conformidade de sistemas, identificando problemas precocemente e fazendo correções regulares. Pode ser aplicada em vários contextos, segurança e desenvolvimento de software."
- R09: "Não se aplica."
- R19: "Não tenho muito conhecimento."
- R22: "Quase nada."

Por fim, no que tange conhecimentos sobre ferramentas de inspeção de *software*, responderam:

- R09: "Não se aplica"
- R16: "Debug da IDE"
- R19: "Azure"
- R24: "Para python: black, isort, prospector"

Com base na análise dos dados coletados em relação aos conhecimentos prévios dos alunos sobre o processo de inspeção contínua e ferramentas de análise, pode-se inferir o seguinte:

1. Alunos com conhecimento avançado: O aluno R02 demonstrou um entendimento aprofundado da inspeção contínua, descrevendo-a como um processo contínuo de monitoramento

para garantir a qualidade e conformidade de sistemas. Ele também reconheceu a importância de identificar problemas precocemente e realizar correções regulares, destacando a aplicabilidade dessa prática em diversos contextos, incluindo segurança e desenvolvimento de *software*.

2. Alunos com conhecimento limitado ou inexistente: Os alunos R09 e R19 indicaram não possuir conhecimento significativo sobre o processo de inspeção contínua. R09 declarou que o tema não se aplica a ele, enquanto R19 admitiu ter um conhecimento limitado sobre o assunto.
3. Alunos com conhecimento restrito sobre ferramentas de inspeção de *software*: O aluno R16 mencionou o "Debug da IDE" como uma ferramenta de inspeção de *software*. R24 apresentou conhecimento específico ao listar algumas ferramentas para Python, incluindo *black*, *isort* e *prospector*. R09 e R19 novamente declararam que o tema não se aplica a eles.

Portanto, pode-se concluir que há uma variedade de níveis de conhecimento entre os alunos sobre inspeção contínua e ferramentas de análise. Alguns alunos demonstraram um entendimento mais aprofundado, enquanto outros têm um conhecimento limitado ou inexistente sobre o assunto. Essas informações são fundamentais para direcionar o ensino e fornecer suporte adequado aos alunos sobre o tema proposto neste trabalho.

## **10.2 Como foi o trabalho em equipe durante o processo de refatoração?**

Em relação a pergunta sobre a experiência de trabalhar em equipe no projeto de refatoração os alunos responderam:

- R02: "Foi bastante didático, por utilizar o github ficou simples de todo mundo usar e testa"
- R05: "Dividimos cada code smell cada um deveria fazer, assim ficando mais fácil e tranquilo para todos."
- R08: "Por ser a primeira vez que utilizei a ferramenta, o processo levou mais tempo do que deveria. Nem todos os membros da equipe utilizavam a linguagem escolhida, além de que a implementação do Sonar Cloud foi um pouco confusa para todos os membros no começo, pois ninguém havia utilizado ela previamente."
- R11: "Interessante, uma amostra do que seria no mercado do trabalho."
- R13: "Foi ótimo. Participei no desenvolvimento do projeto que utilizamos, e poder ver onde eu poderia melhorar o meu código durante a refatoração foi muito bom."

- R16: "A ferramenta ajudou muito nessa parte. Cada membro da equipe resolia as issues encontradas e o sonar já reconhecia a atualização quando era feito o PR."
- R24: "O trabalho em equipe durante a refatoração foi incrível. Todos contribuíram com ideias valiosas e a colaboração foi fundamental para superar desafios técnicos de forma eficiente."

Vários estudantes destacaram aspectos positivos da experiência, especialmente em relação à colaboração e divisão de tarefas. O estudante R05, por exemplo, ressaltou que a equipe dividiu as responsabilidades entre os membros da equipe, o que tornou o processo mais fácil e tranquilo para todos. Outro ponto positivo foi mencionado por R16, que destacou como a ferramenta utilizada, o Sonar Cloud, facilitou o trabalho em equipe ao reconhecer automaticamente as atualizações feitas durante os *pull requests* (PRs).

Entretanto, alguns desafios foram mencionados, como o relatado por R08, que encontrou dificuldades devido à falta de familiaridade com a ferramenta e com a linguagem de programação escolhida. Este comentário sublinha a importância de capacitar os estudantes nas ferramentas e tecnologias utilizadas atualmente e também como foi visto na análise dos dados apresentado na Figura 26 é ponto a ser melhorado na metodologia de ensino para garantir que todos os alunos participantes estejam igualmente preparados para contribuir de maneira eficaz na execução do processo de implantação das práticas de inspeção.

Por fim, a percepção de que a experiência de trabalho em equipe foi uma amostra do que seria o ambiente de trabalho no mercado (R11) é um indicativo de que o projeto não apenas ensinou habilidades técnicas, mas também preparou os estudantes para os desafios colaborativos que encontrarão em suas carreiras futuras. As respostas de R13 e R24 refletem uma experiência extremamente positiva, com um sentimento de satisfação tanto no desenvolvimento do projeto quanto na colaboração entre os membros da equipe. A capacidade de ver melhorias no próprio código (R13) e o valor das contribuições e ideias dos colegas (R24) destacam a importância do *feedback* e da colaboração contínua no processo de aprendizado e desenvolvimento de *software*.

### **10.3 Se fosse utilizar novamente as práticas e ferramentas de análise de código, o que faria diferente?**

Sobre o questionamento os alunos responderam:

- R05: "Buscaria organizar o processo de refatoração hierarquicamente, levando em conta a gravidade das etapas e o prazo final."

- R07: "Estruturaria melhor o repositório no GitHub, para separar melhor o backend do frontend."
- R010: "Utilizaria a ferramenta integrada ao visual Studio code desde o início, evitando assim, a maior quantidade possível de problemas na branch de produção."
- R13: "Dividiria de forma mais adequada os papéis de cada membro de acordo com as habilidades na utilização das ferramentas"
- R16: "Ia utilizar em sistemas maiores para ter a experiência de como a ferramenta ia responder."

A organização do processo de refatoração foi mencionada por R05, que destacou a importância de uma abordagem hierárquica, levando em conta a gravidade dos erros encontrados em seus códigos e o prazo final da entrega. Já o aluno R07 observou a necessidade de estruturar melhor o *backend* do *frontend* dos seus projetos. Esses *feedbacks* revelam a importância de uma organização clara do código-fonte, o que pode facilitar o gerenciamento do projeto e a colaboração entre os membros da equipe. Ao criar uma estrutura bem definida, é possível minimizar a introdução de novos problemas e tornar o desenvolvimento mais eficiente.

R010 sugeriu que integraria a ferramenta ao editor de texto, Visual Studio Code, desde o início do processo de refatoração para evitar problemas na *branch* principal. Com esta resposta observa-se que os alunos perceberam a importância de uma inspeção contínua do seu código, essa integração com a ferramenta de desenvolvimento facilitando a detecção e correção de problemas em tempo real, o que pode melhorar significativamente a produtividade e reduzir o tempo de *feedback*.

A divisão adequada dos papéis de acordo com as habilidades dos membros da equipe foi mencionada por R13, vimos esse mesmo pensamento no questionamento 10.2. Este ponto destaca a importância de uma boa gestão de equipe, onde as responsabilidades são distribuídas de maneira que cada membro possa utilizar suas habilidades ao máximo, esse tipo de organização pode aumentar a eficiência e garantir que todos contribuam de forma produtiva.

Por fim, R16 expressou interesse em utilizar o Sonar Cloud em sistemas maiores para experimentar como a ferramenta se comportaria. Isso demonstra uma curiosidade e uma vontade de expandir o uso das ferramentas para cenários mais complexos, o que pode proporcionar um aprendizado mais profundo e uma melhor compreensão das capacidades e limitações dessas ferramentas.

## 10.4 Qual sua opinião sobre o processo de Inspeção Contínua?

Em relação à pergunta sobre o processo de Inspeção Contínua os alunos responderam:

- R01: "Muito interessante pro mercado de trabalho."
- R03: "É um processo relevante para entrega de software com alta qualidade."
- R08: "Creio que seja uma boa opção no desenvolvimento de sistemas a curto prazo, visto que o processo torna o surgimento de bugs algo menos recorrente."
- R12: "Acho super interessante a fim de evitar o acúmulo de problemas futuros na aplicação."
- R15: "É um processo de extrema importância em qualquer desenvolvimento de software."
- R16: "É um processo essencial pra entrega do software, sendo seu estudo extremamente relevante no âmbito do nosso curso."
- R26: "Acho um processo essencial para a manutenção e desenvolvimento de um software, utilizando as técnicas aprendidas na disciplina, podemos ter menos tempo gasto em refatoração pois muitos problemas são repetidos e tendo os primeiros feedbacks da inspeção, podemos ter um cuidado em não reproduzir os mesmos problemas. Dessa forma, teremos menos retrabalho e um software com uma qualidade superior."

Os alunos R01 e R03 ressaltaram a relevância do processo de inspeção para o mercado de trabalho e a entrega de *software* de alta qualidade. Eles reconhecem que o mercado valoriza práticas que garantem a qualidade do produto final, e que o uso das práticas da inspeção contínua é fundamental para assegurar esse padrão. Os alunos R08, R12 e R26 destacaram a eficácia do processo de inspeção na prevenção de problemas futuros. Já o estudante R08 mencionou que a inspeção contínua torna o surgimento de *bugs* menos recorrente, especialmente em desenvolvimentos a curto prazo. R12 vê o processo como uma medida preventiva para evitar o acúmulo de problemas, enquanto o R26 enfatiza que a aplicação das técnicas de inspeção pode reduzir o retrabalho, economizar tempo na refatoração e resultar em um *software* de qualidade superior.

R15, R16 e R26 enfatizaram a essencialidade do processo de inspeção contínua em qualquer desenvolvimento de *software* e no contexto dos cursos de Engenharia de *Software* e Ciência da Computação. R15 considera o processo de extrema importância, enquanto R16 e R26 observam que o estudo e a aplicação da inspeção contínua são cruciais no contexto acadêmico, preparando os alunos para os desafios do mercado de trabalho e garantindo que os *softwares* desenvolvidos tenham um padrão elevado de qualidade.

## 10.5 O que você achou da ferramenta que foi apresentada?

Em relação a opinião dos alunos sobre a ferramenta apresentada os estudantes responderam:

- R02: "Muito boa! Além de apontar os potenciais problemas, ainda aprendi bastante sobre boas práticas de código."
- R05: "Muito boa, após entender melhor a interface e suas funcionalidades acredito que ela se torna indispensável no desenvolvimento de projetos grandes, pois é mais fácil identificar bugs ou más práticas de programação."
- R08: "É uma ótima opção para quem quer otimizar o tempo e garantir que a refatoração seja validada de forma segura."
- R11: "Muito interessante. Nunca tinha usado nada parecido. Foi uma experiência muito boa porque a ferramenta auxilia muito na manutenção do código."
- R13: "Achei muito boa e útil para utilizar no processo de desenvolvimento de qualquer projeto de software."
- R16: "A ferramenta foi de extrema ajuda no processo de refatoração. Tendo em vista que, a maior parte do trabalho de refatoração é entender o que o código, ou parte dele, faz para que seja realizado uma refatoração que não altere sua funcionalidade. Além de que torna-se necessário verificar o impacto da modificação daquela funcionalidade no contexto geral do sistema. Com a ferramenta, a gente abstrai muita dessas complicações pois ele já da onde está o problema, o motivo dele ser um problema e uma recomendação de solução para esse problema."
- R24: "Uma ferramenta que pode impactar o mercado, quanto mais se adaptar-se ao mundo corporativo de maneira mais prática."

Vários alunos, incluindo R02, R05, e R16, destacaram a capacidade do Sonar Cloud de identificar potenciais problemas no código, bem como de promover boas práticas de programação. R02 mencionou que, além de apontar problemas, a ferramenta ajudou a aprender sobre boas práticas de código. R05 observou que, após entender melhor a *interface* e as funcionalidades, o Sonar Cloud se torna indispensável, especialmente em projetos grandes, onde é crucial identificar *bugs* e más práticas de programação. R16 também enfatizou que a ferramenta simplifica o processo de refatoração ao indicar problemas e fornecer recomendações de solução, o que facilita a manutenção do código sem comprometer sua funcionalidade.

R08 elogiou o Sonar Cloud como uma excelente opção para quem deseja otimizar o

tempo e garantir a segurança na validação das refatorações. Essa visão é compartilhada por outros alunos, que veem a ferramenta como um recurso que agiliza o processo de desenvolvimento, permitindo que o foco seja mantido na qualidade do código. R11 e R24 refletiram sobre o impacto do Sonar Cloud em suas experiências pessoais e no mercado de trabalho. R11, por exemplo, mencionou que nunca havia utilizado uma ferramenta similar e que a experiência foi muito positiva, especialmente pelo auxílio na manutenção do código. R24, por sua vez, considerou o Sonar Cloud uma ferramenta com potencial de impacto significativo no mercado, especialmente à medida que se adapta ao mundo corporativo de forma mais prática.

R13 elogiou a ferramenta como útil para qualquer processo de desenvolvimento de *software*, destacando sua versatilidade. A opinião de R13 sugere que o Sonar Cloud não apenas se adapta bem a diferentes tipos de projetos, mas também pode ser integrado como parte essencial do ciclo de desenvolvimento, independentemente do tamanho ou escopo do projeto.

## 11 CONCLUSÕES E TRABALHOS FUTUROS

Esta seção apresenta as principais conclusões obtidas com a pesquisa, bem como sugestões para trabalhos futuros.

A implementação do processo de contínua e o uso de ferramenta de análise de código em um ambiente acadêmico mostraram-se eficazes na identificação e redução de entrada de novos problemas no projeto dos alunos e também em promover a cultura de revisão entre os estudantes. Os dados coletados usando os inventários IMI e IMMS, evidenciam uma melhoria significativa no conhecimento dos alunos sobre práticas de inspeção e refatoração de código. A metodologia de ensino escolhida também teve sucesso entre os alunos, se mostravam interessados e confortáveis durante a experiência de inspeção contínua e uso do Sonar Cloud. Além disso, o processo incentivou a colaboração em equipe e o aprimoramento das habilidades individuais, refletindo um aumento na confiança dos alunos em aplicar esses conhecimentos nos ambientes acadêmicos e profissionais.

As respostas qualitativas coletadas indicaram que os estudantes valorizaram o processo de implantação de práticas de inspeção em sala de aula, reconhecendo a importância das práticas de inspeção para garantir a entrega de *software* de alta qualidade. A ferramenta Sonar Cloud foi amplamente elogiada por sua capacidade de identificar problemas de código e promover boas práticas de codificação, sendo vista como uma adição essencial ao processo de desenvolvimento de *software*, tanto em projetos acadêmicos quanto no ambiente corporativo.

Contudo, alguns desafios foram identificados, como a necessidade de maior familiaridade pelos alunos com a ferramenta de inspeção contínua e a linguagem de programação escolhida para o projeto de refatoração. Além disso, uma melhor organização dos processos dentro da refatoração no Sonar Cloud, como a priorização das *issues* geradas pela ferramenta. Essas questões apontam para áreas de melhoria em futuras implementações do presente trabalho, sugerindo uma abordagem mais estruturada desde o início do processo de refatoração poderia facilitar o uso da ferramenta entre os alunos.

Outrossim, ponto relevante para futuros trabalhos é expandir a metodologia de ensino com guias mais detalhados sobre a utilização do Sonar Cloud, melhorar a explicação do funcionamento da ferramenta e sua várias funções pode reduzir a curva de aprendizado dos alunos e melhorar a eficiência do processo. Outra questão de melhoria, seria a análise do impacto dessas práticas na qualidade final do *software* produzido pelos estudantes, utilizando métricas mais objetivas para avaliar o código antes e depois da aplicação do processo de inspeção contínua

utilizando ferramenta de análise de código.

Esses trabalhos futuros não só consolidarão os benefícios observados até agora na implantação deste trabalho, mas também continuarão a explorar novas maneiras de melhorar o ensino e a prática do desenvolvimento de *software* dentro da universidade, garantindo que os estudantes estejam bem preparados para os desafios da indústria.

## REFERÊNCIAS

- Ackerman, A. F.; BUCHWALD, L. S.; LEWSKI, F. H. Software inspections: an effective verification process. **IEEE Software**, v. 6, p. 31–36, 1989. Disponível em: <https://api.semanticscholar.org/CorpusID:16850481>.
- Azevedo, P. T. C. C. d.; CAMINHA, M. d. F. C.; ANDRADE, C. R. S. d.; GODOY, C. G. d.; MONTEIRO, R. L. S.; FALBO, A. R. Intrinsic motivation of medical students from a college with active methodology in brazil: a cross-sectional study. **Revista Brasileira de Educação Médica**, v. 43, n. 1, p. 12–23, 2019. ISSN 0100-5502. Disponível em: <https://doi.org/10.1590/1981-5271v43suplemento1-20180229.ING>.
- Duvall, P.; MATYAS, S.; GLOVER, A. **Continuous Integration: Improving Software Quality and Reducing Risk**. First. [S. l.]: Addison-Wesley Professional, 2007. ISBN 9780321336385.
- EILERTSEN, A. M. Refactoring operations grounded in manual code changes. In: **Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Companion Proceedings**. New York, NY, USA: Association for Computing Machinery, 2020. (ICSE '20), p. 182–185. ISBN 9781450371223. Disponível em: <https://doi.org/10.1145/3377812.3381395>.
- Fagan, M. E. Design and code inspections to reduce errors in program development. **IBM Systems Journal**, v. 38, n. 2.3, p. 258–287, 1999.
- FERREIRA, A. M.; BRITO, M. A.; LIMA, J. Continuous inspection of software quality in an automotive project. In: **2023 18th Iberian Conference on Information Systems and Technologies (CISTI)**. [S. l.: s. n.], 2023. p. 1–6.
- FOWLER, M. **Refactoring: Improving the Design of Existing Code**. Addison-Wesley, 1999. Disponível em: [https://books.google.com.br/books?hl=en&lr=&id=2H1\\_DwAAQBAJ&oi=fnd&pg=PT14&dq=+Refactoring:+Improving+the+Design+of+Existing+Code&ots=NhytugkNY&sig=s\\_\\_U0BcBz3pOr-rTrj0u03MZ7NY&redir\\_esc=y#v=onepage&q=Refactoring%3A%20Improving%20the%20Design%20of%20Existing%20Code&f=false](https://books.google.com.br/books?hl=en&lr=&id=2H1_DwAAQBAJ&oi=fnd&pg=PT14&dq=+Refactoring:+Improving+the+Design+of+Existing+Code&ots=NhytugkNY&sig=s__U0BcBz3pOr-rTrj0u03MZ7NY&redir_esc=y#v=onepage&q=Refactoring%3A%20Improving%20the%20Design%20of%20Existing%20Code&f=false).
- GITHUB. **GitHub Documentation**. 2023. Acessado em 6 de novembro de 2023. Disponível em: <https://docs.github.com/pt>.
- Gomes, P. H. d. A. **Inspeção de código-fonte como subsídio para o processo de ensino e aprendizagem de qualidade de software**.: Dissertação (mestrado). São José do Rio Preto, 2021.
- Gomes, P. H. de A.; GARCIA, R. E.; SPADON, G.; ELER, D. M.; OLIVETE, C.; CORREIA, R. C. M. Teaching software quality via source code inspection tool. In: **2017 IEEE Frontiers in Education Conference (FIE)**. [S. l.: s. n.], 2017. p. 1–8.
- Humble, J.; Farley, D. **Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation**. 1st. ed. [S. l.]: Addison-Wesley Professional, 2010. ISBN 0321601912.
- IEEE Standard for Software Reviews and Audits. **IEEE Std 1028-2008**, p. 1–53, 2008.
- ISO/IEC/IEEE. Iso/iec/ieee 14764: Software engineering — software life cycle processes — maintenance. **International Standard**, p. 1–36, 2022.

Júnior, A. C.; GARCIA, V. C. S.; COELHO, D. V.; SAID, C. d. C.; STRAPASSON, A. C. P.; RESENDE, I. S. d. Tradução e adaptação transcultural do instructional materials motivation survey (imms) para o português do brasil. **Revista Brasileira de Educação Médica**, Associação Brasileira de Educação Médica, v. 44, n. 4, p. e179, 2020. ISSN 0100-5502. Disponível em: <https://doi.org/10.1590/1981-5271v44.4-20200142>.

KALINOWSKI, M.; SPÍNOLA, R. Introdução à inspeção de software - aumentando a qualidade através de verificações intermediárias. **Engenharia de Software Magazine**, I, p. 68–74, 01 2008.

LU YAO. MAO, X. W. T. Y. G. L. Z. Improving students' programming quality with the continuous inspection process: a social coding perspective. **Frontiers of Computer Science**, v. 14, 1 2020.

Oliveira, M. F. **Avaliação do Uso de serious games em educação e treinamento corporativo: um case de montadora de veículos**. Dissertação (Dissertação (Mestrado em Engenharia de Produção)) – Faculdade de Engenharia e Ciências, Universidade Estadual Paulista, Guaratinguetá, 2024.

Pressman, R. S. **Engenharia de software: uma abordagem profissional**. 7. ed. Porto Alegre: AMGH, 2011. Recurso eletrônico, tradução de Ariovaldo Griesi, revisão técnica de Reginaldo Arakaki, Julio Arakaki, Renato Manzan de Andrade. ISBN 978-85-8055-044-3.

Sommerville, I. **Engenharia de software**. Pearson Prentice Hall, 2011. ISBN 9788579361081. Disponível em: <https://books.google.com.br/books?id=H4u5ygAACAAJ>.

SonarSource. **SonarCloud Documentation**. 2023. Acessado em 6 de novembro de 2023. Disponível em: <https://docs.sonarsource.com/sonarcloud>.

THEORY, C. for S.-D. **Intrinsic Motivation Inventory (IMI)**. 2015. Disponível em: <https://selfdeterminationtheory.org/intrinsic-motivation-inventory/>.

Tomomitsu, R. H. A. **MELHORIA NA QUALIDADE DE PROJETO JAVA COM SONARQUBE**: Análise em projeto legado. Limeira, 2021.

## APÊNDICE A – QUESTIONÁRIO UTILIZADO ANTES DA IMPLANTAÇÃO DA PESQUISA

**Questão 1.** Nome Completo:

**Questão 2.** Gênero:

- (a) Masculino
- (b) Feminino

**Questão 3.** Faixa etária:

- (a) 17 a 20 anos
- (b) 21 a 23 anos
- (c) 24 a 26 anos
- (d) 27 a 30 anos
- (e) Mais de 30 anos

**Questão 4.** Curso:

- (a) Engenharia de Software
- (b) Ciências da Computação

**Questão 5.** Qual o seu nível de conhecimento atual sobre o processo de Inspeção Continua?

- (a) Nenhum
- (b) Mínimo
- (c) Básico
- (d) Intermediário
- (e) Avançado
- (f) Especialista

**Questão 6.** Qual o seu nível de conhecimento sobre esses temas: bug, vulnerability, code smell e security hotspot rules.

- (a) Nenhum
- (b) Mínimo
- (c) Básico
- (d) Intermediário
- (e) Avançado
- (f) Especialista

**Questão 7.** Você considera esse processo importante para ser estudado dentro da universidade?

- (a) Concordo totalmente
- (b) Concordo
- (c) Não estou decidido
- (d) Discordo
- (e) Discordo totalmente

**Questão 8.** Você utiliza alguma ferramenta de análise de código no seus estudos/trabalho?

- (a) Muito frequente
- (b) Frequentemente
- (c) Eventualmente
- (d) Raramente
- (e) Nunca

**Questão 9.** Você conhece alguma ferramenta para Inspeção/Analise de software?

**Questão 10.** Caso a respostas anteriores sobre o uso de ferramentas seja sim, qual o nome desta ferramenta?

**Questão 11.** Você acha que usar ferramentas de analise de código pode ajudar no processo de refatoração de um software?

- (a) Concordo totalmente
- (b) Concordo
- (c) Nem discordo, nem concordo
- (d) Discordo
- (e) Discordo totalmente

**Questão 12.** Caso você conheça sobre inspeção contínua, fale um pouco sobre seus conhecimentos.

## APÊNDICE B – TERMO ÉTICO DO QUESTIONÁRIO 1

Olá, seja bem-vindo(a)!

Esse projeto visa uma experiência onde o processo de Inspeção Contínua é introduzida no ambiente de sala de aula para melhorar a qualidade da programação dos alunos.

Com esse estudo de caso espera-se mostrar que a inspeção contínua pode auxiliar os alunos a identificar seus maus hábitos de codificação, identificar possíveis bugs já implantados no código e evitar a introdução de mais erros durante o processo de refatoração. Por fim assim, reduzir significativamente a densidade de problemas de qualidade introduzidos no código.

Os dados serão utilizados de forma confidencial e apenas para fins acadêmicos e científicos. Caso você tenha alguma dúvida ou necessite de qualquer esclarecimento, por favor, entre em contato com as pesquisadoras abaixo a qualquer tempo.

Pesquisadora responsável pela pesquisa: Profa. Dra. Jacilane de Holanda Rabelo (jacilane.rabelo@ufc.br) Pesquisadora: Antônia Naelly Freire de Lima (naellyfreire@alu.ufc.br)  
Universidade Federal do Ceará (UFC) - Campus Russas Rua: Felipe Santiago, 411 - Cidade Universitária, Russas - CE.

Agradecemos desde já pela colaboração.

Você poderá desistir da pesquisa a qualquer momento. Ao clicar na opção "Próxima" com a opção "Sim" selecionada, você afirma que entendeu os termos apresentados.

## APÊNDICE C – QUESTIONÁRIO UTILIZADO APÓS A IMPLANTAÇÃO DA PESQUISA

**Questão 1.** Nome Completo:

**Questão 2.** Curso:

- (a) Engenharia de *Software*
- (b) Ciências da Computação

**Questão 3.** Gênero:

- (a) Masculino
- (b) Feminino
- (c) Prefiro não opinar
- (d) Outros...

**Questão 4.** Faixa etária:

- (a) 17 a 20 anos
- (b) 21 a 23 anos
- (c) 24 a 26 anos
- (d) 27 a 30 anos
- (e) Mais de 30 anos

**Questão 5.** Enquanto eu estava fazendo a atividade de refatoração, eu estava pensando o quanto estava aproveitando.

- (a) Concordo totalmente
- (b) Concordo
- (c) Não estou decidido
- (d) Discordo
- (e) Discordo totalmente

**Questão 6.** Eu não me sentia nervoso(a) enquanto fazia a atividade de refatoração.

- (a) Concordo totalmente
- (b) Concordo
- (c) Não estou decidido
- (d) Discordo
- (e) Discordo totalmente

**Questão 7.** Eu senti que foi minha escolha fazer essa atividade de inspeção.

- (a) Concordo totalmente
- (b) Concordo
- (c) Não estou decidido
- (d) Discordo
- (e) Discordo totalmente

**Questão 8.** Eu acho que sou muito bom nas atividades de refatoração.

- (a) Concordo totalmente
- (b) Concordo
- (c) Não estou decidido
- (d) Discordo
- (e) Discordo totalmente

**Questão 9.** Achei que as atividades de Inspeção Continua foi bem agradável.

- (a) Concordo totalmente
- (b) Concordo
- (c) Não estou decidido
- (d) Discordo
- (e) Discordo totalmente

**Questão 10.** Eu me senti tenso durante as atividades de refatoração.

- (a) Concordo totalmente
- (b) Concordo
- (c) Não estou decidido
- (d) Discordo
- (e) Discordo totalmente

**Questão 11.** Fazer as atividades de refatoração foi divertida.

- (a) Concordo totalmente
- (b) Concordo
- (c) Não estou decidido
- (d) Discordo
- (e) Discordo totalmente

**Questão 12.** Eu gostei muito de fazer as atividade de refatoração.

- (a) Concordo totalmente

- (b) Concordo
- (c) Não estou decidido
- (d) Discordo
- (e) Discordo totalmente

**Questão 13.** Eu estou satisfeito (a) com o meu desempenho nas atividades de refatoração

- (a) Concordo totalmente
- (b) Concordo
- (c) Não estou decidido
- (d) Discordo
- (e) Discordo totalmente

**Questão 14.** Eu achei as atividades de refatoração muito chata.

- (a) Concordo totalmente
- (b) Concordo
- (c) Não estou decidido
- (d) Discordo
- (e) Discordo totalmente

**Questão 15.** Eu achei as atividades de Inspeção Continua muito interessante.

- (a) Concordo totalmente
- (b) Concordo
- (c) Não estou decidido
- (d) Discordo
- (e) Discordo totalmente

**Questão 16.** Depois de fazer as atividades de Inspeção Continua por um tempo, me senti muito competente.

- (a) Concordo totalmente
- (b) Concordo
- (c) Não estou decidido
- (d) Discordo
- (e) Discordo totalmente

**Questão 17.** Eu achei o processo de Inspeção Continua muito importante para minha vida acadêmica e profissional.

- (a) Concordo totalmente
- (b) Concordo
- (c) Não estou decidido
- (d) Discordo
- (e) Discordo totalmente

**Questão 18.** Aprendi algumas coisas surpreendentes ou inesperadas durante o processo de Inspeção Contínua.

- (a) Concordo totalmente
- (b) Concordo
- (c) Não estou decidido
- (d) Discordo
- (e) Discordo totalmente

**Questão 19.** O conteúdo e o estilo de escrita neste material dão a impressão de que vale a pena conhecê-lo.

- (a) Concordo totalmente
- (b) Concordo
- (c) Não estou decidido
- (d) Discordo
- (e) Discordo totalmente

**Questão 20.** Havia histórias, figuras ou exemplos que me mostraram como esse processo de Inspeção poderia ser importante para algumas pessoas.

- (a) Concordo totalmente
- (b) Concordo
- (c) Não estou decidido
- (d) Discordo
- (e) Discordo totalmente

**Questão 21.** Esses materiais que envolvem o processo de inspeção são interessantes.

- (a) Concordo totalmente
- (b) Concordo
- (c) Não estou decidido
- (d) Discordo
- (e) Discordo totalmente

**Questão 22.** Gostei muito de estudar essa atividade.

- (a) Concordo totalmente
- (b) Concordo
- (c) Não estou decidido
- (d) Discordo
- (e) Discordo totalmente

**Questão 23.** Gostei tanto do *SonarCloud* que gostaria de saber mais sobre ele.

- (a) Concordo totalmente
- (b) Concordo
- (c) Não estou decidido
- (d) Discordo
- (e) Discordo totalmente

**Questão 24.** Concluir o trabalho de refatoração nesta disciplina usando práticas de inspeção me deu um sentimento satisfatório de realização.

- (a) Concordo totalmente
- (b) Concordo
- (c) Não estou decidido
- (d) Discordo
- (e) Discordo totalmente

**Questão 25.** Recomendo a utilização dessa ferramenta para meus colegas/amigos.

- (a) Concordo totalmente
- (b) Concordo
- (c) Não estou decidido
- (d) Discordo
- (e) Discordo totalmente

**Questão 26.** Eu considero que ela ajudou a identificar erros e *bug* que meu código possuía.

- (a) Concordo totalmente
- (b) Concordo
- (c) Não estou decidido
- (d) Discordo
- (e) Discordo totalmente

**Questão 27.** Eu considero que foi fácil de implantar e de utilizar.

- (a) Concordo totalmente
- (b) Concordo
- (c) Não estou decidido
- (d) Discordo
- (e) Discordo totalmente

**Questão 28.** Eu considero que ela ajudou/auxiliou na refatoração do meu código.

- (a) Concordo totalmente
- (b) Concordo
- (c) Não estou decidido
- (d) Discordo
- (e) Discordo totalmente

**Questão 29.** Eu utilizaria durante o desenvolvimento/manutenção dos meus *software*.

- (a) Concordo totalmente
- (b) Concordo
- (c) Não estou decidido
- (d) Discordo
- (e) Discordo totalmente

**Questão 30.** Considero que meus conhecimentos prévios sobre a linguagem de programação utilizada no projeto influenciou na minha experiência com a ferramenta do *SonarCloud*.

- (a) Concordo totalmente
- (b) Concordo
- (c) Não estou decidido
- (d) Discordo
- (e) Discordo totalmente

**Questão 31.** Aprendi novos padrões/regras de linguagem de programação durante a utilização da ferramenta.

- (a) Concordo totalmente
- (b) Concordo
- (c) Não estou decidido
- (d) Discordo
- (e) Discordo totalmente

**Questão 32.** O *SonarCloud* possibilita aprender sobre as linguagem de programações.

- (a) Concordo totalmente
- (b) Concordo
- (c) Não estou decidido
- (d) Discordo
- (e) Discordo totalmente

**Questão 33.** Otima ferramenta para quem está iniciando na programação.

- (a) Concordo totalmente
- (b) Concordo
- (c) Não estou decidido
- (d) Discordo
- (e) Discordo totalmente

**Questão 34.** Considero que a utilização do SonarCloud reduziu a introdução de novos problemas dentro do meu código.

- (a) Concordo totalmente
- (b) Concordo
- (c) Não estou decidido
- (d) Discordo
- (e) Discordo totalmente

**Questão 35.** Considero que a utilização da ferramenta deixou mais facil o processo de refatoração do meu código.

- (a) Concordo totalmente
- (b) Concordo
- (c) Não estou decidido
- (d) Discordo
- (e) Discordo totalmente

**Questão 36.** Eu considero um processo pode ajudar/auxiliar no processo de refatoração.

- (a) Concordo totalmente
- (b) Concordo
- (c) Não estou decidido
- (d) Discordo
- (e) Discordo totalmente

**Questão 37.** Eu considero esse processo importante para ser estudado dentro da universidade.

- (a) Concordo totalmente
- (b) Concordo
- (c) Não estou decidido
- (d) Discordo
- (e) Discordo totalmente

**Questão 38.** Eu considero um processo importante a ser feito durante a entrega de um software.

- (a) Concordo totalmente
- (b) Concordo
- (c) Não estou decidido
- (d) Discordo
- (e) Discordo totalmente

**Questão 39.** Conhecer as práticas e ferramentas de inspeção mudou minha visão sobre o processo de refatoração.

- (a) Concordo totalmente
- (b) Concordo
- (c) Não estou decidido
- (d) Discordo
- (e) Discordo totalmente

**Questão 40.** Qual o nível de conhecimento você tem sobre a linguagem utilizada no projeto escolhido pela sua equipe?

- (a) **Nenhum** - nunca ouvi falar disso;
- (b) **Mínimo** - ouço falar, mas nunca uso;
- (c) **Básico** - tenho conhecimentos gerais, mas quase nunca os uso;
- (d) **Intermediário** - tenho conhecimento médio e às vezes uso;
- (e) **Avançado** - tenho um conhecimento profundo e costumo usar este;
- (f) **Especialista** - Eu sou um especialista no assunto e uso quase todos os dias.

**Questão 41.** Qual o nível de conhecimento você tem sobre a linguagem utilizada no projeto escolhido pela sua equipe?

- (a) **Nenhum** - nunca ouvi falar disso;
- (b) **Mínimo** - ouço falar, mas nunca uso;
- (c) **Básico** - tenho conhecimentos gerais, mas quase nunca os uso;

- (d) **Intermediário** - tenho conhecimento médio e às vezes uso;
- (e) **Avançado** - tenho um conhecimento profundo e costumo usar este;
- (f) **Especialista** - Eu sou um especialista no assunto e uso quase todos os dias.

**Questão 42.** Qual o seu nível de conhecimento sobre esses temos: *bug, vulnerability, code smell e security hotspot rules*.

- (a) **Nenhum** - nunca ouvi falar disso;
- (b) **Mínimo** - ouço falar, mas nunca uso;
- (c) **Básico** - tenho conhecimentos gerais, mas quase nunca os uso;
- (d) **Intermediário** - tenho conhecimento médio e às vezes uso;
- (e) **Avançado** - tenho um conhecimento profundo e costumo usar este;
- (f) **Especialista** - Eu sou um especialista no assunto e uso quase todos os dias.

**Questão 43.** Como foi o trabalho em equipe durante o processo de refatoração?

**Questão 44.** Se fosse utilizar novamente as práticas e ferramentas de análise de código, o que faria diferente?

**Questão 45.** Qual sua opinião sobre o processo de Inspeção Continua?

**Questão 46.** O que você achou da ferramenta que foi apresentada?

## APÊNDICE D – TERMO ÉTICO DO QUESTIONÁRIO 2

Olá, seja bem-vindo(a)!

Para finalizar a implantação do processo de Inspeção Continua em sala de aula, gostaríamos de solicitar sua avaliação para promover melhorias ao estudo.

A análise da pesquisa é conduzida de maneira anônima. No entanto, para a atribuição dos pontos, solicita-se o fornecimento do nome completo do aluno. Essa seção (doravante "Termo de Consentimento Livre e Esclarecido") visa registrar a manifestação livre e inequívoca pela qual o Titular concorda com o tratamento dos seus dados pessoais para a finalidade específica, em conformidade com a Lei nº 13.709 – Lei Geral de Proteção de Dados Pessoais (LGPD).

Ao manifestar a sua aceitação ao presente Termo, o Titular consente e concorda com as operações que se referem à coleta, produção, recepção, classificação, utilização, ao acesso, à reprodução, transmissão, distribuição, ao processamento, arquivamento, armazenamento, avaliação ou ao controle da informação, à modificação, comunicação, transferência, difusão e extração dos referidos dados de forma anônima.

Pesquisadora responsável pela pesquisa: Profa. Dra. Jacilane de Holanda Rabelo ([jacilane.rabelo@ufc.br](mailto:jacilane.rabelo@ufc.br)) Pesquisadora: Antônia Naelly Freire de Lima ([naellyfreire@alu.ufc.br](mailto:naellyfreire@alu.ufc.br)) Universidade Federal do Ceará (UFC) - Campus Russas Rua: Felipe Santiago, 411 - Cidade Universitária, Russas - CE.

Agradecemos desde já pela colaboração.