



Programa de Pós-Graduação em

# **Computação Aplicada**

**Mestrado/Doutorado Acadêmico**

Ed Wilson Rodrigues Silva Júnior

Estudos Empíricos Sobre o Uso de Gamificação em Modelagem de  
Software Com UML

São Leopoldo, 2025

Ed Wilson Rodrigues Silva Júnior

ESTUDOS EMPÍRICOS SOBRE O USO DE GAMIFICAÇÃO EM MODELAGEM DE  
SOFTWARE COM UML

Tese apresentada como requisito para a  
obtenção do título de Doutor pelo Programa de  
Pós-Graduação em Computação Aplicada da  
Universidade do Vale do Rio dos Sinos —  
UNISINOS

Orientador:  
Prof. Dr. Kleinner Silva Farias de Oliveira

São Leopoldo  
2025

S586e

Silva Júnior, Ed Wilson Rodrigues.

Estudos empíricos sobre o uso de gamificação em modelagem de software com UML / por Ed Wilson Rodrigues Silva Júnior. – 2025.

179 f. : il. ; 30 cm.

Tese (doutorado) — Universidade do Vale do Rio dos Sinos, Programa de Pós-Graduação em Computação Aplicada, São Leopoldo, RS, 2025.

"Orientador: Dr. Kleinner Silva Farias de Oliveira".

1. Linguagem de modelagem unificada (UML).  
2. Gamificação. 3. Estudo experimental. 4. Modelo de qualidade. 5. Modelagem de software. I. Título.

CDU: 004.415.2.045

## RESUMO

A gamificação tem se destacado como uma estratégia promissora para aumentar o engajamento e a motivação em diferentes domínios, incluindo a Engenharia de Software, onde a modelagem com UML permanece como uma atividade central, mas enfrenta desafios recorrentes relacionados à adoção e ao uso efetivo. Apesar de seu potencial, pouco se sabe sobre como elementos gamificados podem apoiar o processo de modelagem ou melhorar a qualidade dos modelos produzidos. Neste contexto, esta tese investiga três lacunas principais: a ausência de conhecimento empírico sobre o uso da UML na indústria, a inexistência de um modelo de qualidade para avaliar atividades de modelagem gamificadas e a escassez de evidências sobre os impactos da gamificação na qualidade dos modelos UML. O objetivo geral é produzir conhecimento empírico sobre o uso da gamificação em modelagem de software, propondo mecanismos para avaliar os modelos gerados e analisando os efeitos de técnicas gamificadas na aprendizagem e na qualidade dos artefatos. Para isso, foram conduzidos um survey com profissionais de TI, o desenvolvimento e avaliação de um modelo de qualidade gamificado e uma série de estudos empíricos — incluindo experimentos controlados e um estudo de caso — para investigar atributos como completude, consistência, motivação e profundidade analítica. Os resultados indicam que, embora amplamente conhecida, a UML ainda enfrenta barreiras de adoção associadas à cultura organizacional e à complexidade da linguagem; mostram também que o modelo de qualidade proposto é percebido como útil para apoiar a aprendizagem; e evidenciam que elementos gamificados podem melhorar o engajamento, a diversidade dos artefatos produzidos e a precisão na detecção de inconsistências. Conclui-se que a gamificação possui potencial para aprimorar a prática e o ensino da modelagem de software, favorecendo a geração de modelos UML mais completos e robustos, além de contribuir para a motivação dos participantes.

**Palavras-chave:** Linguagem de Modelagem Unificada. Gamificação. Estudo experimental. Modelo de qualidade.

## ABSTRACT

Gamification has emerged as a promising strategy to increase engagement and motivation across different domains, including Software Engineering, where UML modeling remains a central activity but faces recurring challenges related to adoption and effective use. Despite its potential, little is known about how gamified elements can support the modeling process or improve the quality of the produced models. In this context, this thesis investigates three main gaps: the lack of empirical knowledge on the use of UML in industry, the absence of a quality model to evaluate gamified modeling activities, and the scarcity of evidence regarding the effects of gamification on the quality of UML models. The general objective is to produce empirical knowledge on the use of gamification in software modeling, proposing mechanisms to evaluate the models generated and analyzing the effects of gamified techniques on learning and on the quality of the artifacts. To achieve this, a survey with IT professionals was conducted, followed by the development and evaluation of a gamified quality model, and a series of empirical studies—including controlled experiments and a case study—aimed at investigating attributes such as completeness, consistency, motivation, and analytical depth. The results indicate that, although well known, UML still encounters adoption barriers associated with organizational culture and language complexity; they also show that the proposed quality model is perceived as useful to support learning; and they provide evidence that gamified elements can improve engagement, the diversity of the produced artifacts, and the accuracy in detecting inconsistencies. It is concluded that gamification has the potential to enhance both the practice and teaching of software modeling, supporting the creation of more complete and robust UML models and contributing to increased participant motivation.

**Keywords:** Unified Model Language. Gamification. Experimental study. Quality model.

## LISTA DE FIGURAS

1	Diagramas UML	28
2	Exemplos de elementos gamificados	34
3	Processo experimental	42
4	Resultados sobre fatores que afetam o uso da UML (QP1.1)	46
5	Resultados sobre dificuldade de uso da UML (QP1.2)	49
6	Resultados dos benefícios percebidos sobre o uso da UML (QP1.3)	50
7	Resultados sobre a frequência de uso da UML (QP1.4)	51
8	Resultados sobre contextos de uso (QP1.5)	53
9	Resultados sobre visão de modelagem (QP1.6)	55
10	Estrutura Genérica do Modelo de Qualidade	69
11	Sintaxe Abstrata do Modelo de Qualidade	70
12	Processo Experimental	77
13	Processo experimental	102
14	Tela de perguntas gamificadas	104
15	Tela do dashboard usado no estudo.	105
16	Processo experimental	139
17	Painel de conquistas utilizado para acompanhamento em tempo real do progresso	140
18	Leaderboard exibido para incentivar a competição saudável e reforçar o feedback contínuo durante os sprints	141
19	Pontuação total das equipes	145
20	Comparação das métricas antes e após a gamificação	149
21	Gráfico de dispersão entre pontuação total e número de artefatos	153
22	Relação entre visualizações do leaderboard e frequência de participação nas cerimônias Scrum	155
23	Frequência de inconsistências entre diagramas por equipe	157

## LISTA DE TABELAS

1	Análise comparativa dos trabalhos relacionados selecionados	39
2	Questões de Pesquisa investigadas neste capítulo	41
3	Dados de perfil dos participantes.	45
4	Análise Comparativa dos Trabalhos Relacionados	68
5	Dados de perfil dos participantes.	79
6	Dados coletados relacionados ao questionário TAM.	81
7	Visão geral das inconsistências utilizadas no estudo	88
8	Resumo dos trabalhos relacionados sobre gamificação e detecção de inconsistências em modelagem de software.	95
9	Características das perguntas utilizadas no estudo.	103
10	Conjunto de dados coletados nas tarefas experimentais.	106
11	Resultados dos testes estatísticos.	108
12	Resultados empíricos e suas implicações na pesquisa	113
13	Achados empíricos e implicações para pesquisa e para o design de ferramentas.	116
14	Agenda de pesquisa sobre gamificação no ensino de modelagem conceitual	121
15	Resumo das ameaças à validade e das estratégias de mitigação	124
16	Caracterização dos participantes e do contexto do estudo	136
17	Variáveis, critérios e métricas de análise	138
18	Pontuação total obtida por cada equipe participante do estudo de caso	144
19	Comparação antes e depois da gamificação (por equipe e indicador)	149
20	Estatística descritiva das variáveis principais (n = 9 equipes)	150
21	Estatística descritiva da Motivação (1–5), antes e após a gamificação (n = 30)	150

## **LISTA DE SIGLAS**

UML	Unified Model Language
OO	Object-oriented
SE	Software Engineering
OMG	Object Management Group
GE	General Electric
OOSE	Object-Oriented Software Engineering
OMT	Object Modeling Technique
PBL	Points, Badges and Leaderboards
TI	Tecnologia de Informação
BPMN	Business Process Modeling Notation
OSS	Open Source Software
CC	Critérios de Comparação
API	Application Programming Interface
RUP	Rational Unified Process
TAM	Technology Acceptance Model



## SUMÁRIO

<b>1 INTRODUÇÃO</b>	<b>17</b>
1.1 Definição do Problema	18
1.2 Objetivo e Questões de Pesquisa	19
1.3 Contribuições da Tese	21
1.4 Estrutura da Tese	23
<b>2 FUNDAMENTAÇÃO TEÓRICA</b>	<b>25</b>
2.1 Modelagem de Software	25
2.2 Unified Modeling Language	26
2.2.1 Diagramas de Estrutura	29
2.2.2 Diagramas de Comportamento	29
2.3 Gamificação	30
2.3.1 Elementos da Gamificação	32
<b>3 UM SURVEY SOBRE O USO DE UML NA INDÚSTRIA</b>	<b>35</b>
3.1 Trabalhos Relacionados	36
3.1.1 Análise dos Trabalhos Relacionados	37
3.1.2 Análise Comparativa e Oportunidades	39
3.2 Metodologia	40
3.2.1 Objetivo e Questões de Pesquisa	40
3.2.2 Processo Experimental	40
3.2.3 Questionário Formulado	42
3.3 Resultados	43
3.3.1 Análise do Perfil dos Participantes	43
3.3.2 QP1.1: Fatores Que Afetam o Uso Eficaz da UML	44
3.3.3 QP1.2: O Que Dificulta a Utilização da UML	47
3.3.4 QP1.3: Quais Benefícios São Percebidos ao Usar a UML	49
3.3.5 QP1.4: A Frequência de Uso da UML	51
3.3.6 QP1.5: Quais as Principais Características da Atual Demanda de Desenvolvimento de Software Limitam o Uso da UML	52
3.3.7 QP1.6: Qual a Visão dos Desenvolvedores em Relação à Modelagem Utilizando UML	54
3.4 Discussão Adicional	56
3.4.1 Análises dos Resultados	56
3.4.2 Adoção de Modelagem Contínua	57
3.4.3 Gamificação e Modelagem de Software	59
3.5 Ameaças à Validade	60
3.6 Considerações Finais do Capítulo	60
<b>4 MODELGAME: MODELO DE QUALIDADE GAMIFICADO PARA APRENDIZAGEM DE MODELAGEM DE SOFTWARE</b>	<b>63</b>
4.1 Background	64
4.1.1 Gamificação e Ensino de Engenharia de Software	64
4.1.2 Modelagem de Software e Qualidade de Modelo	65
4.2 Trabalhos Relacionados	65
4.2.1 Análise de Obras Relacionadas	66
4.2.2 Análise Comparativa e Oportunidades	67

<b>4.3 Modelo de Qualidade</b>	68
<b>4.4 Modelo de Qualidade Proposto</b>	69
4.4.1 Sintaxe Abstrata	70
4.4.2 Noções de Qualidade	73
<b>4.5 Avaliação</b>	74
4.5.1 Objetivo e Questões de Pesquisa	74
4.5.2 Questionário	75
4.5.3 Seleção de Participantes	75
4.5.4 Processo Experimental	76
4.5.5 Análise de Resultado	78
<b>4.6 Ameaças à Validade</b>	80
<b>4.7 Considerações Finais do Capítulo</b>	82
<b>5 ESTUDO EMPÍRICO SOBRE O IMPACTO DA GAMIFICAÇÃO NA MODELAGEM DE SOFTWARE COM UML</b>	<b>85</b>
<b>5.1 Background</b>	87
5.1.1 Inconsistências em Modelos UML	87
5.1.2 Gamificação em Modelagem de Software	89
<b>5.2 Trabalhos Relacionados</b>	90
5.2.1 Análise dos Trabalhos Relacionados	90
5.2.2 Análise Comparativa e Oportunidades	96
<b>5.3 Metodologia</b>	97
5.3.1 Objetivo e Questões de Pesquisa	97
5.3.2 Formulação de Hipóteses	98
5.3.3 Seleção dos Participantes	100
5.3.4 Variáveis e Método de Quantificação	100
5.3.5 Processo Experimental	101
5.3.6 Operação	102
5.3.7 Tarefas Experimentais	103
5.3.8 Pacote de Replicação	105
<b>5.4 Resultados</b>	105
5.4.1 QP3.1: Taxa de Gamificação e Detecção	106
5.4.2 QP3.2: Taxa de Esforço de Gamificação e Detecção	108
5.4.3 QP3.3: Gamificação e Taxa de má interpretação	110
<b>5.5 Discussão</b>	112
5.5.1 Usando a Teoria da Autodeterminação	112
5.5.2 Equilibrando Engajamento e Precisão	114
5.5.3 Sobre o Impacto da Gamificação nas Taxas de Má Interpretação	115
5.5.4 Implicações para a Pesquisa sobre Gamificação em Engenharia Baseada em Modelos	117
5.5.5 Desafios e Direções para Pesquisas Futuras	120
5.5.6 Limitações do Estudo	122
<b>5.6 Ameaças à Validade</b>	123
<b>5.7 Considerações Finais do Capítulo</b>	124
<b>6 GAMIFICAÇÃO NO ENSINO DE ENGENHARIA DE SOFTWARE: UM ESTUDO DE CASO SOBRE SUA APLICAÇÃO NO DESENVOLVIMENTO ÁGIL E NA MODELAGEM UML</b>	<b>127</b>
<b>6.1 Fundamentação Teórica</b>	128

6.1.1 Inconsistências em Modelos UML	129
6.1.2 Gamificação no Ensino de Engenharia de Software	130
6.1.3 Limitações dos Estudos Anteriores	130
6.1.4 Justificativa Para o Estudo de Caso	132
<b>6.2 Metodologia</b>	133
6.2.1 Objetivo e Questões de Pesquisa	134
6.2.2 Hipóteses	134
6.2.3 Participantes e Contexto	135
6.2.4 Variáveis e Critérios de Análise	137
6.2.5 Desenho do Estudo e Fases do Processo	138
6.2.6 Materiais Utilizados e Replicabilidade	142
6.2.7 Análise dos Dados	142
6.2.8 Considerações Éticas	143
<b>6.3 Resultados e Discussão</b>	143
6.3.1 Análise Comparativa dos Cenários Antes e Após o Uso da Gamificação	146
6.3.2 QP3.4: A Gamificação Contribui Para o Aumento da Completude e Diversidade dos Artefatos Produzidos?	151
6.3.3 QP3.5: A Gamificação Melhora o Engajamento e a Colaboração Entre os Membros das Equipes?	154
6.3.4 QP3.6: A Gamificação Afeta a Profundidade Analítica da Modelagem UML?	156
<b>6.4 Ameaças à Validade</b>	158
<b>6.5 Considerações Finais do Capítulo</b>	159
<b>7 CONCLUSÕES E TRABALHOS FUTUROS</b>	161
7.1 Artigos Publicados	162
<b>REFERÊNCIAS</b>	163
<b>APÊNDICE A - ARTIGOS PUBLICADOS</b>	179

## 1 INTRODUÇÃO

Ao longo dos anos, os avanços ocorridos na Engenharia de Software (Software Engineering – SE) fizeram com que novas tecnologias fossem desenvolvidas, acompanhadas por técnicas e modelos que poderiam ajudar no desenvolvimento de softwares de alta qualidade (ROMEO et al., 2025; MARAND et al., 2025; OZKAYA; ERATA, 2020; BOOCH et al., 2008). Diante disso, a modelagem de software se tornou uma atividade importante para que o desenvolvimento pudesse ocorrer de forma aderente aos requisitos estabelecidos pelo solicitante, ela abrange um conjunto de princípios, conceitos e práticas que levam ao desenvolvimento de um sistema ou produto de alta qualidade. Os princípios desta atividade estabelecem práticas que guiam todo o processo de desenvolvimento de software, por exemplo, documentar as decisões do projeto e promover uma melhor comunicação entre as equipes.

A UML (Linguagem de Modelagem Unificada) foi criada para modelar sistemas em diversas linguagens de programação, sendo uma unificação gerada a partir dos principais métodos de modelagem de software e fornecendo design de notação comum que cobre a análise de software à implantação de software (OMG, 2017; BOOCH, 1999; BOOCH et al., 2008; RUMBAUGH et al., 1991). Essa é uma linguagem padrão para modelagem orientada a objetos que surgiu da fusão de três grandes métodos: o BOOCH, o OMT (Rumbaugh) e o OOSE (Jacobson).

O método BOOCH definiu a noção de que um sistema é analisado a partir de um número de visões, onde cada visão é descrita por um número de modelos e diagramas. O OMT (*Object Modeling Technique*) foi desenvolvido pela *General Electric* (GE) e é especialmente voltado para o teste dos modelos, baseado nas especificações da análise de requisitos do sistema e é muito utilizado para criar modelos de sistemas de software. O método OOSE (*Object-Oriented Software Engineering*) é utilizado para descrever o sistema.

A linguagem UML não possui um processo que define como o trabalho deve ser desenvolvido; ou seja, ela apenas indica que tipo de trabalho deve ser feito. Seu papel é fornecer mecanismos formais e padronizados para especificar, visualizar e documentar aspectos estruturais e comportamentais de sistemas de software, apoiando a comunicação, a análise e o design ao longo do desenvolvimento. A UML possui elementos visuais e textuais que permitem a especificação de diversos aspectos sobre como sistemas de software devem se estruturar e se comportar, incluindo relações entre componentes, fluxos de interação e fatores temporais (ROBLES et al., 2017; MASON; RENNIE, 2006; OMG, 2017). Ela é composta por diagramas classificados em diagramas estruturais e diagramas comportamentais. Os diagramas estruturais, tratam o aspecto estrutural tanto do ponto de vista do sistema quanto das classes. Existem para visualizar, especificar, construir e documentar os aspectos estáticos de um sistema, ou seja, a representação de seu esqueleto e estruturas “relativamente estáveis”. Os aspectos estáticos de um sistema de software abrangem a existência e a colocação de itens como classes, interfaces, colaborações, componentes.

Os diagramas comportamentais são voltados para realizar a descrição do sistema computa-

cional modelado quando em execução, isto é, como a modelagem dinâmica do sistema. Eles são usados para visualizar, especificar, construir e documentar os aspectos dinâmicos de um sistema que e a representação das partes que “sofrem alterações”, como por exemplo o fluxo de mensagens ao longo do tempo e a movimentação física de componentes em uma rede.

Aprender a usar todo o potencial da UML pode ser uma tarefa complexa, devido às dificuldades teóricas e práticas relacionadas. Isso pode fazer com que os desenvolvedores se sintam desestimulados e cada vez menos engajados com o passar do tempo e isto poderia acarretar, por exemplo, no desenvolvimento de modelos incompletos, descontextualizados e com má qualidade. (LANGE; CHAUDRON; MUSKENS, 2006; FEICHAS; SEABRA, 2023; MARCHEZAN et al., 2023a) reforça que estes defeitos trazem riscos potenciais que podem causar má interpretação e falha de comunicação, ocasionando prejuízos à qualidade dos softwares. Portanto, encontrar formatos que favoreçam o aprendizado do desenvolvedor, gere engajamento e consequentemente a geração de modelos UML cada vez mais eficazes, pode se tornar um dos principais desafios encontrados na indústria atualmente. Diante deste cenário, surge a gamificação como uma possível alternativa para mitigar estes problemas, potencializando a adoção da UML, melhorando os modelos gerados pelos desenvolvedores e consequente gerando softwares de alta qualidade.

## 1.1 Definição do Problema

Atualmente a gamificação tem sido empregada em diversas áreas como forma de melhorar processos através do engajamento de seus usuários, elevando suas emoções (SANTOS et al., 2024; WEBB, 2013; GARACCIONE; COPPOLA; ARDITO, 2024). As emoções são conhecidas por afetar a capacidade cognitiva e, portanto, o desempenho no trabalho. O mesmo vale para o desenvolvimento de software, que exige criatividade e habilidades na resolução de problemas. Em (GIRARDI et al., 2021) são fornecidas evidências empíricas de que existe uma ligação entre as emoções e a produtividade percebida no local de trabalho, além de apresentar uma taxonomia de gatilhos para as emoções positivas e negativas dos desenvolvedores.

Apesar da gamificação ser um tema de interesse recente, alguns estudos apontam dificuldades e desafios para sua implementação (PAULA PORTO et al., 2021; MARÍN, 2021; RODRIGUES; SOUZA; FIGUEIREDO, 2018; PEDREIRA et al., 2015). Em determinados contextos, especialmente em sistemas de maior criticidade ou que demandam especificações mais rigorosas, tarefas como a análise detalhada de requisitos e a verificação manual de múltiplos diagramas UML podem se tornar exaustivas e propensas a erros. Esses processos, quando realizados de forma repetitiva e sob pressão de prazos, podem reduzir o engajamento dos desenvolvedores e comprometer a qualidade dos modelos produzidos. Nesse contexto, a gamificação pode ser vista como um recurso com grande potencial que pode melhorar o interesse e a motivação dos desenvolvedores na realização de suas práticas. Este estudo identificou alguns problemas (P), sendo eles:

**P1: Ausência do conhecimento empírico sobre o uso da UML nas empresas.** A literatura atual ainda carece de uma ampla e compreensão exploratória das percepções dos desenvolvedores sobre fatores que afetam ou mesmo comprometem a adoção da modelagem UML em projetos do mundo real (LI et al., 2025; AKDUR; SAY; DEMIRÖRS, 2021; FERNÁNDEZ-SÁEZ; CHAUDRON; GENERO, 2018; CHAUDRON; HEIJSTEK; NUGROHO, 2012). Desta forma, existe a necessidade de fornecer uma visão de como está o uso da UML na indústria.

**P2: A inexistência de um modelo de qualidade para modelagem de software gamificada.** A ampla gama de aplicações acarreta o custo da complexidade de adaptar a gamificação para etapas específicas do processo de software (MARCHEZAN et al., 2024a; JÚNIOR; FARIAS, 2021; JÚNIOR; FARIAS; SILVA, 2021, 2022; SHARMA et al., 2018) mesmo com o fato dela ser considerada uma ferramenta viável em engenharia de software, incentivando os envolvidos a participarem e interajam efetivamente no processo, melhorando assim a interação social, aumento do compartilhamento de conhecimento, desenvolvimento de *soft skills*, melhoria do desempenho da equipe e aumento da qualidade do software (HERNÁNDEZ et al., 2016). Atualmente, não existem critérios sistematizados nem métricas consolidadas que permitam avaliar, de forma objetiva e reproduzível, atributos de qualidade de modelos UML produzidos em atividades de modelagem gamificadas, o que dificulta a comparação de resultados e a análise de seus efeitos.

**P3: Falta de conhecimento empírico sobre o uso de técnicas de gamificação em modelagem de software.** Na maioria dos casos não existe o planejamento e entendimento de quais elementos integrar no uso da gamificação, uma vez que cada um deles surge com objetivos específicos. Apesar desta ferramenta contar com vários elementos, os mais populares apresentados na literatura são *points*, *badges* e *leaderboards*, definidos como a tríade PBL (MORSCHHEUSER et al., 2017). Não existem trabalhos na literatura que (1) visem estabelecer parâmetros para avaliar os modelos UML desenvolvidos a partir de atividades gamificadas; (2) forneçam diretrizes para melhorar a qualidade desses artefatos; (3) analisem quais elementos da gamificação poderiam ser incluídos em cada uma das fases da modelagem usando UML; (4) identifiquem aspectos intrínsecos, como motivação, senso de autonomia e percepção de competência, e extrínsecos, como recompensas, feedback e mecanismos de reconhecimento, que influenciam o desempenho dos desenvolvedores durante as atividades de modelagem; (5) compararem teorias validadas sobre a inclusão da gamificação em modelagem de software com UML e contribuam com a identificação dos objetivos de uso da gamificação nas atividades de modelagem.

## 1.2 Objetivo e Questões de Pesquisa

O principal objetivo desta tese é produzir conhecimento empírico sobre o uso de gamificação em modelagem de software, definir uma abordagem de avaliação dos modelos UML a partir de atividades gamificadas, reunindo assim o conhecimento empírico sobre a melhoria dos artefatos gerados. Com base nesse conhecimento empírico, busca-se gerar *insights* sobre como reduzir o

inconsistência e incompletude dos modelos e consequentemente contribuir com a aprendizagem dos estudantes em modelagem de software com o uso da UML. Este objetivo será alcançado através da compreensão dos efeitos gerados a partir das técnicas de gamificação. Com isso em mente, o objetivo deste estudo é formulado da seguinte forma:

**Objetivo geral:** Produzir conhecimento empírico sobre o uso da gamificação em modelagem de software, por meio da condução de estudos experimentais que permitam analisar seus efeitos sobre a qualidade dos modelos UML e o engajamento dos participantes.

Para explorar múltiplas facetas desse objetivo geral, uma Questão de Pesquisa Geral (QPGERAL) foi formulada, sendo:

- **QPGERAL: Como produzir conhecimento empírico sobre o uso da gamificação em modelagem de software?**

Esta questão de pesquisa geral é subdividida em questões de pesquisa mais específicas, que requerem meios de medição adequados e estudos empíricos sobre a aplicabilidade da gamificação em modelagem de software.

A primeira questão de pesquisa (QP1) explora o primeiro problema (P1) desta tese, onde aborda a necessidade de fornecer uma visão de como está o uso da UML na indústria e os fatores que são fundamentais para entender quais são os principais desafios e limitações da Linguagem de Modelagem Unificada, sendo eles, o que interfere no uso efetivo dela, o que dificulta seu uso, benefícios percebidos ao se utilizar, frequência de uso, principais características da demanda atual de desenvolvimento de software e a visão dos desenvolvedores sobre a modelagem usando UML. Sendo assim, para responder esta questão um *survey* foi desenvolvido, aplicado e detalhado no Capítulo 3. A QP1 foi projetada da seguinte forma:

- **QP1: Como a modelagem UML é adotada na indústria?**

A segunda questão de pesquisa (QP2) consiste na tentativa de identificar como as atividades de modelagem de software gamificadas podem ser avaliadas, visto que alguns trabalhos anteriores (GARACCIONE et al., 2024; BUCCHIARONE et al., 2023a; JURGELAITIS; DRUNGILAS; ČEPONIENĖ, 2018; YOHANNIS, 2016) buscaram entender como gamificação pode ser incorporada no ensino de modelagem de software a partir de alguns elementos (por exemplo, pontos, níveis e emblemas), porém, não existe nenhum modelo na literatura que vise servir como um quadro de referência para avaliação dessas atividades. Deste modo, a QP2 especificada abaixo busca responder o segundo problema (P2). Para isso, o presente trabalho propõem um modelo de qualidade a partir da aplicação de um estudo empírico com 19 participantes para compreender visão deles em relação à gamificação e a aceitação do modelo proposto apresentado no capítulo 4.

- **QP2: Como avaliar a qualidade de modelos UML produzidos em atividades de modelagem gamificadas?**



A terceira questão de pesquisa (QP3) pretende explorar a eficácia dos elementos de gamificação na melhoria dos modelos a partir de seus atributos de qualidade. Alguns estudos (GARACCIONE et al., 2025a; YIGITBAS et al., 2024a; BUCCHIARONE et al., 2024; CAGNAZZO et al., 2023; PORTO et al., 2020; BISTA et al., 2014; DESTEFANIS et al., 2016; BISTA; NEPAL; PARIS, 2012; KLOCK et al., 2014; FRANÇA, 2014) procuraram identificar quais destes elementos podem ser utilizados em modelagem de software e como gerar engajamento dos participantes a partir deles. Todavia são desconhecidos estudos experimentais para analisar os impactos da gamificação de formas significativamente diferentes para medir a qualidade do modelos UML. A QP3 busca responder o terceiro problema (P3) desta tese através da proposta de uma série de estudos empíricos, incluindo experimentos controlados e quase-experimentos, os quais seguirão as boas práticas estabelecidas em (WOHLIN et al., 2012). Os resultados e análises referentes à QP3 são apresentados e discutidos nos Capítulos 5 e 6.

- **QP3: Quais são os efeitos da gamificação em atributos de qualidade dos modelos UML?**

Os estudos visam responder essas questões de pesquisa e são considerados importantes contribuições originais deste trabalho. Nenhum trabalho anterior estudou essas diferentes dimensões da aplicabilidade da gamificação em modelagem UML até agora. A próxima seção discute as contribuições da tese de forma detalhada.

### 1.3 Contribuições da Tese

As seções anteriores definiram o contexto da tese, problema a ser retratado, em seguida, o objetivo e questões de pesquisa. Esta seção descreve as contribuições que a presente tese pretende exibir a partir de uma abordagem na avaliação e na produção de conhecimento empírico sobre a aplicação da gamificação como forma de gerar engajamento dos desenvolvedores e consequentemente qualidade dos modelos UML gerados por eles. Todas as contribuições serão provenientes de uma série de estudos empíricos, incluindo experimentos controlados, entrevistas, estudos de caso e observacionais. Esses estudos qualitativos e quantitativos avaliam os esforços para combinar diversas perspectivas e reunir múltiplas fontes de evidências em cenários realistas e controlados. Mais especificamente, as contribuições desta tese são as seguintes:

1. **Conhecimento empírico sobre a adoção de UML na indústria.** Alguns estudos foram executados para analisar a aplicabilidade da UML na indústria (ROMEO et al., 2025; LI et al., 2025; EGYED, 2010; FARIAS et al., 2015; FERNÁNDEZ-SÁEZ; CHAUDRON; GENERO, 2018; HO-QUANG et al., 2017; OZKAYA; ERATA, 2020; PETRE, 2014; STÖRRLE, 2017), no entanto, a literatura atual carece de estudos que explorem o impacto da questão da cultura organizacional, analisem os fatores que dificultam a utilização da UML e comprovem se esta linguagem é de fato o padrão das linguagens de modelagem.



Estes estudos se concentraram na coleta de feedback dos participantes sobre os diagramas UML mais usados (AKDUR; SAY; DEMIRÖRS, 2021; CHAUDRON; HEIJSTEK; NUGROHO, 2012; FERNÁNDEZ-SÁEZ; CHAUDRON; GENERO, 2018; PETRE, 2014). Isso se torna um problema porque pressupõe que as visões e fragmentos de opiniões dos participantes ao redor do mundo são válidos em nível local e regional. Por exemplo, não exploraram se o contexto do projeto afeta a adoção de modelos UML, nem discutem as percepções dos participantes sobre a utilidade percebida de usar UML. Portanto, a contribuição desta tese se concentra em identificar o estado da prática da indústria, no que diz respeito ao uso de UML em empresas de nível mundial. Especificamente, busca investigar como a UML está sendo utilizada na prática, em relação à relevância de seu uso em projetos de software (QP1).

2. **Modelo de qualidade para apoiar a modelagem de software gamificada.** Estudos como (MARCHEZAN et al., 2023b; AKDUR; SAY; DEMIRÖRS, 2021; JURGELAITIS; DRUNGILAS; ČEPONIENĖ, 2018; YOHANNIS, 2016) buscaram entender como aplicar a gamificação no ensino de modelagem de software a partir de alguns elementos como pontos, emblemas e níveis. No entanto, instrutores e pesquisadores ainda encontram limitações ao aplicar, avaliar e mensurar o uso dessa ferramenta na aprendizagem de alunos de modelagem de software e, conseqüentemente, nos modelos por eles desenvolvidos, uma vez que na literatura atual não existe “guia” que os oriente. Este estudo conjectura que a mecânica de gamificação pode melhorar o engajamento do aluno enquanto aprende modelagem de software, mitigando tais problemas relativos aos modelos UML. A literatura atual carece de estudos que explorem a gamificação e a qualidade do modelo no contexto de modelagem de software (QP2), esta tese propõe uma solução relativa à ausência de um modelo de qualidade que sirva como um guia para avaliar as atividades gamificadas em modelagem de software.
3. **Conhecimento empírico sobre os impactos da gamificação para medir a qualidade dos modelos UML.** Vários esforços de pesquisa têm se concentrado na identificação das fases que compõem o projeto de gamificação (MARAND et al., 2025; GARACCIONE et al., 2025a; YIGITBAS et al., 2024b; MARCHEZAN et al., 2024a; CAGNAZZO et al., 2023; CHEN, 2023; ALMEIDA et al., 2023; MORA et al., 2015; WEBB, 2013). O potencial de gamificação no contexto da modelagem de software pode mitigar alguns dos desafios atuais ligados ao comprometimento, comunicação e a motivação das equipes de desenvolvimento de software, impactando sua produtividade. Este estudo tenta essencialmente avaliar os efeitos de uso de técnicas de gamificação nos modelos a partir de seus atributos de qualidade. Esses efeitos serão investigados a partir de cenários envolvendo atividades gamificadas de modelagem com UML para que resultados empíricos possam ser gerados.

Essas contribuições são apresentadas e discutidas ao longo dos próximos capítulos e refi-

nadas no Capítulo 6. Elas foram relatadas em artigos, onde parte delas já foram publicadas em conferências e se relacionam diretamente com a tese.

## 1.4 Estrutura da Tese

Esta proposta de tese está estruturada em cinco capítulos, além deste. Esta seção descreve como as contribuições são relatadas em cada capítulo e explica como cada capítulo se relaciona com a questão de pesquisa.

**Capítulo 2: Fundamentação Teórica.** São apresentadas as principais teorias que embasam este estudo como Modelagem de software, Linguagem de Modelagem Unificada e Gamificação. Essas definições são essenciais para entender as contribuições e os resultados a serem alcançados.

**Capítulo 3: Um Survey Sobre o Uso de UML Na Indústria (QP1).** Este capítulo apresenta uma revisão da literatura cujos resultados são baseados em um *survey* sobre o uso de UML na indústria. É importante saber como a UML tem sido utilizada por profissionais de TI (Tecnologia de Informação) e o quanto tem sido efetivamente utilizada no ambiente de trabalho possibilitando que as empresas reflitam sobre o assunto e verifiquem se podem usufruir mais dela.

**Capítulo 4: *Modelgame*: Um Modelo de Qualidade Para Avaliar a Aprendizagem Gamificada de Modelagem de Software (QP2).** Este capítulo estabelece o contexto para propor uma estrutura de referência que pode ser parametrizada por instrutores para avaliar modelos UML elaborados por alunos. A hipótese defendida no capítulo é que a qualidade dos modelos UML pode ser melhorada com a aplicação de atividades com o uso de técnicas de gamificação. A qualidade dos modelos UML pode ser melhorada aplicando atividades gamificadas e fornecendo diretrizes cientes dos problemas identificados nestes artefatos.

**Capítulo 5: Estudo Empírico Sobre o Impacto da Gamificação na Modelagem de Software com UML (QP3).** Este capítulo investiga os efeitos de formas significativamente diferentes das mecânicas de gamificação cujo objetivo é apresentar os resultados de uma série de estudos experimentais sobre identificação de fatores que podem melhorar a qualidade dos modelos UML com o uso de elementos gamificados.

**Capítulo 6: Gamificação no Ensino de Engenharia de Software: Um Estudo de Caso Sobre Sua Aplicação no Desenvolvimento Ágil e na Modelagem UML (QP3).** Este capítulo apresenta um estudo de caso que avalia os efeitos da gamificação no engajamento dos alunos, na colaboração entre equipes e na qualidade dos artefatos produzidos em cenários reais de desenvolvimento ágil e modelagem UML.

**Capítulo 7: Conclusões e Trabalhos Futuros.** Este capítulo integra os principais achados dos estudos empíricos realizados, destacando as contribuições teóricas e práticas da tese para o ensino e a prática de modelagem de software em contextos gamificados. Também discute limitações que podem influenciar a generalização dos resultados e aponta direções relevantes para pesquisas futuras, reforçando oportunidades de aprofundamento sobre qualidade de modelos UML e o papel da gamificação nesse processo.

## 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo sistematiza os principais conceitos utilizados para desenvolvimento dos estudos empíricos propostos nesta tese. Inicialmente é apresentado o estado da arte sobre a UML, e posteriormente são apresentados os conceitos referentes a gamificação para contextualização da proposta.

### 2.1 Modelagem de Software

A modelagem de software constitui um processo fundamental na Engenharia de Software, pois permite representar de forma abstrata estruturas, comportamentos e relações essenciais do sistema antes de sua implementação, favorecendo a compreensão, a comunicação entre stakeholders e a redução da complexidade inerente ao desenvolvimento (BOOCH, 2005, 1999; RUMBAUGH et al., 1994). Modelos fornecem uma abstração crítica que possibilita omitir detalhes não essenciais e analisar o sistema de maneira incremental, garantindo que os desenvolvedores verifiquem requisitos, validem decisões de projeto e evoluam as soluções de modo sistemático. Nesse contexto, a Engenharia de Software é compreendida como um procedimento organizado que emprega técnicas predefinidas e convenções notacionais para orientar o desenvolvimento de sistemas, estruturando-se em etapas com métodos e modelos específicos (RUMBAUGH et al., 1994; BOOCH, 1999). Entre essas etapas, destaca-se o desenvolvimento de dados, no qual a adoção da modelagem orientada a objetos traz benefícios significativos, como confiabilidade, interoperabilidade, reusabilidade, adaptabilidade e extensibilidade (BOAVENTURA, 2007), reforçando seu papel estratégico na construção de software de qualidade. Assim, a modelagem de software não apenas estrutura o raciocínio técnico, mas atua como eixo central para garantir coesão conceitual, precisão na especificação e maior controle sobre a evolução do projeto.

Modelo é uma abstração de algo, com a finalidade de entendê-lo antes de construí-lo. Permite omitir detalhes não essenciais, e é mais fácil manipulá-lo do que à entidade original (BLAHA; RUMBAUGH, 2006). Segundo os mesmos autores, a abstração provida pelos modelos permite lidar com complexidade da montagem de sistemas de softwares visto que os desenvolvedores precisam abstrair visões do sistema, montar e verificar se os modelos satisfazem aos requisitos estabelecidos, e acrescentar detalhes gradativamente, para que no fim, os mesmos sejam transformados em implementações.

A montagem de modelos, ou modelagem, permite com que objetivos de grande importância sejam alcançados. Ajuda na visualização do sistema, permite a especificação da estrutura e comportamento do mesmo, direciona a sua construção, documenta as decisões tomadas (BOOCH, 2005), e ainda permite o teste de entidades antes de construí-las, facilita a comunicação com clientes, reduz a complexidade, separando um pequeno número de aspectos importantes a serem tratados de uma só vez (BLAHA; RUMBAUGH, 2006).

(TORCHIANO et al., 2013) apontam que os modelos ajudam a melhorar o design e a documentação. No entanto, eles também descobriram que o uso do modelo está conectado a um esforço extra, especialmente devido à falta de ferramentas de suporte. (FORWARD; BADRED-DIN; LETHBRIDGE, 2010) reforçam que os modelos são usados principalmente para design e documentação, enquanto a geração de código raramente é feita.

(BOOCH, 2005) sugere quatro princípios da modelagem:

- Modelos devem ser bem escolhidos: modelos escolhidos corretamente solucionaram eficientemente os problemas de desenvolvimento, ao passo que os inadequados podem causar confusões e priorizar questões irrelevantes.
- Cada modelo poderá ser expresso em diferentes níveis de precisão: o nível de detalhamento dos modelos pode mudar de acordo com a necessidade dos *stakeholders* ao qual o modelo será exibido.
- Os melhores modelos estão relacionados à realidade: os modelos devem ser feitos de modo que exista clara conexão com a realidade, e, além disso, caso a conexão não seja fiel, deve-se saber exatamente como esses modelos diferem-se do “mundo real”.
- Nenhum modelo único é suficiente: para compreender a estrutura de sistemas de software complexos, é preciso recorrer a várias visões complementares e inter-relacionadas, que poderão conter aspectos estruturais e comportamentais, onde seu conjunto representará a base do projeto.

## 2.2 Unified Modeling Language

Na atualidade em que ocorrem mudanças do paradigma tradicional para o orientado a objetos, a linguagem unificada de modelagem (*UML - Unified Modeling Language*) é o padrão adotado pelo Grupo de Gerenciamento de Objetos (*OMG - Object Management Group*) desde janeiro de 1997, em resposta a sua própria solicitação de propostas para uma linguagem padrão de modelagem (LI et al., 2025; ROMEO et al., 2025; CARVALHO et al., 2003).

A UML (do inglês *Unified Modeling Language*, ou linguagem de modelagem unificada) é uma linguagem de modelagem visual padrão destinada a ser usada tanto para a modelagem de negócios e processos similares como para a análise, projeto e implementação de sistemas baseados em software. É uma linguagem comum para analistas de negócios, arquitetos e desenvolvedores de software com experiência em descrever, especificar, projetar e documentar processos de negócios existentes ou novos e a estrutura e o comportamento de artefatos de sistemas de software (UML-DIAGRAMS, c2020).

Para (LARMAN; UTILIZANDO, 2007), a palavra visual na definição é um ponto-chave. A UML é a notação diagramática padrão, de fato, para desenhar ou apresentar figuras (com

algum texto) relacionadas a software – principalmente software orientado a objetos (OO). Em (FOWLER; SCOTT, 2003), são destacados três modos pelos quais as pessoas aplicam UML:

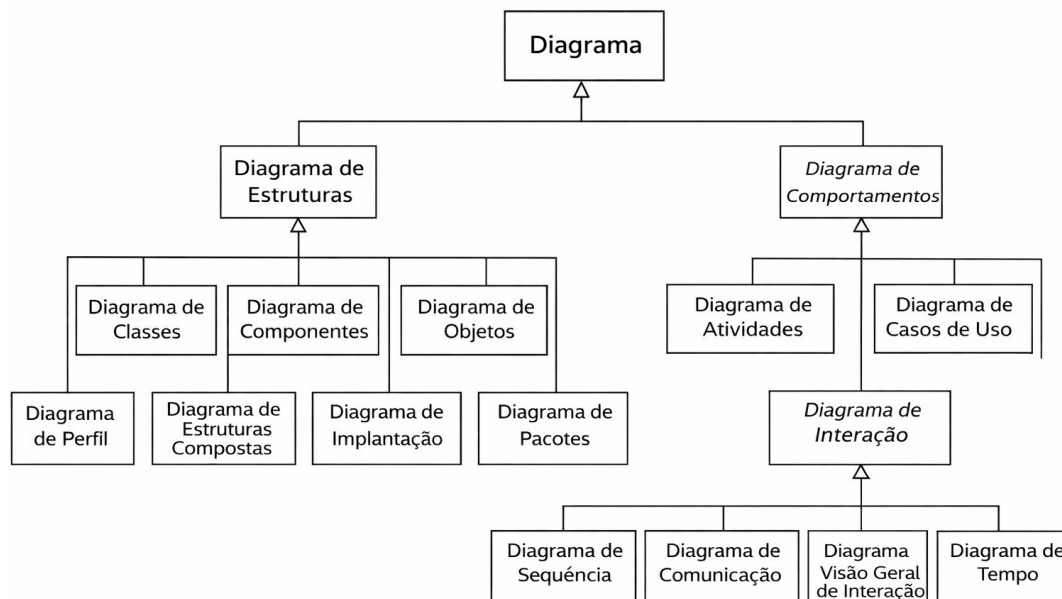
- **UML como rascunho:** diagramas incompletos e informais (frequentemente rascunhados à mão em quadros brancos) criados para explorar partes difíceis do problema ou do espaço de soluções, aproveitando o poder das linguagens visuais.
- **UML como planta de software:** diagramas de projeto relativamente detalhados, utilizados tanto em Engenharia Reversa — para visualizar e compreender melhor o código existente por meio de diagramas UML — quanto em Engenharia Avante, para apoiar a geração de código.
- **Em Engenharia Reversa:** uma ferramenta UML lê o código-fonte ou binário e gera (tipicamente) diagramas UML de pacotes, classes e sequência. Essas “plantas de software” auxiliam na compreensão da estrutura, dos elementos e das colaborações globais no sistema.

Antes da programação, alguns diagramas detalhados podem servir como diretrizes para a geração de código (por exemplo, em Java), seja manualmente ou automaticamente, com o uso de ferramentas. É comum que parte do código seja gerada a partir desses diagramas e outra parte seja completada por um desenvolvedor, possivelmente combinando também rascunhos UML.

- **UML como linguagem de programação:** especificação executável completa de um sistema de software em UML. O código executável é gerado automaticamente e geralmente não é visualizado ou modificado pelos desenvolvedores, que trabalham diretamente na “linguagem de programação” UML. Esse uso pressupõe uma forma prática de diagramar todo o comportamento e lógica do sistema (provavelmente com diagramas de interação ou de estado) e ainda está em evolução quanto à teoria, ferramentas robustas e usabilidade.

Ainda sob o ponto de vista do autor, “a UML descreve tipos de esboço de diagramas, tais como diagramas de classe e diagramas de sequência. Ela não superpõe a eles uma perspectiva de modelagem. Por exemplo, a mesma notação UML de diagrama de classes pode ser usada para desenhar imagens de conceitos do mundo real ou de classes de software em Java” (LARMAN; UTILIZANDO, 2007).

A UML é resultante da unificação de notação empregada em três metodologias bastante utilizadas na década de 90: Booch (de Grady Booch da Ration Software Corporation), OOSE - Object-Oriented Software Engineering (de Jacobson da Objectory) e OMT - Object Modeling Technique (de James Rumbaugh da General Eletrics) (JACOBSON; BOOCH; RUMBAUGH, 1999). Posteriormente, foi desenvolvida uma metodologia orientada a objetos conhecida como *Rational Unified Process* (RUP) que abrange todo o processo de desenvolvimento de um software.

**Figura 1: Diagramas UML**

Fonte: Adaptado de (OMG, 2017)

Contudo, apesar da UML ter sido projetada para o desenvolvimento de sistema OO, independentemente da linguagem de programação a ser utilizada, ela permite especificar, visualizar e documentar artefatos e assim construir uma estrutura consistente de um sistema de software, além de possibilitar modelar elementos, relacionamentos e diagramas.

Os elementos especificados com a UML podem ser estruturais, como classes, interfaces comportamentais, grupos de elementos dentre outros. Os relacionamentos que são a relação entre os blocos de elementos podem ser subdivididos em: dependência, associação, generalização e realização. Já os diagramas que são as representações gráficas dos elementos e sua relação são divididos em nove diagramas: de classes, objetos, caso de uso, sequência, colaboração, gráficos de estados, atividades, componentes, implantação (BOOCH, 1999).

Os diagramas da UML se dividem em dois grandes grupos: diagramas estruturais e diagramas comportamentais (Figura 1). Os estruturais devem ser utilizados para especificar detalhes da estrutura do sistema (parte estática), por exemplo: classes, métodos, interfaces, *namespaces*, serviços, como componentes devem ser instalados, como deve ser a arquitetura do sistema etc. Os comportamentais devem ser utilizados para especificar detalhes do comportamento do sistema (parte dinâmica), por exemplo: como as funcionalidades devem funcionar, como um processo de negócio deve ser tratado pelo sistema, como componentes estruturais trocam mensagens e como respondem às chamadas etc.

### 2.2.1 Diagramas de Estrutura

A UML define sete tipos de diagramas de estrutura (OMG, 2017).

- Diagrama de classe: os diagramas de classes capturam a estrutura lógica do sistema, as classes e os objetos que compõem o modelo, descrevendo o que existe e quais atributos e comportamento ele possui.
- Diagrama de estrutura: os diagramas de estrutura refletem a colaboração interna de classes, interfaces e componentes (e suas propriedades) para descrever uma funcionalidade
- Diagrama de componentes: os diagramas de componentes ilustram as partes do software, controles internos e outros elementos que compõem um sistema, sua organização e dependências.
- Diagrama de implantação: os diagramas de implantação mostram como e onde o sistema deve ser implantado; isto é, sua arquitetura de execução.
- Diagrama de objeto: os diagramas de objetos descrevem instâncias de classes ou objetos e seus relacionamentos em um determinado momento.
- Diagrama de pacotes: os diagramas de pacotes descrevem a organização dos elementos de um modelo em pacotes e suas interdependências.
- Diagrama de pacotes: os diagramas de pacotes descrevem a organização dos elementos de um modelo em pacotes e suas interdependências.
- Diagrama de perfil: os diagramas de perfil são aqueles criados para estender elementos, conectores e componentes UML.

### 2.2.2 Diagramas de Comportamento

Sete diagramas de comportamento também são definidos na UML (OMG, 2017).

- Diagramas de atividades: os diagramas de atividades modelam os comportamentos de um sistema e a maneira como esses comportamentos estão relacionados em um fluxo geral de funcionamento.
- Diagramas de casos de uso: os diagramas de casos de uso capturam interações e relacionamentos entre os atores e o sistema; eles descrevem os requisitos funcionais do sistema, a maneira pela qual operadores externos interagem nos limites do sistema e a resposta deste.



- Diagramas de máquina de estado: ilustram como um elemento pode se mover entre estados, classificando seu comportamento de acordo com gatilhos de transição e restrições de proteção.
- Diagramas de tempo: os diagramas de tempo definem o comportamento de diferentes objetos em uma escala de tempo, fornecendo uma representação visual dos objetos que mudam de estado e interagem ao longo do tempo.
- Diagramas de sequência: os diagramas de sequência são representações estruturadas do comportamento como uma série de etapas sequenciais ao longo do tempo. Eles são usados para descrever o fluxo de trabalho, a transmissão de mensagens e como os elementos em geral cooperam ao longo do tempo para alcançar um resultado.
- Diagramas de comunicação: os diagramas de comunicação mostram as interações entre os elementos no tempo de execução, visualizando os relacionamentos entre objetos
- Diagramas de visão geral da interação: possibilitam a visualização da cooperação entre os diagramas de interação (tempo, sequência, comunicação e outros diagramas de visão geral da interação) para ilustrar um fluxo de controle que serve a um objetivo abrangente.

## 2.3 Gamificação

A gamificação pode ser compreendida, em sua essência, como “o uso de elementos de design de jogos em contextos não jogáveis” (DETERDING et al., 2011a; WERBACH; HUNTER, 2012, 2015). Essa perspectiva enfatiza não apenas a inserção de componentes típicos de jogos, mas sobretudo o uso estratégico de dinâmicas e mecanismos capazes de promover motivação, engajamento e comportamentos desejados em diferentes domínios. Embora o termo tenha ganhado notoriedade na última década, sua origem conceitual remonta ao final dos anos 1970, quando práticas de ludificação começaram a surgir em contextos de programação e desenvolvimento de software (ALVES, 2015). O uso contemporâneo do termo, entretanto, consolidou-se por volta de 2008 no setor de mídia digital e, desde então, a gamificação evoluiu como um campo de investigação robusto, expandindo sua aplicação para áreas como educação, negócios, saúde e Engenharia de Software, sem apresentar sinais de desaceleração em sua adoção ou relevância científica. Nesse cenário, compreender o conceito de gamificação e suas bases teóricas torna-se fundamental para analisar criticamente como seus elementos podem apoiar processos complexos, como a aprendizagem de modelagem de software e a melhoria da qualidade dos artefatos gerados (GARACCIONE; COPPOLA; ARDITO, 2024; GARACCIONE et al., 2025b)

Para compreender se de fato a gamificação funciona foram realizadas pesquisas empíricas testando uma ampla diversidade de sistemas gamificados com uma gama igualmente ampla de medidas de efeitos (HAMARI; KOIVISTO; SARSA, 2014; SEABORN; FELS, 2015). Enquanto os estudos neste modo ajudaram a estabelecer a validade de face da gamificação, seus

retornos de conhecimento diminuíram rapidamente. Para a pesquisa somar uma estrutura de conhecimento conjunta, ela precisa fluir de e para teorias. Essas teorias integram e explicam descobertas empíricas divergentes, identificam hipóteses relevantes para testar a seguir e fornecem aos profissionais uma forma de conhecimento que ajuda a compreender e prever quando e como qual design específico será eficaz ou não (WHITLEY JR; KITE, 2012; DETERDING, 2014). E precisa avançar a partir do teste de sistemas gamificados que combinam e, assim, combinam os efeitos de) vários elementos de design de jogo para estudar paradigmas que revelam os efeitos, moderadores e mediadores de elementos individuais (HAMARI; KOIVISTO; SARSA, 2014; DETERDING, 2014).

(HERNÁNDEZ et al., 2016) consideram a gamificação uma ferramenta viável para o ensino de engenharia de software, pois incentiva os alunos a participarem e interagirem efetivamente no processo de aprendizagem, melhorando assim a interação social, aumento do compartilhamento de conhecimento, desenvolvimento de *soft skills*, melhoria o desempenho da equipe e aumento da qualidade do software. No entanto, essa ampla gama de aplicações acarreta o custo da complexidade de adaptar a gamificação para etapas específicas do processo de software (SHARMA et al., 2018).

Cada vez mais, a gamificação tem se tornado parte das fases de desenvolvimento das aplicações, na busca por melhorias da qualidade dos modelos de design, através da motivação e maior engajamento entre usuário e desenvolvedores, todavia, ainda existem poucos estudos que discutam a integração dos elementos desta ferramenta na modelagem de software (PEDREIRA et al., 2015). A gamificação oferece vários mecanismos (ZICHERMAN; CUNNINGHAM, 2011; HUGOS, 2012) que podem ser utilizados em cada fase do contexto da engenharia de software, baseando na hipótese de que podem ajudar a enfrentar os desafios relacionados ao tratamento de fatores sociais e humanos (MACHUCA-VILLEGAS et al., 2020). Existe uma demanda significativa por estudos empíricos para a compreensão dos impactos e eficácia da gamificação. No entanto, a falta de procedimentos e modelos padronizados para a avaliação desta ferramenta é um desafio para o desenho, comparação e relato de resultados relacionados à avaliação das abordagens de gamificação e seus efeitos (MONTEIRO et al., 2021). Buscar integrar a gamificação como um ativo-chave na modelagem de software é um dos grandes desafios atuais, inclusive no campo educacional.

Apesar desta ferramenta ser um tema de interesse recente, alguns estudos apontam dificuldades e desafios para sua implementação (PAULA PORTO et al., 2021; MARÍN, 2021; RODRIGUES; SOUZA; FIGUEIREDO, 2018; PEDREIRA et al., 2015). Na área de engenharia de software, algumas tarefas consistem na aplicação de alguns processos que podem ser considerados exaustivos e demorados. Nesse contexto, a gamificação pode ser vista como um recurso com grande potencial que pode melhorar o interesse e a motivação dos desenvolvedores na realização de suas práticas.

### 2.3.1 Elementos da Gamificação

Primeiramente, é importante entender o conceito de uso dos termos "mecânica do jogo" (SICART, 2008; MARCZEWSKI, 2013) e "dinâmica do jogo" (FORRESTER, 1971; WERBACH; HUNTER, 2015). Mecânica do jogo equivale às regras (ou diretrizes) que garantem que a atividade passível à gamificação seja adquirida em um jogo. Dinâmica de jogo são as características e valores que afetam a percepção do jogador. Mecânica busca as regras do jogo para identificar e estabelecer a configuração dos elementos a serem transferidos para a atividade gamificada; portanto, deixam pronta a função que deve cumprir ou a mesma: a dinâmica que envolve.

Os elementos do jogo são ferramentas por meio das quais um jogo é construído. Eles constituem as peças ou partes que o compõem. A partir do contexto da gamificação, esses elementos estão embutidos nas atividades a serem gamificadas, uma vez que a ideia não é criar um jogo completo, mas utilizar alguns elementos para serem aplicados no mundo real. Dessa forma, é fornecida mais flexibilidade do que um jogo (WERBACH; HUNTER, 2012).

Elementos da gamificação é o meio que é usado para chamar a atenção, entreter e manter o usuário em determinada tarefa. Elementos esses que precisam ser estudados à risca, para não simplesmente usar qualquer um ou até todos, até porque cada um tem sua finalidade.

Os elementos mais populares são points, badges e leaderboards, definidos como a tríade PBL (MORSCHHEUSER et al., 2017). Porém, é válido destacar a importância de planejar e entender quais elementos integrar no processo da gamificação, uma vez que cada um deles surge com objetivos específicos. A gamificação é apontada como uma forma de realizar um estudo detalhado em relação aos jogos (WERBACH; HUNTER, 2012), aprendendo com eles, para entender o que neles engaja seus usuários e motiva ações na resolução de problemas.

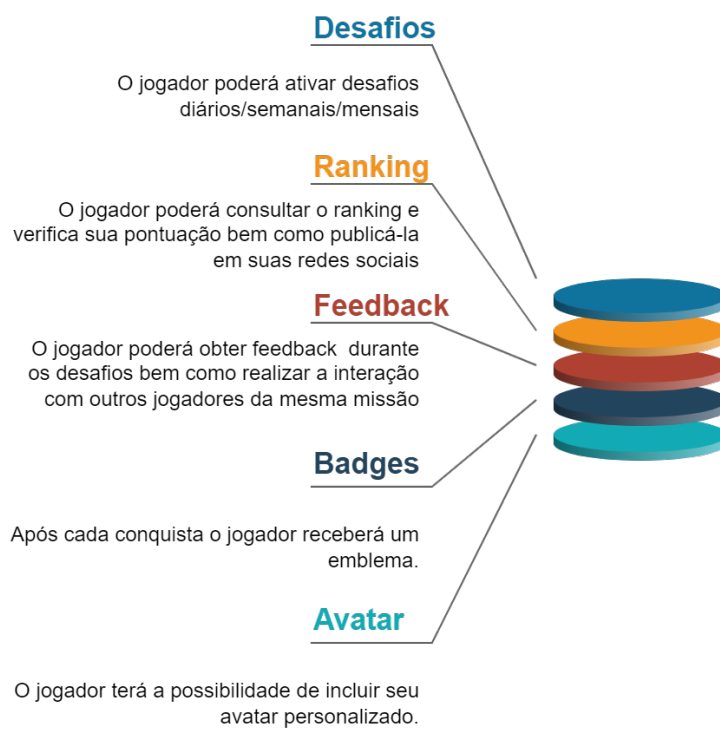
Podemos ver essa relação com algumas das principais técnicas apresentadas abaixo segundo (WERBACH; HUNTER, 2012; DETERDING et al., 2011a; PAHARIA, 2013):

- **Prêmios:** Os jogadores recebem recompensas específicas por completar certas ações.
- **Missões:** As missões que o jogador precisa completar são apresentadas na forma de missões com elementos de jogabilidade adicionais (como uma história) para torná-la mais envolvente.
- **Sistema de recompensa baseado em pontos:** Os jogadores recebem recompensas na forma de pontos por realizar uma determinada ação.
- **Distintivos:** Representam certas conquistas do usuário.
- **Níveis:** Relacionado à recompensas baseadas em pontos; os usuários têm um nível elevado à medida que atingem um certo número de pontos.
- **Votação:** Os jogadores podem votar no comportamento de outro jogador. Os próprios votos representam as recompensas obtidas por cada jogador.

- **Classificação:** Apresenta uma relação com os melhores jogadores para aumentar a competitividade. A posição no ranking pode ser definida pontos, níveis ou número de votos, por exemplo.
- **Apostas:** Os usuários apostam em um determinado evento, como uma estimativa, por exemplo. O vencedor da aposta recebe alguma recompensa em troca.

A Figura 2 sintetiza um conjunto de elementos de gamificação amplamente reconhecidos na literatura — desafios, ranking, feedback, badges e avatares — os quais constituem componentes fundamentais para promover engajamento, motivação e autonomia em ambientes de aprendizagem e prática profissional (FEICHAS; SEABRA, 2023; DETERDING et al., 2011b). Esses elementos, quando integrados a processos educacionais ou de desenvolvimento de software, atuam sobre diferentes dimensões motivacionais: os desafios oferecem estrutura para a progressão orientada a metas, estimulando a superação de etapas cognitivas progressivamente mais complexas; rankings introduzem comparação social e senso de desempenho relativo, reforçando a visibilidade do esforço individual e coletivo; mecanismos de feedback fornecem informação imediata sobre a qualidade da ação realizada, promovendo autorregulação e aprendizagem contínua; badges funcionam como marcadores simbólicos de conquistas, valorizando trajetórias de competência; e avatares contribuem para a personalização e construção de identidade no contexto da atividade, fortalecendo o sentimento de presença e pertencimento. Em conjunto, esses componentes formam uma arquitetura gamificada capaz de estimular comportamentos esperados, intensificar o envolvimento do participante e sustentar ciclos de interação mais longos e significativos (ALVES, 2015). No contexto da modelagem de software com UML, tais elementos podem potencializar a qualidade dos modelos ao influenciar fatores como persistência, atenção ao detalhe, colaboração e profundidade analítica, estabelecendo uma ponte entre mecanismos motivacionais e atributos formais de qualidade de artefatos.

**Figura 2:** Exemplos de elementos gamificados



Fonte: Próprio autor

### 3 UM SURVEY SOBRE O USO DE UML NA INDÚSTRIA

Os modelos UML desempenham um papel crucial nas tarefas de desenvolvimento de software, por exemplo, documentar decisões do projeto e promover uma melhor comunicação entre as equipes (OMG, 2017). No contexto do processo de desenvolvimento, a UML defende que sua utilização implica em inúmeros benefícios (BUCCHIARONE et al., 2021; CHAUDRON; HEIJSTEK; NUGROHO, 2012; FERNÁNDEZ-SÁEZ; CHAUDRON; GENERO, 2018), como fornecer um entendimento comum entre os membros da equipe, entendendo os detalhes do desenvolvimento, e maior eficiência no desenvolvimento de software. Infelizmente, na prática, tais benefícios são muitas vezes negligenciados. Alguns estudos (CHAUDRON; HEIJSTEK; NUGROHO, 2012; FERNÁNDEZ-SÁEZ; CHAUDRON; GENERO, 2018; STÖRRLE, 2017) argumentam que tais benefícios são consequência de uma completa e informal aplicação de modelagem, onde a UML deve ser aplicada em todo o projeto de software, e os praticantes têm um controle claro sobre o uso da mesma. Como essa realidade não existe comumente, várias pesquisas (FERNÁNDEZ-SÁEZ; CHAUDRON; GENERO, 2018; PETRE, 2014) sobre o uso de UML foram propostas nos últimos anos para investigar se ela é de fato, utilizada no contexto de real projetos da indústria.

No entanto, conforme apontado no primeiro problema (P1) identificado neste estudo apresentado no Capítulo 1, a literatura atual carece de uma ampla e compreensão exploratória das percepções dos praticantes sobre fatores que afetam ou mesmo comprometem a adoção da modelagem UML em projetos do mundo real (AKDUR; SAY; DEMIRÖRS, 2021; CHAUDRON; HEIJSTEK; NUGROHO, 2012; FERNÁNDEZ-SÁEZ; CHAUDRON; GENERO, 2018; PETRE, 2014). Pouco se sabe sobre como os praticantes lidam com a modelagem de software no contexto dos projetos das empresas, destacando a heterogeneidade delas, o nível da educação dos participantes, e da frequência de uso. Estudos anteriores têm focado em coletar opiniões dos participantes para entender quais diagramas UML são mais utilizados (PETRE, 2014). Isso se torna um problema, porque ele assume que as percepções e fragmentos de as opiniões dos participantes em todo o mundo são válidas no local e nível regional. Esses estudos não exploram, por exemplo, se o contexto do projeto pode influenciar a adoção de modelos UML, nem discutem as opiniões dos participantes sobre a utilidade percebida ao utilizar a UML.

Alguns trabalhos foram executados para analisar a aplicabilidade da UML na indústria (CICCOZZI; MALAVOLTA; SELIC, 2019; FERNÁNDEZ-SÁEZ et al., 2015; FERNÁNDEZ-SÁEZ; CHAUDRON; GENERO, 2018; HO-QUANG et al., 2017; OZKAYA; ERATA, 2020; PETRE, 2014; STÖRRLE, 2017), no entanto, a literatura atual carece de estudos que: (1) explorem o impacto da questão da cultura organizacional, (2) analisem os fatores que dificultam a utilização da UML e (3) comprovem se esta linguagem é de fato o padrão das linguagens de modelagem, já que as pesquisas existentes têm se concentrado na coleta de opiniões de profissionais em nível mundial, o que pressupõe que tais percepções apesar de fragmentadas podem não ser válidas no âmbito local e regional.

Dessa forma, este capítulo se concentra em identificar o estado da prática da indústria no que diz respeito ao uso de UML em empresas de nível mundial respondendo a QP1 apresentada no Capítulo 1 (Introdução) desta tese. Especificamente, busca-se investigar como a UML está sendo utilizada na prática, em relação à relevância de seu uso em projetos de software. Para o alcance dos objetivos propostos, este estudo realizou uma pesquisa com 376 profissionais de 210 empresas brasileiras de Tecnologia da Informação, na qual estes participaram respondendo a um questionário online, via Google Forms, sobre suas experiências de uso de UML, as dificuldades em adotar esta linguagem, os fatores que afetam o uso eficaz dela, frequência de uso e benefícios trazidos por ela. Os participantes foram selecionados obedecendo dois critérios: 1) nível de conhecimento teórico e experiência prática relacionada à modelagem de software e; 2) programação em projetos convencionais.

Como resultados, observou-se o reforço de algumas evidências já encontradas na literatura do estado da arte sobre a UML na prática, especificamente no que diz respeito ao uso da mesma, que acaba não utilizando todo o seu potencial em projetos de software. No geral, grande parte dos participantes deste estudo conhece a UML, mas não utiliza em seus projetos. Esses resultados podem contribuir para apoiar a tomada de decisão em organizações de software quanto ao uso de práticas de modelagem UML, especialmente no que se refere à compreensão de seus possíveis impactos sobre a qualidade dos artefatos produzidos. Além disso, eles fornecem indícios que podem subsidiar estratégias de incentivo à adoção da modelagem UML em determinados contextos e etapas do ciclo de desenvolvimento, sem pressupor efeitos diretos ou universais sobre custos, produtividade ou motivação dos desenvolvedores. Isso facilitaria e geraria economia, por exemplo, em tarefas de manutenções futuras que consomem a maior parte dos recursos de desenvolvimento de software, conforme discutido em (PRESSMAN, 2005) e (GHAI, 2003): “A manutenção normalmente consome 40 a 80% dos custos de software, portanto, é provavelmente a fase mais importante do ciclo de vida do software”. Melhorar o software antigo é, portanto, um grande ideal. Além do mais identificar padrões de modelagem de indivíduos e/ou projetos antes de uma avaliação organizacional das práticas de modelagem de software pode ser útil para apontar as ameaças potenciais para a institucionalização, como a diversidade de técnicas utilizadas (AKDUR; SAY; DEMIRÖRS, 2021).

Este capítulo está estruturado em sete seções. A seção 3 contextualiza o tema e apresenta o cenário atual. Por sua vez, a seção 3.1 define os trabalhos relacionados. A seção 3.2 detalha a metodologia adotada. A seção 3.3 realiza uma análise do perfil dos participantes e comenta os resultados para cada questão de pesquisa. A seção 3.4 exibe uma discussão adicional. A seção 3.5 apresenta as ameaças à validade e por fim, a última seção as considerações finais.

### 3.1 Trabalhos Relacionados

Na última década, a literatura, na área de modelagem de software, tem investido um grande esforço para entender como a UML tem sido adotada na indústria. Neste sentido, alguns tra-



balhos investigaram vários aspectos sobre como os diagramas comportamentais e estruturais da UML foram utilizados na prática. A Subseção 3.1.1 apresenta uma análise de tais trabalhos publicados na literatura atual. A Subseção 3.1.2 compara o trabalho proposto com a literatura atual, bem como introduz oportunidades de pesquisa.

### 3.1.1 Análise dos Trabalhos Relacionados

(PETRE, 2014). Este trabalho realizou um estudo empírico envolvendo uma série de entrevistas conduzidas ao longo de 2 anos com mais de 50 desenvolvedores de software. Os participantes estavam principalmente na América do Norte e na Europa, mas alguns eram do Brasil, Índia e Japão sendo que muitos trabalhavam em mais de um país. Petre observou que a UML não é adotada de forma universal no contexto da Engenharia de Software, sendo utilizada de maneira consistente apenas em determinados cenários e domínios específicos, de acordo com as necessidades e características dos projetos. O estudo reporta uma heterogeneidade em relação ao uso da UML, a qual tende a ser adotada por pessoas assumindo diferentes papéis para solucionar diferentes problemas ao longo do processo de desenvolvimento. Petre reportou também que a forma de adoção dos diagramas.

(OZKAYA; ERATA, 2020). Este artigo buscou compreender como profissionais usam UML para modelar arquiteturas de software, a partir de diferentes pontos de vista: funcional, simultaneidade, desenvolvimento, operacional, implantação e informação. Cada ponto de vista foi considerado em termos de um conjunto de modelos de software que podem ser criados. Para esse fim, contou com a participação de 109 profissionais de 34 países, que representam os diferentes perfis, cargos, tipos de projetos de software envolvidos e anos de experiência. Os resultados da pesquisa apontam que a maioria dos profissionais (88%) usa UML na modelagem de seus sistemas de software de diferentes pontos de vista de arquitetura. Os principais pontos de vista são os de informação e funcional.

(FERNÁNDEZ-SÁEZ et al., 2015). Os resultados desta pesquisa mostram como as empresas podem tornar os sistemas que estão sendo mantidos mais fáceis de entender, utilizando diagramas da UML. O estudo aponta que o uso de diagramas da UML facilitaria a execução de tarefas de manutenção de software, tornando as manutenções mais assertivas. Para isso, as equipes de desenvolvimento precisariam enfrentar o desafio de atualizar os diagramas da UML, à medida que o código da aplicação fosse alterado. Além disso, o estudo investiga questões importantes sobre como os diagramas da UML estão sendo usados em projetos de manutenção na indústria de software, verifica quais diagramas UML são mais eficazes para manutenção de software, aponta quais são os benefícios percebidos do uso de diagramas da UML na prática, e contextualiza o tipo de empresas que usam a documentação UML na manutenção de software.

(FARIAS et al., 2018). Os resultados desta pesquisa mostram como as empresas podem tornar os sistemas que estão sendo mantidos mais fáceis de entender, utilizando diagramas da UML. O estudo aponta que o uso de diagramas UML pode facilitar a execução de tarefas de



manutenção de software, contribuindo para uma melhor compreensão do sistema e para a realização de intervenções mais corretas e alinhadas à estrutura e ao comportamento do software. Para isso, as equipes de desenvolvimento precisariam enfrentar o desafio de atualizar os diagramas da UML, à medida que o código da aplicação fosse alterado. Além disso, o estudo investiga questões importantes sobre como os diagramas da UML estão sendo usados em projetos de manutenção na indústria de software, verifica quais diagramas UML são mais eficazes para manutenção de software, aponta quais são os benefícios percebidos do uso de diagramas da UML na prática, e contextualiza o tipo de empresas que usam a documentação UML na manutenção de software.

(CICCOZZI; MALAVOLTA; SELIC, 2019). Este trabalho realizou uma revisão sistemática na qual foi selecionado 63 estudos de pesquisa e 19 ferramentas entre mais de 5400 entradas, aplicando um processo sistemático de busca e seleção. O objetivo era identificar, classificar e avaliar as soluções existentes para a execução de modelos UML. Os principais resultados deste estudo foram: (i) há um interesse científico crescente na execução de modelos UML; (ii) a depuração em nível de modelo é suportada em muito poucos casos; (iv) apenas algumas pesquisas fornecem evidências de uso industrial, com avaliações empíricas muito limitadas; e (v) a limitação mais comum trata da cobertura da linguagem UML.

(STÖRRLE, 2017). Este artigo conduziu uma pesquisa on-line entre profissionais da indústria que se baseava em descobrir se eles usam os diagramas da UML e para quais propósitos. O survey aplicado a 82 profissionais de TI investigou (i) se as modelagens feitas em UML e BPMN (Business Process Modeling Notation), são realmente utilizados no mercado, (ii) para quais propósitos estas linguagens de modelagem são utilizadas, (iii) quais as formas distintas de se utilizar estes modelos existentes no mercado e (iv) qual a frequência com que são utilizadas estas linguagens de modelagem. Foi constatado que a UML é utilizada na maioria das situações e com maior frequência. Ainda reforçou que linguagens de modelagem são usadas principalmente para o desenvolvimento de sistemas, sendo utilizadas com frequência também na modelagem de processos.

(FERNÁNDEZ-SÁEZ; CHAUDRON; GENERO, 2018). Os resultados deste trabalho mostraram que a UML é utilizada principalmente para se ter uma visão geral do sistema. Além disso, a maioria dos entrevistados considera que o aprendizado da UML pode ser feito em um treinamento pontual, que deve ser contínuo. Ainda consideraram que a utilização da UML é importante. Contudo, ao longo das entrevistas, observou-se que o uso da UML pode levar as equipes a despendar esforço excessivo em detalhes de baixo impacto para os objetivos do projeto, como aspectos estéticos dos diagramas. Esse comportamento evidencia uma das limitações frequentemente associadas à UML, relacionada à sua riqueza de elementos e ao nível de detalhamento oferecido, que, em determinados contextos, pode dificultar a produtividade durante as atividades de modelagem.

(HO-QUANG et al., 2017). Fizeram uma pesquisa em larga escala com 485 respostas de contribuidores de 458 projetos Open Source diferentes. direcionada aos projetos que usam

UML, com foco em como ela é usada e impacta as atividades de desenvolvimento. As contribuições desta pesquisa são: (i) a identificação de um grande conjunto de projetos OSS que usam UML e (ii) percepções de uma pesquisa em grande escala de desenvolvedores OSS que usam UML. Entre outros insights, foi descoberto que a UML é usada para coordenar o desenvolvimento. Além disso, o uso dela parece ajudar novos contribuidores a começar, embora não pareça atrair novos contribuidores.

### 3.1.2 Análise Comparativa e Oportunidades

Foram definidos cinco Critérios de Comparação (CC) para auxiliar no processo de identificação das similaridades e diferenças entre o trabalho proposto e os artigos selecionados. Esta comparação é crucial para tornar o processo de identificação de oportunidades de pesquisa utilizando critérios objetivos, ao invés de subjetivos. Os critérios são descritos a seguir:

- Contexto (CC01): Estudos realizados com profissionais da indústria brasileira.
- Perfil do participante (CC02): Estudos que coletaram dados dos participantes para triagem e caracterização de perfil.
- Região geográfica específica (CC03): Trabalhos que exploram o uso da UML em escopo regional específico.
- Aplicabilidade da UML (CC04): Estudos que avaliaram quais fatores impedem a adoção da UML na indústria.
- Entrevistas com participantes (CC05): Estudos que triangularam dados quantitativos com qualitativos.

Trabalho Relacionado	Critério de Comparação				
	CC1	CC2	CC3	CC4	CC5
Trabalho Proposto	●	●	●	●	●
(PETRE, 2014)	◐	○	○	●	●
(OZKAYA; ERATA, 2020)	◐	●	○	○	○
(FERNÁNDEZ-SAEZ et al., 2015)	○	●	○	○	◐
(FARIAS et al., 2018)	●	○	●	◐	○
(CICCOZZI; MALAVOLTA; SELIC, 2019)	○	○	○	●	○
(STORRLE, 2017)	○	●	○	○	○
(FERNÁNDEZ-SÁEZ; CHAUDRON; GENERO, 2018)	○	●	○	○	◐
(HO-QUANG et al., 2017)	○	●	○	○	○

● Atende Completamente   ◐ Atende Parcialmente   ○ Não atende

**Tabela 1:** Análise comparativa dos trabalhos relacionados selecionados

A Tabela 1 apresenta a comparação dos trabalhos selecionados, contrapondo os mesmos com este trabalho. Algumas lacunas e oportunidades de pesquisas são observadas:

(1) apenas o trabalho proposto foi o único a atender completamente todos os critérios de comparação;

(2) a maioria deles não tinha como alvo a prática UML especificamente em projetos industriais de código fechado ou estavam restritos a participantes de uma única empresa.

(3) nenhum estudo explorou detalhadamente os fatores que afetam o uso eficaz da UML e as principais dificuldades enfrentadas pelos profissionais da indústria de uma região específica. Portanto, a seguinte oportunidade de pesquisa foi identificada: um estudo experimental que contemple os cinco critérios de comparação estabelecidos. Esta oportunidade é explorada nas próximas seções.

## 3.2 Metodologia

Esta seção apresenta a metodologia seguida para a execução do survey. Esta metodologia é baseada em protocolos de estudos experimentais previamente validados e publicados, tais como (KITCHENHAM; PFLEEGER, 2008) e (WOHLIN et al., 2012). Estes estudos foram escolhidos, pois eles exploraram um desenho experimental semelhante ao estudo aqui reportado. Para isso, a Subseção 3.2.1 introduz o objetivo e as questões de pesquisas investigadas. A Seção 3.2.2 descreve o processo experimental adotado. A Seção 3.2.3 descreve o questionário formulado para o estudo.

### 3.2.1 Objetivo e Questões de Pesquisa

Os objetivos (O) deste estudo são duplo: (O1) compreender a difusão e relevância do uso de UML nas empresas brasileiras; e (O2) analisar o nível de compreensão dos desenvolvedores acerca dos benefícios da UML em projetos do mundo real. Para analisar diferentes aspectos destes objetivos, seis Questões de Pesquisa (QPs) foram formuladas:

### 3.2.2 Processo Experimental

O processo experimental (Figura 4) é formado por um conjunto de atividades agrupadas em três fases discutidas da seguinte forma:

**Fase 1. Seleção dos participantes.** Os participantes foram selecionados com base nos seguintes critérios: nível de conhecimento teórico, experiência prática relacionada à modelagem de software e programação em projetos de desenvolvimento de software reais. Usando tais critérios, buscou-se selecionar participantes com formação acadêmica e com experiência prática na indústria em projetos reais. Este conjunto finito de todos os possíveis participantes representa a população-alvo (KITCHENHAM; PFLEEGER, 2008). Mais especificamente, a população-alvo consiste de profissionais brasileiros - incluindo desenvolvedores, analistas, arquitetos de software e gerentes de projeto - com formação acadêmica em universidades brasileiras. Essa

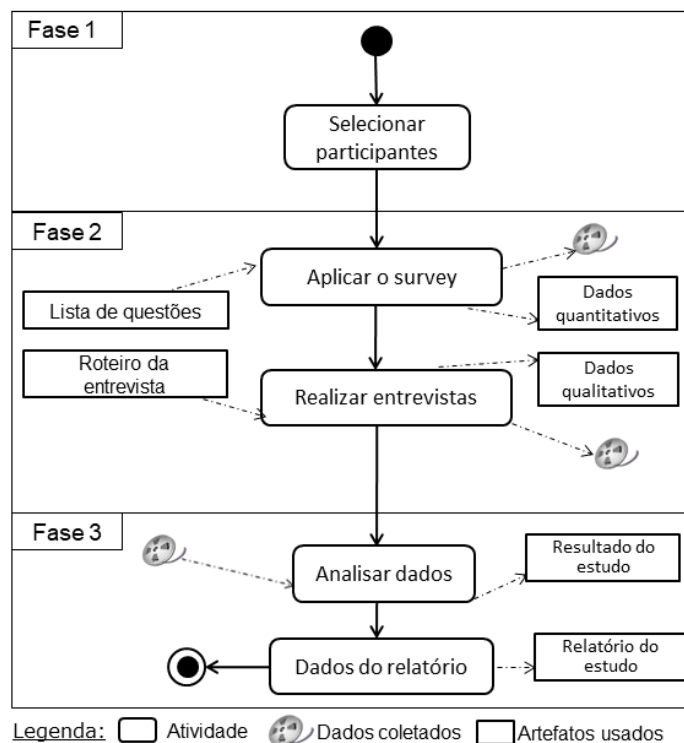
**Tabela 2:** Questões de Pesquisa investigadas neste capítulo

Questões de Pesquisa	Motivação	Variável
<b>QP1.1:</b> Quais fatores influenciam o uso efetivo da UML?	Revelar os fatores influenciadores em um uso mais amplo de modelos UML na prática.	Fatores que influenciam o uso
<b>QP1.2:</b> O que torna a modelagem da UML uma prática desafiadora?	Entender os desafios enfrentados pelos participantes nas práticas que dificultam a adoção da modelagem UML.	Dificultam a adoção
<b>QP1.3:</b> Quais benefícios são percebidos ao usar UML?	Revelar os benefícios mais frequentes ao usar modelagem UML.	Benefícios percebidos
<b>QP1.4:</b> Com que frequência a UML é utilizada?	Entender com que frequência a modelagem UML é usada.	Frequência de uso
<b>QP1.5:</b> Como o contexto de projetos de software em empresas limita o uso de UML?	Identificar fatores de contexto que limitam o uso da modelagem UML.	Contexto do projeto
<b>QP1.6:</b> Como os praticantes veem a modelagem da UML?	Revelar a visão dos praticantes considerando a adoção da modelagem UML.	Visão do participante

população representa aquelas pessoas que estão em posição de responder às questões formuladas e à quem os resultados da pesquisa se aplicam (KITCHENHAM; PFLEEGER, 2008). Ao todo, 376 pessoas (n) responderam ao questionário.

**Fase 2. Aplicação do questionário e entrevistas.** Esta fase focou na aplicação do questionário e na execução de entrevistas. O questionário do estudo foi enviado por e-mail para a população-alvo, totalizando mais de 342 pessoas convidadas. Ao todo, o estudo teve 376 participantes. A população-alvo foi cuidadosamente selecionada pelos autores, visando evitar a coleta de dados de pessoas com um perfil indesejado. Graduandos, pós-graduandos (mestrado e doutorado) e profissionais da indústria, com formação acadêmica reconhecida, foram convidados, bem como profissionais identificados na rede social de profissionais LinkedIn. Os 376 participantes trabalham em 210 diferentes empresas e estão localizados em diferentes estados brasileiros, a exemplo: sul, sudeste, centro-oeste e norte. Finalizadas as etapas de envio e coleta das respostas ao questionário, 20 participantes foram selecionados de forma aleatória para a realização de entrevistas semiestruturadas. As entrevistas seguiram um roteiro previamente definido, alinhado aos objetivos do estudo, permitindo ao mesmo tempo a exploração de percepções adicionais emergentes durante as interações. Todas as entrevistas foram conduzidas de forma remota, gravadas com o consentimento dos participantes por meio da ferramenta Microsoft Teams e posteriormente transcritas para análise. A análise dos dados qualitativos seguiu um processo sistemático, no qual as transcrições foram examinadas por meio de codificação temática, visando identificar padrões, recorrências e tópicos relevantes relacionados ao uso da UML. As categorias emergentes foram então organizadas e confrontadas com os dados quantitativos obtidos a partir do questionário, permitindo a triangulação dos resultados e a exploração de aspectos complementares entre as diferentes fontes de dados.

**Fase 3. Análise dos dados.** Esta fase buscou realizar uma análise cuidadosa dos dados co-

**Figura 3:** Processo experimental

Fonte: Próprio autor

letados através do questionário e das entrevistas. Para isso, os dados coletados foram analisados isoladamente, bem como confrontados, visando realizar uma triangulação dos dados obtidos. Primeiramente, os dados coletados via questionário foram analisados e tabulados, sendo os resultados obtidos utilizados como base para a formulação das questões das entrevistas. Neste sentido, os entrevistados responderam questões que buscaram explorar mais profundamente os resultados obtidos, buscando uma consistência na análise feita dos dados. A análise dos dados partiu de uma visão ampla para uma visão mais focada. Assim, a investigação proporcionou interação através de um processo dialético via interação e reflexão entre pesquisador e participantes.

### 3.2.3 Questionário Formulado

Os dados foram coletados por meio de um questionário online criado no Formulários Google <sup>1</sup> seguindo diretrizes bem estabelecidas, como (KITCHENHAM; PFLEEGER, 2008) e (WOHLIN et al., 2012). Essa estratégia foi escolhida porque o questionário poderia ser aplicado rapidamente e também poderia coletar facilmente dados de um grande número de indivíduos em locais geograficamente diversos. As questões do questionário tratavam de examinar lacunas de pesquisas em estudos anteriores e apreender as estruturas do questionário desenvolvido an-

<sup>1</sup>Questionário: <https://forms.gle/TFRwsgJ7UFUcpafN7>

teriormente. Além disso, o desenho do questionário foi baseado nas descobertas relatadas por (PETRE, 2014).

### 3.3 Resultados

Esta seção apresenta os resultados das questões de pesquisa após a análise das respostas de 376 profissionais e das entrevistas com 20 participantes sobre o uso da UML em ambientes do mundo real. Os resultados obtidos durante a pesquisa foram representados por meio de gráficos e tabela de perfil dos participantes.

#### 3.3.1 Análise do Perfil dos Participantes

A Tabela 3 descreve o perfil dos participantes, relatando suas principais características, incluindo formação, curso de graduação, experiência geral, experiência profissional com modelagem de software e desenvolvimento. Ao todo, 376 participantes brasileiros responderam ao questionário formulado e trabalhavam para 210 empresas diferentes.

**Educação.** A maioria (68,1%) possuía pós-graduação completa (ou incompleta), enquanto 7,9% haviam concluído alguma pós-graduação na área de computação. Além disso, 20,6% dos participantes eram praticantes de nível técnico e cursavam graduação. 3,7% possuíam mestrado na área de computação. Apenas um participante não fez o curso de graduação em computação, mas sim em matemática, cursando posteriormente o mestrado em computação aplicada. Independentemente do nível de escolaridade, todos os participantes eram profissionais com experiência no setor.

**Curso de graduação.** A maioria dos participantes (91,8%) fez graduação em computação, incluindo análise de sistemas (51,9%), ciência da computação (28,7%) e sistemas de informação (11,2%), apenas quatro graduaram-se em outras áreas. Isso mostra uma sólida formação acadêmica, que complementa a experiência prática dos participantes. Considerando a posição dos participantes, 50,7% eram desenvolvedores, enquanto 23,6% analistas de sistemas. 2,4% Arquitetos e gerentes de software representaram 1,9% da amostra. Assim, 80% dos participantes ocupavam cargos diretamente relacionados às práticas de desenvolvimento de software.

**Experiência geral.** Mais de 74% indicaram ter dois anos (ou mais) de experiência profissional, dos quais mais de cinco anos de experiência prática. O nível de experiência é diversificado, evidenciando uma maior concentração de experiência na faixa de 2 a 6 anos de experiência. Além disso, 12,2% tinham mais de 7 anos de experiência. Assim, os participantes tiveram um nível de experiência diferente, com maior concentração de até seis anos de experiência.

**Experiência em modelagem.** Em relação às características da experiência de modelagem, os participantes tinham experiência, mas não muito, com modelagem de software. O resultado esperado seria a falta de experiência, uma vez que estudos empíricos anteriores apontam para a baixa adoção de modelos UML na indústria. Cerca de 38% dos participantes tinham

mais de dois anos de experiência profissional em modelagem de software, enquanto os demais afirmaram ter menos de dois anos de experiência.

**Experiência em desenvolvimento.** O tempo dos participantes aumentou considerando o nível de experiência com desenvolvimento de software. Todas as escalas aumentaram o percentual em relação à experiência com modelagem, observando uma redução expressiva (34,1%) na faixa de menos de 2 anos. A escala de menos de dois anos aumentou de 40% para 68%. Nesse sentido, isso indica mais experiência com desenvolvimento do que modelagem.

Dada a descrição acima, o perfil dos participantes é adequado para responder às questões de pesquisa de uma pesquisa exploratória por dois motivos. Primeiro, os participantes trabalham em empresas predominantemente diferentes. Isso evita respostas enviesadas por experiências obtidas em empresas específicas, além de potencializar a captura de experiências em diversos contextos de negócios e culturas.

Em segundo lugar, os participantes têm formação acadêmica em cursos de graduação em informática, garantindo uma formação teórica. Isso evita o risco de a não adoção da modelagem UML ser motivada pela falta de conhecimento. Os 20 participantes entrevistados apresentaram nível de conhecimento em modelagem superior a cinco anos e atuaram no desenvolvimento de software em áreas como educação (4 participantes), agronegócio (3), e-commerce (2), governo (3), comércio (3), exportação de produtos (2) e finanças (3). As seções a seguir discutem os resultados obtidos. As seções a seguir analisam os resultados das questões de pesquisa.

### 3.3.2 QP1.1: Fatores Que Afetam o Uso Eficaz da UML

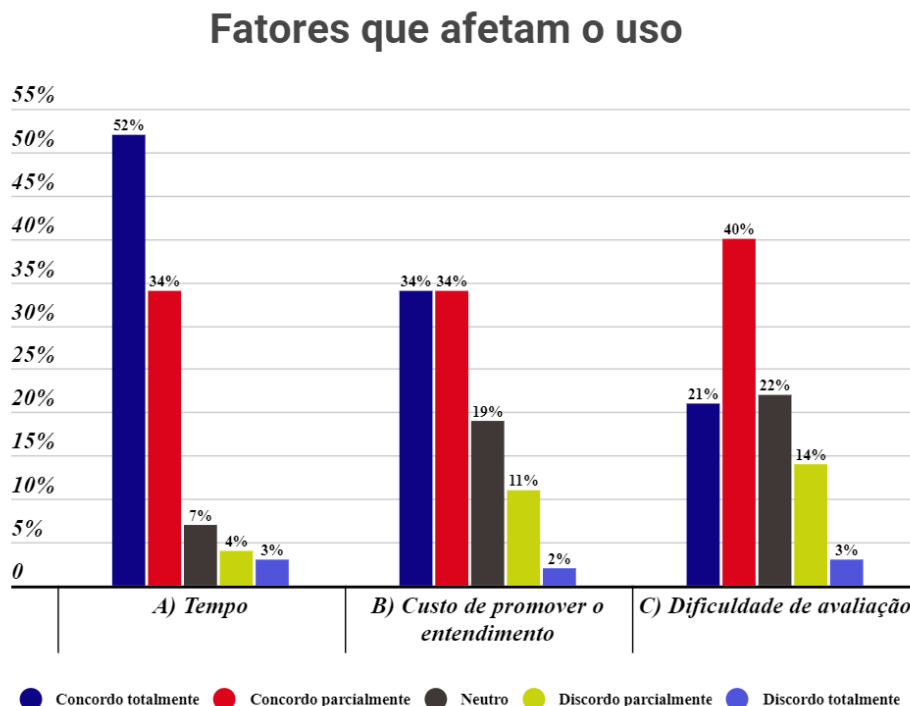
Para fazer a análise desta questão foram identificadas as seguintes variáveis para os profissionais que responderam o *survey*: (a) o curto espaço de tempo para desenvolver leva os desenvolvedores a não utilizarem os modelos, focando apenas no Código, (b) o custo de promover o correto entendimento entre diferentes pessoas com diferentes níveis de formação/experiência e formas de pensar é alto e (c) a dificuldade em avaliar a qualidade dos modelos criados.

A Figura 4 apresenta os dados coletados, entre os 376 participantes, que responderam a essa questão do estudo, apontando duas situações: (a) 52% dos participantes responderam que concordam plenamente; 34% que concordam parcialmente. Dos 20 entrevistados, 18 reforçaram que o curto tempo e as altas demandas atuais são os principais fatores que influenciam o uso da UML, uma vez que os sistemas de software desenvolvidos estão ficando maiores e mais complexos a cada dia devido à procura cada vez maior dos clientes e ao aprimoramento da tecnologia. Isso também leva a projetos complexos de sistema de software que não podem ser facilmente gerenciados pelas partes interessadas do projeto e fazem com que os sistemas de software sejam entregues com atraso (ou orçamento acima do normal) ou desenvolvidos incorretamente. Consequentemente, eles acabam optando por métodos informais, como prototipagem de tela, ou muitas vezes acabam nem modelando/documentando o projeto. A questão (b) obteve as seguintes respostas: 34% concorda totalmente; 34% concorda parcialmente; 19%

**Tabela 3:** Dados de perfil dos participantes.

<b>Característica (n=376)</b>	<b>Resposta</b>	<b>#</b>	<b>%</b>
Escolaridade	Técnico	77	20.6%
	Graduando	117	31.2%
	Graduado	138	36.9%
	Especialização	22	7.9%
	Mestrado	14	3.7%
Formação acadêmica	Análise de Sistemas	195	51.9%
	Ciência da Computação	108	28.7%
	Sistemas de Informação	42	11.2%
	Outros	31	8.2%
Posição	Desenvolvedor	187	50.7%
	Analista de sistemas	87	23.6%
	Arquiteto de software	9	2.4%
	Gerente	7	1.9%
	Outros	79	19.6%
Experiência Geral	< 2 anos	138	37.5%
	2-4 anos	129	35.1%
	5-6 anos	56	15.2%
	7-8 anos	20	5.4%
	> 8 anos	25	6.8%
Experiência profissional com modelagem de software	< 2 anos	227	61.2%
	2-4 anos	91	24.5%
	5-6 anos	25	6.7%
	7-8 anos	10	2.7%
	> 8 anos	18	4.9%
Experiência profissional com desenvolvimento de software	< 2 anos	126	34.1%
	2-4 anos	120	32.5%
	5-6 anos	54	14.6%
	7-8 anos	28	7.6%
	> 8 anos	41	11.1%



**Figura 4:** Resultados sobre fatores que afetam o uso da UML (QP1.1)

Fonte: Próprio autor

foram neutros. Sobre essa questão, quando perguntada aos entrevistados, a maioria (12) considerou que o custo de promover a compreensão correta entre diferentes pessoas com níveis de escolaridade/experiência e formas de pensar é baixo, divergindo dos dados quantitativos. Isso se deve possivelmente ao fato de que a maioria dos profissionais trabalha em equipes onde todos os membros têm o mesmo nível de experiência/treinamento. O conjunto de habilidades educacionais afeta como as partes interessadas aprenderam a modelagem de software, portanto, abordagens de modelagem e suas práticas relevantes através da experiência de modelagem

Na questão (c) 21% concorda totalmente; 40% concorda parcialmente; 22% foram neutros; 14% discorda parcialmente. Os entrevistados reforçaram que existe muita dificuldade em avaliar os modelos criados e identificaram que este é um dos fatores que afetam o uso eficaz da UML na indústria.

Além disso, esses resultados sobre os fatores que influenciam o uso reforçam os achados anteriores do estudo (CHAUDRON; HEIJSTEK; NUGROHO, 2012; FERNÁNDEZ-SÁEZ et al., 2015; BUCCHIARONE et al., 2021; STÖRRLE, 2017). Bucchiarone (BUCCHIARONE et al., 2021) defende que, nos estágios iniciais de desenvolvimento, as partes interessadas poderão modelar informalmente para apoiar processos comunicativos e cognitivos usando notações gráficas emergentes e flexíveis. (STÖRRLE, 2017) também indica que a modelagem informal (por exemplo, esboçar em um quadro branco) é considerada mais eficaz para promover a comunicação, a colaboração e a compreensão. (JACKSON, 2019) ressalta que as representações informais podem ser um bom começo para a modelagem, porém é limitada e dão interpre-

tações inconsistentes não podendo ser analisadas mecanicamente. Aprendemos com estudos experimentais anteriores, como (FARIAS et al., 2018; HO-QUANG et al., 2017; PETRE, 2013; SCANNIELLO et al., 2014), que há um conjunto de questões que desafiam o eficácia da UML. No entanto, os profissionais elaboraram práticas ad hoc que empregam UML de maneira eficaz no raciocínio e na comunicação sobre design, tanto individualmente quanto em diálogos colaborativos.

**Conclusão da QP1.1:** *Os resultados apontam que os participantes, em sua maioria, enxergam três pontos que afetam o uso dos diagramas da UML: (1) curto espaço de tempo para criar e manter os diagramas, em especial para desenvolver os sistemas de uma forma mais ampla; (2) a comunicação entre profissionais pode ser potencializada ao elaborar os diagramas; e (3) há uma dificuldade para avaliar a qualidade dos diagramas criados. Entendemos que as empresas podem precisar de diferentes padrões de abordagem de modelagem para diferentes projetos ou mesmo para diferentes funções individuais de engenharia de software dentro dos projetos. Esses três pontos devem ser observados pelas empresas ao implementar processos suportados por diagramas da UML.*

### 3.3.3 QP1.2: O Que Dificulta a Utilização da UML

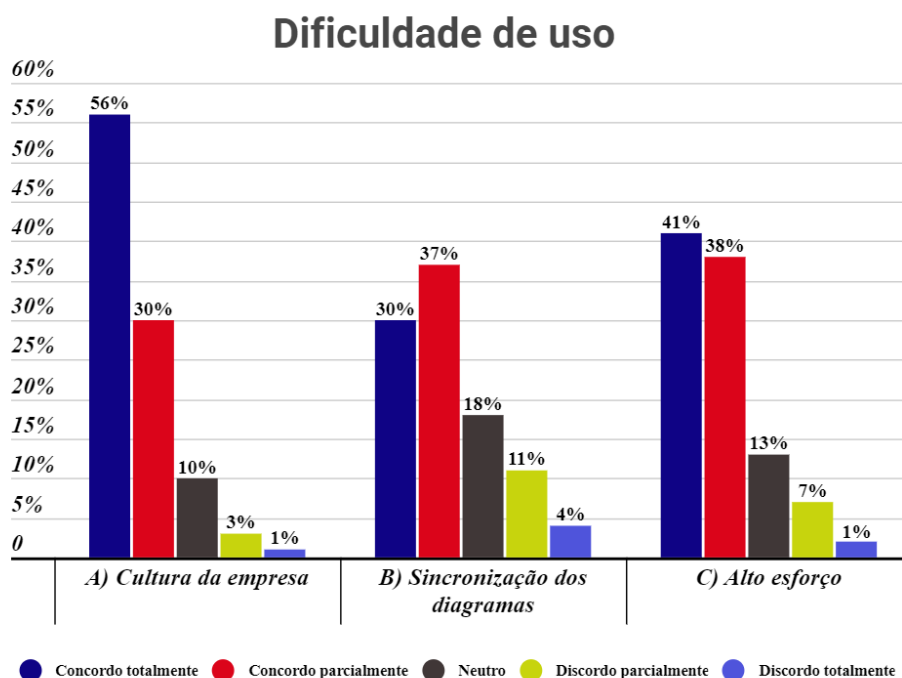
Para responder a esta segunda questão, foram aplicadas ao survey três variáveis (resultados apresentados respectivamente na Figura 5): (a) a cultura da empresa afeta a forma como a UML é utilizada, (b) os diagramas UML representam visões complementares do sistema), mantê-los sincronizados representa um dos fatores que dificultam o uso da UML em empresas e (c) o alto esforço para criar e manter os modelos.

Em relação a resposta (a): 56% dos entrevistados responderam que concordam totalmente; 30% que concorda parcialmente, e; 10% foram neutros. Um ponto evidente durante as entrevistas é que alguns participantes ressaltaram que isso se deve ao fato de uma “nova cultura” imposta que valoriza o erro para o aprendizado permitindo que a demanda seja produzida de forma incremental e corrigindo inúmeras vezes a falha, desta forma, diminuindo os aspectos de planejamento/modelagem. Importante ressaltar que as organizações podem precisar de diferentes padrões de abordagem de modelagem para diferentes projetos (mencionado na QP1.1) ou mesmo para diferentes funções individuais de engenharia de software dentro dos projetos. Conforme descrito em (HELDAL et al., 2016), diferentes unidades dentro da mesma empresa podem usar diferentes abordagens de modelagem. Além disso, no mesmo projeto de desenvolvimento de software, as mesmas funções de engenharia de software podem usar diferentes práticas de modelagem, dependendo de suas tarefas e responsabilidades para diferentes características de modelagem. Outros pontos relatados foram: a prática de uso/treinamentos, visão de valor e mudanças de paradigmas.

Para a questão (b): 37,7% respondeu que concorda parcialmente; 30% concorda totalmente; 18% neutros; 11% discorda parcialmente, 4% discorda totalmente. Este resultado reforça o achado relatado em alguns estudos realizados na indústria (CHAUDRON; HEIJSTEK; NUGROHO, 2012; CICHETTI; CICOZZI; CARLSON, 2016; KUHN; MURPHY; THOMPSON, 2012; LIEBEL et al., 2018), no qual os participantes relataram a falta de ferramentas de modelagem que pudessem apoiar a colaboração entre as equipes. Por exemplo, isso permitiria aos desenvolvedores estar cientes das mudanças que estão fazendo em tempo de execução, algo que já é suportado por Documentos Google. Além disso, as próximas ferramentas de modelagem precisam oferecer suporte à engenharia de ida e volta para sincronizar diagramas UML e código-fonte relacionados. Dado que modelar todos os aspectos estruturais e comportamentais de um sistema de software dentro de um único modelo não é uma tarefa trivial, a UML propôs um conjunto de diagramas para apoiar uma abordagem de modelagem de múltiplas visões. Assim, diferentes aspectos de um sistema em desenvolvimento são representados por vários modelos, as visualizações.

Para a questão (c): 41% responderam que concordam totalmente; 38% concordam parcialmente; 13% foram neutros; 7% discordam parcialmente, e 2% discordam totalmente. Sendo assim a grande maioria considera alto o esforço investido para criar e manter os modelos UML. Em (OZKAYA; ERATA, 2020), os autores mencionaram que uma razão importante neste contexto pode ser a falta de apoio para modelar os problemas de concorrência e desenvolvimento, uma vez que não fornece nenhum tipo de diagrama para modelar *threads* e processos de software e suas interações simultâneas. Neste contexto, outro fator que interfere apontado por (FERNÁNDEZ-SÁEZ et al., 2015) são as ferramentas de modelagem usadas para manter/modificar os diagramas UML no processo de desenvolvimento de software baseado em UML. Existem diferentes tipos de ferramentas com diferentes benefícios: ferramentas licenciadas (o que implica um investimento, mas também retorno com possíveis treinamentos, customizações, etc.) vs. ferramentas abertas, ou ferramentas específicas para modelagem em UML (que verificam a correção da sintaxe) ou ferramentas gerais de modelagem (estas são mais “acessíveis”).

A UML foi identificada como a notação dominante em (FORWARD; LETHBRIDGE, 2008), com 52% de uso. Eles descobriram que as ferramentas de modelagem são usadas principalmente para o trabalho de projeto inicial; geração de código não é amplamente utilizado. Os participantes deste estudo pareciam realmente querer incorporar a modelagem em seus processos, mas não conseguiam, pois o maior problema percebido das abordagens centradas no modelo é manter ele atualizado com o código (68% de concordância). Esta análise de submostra é particularmente interessante, descobrindo que programadores são mais propensos a concordar que as ferramentas de modelagem são muito “pesadas”.

**Figura 5:** Resultados sobre dificuldade de uso da UML (QP1.2)

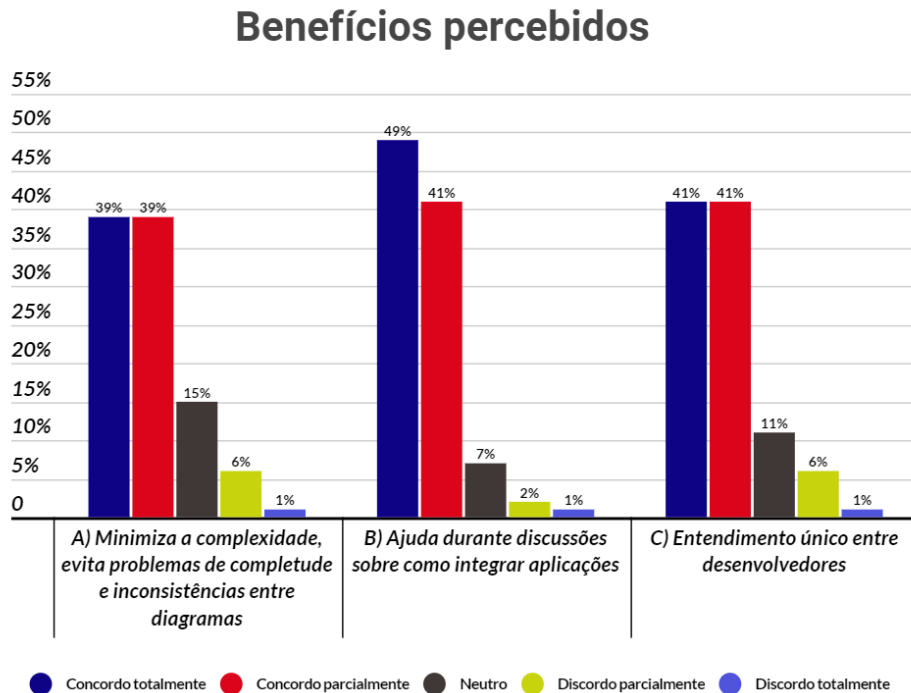
Fonte: Próprio autor

**Conclusão da QP1.2:** Ao confrontar os profissionais com as três variáveis analisadas, observou-se que eles identificam como dificuldades no uso da UML fatores relacionados ao contexto organizacional e às práticas adotadas em projetos de software, especialmente no que se refere ao nível de priorização dado às atividades de documentação. Além disso, foram apontados desafios associados à manutenção da consistência entre os diferentes diagramas UML ao longo do projeto, bem como ao esforço necessário para criar e manter os modelos.

### 3.3.4 QP1.3: Quais Benefícios São Percebidos ao Usar a UML

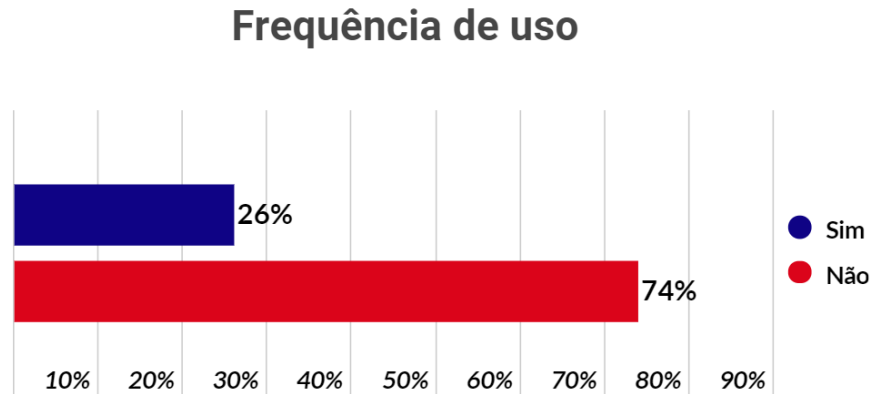
Para analisar esta questão foram apresentadas as três variáveis aos profissionais (cujos resultados estão respectivamente nas Figura 7): (a) usar a UML seletivamente (apenas alguns diagramas) ajuda a minimizar a complexidade, evitar problemas de completude e inconsistência entre os diagramas, (b) modelos UML ajudam durante discussões sobre como integrar aplicações e (c) A UML ajuda na formação de um entendimento único entre os desenvolvedores.

É apresentado abaixo as respostas de 376 participantes sobre quais os benefícios são percebidos ao usar a UML. Para a questão (a): 39% concorda totalmente; 39% concorda parcialmente, e; 15% são neutros. Para a questão (b): 49% concorda totalmente; 41% concorda parcialmente, e; 7% são neutros. Para a questão (c) 41% concorda totalmente; 41% concorda parcialmente; e 11% são neutros.

**Figura 6:** Resultados dos benefícios percebidos sobre o uso da UML (QP1.3)

Fonte: Próprio autor

Todos os entrevistados concordaram por unanimidade que o uso da UML (mesmo que seletivamente) beneficia o desenvolvimento de software, pois ajuda no nível de compreensão geral do contexto, além de facilitar a comunicação entre a equipe, a detecção de defeitos e a integração com aplicações heterogêneas. Interpretações inconsistentes de artefatos de software podem comprometer o alinhamento entre as partes envolvidas no projeto. Em contextos nos quais equipes de desenvolvimento, áreas organizacionais e órgãos reguladores utilizam esses artefatos como base para tomada de decisão, divergências de entendimento podem resultar na necessidade de revisões e ajustes posteriores, impactando prazos e, em determinados casos, gerando implicações administrativas, financeiras ou legais. Esse risco é exacerbado pelo fato de que a verificação da conformidade geralmente é realizada no final do processo de desenvolvimento de software. Consequentemente, qualquer problema descoberto na verificação de conformidade é caro para reparar (USMAN et al., 2020). Esses fatores são identificados em práticas (HO-QUANG et al., 2017) onde a maioria dos participantes (79%) achou a UML útil para entender sistemas, melhorar a comunicação entre desenvolvedores, orientar a implementação e gerenciar a qualidade do projeto. In (PETRE, 2014), os participantes relataram usar UML com mais entusiasmo, trabalhando de forma mais focada em escopo, mantendo os artefatos gerenciáveis em tamanho e adequados como forma de evitar problemas de sincronização e consistência. Uma área que merece investigação adicional é como o uso de UML é moldado pelo contexto do domínio - uma investigação que requer muito mais acesso a uma variedade de indústrias de software.

**Figura 7:** Resultados sobre a frequência de uso da UML (QP1.4)

O contexto atual demonstra que é preciso compreender com que de fato facilita a aplicabilidade eficaz do desenvolvimento de software fugindo de estagnar apenas na teoria. Todas essas evidências destacam a necessidade de considerar a relação das ferramentas, incluindo a notação, tanto com a comunidade de prática quanto com o domínio de aplicação. Os participantes reforçaram o fato de que desenvolvedores de software estão abertos a entenderem os conceitos e que ao mesmo tempo querem utilizar ferramentas que tornem o processo eficaz, caso contrário tendem a descartá-las se estiverem em desacordo com as suas práticas.

**Conclusão da QP1.3:** *Usar apenas alguns diagramas da UML auxiliam ao minimizar a complexidade e evitar problemas de completude e inconsistências entre estes; Na visão destes profissionais a utilização da UML é benéfica para os projetos e evita problemas ao permitir a visualização da resolução e auxilia nas discussões da integração de aplicações.*

### 3.3.5 QP1.4: A Frequência de Uso da UML

A Figura 7 apresenta as respostas dos participantes sobre a utilização da UML, no seu dia a dia. Como a pergunta não foi de cunho obrigatório, 365 dos 376 participantes responderam. Para esta questão, 74% responderam que não utilizam a UML no seu cotidiano, enquanto 26% responderam que a utilizam frequentemente.

Esse resultado reforça o achado em (PETRE, 2014), no qual a autora relata que 35 dos 50 sujeitos em seu estudo não usam UML na prática e (GORSCHKE; TEMPERO; ANGELIS, 2014) onde valida que os modelos de design não são usados extensivamente na indústria, porém, quando são, o uso é informal, com o mínimo ou nenhum suporte de ferramenta e a notação não é necessariamente UML.

Os 20 entrevistados afirmaram que não usavam UML frequentemente, mas reconheceram os diversos benefícios de usá-lo no desenvolvimento de um projeto de sistema de software, incluindo o nível de compreensão de todo o processo, integração/comunicação entre a equipe, redução de custos em manutenção e rastreabilidade. Este estudo reforça os achados de Petre (PETRE, 2014) onde 35 dos 50 afirmaram que não usavam UML na prática. Neste estudo, também é apontada a necessidade de compreender a relação entre método e prática. Hoje, é necessário compreender as novas demandas da indústria de software, como aponta em (STÖRRLE, 2017) que contrasta com os resultados aqui apresentados indicando que isso pode estar ligado a diferenças culturais mais gerais. Da mesma forma, os resultados de (OZKAYA; ERATA, 2020), mostram que a maioria dos profissionais (88%) usa UML na modelagem de seus sistemas de software de diferentes pontos de vista de arquitetura. Entre os pontos de vista da arquitetura considerados (ou seja, funcional, informação, simultaneidade, desenvolvimento, implantação e operacional), os principais pontos de vista populares nos quais os participantes modelam seus sistemas usando UML são os pontos de vista funcional e de informação (96–99%). O ponto de vista operacional é o menos popular, ignorado por 61% dos participantes em sua modelagem de software com UML. Estudos como (KOBRYN, 2002; DORI, 2002; THOMAS, 2004) argumentam que a UML não está cumprindo um papel de ser uma língua franca ou padrão por causa de questões como tamanho, complexidade, semântica, consistência e transformação do modelo.

**Conclusão da QP1.4:** *Observa-se que, no contexto industrial analisado, a UML não é utilizada com alta frequência pela maioria dos participantes da pesquisa. Além disso, quando adotada, sua utilização ocorre de formas variadas e restritas a contextos ou finalidades específicas, não havendo um padrão predominante de uso.*

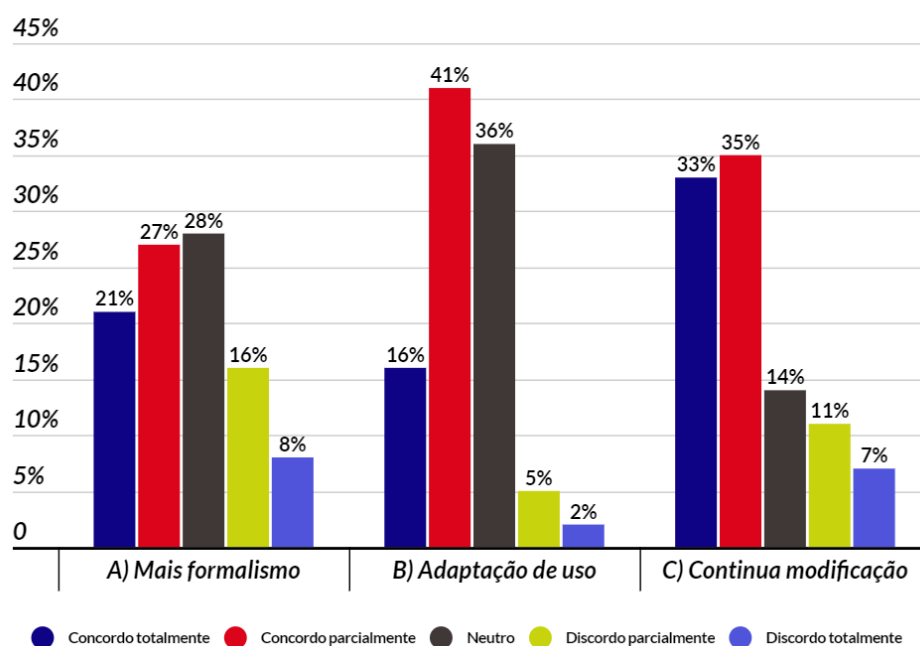
### 3.3.6 QP1.5: Quais as Principais Características da Atual Demanda de Desenvolvimento de Software Limitam o Uso da UML

Para analisar esta questão foram apresentadas as três variáveis aos profissionais (cujos resultados estão respectivamente nas Figura 9): (a) Se a UML tivesse mais formalismo os desenvolvedores passariam a utilizá-la com mais frequência, (b) A utilidade da UML para os profissionais da indústria surge do fato de adaptar o seu uso para um propósito específico e (c) As empresas tendem a desenvolver software relativamente pequeno que passa por contínua modificação. Desenvolvedores passam a guardar as decisões de projeto "na mente" e a se comunicarem efetivamente sem qualquer diagrama formal.

É apresentado abaixo as respostas de 376 participantes sobre as principais características da atual demanda de desenvolvimento de software acabam limitando o uso da UML. Para a questão (a): 21% concordam totalmente, 27% concordam parcialmente, 28% são neutros, 16% discordam parcialmente e 8% discordam totalmente. Dos 20 participantes entrevistados, 15 consideram que o alto grau de formalismo torna-se um fator negativo para a aplicabilidade da



**Figura 8:** Resultados sobre contextos de uso (QP1.5)  
**Contextos de uso**



Fonte: Próprio autor

UML no contexto do desenvolvimento atual, uma vez que os processos são altamente dinâmicos e ágeis, o que acaba exigindo um menos formal e mais interativo. O contexto de projetos atuais (principalmente os ágeis) passam por processos de atualizações rapidamente e normalmente ter que remodelar toda a documentação gerada pela UML acaba tornando o custo mais elevado. Diante disso, mesmo sendo o ideal, na maioria dos projetos de software não há recursos suficientes para levar em conta esse custo por isso a popularidade de uma modelagem informal (por exemplo, esboçando em um quadro branco) é particularmente útil para comunicação, colaboração e compreensão. Nesse contexto, soluções baratas como quadro branco ou ferramentas de desenho são suficientes para colher benefícios significativos e tornar a fase de design mais rápida e eficaz sem investimentos caros. Além disso, sendo mais flexível e menos acorrentado a formalismos e restrições específicos, a modelagem informal empurra para baixo a curva de aprendizado /treinamento de iniciantes e apoia as necessidades de uma variedade maior de partes interessadas.

Para a questão (b): 16% concorda totalmente; 41% concorda parcialmente, e; 36% são neutros. A maioria dos entrevistados (12) ressaltaram que a UML pode se tornar adaptável a um propósito específico, porém relataram que no cotidiano é complicado encontrar diferentes técnicas para se tornar essa adaptação efetiva.

Para a questão (c) 33% concorda totalmente; 35% concorda parcialmente; e 14% são neutros, estes dados corroboram com a visão dos entrevistados sobre as exigências da demanda atual de gerar um processo incremental.



Em (LI et al., 2025; FARIAS et al., 2018; PETRE, 2014) fica evidenciado que embora os participantes do estudo acreditem, em sua maioria, que a UML seja uma “língua franca” nas empresas e que eles tenham conhecimento teórico sobre este tipo de modelagem acabam não a utilizando com frequência. Os resultados de (FERNÁNDEZ-SÁEZ et al., 2015) revelaram os desenvolvedores de software que usam diagramas UML acabam enfrentando dificuldades no que diz respeito à leitura dos mesmos. A maioria das empresas pesquisadas, portanto, usa os diagramas UML “mais compreensíveis”. Os mantenedores nem sempre utilizam a documentação disponível e trabalham diretamente com o código-fonte; mesmo que a documentação esteja disponível, ela não é utilizada. Isso sugere que a adoção da UML pode estar relacionada às características do contexto organizacional e aos processos de desenvolvimento adotados pelas empresas, os quais influenciam o nível e a forma de utilização da linguagem ao longo dos projetos.

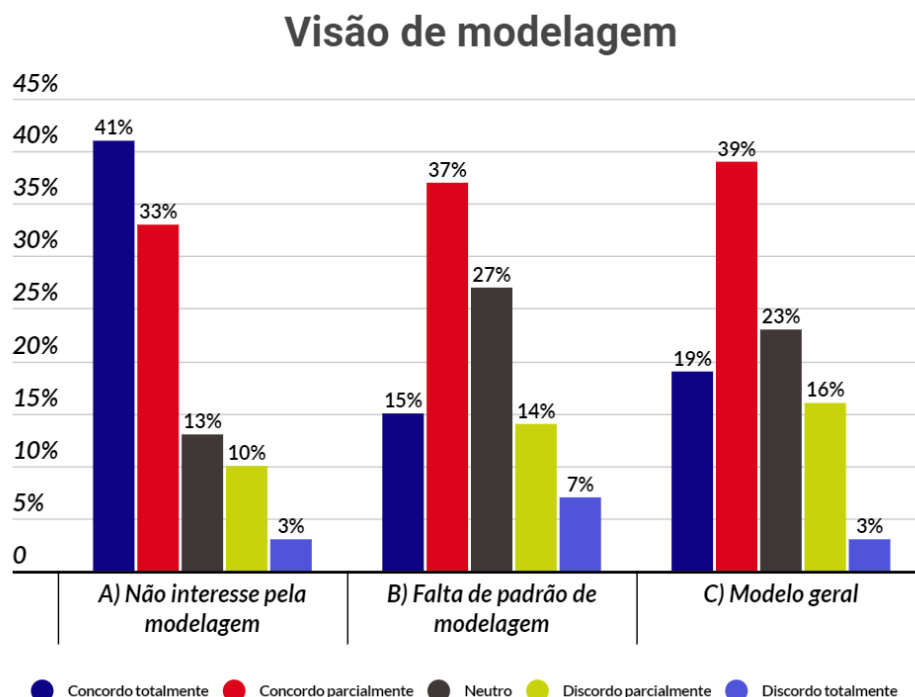
**Conclusão da QP1.5:** *É necessário entender o contexto de desenvolvimento de software atual para buscar novas estratégias de aplicar a UML na indústria. As três variáveis apresentadas, nesta questão, reforçam fatores de que novas atualizações precisam ser realizadas nesta linguagem de modelagem para potencializar o seu uso.*

### 3.3.7 QP1.6: Qual a Visão dos Desenvolvedores em Relação à Modelagem Utilizando UML

Para analisar esta questão foram apresentadas as três variáveis aos profissionais (cujos resultados estão respectivamente nas Figura 10): (a) Desenvolvedores não gostam de modelar, (b) A falta de padrões de modelagem que guiem os desenvolvedores com boas práticas de como representar os modelos e (c) Dificuldade de criar um “modelo geral” de forma fácil.

Os dados a seguir são baseados nas respostas de 376 participantes que responderam o *survey* e confrontados com os dados levantados na fase de entrevista (20 participantes). Para a questão (a): 41% concordam totalmente, 33% concordam parcialmente e 13% são neutros. Estes dados convergem com a resposta dos 20 participantes entrevistados, sendo que 13 deles ressaltaram que os desenvolvedores gostam e entendem a importância da modelagem, todavia por alguns motivos justificados nas próximas questões (a e b) e até mesmo em outras Questões de Pesquisa (QP1.1 e QP1.2) esses fatores acabam limitando e até mesmo anulando esta fase. Em (PETRE, 2013) a UML é considerada “desnecessariamente complexa” por vários participantes deste estudo que relataram variações na compreensão e interpretação entre os desenvolvedores acarretando em problemas como por exemplo, desafios na semântica formal da linguagem. Outros notaram que as complexidades da notação limitavam sua utilidade – ou exigiam uso direcionado – em discussões com as partes interessadas (incluindo as partes interessadas altamente técnicas).

Para a questão (b): 15% concorda totalmente; 37% concorda parcialmente, e; 27% são neu-

**Figura 9:** Resultados sobre visão de modelagem (QP1.6)

Fonte: Próprio autor

tros. Segundo os entrevistados essa falta de padrão de modelagem acaba desestimulando os desenvolvedores a realizar a modelagem com UML. Em seu estudo de engenharia orientada a modelos na indústria (HUTCHINSON et al., 2011; HUTCHINSON; ROUNCEFIELD; WHIT-TLE, 2011), descobriram que as pessoas tendem a usar várias linguagens de modelagem. As empresas que usam MDE tendem a desenvolver linguagens específicas de domínio (DSLs), e essa noção é muito focada no produto/implementação.

Para a questão (c) 19% concorda totalmente; 39% concorda parcialmente; e 23% são neutros, estes dados são reforçados com as respostas da maioria dos participantes entrevistados (16) que também expuseram o nível de dificuldade em modelar o software apresentando uma visão na íntegra.

Em (FERNÁNDEZ-SÁEZ; CHAUDRON; GENERO, 2018) foi considerado fornecer uma visão abrangente e sistemática dos principais desafios na modelagem de software e compreender as diferentes categorias deles junto com as discussões dos desafios concretos em cada categoria que os profissionais podem enfrentar. Neste estudo foi proposto 8 tipos diferentes de desafios, entre eles (i) gerenciamento da complexidade da linguagem, (ii) extensas linguagens de modelagem, (iii) ambientes de modelagem de domínio específico, (iv) desenvolver linguagens de modelagem formal, (v) analisando modelos, (vi) separação de preocupações, (vii) transformando modelos, e (viii) modelos de gestão.

**Conclusão da QP1.6:** *Fica evidente que a maioria dos desenvolvedores não gosta de modelar e esse desinteresse é justificado por fatores cruciais como a ausência de padrões e também a dificuldade em trazer os aspectos desta fase para o desenvolvimento do software como um todo. É necessário pensar em novas formas de facilitar a modelagem e aproximar os desenvolvedores dela tornando o processo mais simples, dinâmico e motivador.*

### 3.4 Discussão Adicional

Para fornecer uma discussão adicional, a Seção 3.4.1 traz reflexões e direções futuras sobre os resultados obtidos. Seção 3.4.2 discute questões relacionadas à adoção de modelagem contínua. Seção 5.1.2 descreve algumas discussões sobre modelagem de software gamificado como uma maneira de melhorar a adoção de modelos UML.

#### 3.4.1 Análises dos Resultados

**Restrição de tempo e falta de conhecimento.** Os resultados apontam a restrição de tempo como um dos principais fatores que afetam o uso dos diagramas da UML. Embora os participantes reconheçam a importância e os benefícios de criar os diagramas da UML, o curto espaço de tempo em projetos levam aos profissionais, ou a não utilizarem os diagramas, ou utilizarem de forma restritiva. Além disso, a falta de conhecimento aprofundado sobre os diagramas da UML seria um impeditivo para o uso, visto que o custo de promover a disseminação dos conhecimentos entre pessoas com diferentes níveis de formação e de experiência é muito alto para as empresas. Ao criar diagramas, mesmo com restrição de tempo e com pouco conhecimento, a dificuldade de avaliar a qualidade dos diagramas criados seria outro desafio ao utilizar os diagramas.

**Cultura da empresa e agilidade.** A cultura de agilidade desenvolvida nas empresas foi identificada como um impedimento ao uso da UML. Atualmente, preparar e manter diagramas UML são duas atividades manuais que requerem conhecimento de representações abstratas de domínio e conceitos tecnológicos. Neste contexto, cresce a popularidade da modelagem informal (por exemplo, criando esboços em lousas) como forma de tentar melhorar a colaboração, comunicação e entendimento de todos os envolvidos no desenvolvimento de software podem acabar se tornando soluções com custos menores (lousas, ferramentas de prototipagem, entre outras). Um outro ponto a ser discutido sobre isso, é que esses modelos se tornam menos formais e mais flexíveis visto que o aprendizado é mais simplificado, bastando poucos treinamentos e dando suporte para as necessidades de uma variedade maior de partes interessadas. Normalmente, trabalhar com a representação de abstrações no contexto da cultura de agilidade não tem se mostrado apropriado. Pelo contrário, fazê-lo rapidamente, entregar e receber

feedback de clientes de código executáveis, mesmo com problemas conceituais, tornou-se atividades prioritárias. Entregar rapidamente (sem grandes planejamentos) e considerar erros como um processo natural para chegar ao produto de software final também se tornaram atividades prioritárias. Neste contexto, a modelagem multi-visão proposta pela UML não encontra espaço de aplicação, embora seja reconhecida como algo importante.

**Uso seletivo de diagramas e complexidade.** Quando questionados sobre quais benefícios são percebidos ao usar uml, a maioria dos participantes respondeu que o uso de diagramas UML seletivamente, ou seja, apenas alguns de seus diagramas ajuda a minimizar a complexidade, evitar problemas de completude e inconsistência entre os diagramas dos usuários e ajudar a formar uma compreensão única entre os desenvolvedores e mesmo que os diagramas uml ajudem durante as discussões sobre como integrar aplicativos. Esta conclusão também foi verificada por (DZIDEK; ARISHOLM; BRIAND, 2008; GONÇALES; FARIAS; BISCHOFF, 2019). Os desenvolvedores tendem a concentrar os maiores esforços da fase de modelagem nas partes mais complexas ou críticas dos sistema. A generalidade e a liberdade que permitem à UML atender a essa ampla gama de propósitos também são a fonte de sua fraqueza. UML não tem semântica formal. Isso representa um problema quando pessoas diferentes usam um modelo UML; e como um dos principais objetivos da UML é comunicar sobre um projeto, diferentes maneiras de usar a UML são causas potenciais de problemas de comunicação (LANGE; CHAUDRON; MUSKENS, 2006).

### 3.4.2 Adoção de Modelagem Contínua

Atualmente, as empresas buscam não apenas agilizar seus processos, mas principalmente encontrar continuidade ao longo do ciclo de desenvolvimento de software (CHEN, 2015, 2017; LAUKKANEN; ITKONEN; LASSENIUS, 2017; FITZGERALD; STOL, 2017; ELAZHARY et al., 2021). Fitzgerald e Stol(2017) argumentaram que alcançar fluxo e continuidade ao longo do ciclo de desenvolvimento de software é muito mais importante em uma primeira instância do que a velocidade. Como as empresas estão cada vez mais priorizando a implementação de práticas de entregas contínuas (CHEN, 2015), para se beneficiar da adoção de modelos UML (BUCCHIARONE et al., 2021; CHAUDRON; HEIJSTEK; NUGROHO, 2012; DZIDEK; ARISHOLM; BRIAND, 2008), devem aumentar seus investimentos para inserir modelos UML ao longo de seu ciclo de desenvolvimento. Essa expansão da adoção de modelos UML nas empresas requer mudanças, por exemplo, no pipeline e na plataforma de entrega contínua, dando origem a *modelagem contínua de software*. No entanto, a implementação da modelagem contínua de software representa desafios consideráveis.

**Desafios técnicos.** Faltam soluções robustas, ferramentas e boas práticas fora da caixa e altamente adaptáveis às realidades das empresas. A ausência dessa solução levou à adoção isolada, ao contrário da adoção contínua de modelos UML em todo o processo de entrega contínua. Ferramentas de modelagem que preenchem essa lacuna podem trazer os benefícios já documen-

tados do uso de modelos UML para a realidade das empresas, destacando também a economia de recursos. Ao construir uma plataforma de modelagem contínua, diferentes ferramentas e tecnologias podem ser usadas como blocos de construção para o pipeline de entrega contínua (CHEN, 2015, 2017). Mas as empresas não podem ser presas por fornecedores de tais ferramentas ou tecnologias. A comunidade científica poderia propor padrões de modelagem amplamente aceitos, boas práticas sensíveis às realidades das empresas, definir APIs abertas (modelagem de software como serviço) e construir um ecossistema de ferramentas para a construção de um pipeline de modelagem contínua de software.

Normalmente, os desenvolvedores de software precisam lidar com empresas com modelos UML inadequados (por exemplo, grandes modelos monolíticos) para as práticas de processo adotadas pelas empresas. Hoje, os ciclos de desenvolvimento são curtos para entregar recursos recém-solicitados rapidamente, estabelecendo um ciclo contínuo de obtenção de feedback. Modelos monolíticos grandes precisam ser caracterizados e repensados como modelos UML orientados a recursos. Esse alinhamento entre os modelos UML e a realidade dos processos das empresas pode aumentar a adoção da modelagem de software. Nesse sentido, um grande desafio hoje seria a criação de abordagens de modelagem contínua orientadas às características, bem como a produção de evidências empíricas sobre as vantagens e desvantagens da adoção da modelagem contínua da UML. Resolver esse desafio exigirá estreita colaboração entre pesquisadores e profissionais e permitirá que os benefícios da modelagem UML sejam trazidos para a realidade de mais empresas.

**Desafio de processo.** A modelagem UML ainda está associada às práticas de processo tradicionais, e não está claramente inserida em práticas ágeis. Por exemplo, um diagrama de classe UML (modelo conceitual) para estar pronto para uso pelos desenvolvedores precisaria representar os conceitos e seus relacionamentos encontrados nos requisitos. Para esse fim, a elaboração de um modelo conceitual pode ser apoiada por atividades de engenharia de requisitos em níveis variados de detalhamento, conforme os objetivos do projeto. Em alguns contextos, isso pode incluir a produção de especificações de requisitos e diagramas de casos de uso que representem as funcionalidades consideradas relevantes — tarefas predominantemente manuais e propensas a erros. Isso pode atrasar ou impedir a elaboração de modelos em um contexto de curto ciclo de desenvolvimento orientado a poucas características. Em determinados contextos, o tempo necessário para a preparação de modelos UML pode representar uma parcela significativa do esforço de uma iteração de desenvolvimento, especialmente quando considerado em relação à duração da *sprint*, à complexidade do sistema e às práticas adotadas pela equipe. Há lacunas de pesquisa para buscar alternativas, visando o alinhamento entre processos de negócios, práticas ágeis de desenvolvimento e questões técnicas.

**Documentação e sistemas monolíticos legados.** Promover a modelagem de grandes sistemas sem documentação ainda é um desafio há décadas, pode haver também uma tendência cultural de assumir que o status quo é o único caminho possível. A ausência de documentação é um fator complicador na reestruturação de sistemas monolíticos legados em sistemas baseados

em uma arquitetura de microsserviço. Portanto, também dificulta a implementação de práticas contínuas de entrega. Os sistemas legados normalmente têm dezenas de subsistemas bem acoplados que interagem para fornecer diferentes serviços para clientes internos e externos dentro das empresas. (FITZGERALD; STOL, 2017) ressaltam que a falta de documentação leva em conta apenas o conhecimento tácito de engenheiros de software que trabalham em diferentes departamentos.

A modelagem de sistemas legados com base na criação de uma *"big picture view"* ainda é difícil de executar na prática devido ao tamanho dos sistemas, geralmente consistindo de centenas de milhares de linhas de código em Java, por exemplo. Atualizações contínuas para esses modelos podem ser altamente desafiadoras. A modelagem multivisão da UML permite atualização de modelos complementares, como diagramas de classe e diagramas de sequência. Isso pode levar a inconsistências entre tais modelos (KRETSCHMER et al., 2021; KHELLADI; KRETSCHMER; EGYED, 2019; REDER; EGYED, 2013).

### 3.4.3 Gamificação e Modelagem de Software

Gamificação (DETERDING et al., 2011a; HUOTARI; HAMARI, 2017; LIU; SANTHANAM; WEBSTER, 2017) é definida como "o uso de elementos de design de jogo em contextos não-jogos". Esta técnica utiliza a filosofia, elementos e mecânicas do design de jogos em ambientes não-jogos, trazendo assim todos os aspectos positivos que eles fornecem. A literatura atual reconhece a importância da aplicação da gamificação no contexto da prática de engenharia de software; no entanto, a forma de projetar e utilizar a gamificação no contexto da modelagem aplicada às necessidades industriais ainda é uma questão em aberto. Até onde sabemos, há poucos estudos sobre a aplicação da gamificação em práticas de engenharia de software — a maioria das quais estão relacionadas a contextos mais amplos de engenharia de software (PORTO et al., 2020; PEDREIRA et al., 2015; REN; BARRETT; DAS, 2020). Em atividades mais específicas, como gerenciamento de projetos (PORTO et al., 2020) ou desenvolvimento (DUBOIS; TAMBURRELLI, 2013), o cenário é pior.

Esta seção conjectura que a inserção de técnicas de gamificação, como feedback, progresso e desafios, na modelagem de software poderia mitigar os fatores que influenciam o uso e os desafios que dificultam a adoção. Por exemplo, a incompletude dos modelos UML é um problema crítico (LANGE; CHAUDRON; MUSKENS, 2006; FERNÁNDEZ-SÁEZ; CHAUDRON; GENERO, 2018). O uso de técnicas de gamificação, como desafios, pontos, feedback e progresso, poderia motivar os desenvolvedores a criar modelos mais completos em troca de pontos, por exemplo. Quadros de classificação poderiam ser criados para avaliar os desenvolvedores em termos da qualidade dos modelos criados. Além disso, o feedback constante durante a edição dos modelos poderia fomentar o aprendizado, além de estimular o desenho de diagramas. Estudos empíricos podem ser realizados como forma de analisar a integração entre gamificação e modelagem de software utilizando UML com base nos fatores mencionados no QP1.1 e

QP1.2, para aumentar a percepção de benefícios por profissionais (QP1.3) e a frequência de uso (QP1.4). Portanto, o uso de técnicas de gamificação pode motivar os desenvolvedores, melhorar a qualidade dos modelos UML criados e fomentar o aprendizado.

### 3.5 Ameaças à Validade

Nesta seção, serão analisadas as possíveis ameaças à validade do survey.

**Validade interna.** A validade interna está relacionada a questões que podem afetar a relação causal entre o tratamento e o resultado. Os principais pontos que afetam a validade interna do estudo referem-se ao perfil e experiência dos participantes. Ao analisar o perfil dos participantes, conforme apresentado na seção 3.3, nota-se que pelo menos 30% deles têm baixa (até 4 anos) experiência geral, baixa experiência com modelagem de software e baixa experiência com desenvolvimento de software. Isso provavelmente porque o nível de escolaridade de cerca de 50% deles é de ensino técnico ou ainda estão cursando a graduação. No entanto, devido ao tamanho da amostra e também às entrevistas realizadas, acreditamos que os dados coletados não são afetados por essa ameaça. Outro fator de ameaça interna está ligado ao processo aleatório de seleção dos participantes para a entrevista, o que pode ter causado uma potencial semelhança de perfil dos entrevistados. Embora os participantes entrevistados tenham trabalhos de desenvolvimento de software nas áreas de educação, agronegócio, *e-commerce*, governo, comércio, exportação de produtos e finanças, reconhecemos que os dados qualitativos poderiam ser ainda mais explorados se tivéssemos uma maior participação de profissionais ligados a outros setores. No entanto, temos uma variedade maior de setores através de participantes da pesquisa.

**Validade externa.** A validade externa está preocupada com a capacidade de generalizar os resultados além do estudo real. Para realizar a interpretação correta dos resultados da pesquisa, embora os dados demográficos da amostra sejam diversificados, entendemos que a generalização deles para toda a população pode não ser adequada. Neste estudo, os participantes pertenciam a uma variedade geográfica e trabalham em empresas de diferentes domínios e tamanhos, no entanto, não podemos ter certeza de que essa amostra é representativa da indústria em geral.

### 3.6 Considerações Finais do Capítulo

Este capítulo respondeu à QP1 desta tese ao apresentar uma investigação empírica sobre como a modelagem UML tem sido utilizada na indústria. Os principais resultados mostram que 74% dos participantes afirmaram não utilizar UML em seu cotidiano, enquanto apenas 26% a utilizam com frequência. Além disso, 54% dos participantes concordam que fatores como prazos reduzidos de desenvolvimento, custos de disseminação da prática entre equipes heterogêneas e dificuldades na avaliação da qualidade dos modelos afetam diretamente o uso eficaz da UML. Esses achados reforçam evidências já identificadas na literatura sobre a subutilização ou aplicação inadequada da UML em projetos de software (FARIAS et al., 2018; FARIAS, 2016;



PETRE, 2014; GORSCHER; TEMPERO; ANGELIS, 2014). Embora a maioria dos profissionais conheça a linguagem, sua adoção prática permanece limitada, o que sugere oportunidades de melhoria tanto no ensino quanto na aplicação industrial. Tais resultados podem orientar empresas na adoção de estratégias para reduzir custos futuros de manutenção e apoiar o uso sistemático da modelagem ao longo do ciclo de desenvolvimento.

Diante desses desafios evidenciados no estado atual da prática, surge a necessidade de investigar mecanismos que possam apoiar a aprendizagem e contribuir para a melhoria da qualidade dos modelos produzidos. A ausência de ferramentas e diretrizes que auxiliem instrutores e desenvolvedores a avaliar e aprimorar diagramas UML indica uma lacuna concreta que precisa ser enfrentada. Nesse sentido, os resultados apresentados neste capítulo não apenas evidenciam a necessidade de novas abordagens, mas também motivam diretamente a investigação proposta no próximo capítulo, que apresenta o *ModelGame*, um modelo de qualidade gamificado concebido para apoiar a aprendizagem de modelagem de software e oferecer uma estrutura de referência sistematizada para avaliação de modelos UML em atividades gamificadas.



## 4 MODELGAME: MODELO DE QUALIDADE GAMIFICADO PARA APRENDIZAGEM DE MODELAGEM DE SOFTWARE

A gamificação foi adotada em tarefas de desenvolvimento de software nos últimos anos. Essa adoção visa, por exemplo, melhorar o engajamento dos desenvolvedores ao criar modelos UML ou escrever código. Estudos empíricos (FARIAS; GARCIA; LUCENA, 2012; FARIAS et al., 2015; LANGE; CHAUDRON; MUSKENS, 2006) relatam que os modelos UML sofrem de problemas de incompletude e inconsistência. (LANGE; CHAUDRON; MUSKENS, 2006) reforçam ainda que esses defeitos trazem riscos potenciais que podem causar erros de interpretação e falha na comunicação, representando um risco à qualidade do software. Assim, encontrar formatos que favoreçam o aprendizado do aluno e, consequentemente, na geração de modelos UML cada vez mais eficazes, pode se tornar um dos principais desafios enfrentados por instrutores que incluem UML como parte do conteúdo de modelagem de software.

Alguns estudos (PEDREIRA et al., 2015; JURGELAITIS; DRUNGILAS; ČEPONIENĖ, 2018; YOHANNIS, 2016; BUCCHIARONE et al., 2023b; GARACCIONE et al., 2025a; PORTO et al., 2020) buscaram entender como aplicar a gamificação no ensino de modelagem de software a partir de alguns elementos como pontos, emblemas e níveis. Porém, ainda existem limitações para que professores e pesquisadores possam aplicar, avaliar e mensurar o uso dessa ferramenta na modelagem de softwares de aprendizagem dos alunos e, consequentemente, nos modelos que eles desenvolvem, pois não há "diretrizes" na literatura atual que os oriente. Este estudo pressupõe que os mecanismos de gamificação podem mitigar esses problemas associados aos modelos UML, aumentando o envolvimento dos alunos enquanto aprendem sobre modelagem de software. A literatura atual carece de estudos que explorem a gamificação e a qualidade do modelo no contexto de aprendizagem de modelagem de software.

Este capítulo, portanto, explora a QP2 formulada na introdução (capítulo 1) deste trabalho, através da apresentação do *ModelGame*, um modelo de qualidade para apoiar a aprendizagem de modelagem de software gamificada. Ele serve como uma estrutura de referência para que os instrutores possam obter uma maneira parametrizada de avaliar os modelos UML criados pelos alunos. A qualidade dos modelos UML pode ser melhorada aplicando atividades gamificadas e fornecendo diretrizes cientes dos problemas de qualidade. Uma estrutura de referência ajudaria a (1) estabelecer parâmetros para avaliar modelos UML criados por alunos; (2) fornecer diretrizes para melhorar a qualidade desses artefatos; (3) analisar quais elementos de gamificação poderiam ser incluídos em cada uma das fases de modelagem usando UML; (4) identificar aspectos intrínsecos e extrínsecos dos alunos durante as etapas de modelagem, para aprimorar os modelos; (5) comparar teorias validadas sobre a inclusão da gamificação no ensino de modelagem de software, tendo em conta os tipos de aprendizagem e metodologias utilizadas; e (6) contribuir para a identificação dos objetivos do uso da gamificação nas atividades de modelagem.

Um questionário qualitativo foi respondido por 19 professores que ensinam modelagem de

software em instituições de ensino superior. Os resultados mostram que (1) 94,7% reconhecem que o modelo proposto pode melhorar a qualidade dos modelos UML, indicando que o adotariam em suas práticas de aprendizagem; e (2) 47,4% não usam nenhuma mecânica de gamificação em suas aulas. Esses resultados são animadores, mostrando o potencial de aplicação e aprimoramento do ensino e aprendizagem da modelagem de software.

O restante do capítulo está organizado da seguinte forma. A seção 4.2 apresenta os principais conceitos discutidos ao longo do artigo. A seção 4.3 discute os trabalhos relacionados, destacando as oportunidades de pesquisa. A seção 4.4 apresenta o modelo de qualidade proposto. A seção 4.5 apresenta como o modelo de qualidade foi avaliado. A seção 4.6 aponta algumas ameaças à validade. Finalmente, a Seção 4.7 apresenta algumas considerações finais e trabalhos futuros.

## 4.1 Background

Esta seção apresenta os conceitos essenciais para a compreensão deste trabalho, incluindo Gamificação e Ensino de Engenharia de Software (Subseção 4.1.1), Modelagem de Software e Qualidade de Modelo (Subseção 4.1.2).

### 4.1.1 Gamificação e Ensino de Engenharia de Software

A gamificação visa utilizar elementos do jogo no contexto do não jogo (DETERDING et al., 2011a), trazendo todos os aspectos positivos que fornecem como forma de incentivar e engajar os “jogadores”, ampliando assim suas motivações. (WERBACH; HUNTER, 2012) classifica a gamificação em três dimensões: Dinâmica, Mecânica e Componentes. Dinâmica incluem todos os aspectos do jogo relacionados às respostas emocionais dos “jogadores” (por exemplo, relacionamento, progressão e narrativa). Mecânica oferecem elementos que promovem a ação de um jogo - geralmente elaborados por meio de um desenvolvimento baseado em regras -, de forma que o jogador possa interagir com tais elementos, por exemplo, desafios, feedback e recompensas. Componentes representam os elementos estéticos da gamificação, cujo objetivo é apresentar aspectos visuais com os quais os jogadores podem realizar a interação, por exemplo, pontos, pontuações e emblemas (distintivos).

Saber que o ensino de Engenharia de Software deve envolver os alunos na vivência das práticas profissionais da área para que possam compreender quais práticas e técnicas são úteis nas mais diversas situações (ARDIS et al., 2015). Os desafios de ensinar novos engenheiros de software não se limitam a aprender programação, mas também incluem atenção aos detalhes, considerando a qualidade dos modelos criados, cronograma estabelecido e orçamentos definidos (ADCOCK et al., 2009). Além de compreender os desafios técnicos, esses futuros profissionais devem estar atualizados com as questões não técnicas, como trabalho em equipe, comunicação e gestão.

Para atender a essas novas demandas do contexto atual, o formato com aulas expositivas não é mais considerado suficiente e pode até se tornar desmotivador e ineficaz na aprendizagem dos alunos. Nesse sentido, a gamificação tem sido cada vez mais utilizada no ensino de engenharia de software como forma de promover mudanças comportamentais e psicológicas (HAMARI; KOIVISTO; SARSA, 2014) proporcionando um ambiente que favoreça a comunicação, cooperação, feedback, recompensa, realização e outros elementos recorrentes capazes de melhorar desempenho, eficiência e engajamento em atividades educacionais, podendo aprimorar, por exemplo, o aprendizado de modelagem de software.

#### 4.1.2 Modelagem de Software e Qualidade de Modelo

A modelagem de software engloba o conjunto de princípios, conceitos e práticas que levam ao desenvolvimento de um sistema de alta qualidade ou produtos. Os princípios desta atividade estabelecem uma filosofia que norteia todo o processo de desenvolvimento de software.

Nesse cenário, os modelos UML desempenham um papel crucial nas tarefas de desenvolvimento de software, por exemplo, documentar decisões de projeto, compreender detalhes de desenvolvimento, promover melhor comunicação entre as equipes e gerar maior eficiência no desenvolvimento de software (OMG, 2017). No entanto, esses modelos sofrem problemas de inconsistência e incompletude (FARIAS et al., 2019; OLIVEIRA; GARCIA; WHITTLE, 2008), bem como acabam sendo negligenciados dentro do processo de modelagem, conforme apontado em alguns estudos empíricos na literatura (FRANCE; RUMPE, 2007; LINDLAND; SINDRE; SOLVBERG, 1994). Os diagramas de classe e sequência, por exemplo, apresentam inconsistências quando os objetos do diagrama de sequência não são encontrados no diagrama de classes, consequentemente os desenvolvedores acabam convivendo com inconsistências ao longo do processo de desenvolvimento. Um desafio de pesquisa ainda aberto é como avaliar esses diagramas, tanto na indústria quanto no processo de ensino, em termos de qualidade, como sintática e semântica, por exemplo.

## 4.2 Trabalhos Relacionados

A seleção dos trabalhos relacionados foi realizada seguindo duas etapas: (1) busca em repositórios digitais, como Google Scholar e Scopus (Elsevier) de artigos relacionados à gamificação, modelagem de qualidade e aprendizagem de modelagem; e (2) filtrar os artigos selecionados considerando o alinhamento desses trabalhos com o objetivo do trabalho (Seção 4). Após a seleção dos trabalhos, estes foram analisados (Subseção 4.3.1) e comparados (Seção 4.3.2), buscando identificar oportunidades de pesquisa.

#### 4.2.1 Análise de Obras Relacionadas

(PAULA PORTO et al., 2021). Este trabalho realizou um mapeamento sistemático com o objetivo de caracterizar como a gamificação tem sido adotada em contextos não educacionais de atividades de engenharia de software. Os principais resultados deste estudo mostram que a gamificação proporcionou benefícios para atividades como especificação de requisitos, desenvolvimento, testes, gerenciamento de projetos e processo de suporte. Além disso, ele destacou que o número de publicações e novas iniciativas de pesquisa tem aumentado ao longo dos anos, muitos resultados positivos têm sido alcançados nas atividades de engenharia de software. No entanto, o estudo reforçou que a gamificação ainda pode ser explorada para outras tarefas nesta área, visto que as evidências empíricas são muito limitadas.

(MARÍN, 2022). Realizou a aplicação da gamificação em alguns tópicos de um curso de engenharia de software para engajar os alunos e aumentar sua motivação e defendeu que, com a devida motivação, os alunos podem exercitar melhor os tópicos e obter conhecimentos mais sólidos. Foram cinco jogos relacionados ao gerenciamento de risco, modelagem BPMN, processo Scrum, projeto e inspeção de diagramas de classe e medição de tamanho funcional cósmico para auxiliar no processo de aprendizagem do curso de engenharia de software. Este estudo também apresentou as lições aprendidas sobre a aplicação de gamificação e serious games na engenharia de software, incluindo limitações ou desvantagens.

(JURGELAITIS; DRUNGILAS; ČEPONIENĖ, 2018). Este trabalho conduziu uma pesquisa para investigar como a gamificação poderia ser inserida em um curso de Modelagem de Sistemas de Informação, que cobre uma variedade de tópicos sobre UML. Como resultado, foi apresentada uma implementação do curso de modelagem de sistema gamificado no ambiente Moodle, utilizando plugins para o uso dos elementos gamificados necessários. O estudo apresentou bons resultados e obteve uma aceitação positiva por parte dos alunos participantes.

(RODRIGUES; SOUZA; FIGUEIREDO, 2018). Investigaram o uso de jogos e elementos de jogos no ensino de engenharia de software, por meio de uma pesquisa que contou com a participação de 88 instrutores dessa disciplina. Os resultados mostraram que a maioria dos instrutores está ciente dessas abordagens educacionais, entretanto, os jogos foram adotados por apenas 21 participantes e os elementos do jogo foram adotados apenas por 19. Os jogos são mais frequentemente usados para cobrir “Processo de Software” e “Gerenciamento de Projetos”. Os elementos de jogo mais comumente usados são pontos, questionários e desafios. Os resultados também mostram que os principais motivos para a não adoção dos recursos são a falta de conhecimento, informações sobre jogos relevantes para a engenharia de software de ensino e a falta de tempo para planejar e incluir essas abordagens em sala de aula.

(COSENTINO; GÉRARD; CABOT SAGRERA, 2017a). Apresentam uma abordagem baseada em modelo para a modelagem de aprendizagem de uma forma gamificada. A abordagem inclui uma nova linguagem para modelar o próprio processo de gamificação e um ambiente onde pode ser incorporado às ferramentas de modelagem atuais para permitir que instrutores e

alunos projetem e usem uma estrutura de modelagem completa, incluindo elementos de gamificação. Além disso, a abordagem também teve como proposta fornecer suporte para coleta e análise de dados de gamificação, facilitando as atividades de monitoramento.

([YOHANNIS, 2016](#)). Esta pesquisa apresentou uma exploração do design de jogos como uma abordagem para fortalecer o domínio do aluno em modelagem de software, desenvolvendo suas habilidades de abstração. Reuniu conceitos de desenvolvimento de gamificação, como a lente dos átomos de habilidade intrínseca e princípios de design pedagógico de várias teorias e modelos de aprendizagem. A pesquisa seguiu a Metodologia de Pesquisa da Ciência do Design e explora as melhores práticas da Engenharia Orientada a Modelos. Como resultado, um *framework* de modelagem de design de jogo e estrutura de geração e uma série de jogos produzidos são apresentados.

([PEDREIRA et al., 2015](#)). Desenvolveram um mapeamento sistemático da gamificação em Engenharia de Software com base em 29 estudos. O mapeamento revelou que a implementação de software é a área em que a maioria dos estudos se concentra, seguida pelos requisitos de software, poucas outras em áreas distintas, como planejamento de projetos e teste de software, e ainda em menor grau nas atividades que envolvem modelagem de software. No entanto, o destaque deste trabalho foi destacar que a gamificação na engenharia de software ainda está em um estágio muito inicial e as evidências sobre seu impacto neste campo permanecem inconclusivas.

#### 4.2.2 Análise Comparativa e Oportunidades

Cinco Critérios de Comparação (CC) foram definidos selecionando as variáveis mais relevantes para auxiliar no processo de identificação de semelhanças e diferenças entre o trabalho proposto e os artigos selecionados. Esta comparação é crucial para tornar o processo de identificação de oportunidades de pesquisa usando critérios objetivos em vez de critérios subjetivos. Os critérios são descritos abaixo:

- **Contexto (CC01):** Trabalhos que exploram o uso da gamificação no ensino/aprendizagem de modelagem de software.
- **Perfil do participante (CC02):** Estudos que coletaram dados dos participantes para triagem e caracterização do perfil.
- **Aplicabilidade da gamificação em UML (CC03):** Estudos que avaliaram como a gamificação pode contribuir para os modelos UML.
- **Criação do modelo (CC04):** Estudos que desenvolveram um modelo para melhorar os fatores que implicam na não adoção da UML.
- **Participação do instrutor (CC05):** Estudos que coletaram dados qualitativos por meio da participação de instrutores de modelagem de software.

**Tabela 4:** Análise Comparativa dos Trabalhos Relacionados

Trabalho Relacionado	Critério de comparação				
	CC1	CC2	CC3	CC4	CC5
Trabalho proposto	●	●	●	●	●
(PAULA PORTO et al., 2021)	○	○	●	○	○
(JURGELAITIS; DRUNGILAS; ČEPONIENĖ, 2018)	●	○	●	●	○
(YOHANNIS, 2016)	●	○	◐	◐	○
(DUBOIS; TAMBURRELLI, 2013)	●	○	○	○	○
(RODRIGUES; SOUZA; FIGUEIREDO, 2018)	●	●	◐	○	●
(COSENTINO; GÉRARD; CABOT SAGRERA, 2017a)	●	○	●	●	○
(MARÍN, 2022)	●	○	◐	◐	○
(PEDREIRA et al., 2015)	○	○	◐	○	○

● Atende completamente   ◐ Atende parcialmente   ○ Não atende

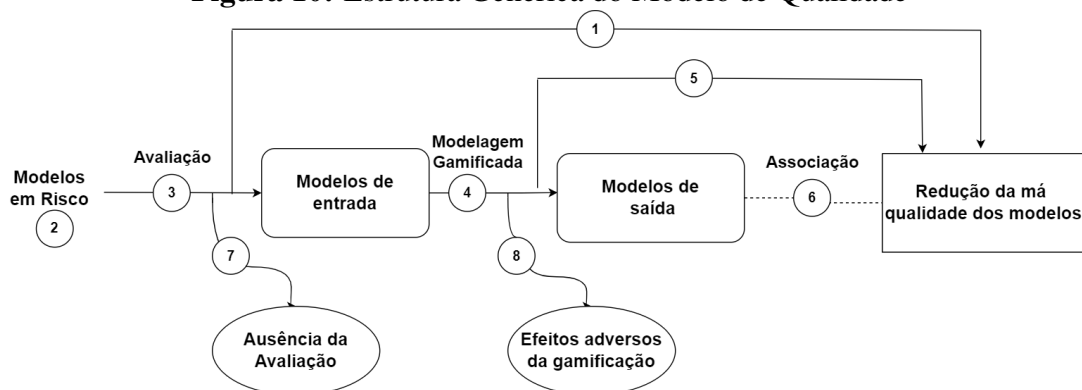
Fonte: Elaborado pelo autor.

A Tabela 4 mostra a comparação dos trabalhos selecionados, confrontando este trabalho. Observam-se algumas lacunas e oportunidades de pesquisa: (1) apenas o trabalho proposto foi o único a atender plenamente todos os critérios de comparação; (2) embora a maioria deles visasse a aplicação de gamificação no ensino de modelagem de software, eles não eram direcionados ao uso de UML; (3) nenhum estudo desenvolveu um modelo para avaliar a aprendizagem e o aprimoramento de modelos UML desenvolvidos por alunos; e (4) a maioria deles não contou com a participação de instrutores para identificar as dificuldades e oportunidades na aplicação da gamificação no ensino de modelagem de software. Assim, a próxima seção apresenta um modelo de qualidade para explorar essas oportunidades identificadas.

### 4.3 Modelo de Qualidade

Esta seção apresenta o modelo de qualidade proposto para apoiar a aprendizagem da modelagem de software de forma gamificada. Ele serve como um quadro de referência para que os instrutores possam avaliar os modelos UML criados pelos alunos por meio de atividades gamificadas. A seção 4.4 apresenta uma proposta de estrutura analítica genérica. A Subseção 4.4.1 detalha a sintaxe abstrata do modelo de qualidade proposto. A Subseção 4.4.2 explica as noções de qualidade relacionadas ao aprendizado de modelagem de software gamificado.

A Figura 10 apresenta a estrutura analítica genérica para melhorar a qualidade dos modelos e serve de base para a criação de um esquema de avaliação. As setas ("links"), rotuladas como Avaliação e Modelagem Gamificada, representam as questões que a evidência deve responder; as linhas pontilhadas representam associações; retângulos representam os Modelos (cantos arredondados) ou os estados de qualidade (cantos quadrados) pelos quais essas ligações são medidas. As elipses representam os efeitos adversos que podem ser gerados a partir da avaliação e uso da gamificação.

**Figura 10:** Estrutura Genérica do Modelo de Qualidade

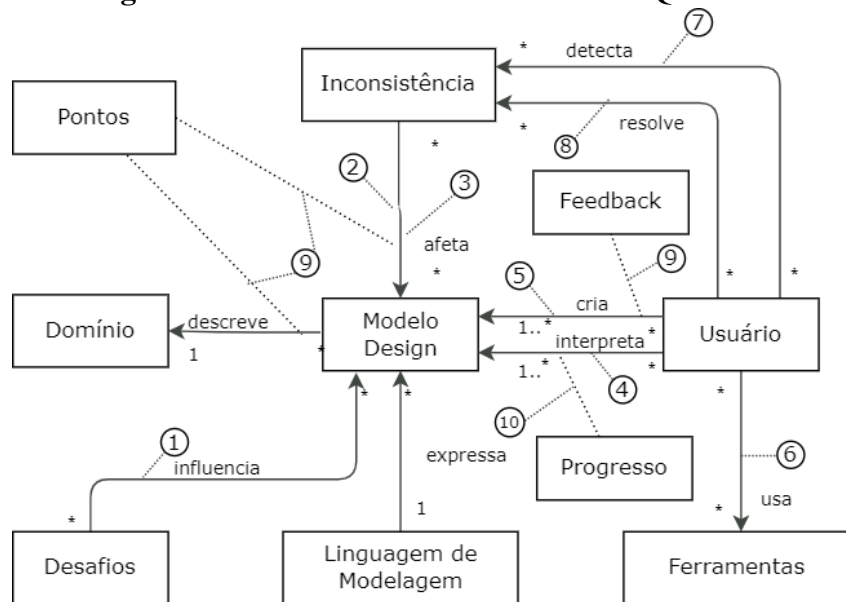
Fonte: Próprio autor

#### 4.4 Modelo de Qualidade Proposto

Os números apresentados na Figura 10 referem-se às questões-chave utilizadas para operacionalizar os conceitos e as relações definidos na sintaxe abstrata do Modelo de Qualidade, apresentada na Seção 4.2. Essas questões foram organizadas de forma a refletir diferentes dimensões do modelo e orientar sua aplicação prática. Um primeiro conjunto de questões (1 a 3) está relacionado à existência de ferramentas, critérios e noções de qualidade que apoiem a avaliação de modelos UML, bem como à identificação de características que possam comprometer sua qualidade. Em seguida, as questões (4 a 6) concentram-se na investigação dos efeitos da gamificação, analisando se sua aplicação pode contribuir para a identificação de fatores associados à melhoria da qualidade dos modelos e se há diferenças entre modelos produzidos com e sem o uso de gamificação. Por fim, as questões (7 e 8) abordam possíveis efeitos adversos associados à ausência de avaliação sistemática e à aplicação da gamificação, considerando tanto a aceitabilidade do processo quanto os danos potenciais e sua frequência. Dessa forma, o conjunto de questões estabelece um vínculo direto entre os elementos da sintaxe abstrata e a aplicação do Modelo de Qualidade, tornando explícita a lógica de avaliação proposta.

Fato é que não basta incluir esta caixa de ferramentas no processo de aprendizagem da UML, é necessário fornecer ao instrutor um modelo (guia) que possa servir de referência para avaliar a qualidade dos diagramas elaborados por meio de atividades gamificadas. Por exemplo, o instrutor poderia criar modelos predefinindo inconsistências, fazendo uso dessas questões levantadas para avaliar os modelos criados pelos alunos. O conjunto de questões serve de ponto de partida para esta avaliação. Sabendo que a adaptação da abordagem de gamificação requer um esforço significativo (OUHBI; POMBO, 2020), neste estudo apresentamos o *ModelGame* como uma forma de identificar fatores que contribuem para a qualidade desses artefatos e, consequentemente, para a aprendizagem dos alunos.



**Figura 11:** Sintaxe Abstrata do Modelo de Qualidade

Fonte: Próprio autor

#### 4.4.1 Sintaxe Abstrata

Seguindo o padrão de especificação do metamodelo UML, a Figura 12 apresenta a sintaxe abstrata do Modelo de Qualidade proposto para aprendizagem de modelagem de software gamificada (*ModelGame*). Identifica os principais conceitos e relacionamentos. Os números representam as noções de qualidade discutidas na Subseção 4.5.2. A seguir estão detalhados cada um desses conceitos e relacionamentos.

**Domínio.** O primeiro conceito apresentado neste estudo é o domínio, que corresponde a um contexto específico da aplicação a ser desenvolvido para resolver o problema. Nesse processo, o template de design representa a solução dada ao domínio.

Associação

- contextualiza: Desafios [\*]

Cada contextualização refere-se ao domínio que servirá de base para os desafios lançados.

**Desafios.** Este conceito representa a fase em que o problema é contextualizado (baseado no domínio), bem como quais serão as missões, fases, cenários e outros elementos apresentados aos jogadores, neste caso os alunos, que devem utilizar os princípios de engenharia de software para realizar a modelagem e atingir o objetivo final.

Associação

- influências: Modelo de Design [\*]

Cada influência representa que o desafio proposto interfere em aspectos do modelo de design, fazendo com que o usuário busque fazer uma melhoria contínua.



Linguagem de modelagem. A modelagem de software é uma etapa importante para que o desenvolvimento aconteça de forma que atenda aos requisitos estabelecidos pelo solicitante, para isso existe a linguagem de modelagem, que oferece uma forma padronizada de documentar e projetar software. Por meio da utilização de linguagens de modelagem, é possível atingir um alto nível de conhecimento sobre o software em questão, melhorando a comunicação entre todos os envolvidos no processo, evitando erros de implementação. Ele aponta que os engenheiros de software usam essas linguagens para comunicar as decisões de design e verificar a viabilidade de implementação do design pretendido. A UML consolidou-se como a Linguagem de Modelagem no paradigma da orientação a objetos.

#### Associação

- expressa: Modelo de Design [\*]

Executa a representação dos modelos de design pretendidos, nos quais a Linguagem de Modelagem deve ser aplicável ao tipo de domínio.

Do utilizador. Este conceito corresponde ao indivíduo que realiza a interpretação dos modelos de design desenvolvidos, cujo objetivo é ser capaz de compreender o domínio em questão. No contexto gamificado, o usuário tem o papel de jogador e é ele quem realiza todo o processo, podendo realizar a interpretação dos modelos existentes ou mesmo criando novos. O usuário também pode identificar e resolver inconsistências que surgem de composições entre modelos.

#### Associação

- cria: Modelo de Design [1 .. \*]

Representa o processo no qual o usuário cria um modelo de design, que pode ser um ou mais.

- interpreta: Modelo de Design [1 .. \*]

Nesta associação, o usuário realiza a interpretação do template de design. Ao interpretar o modelo, caminhos para a resolução de inconsistências podem ser identificados.

- detecta: Inconsistência [\*]

Representa a descoberta do usuário de inconsistências do modelo de design, por exemplo, aquelas que são geradas a partir da identificação de conflitos, seja uma classe abstrata ou não.

- resolve: Inconsistência [\*]

Cada resolução equivale à representação de resolução das inconsistências pelo usuário que ocorre após ele analisar e determinar a melhor alternativa para realizar esta ação.

- usa: Ferramentas de modelagem [\*]

Determina que o usuário pode usar ferramentas de modelagem para gerar / atualizar modelos de design.

Associação

- Sem um relacionamento direcionado.

Ferramenta de modelagem. Este conceito representa as aplicações que são utilizadas para realizar a construção de modelos de design. São diversas ferramentas disponíveis, online e desktop, cabendo ao usuário escolher aquela que melhor atenderá suas necessidades e se adaptará ao contexto em questão, ou seja, eles funcionam em qualquer domínio que está sendo considerado.

Modelo de design. O modelo de design se refere a uma notação visual (diagrama) para representar aspectos estáticos e dinâmicos. Esses modelos são construídos de acordo com um objetivo ou tarefa específica e tendem a facilitar a interpretação lógica do software em vários aspectos. Os diagramas mais populares são Casos de Uso e Classes, sendo o primeiro estático e representando um conjunto de ações geradas a partir de requisitos funcionais (casos de uso) e apresentando as interações geradas com usuários externos (atores). O segundo é um diagrama estático e faz a representação da estrutura lógica do software envolvendo as classes, seus atributos, métodos e relacionamentos entre elas (OMG, 2017).

Associação

- descreve: Domínio [1]

Cada descrição faz a representação de um domínio específico e significa que todo modelo de design deve descrevê-lo.

Inconsistência. Corresponde aos defeitos encontrados nos modelos desenvolvidos pelos usuários. Podem ocorrer por causa da não identificação e correção de possíveis conflitos e até mesmo por uma interpretação errônea.

Associação

- afeta: Modelo de Design [\*]

Essa associação indica que a cada ocorrência do afeto, é apresentado um problema que prejudica a qualidade do modelo de design.

Pontos. Este conceito representa uma das mecânicas de jogo mais utilizadas na engenharia de software e funciona como uma recompensa quantitativa para cada ação desenvolvida, na qual é possível regular o número de pontos recompensados do jogador, aqui definido como usuário, com base na importância de cada ação. Por meio desse conceito, é possível estimular a competição, a colaboração e a criatividade entre os usuários, estimulando o aprendizado. Os pontos aparecem como uma derivação dos afetos de associação, pois quando cada erro de inconsistência for identificado ou não, o usuário receberá uma pontuação e a associação descreve, porque os pontos também serão aplicados ao fazer conexões entre o modelo e o domínio.

**Progresso.** O conceito de progresso surge como um fator que torna o usuário capaz de perceber sua evolução no processo, no caso, a modelagem de software. O progresso surge como uma derivação da interpretação da associação, fazendo com que o usuário saiba quando realizou uma interpretação correta do modelo de design proposto ou o que ainda precisa ser melhorado. **Comentários.** O feedback tem a função de fazer com que o usuário perceba que o objetivo proposto pode ser alcançado e acompanhe sua evolução, incluindo analisar como mudar ou criar novas estratégias para atingir o objetivo. Este conceito surge como uma derivação entre as associações que cria, fazendo com que o usuário receba um retorno ao processo de criação do modelo.

#### 4.4.2 Noções de Qualidade

Conforme discutido na Seção 4.2, a gamificação pode trazer elementos importantes para o aprendizado da modelagem de software e, portanto, o objetivo desta seção é produzir as noções de qualidade do modelo deste estudo. O *ModelGame* é composto por dez contagens, quatro das quais propostas neste estudo - escopo, uso, motivacional e engajamento - extraídas dos principais benefícios que os elementos de gamificação apresentados na Figura 13 podem trazer aos modelos. Os demais são adaptações de trabalhos anteriores (FERNÁNDEZ-SAEZ et al., 2012; LANGE; CHAUDRON; MUSKENS, 2006; LINDLAND; SINDRE; SOLVBERG, 1994), são eles, sintáticos, semânticos, sociais, esforço, detecção e resolução.

- **Qualidade do escopo (1).** Busca determinar o quanto o desafio proposto está contextualizado com o modelo de design, bem como a definição do domínio, problema, competências, conceitos, comportamentos e atitudes que serão desenvolvidos ao longo do processo.
- **Qualidade sintática (2).** Essa noção faz a representação do processo de correção dos modelos de design que são produzidos pela linguagem de modelagem, pois se não for usada corretamente, irão surgir inconsistências. É importante inserir essa noção de qualidade no estudo, pois durante o processo de desenvolvimento dos modelos, os usuários podem se deparar com a composição de dois diagramas de classes, por exemplo.
- **Qualidade semântica (3).** É necessário verificar se o modelo de projeto e o domínio do problema correspondem, para que esta noção realize este tipo de análise. Podem ocorrer problemas de comunicação entre os usuários se os elementos semânticos do modelo forem afetados.
- **Qualidade social (4).** Os modelos de design são usados para a comunicação entre os membros de uma equipe para informar todas as decisões estabelecidas sobre o desenvolvimento de software (FARIAS et al., 2014). Se ocorrerem interpretações divergentes, essa comunicação será muito prejudicada.

- **Qualidade do esforço (5).** Essa noção se refere aos desafios de produção do modelo que será gerado, incluindo fatores como tempo e custo.
- **Qualidade de uso (6).** Para produzir modelos de design, os usuários podem usar ferramentas incomuns, como papel, quadro branco e muito mais. No entanto, na maioria das vezes eles optam por usar ferramentas formais (CASES) e podem ser online ou *desktop*. Essa noção corresponde ao nível de facilidade e aplicabilidade dos modelos elaborados ao fazer uso dessas ferramentas, também é importante contribuir para a comunicação entre os usuários por meio de funcionalidades relacionadas à colaboração.
- **Qualidade de detecção (7).** Essa noção está relacionada ao processo de localização de inconsistências nos modelos. Quando inconsistências ocorrem, é importante que elas possam ser identificadas e rastreadas de forma eficiente. Caso a detecção seja complexa ou demorada, o processo de correção dos modelos pode ser comprometido.
- **Qualidade de resolução (8).** Corresponde ao nível de qualidade relacionado ao esforço que os usuários realizam na busca por alternativas para solucionar o problema identificado.
- **Qualidade motivacional (9).** Essa noção se refere aos fatores motivacionais envolvidos durante o aprendizado e desenvolvimento de modelos de design, que podem ser intrínsecos ou extrínsecos. Elementos de gamificação como pontos, feedback e progresso trazem ao usuário um grau de satisfação em continuar sua descoberta e transformações ao longo do processo.
- **Qualidade do engajamento (10).** O usuário ao acompanhar seu progresso pode se sentir comprometido com o objetivo em questão, e essa noção representa a medida do nível de comprometimento deles durante o desenvolvimento de modelos de design.

## 4.5 Avaliação

Esta seção descreve a metodologia seguida para avaliar o modelo de qualidade proposto. Esta metodologia segue diretrizes empíricas bem estabelecidas (WOHLIN et al., 2012). A Subseção 4.6.1 detalha o objetivo e as questões de pesquisa (QPs). A Subseção 4.6.2 apresenta o questionário formulado para avaliar o modelo de qualidade proposto. A Subseção 4.6.3 explica o contexto e a seleção dos participantes. A Subseção 4.6.4 descreve o processo experimental. A Subseção 4.6.5 realiza a análise do resultado.

### 4.5.1 Objetivo e Questões de Pesquisa

O objetivo (O) deste estudo é duplo: (O1) Apresentar o *ModelGame* como uma ferramenta para o ensino de Modelagem de Software; e (O2) Analisar a aplicabilidade do modelo de qua-

lidade quanto ao aprimoramento de modelos UML.

Para analisar as diferentes facetas dos objetivos, duas Questões de Pesquisa (QPs) foram formuladas:

- QP2.1: Como os instrutores avaliam o uso de gamificação na modelagem de software?
- QP2.2: Qual é a aceitação do *ModelGame* por instrutores de modelagem de software?

#### 4.5.2 Questionário

Os dados foram coletados por meio de um questionário online criado por meio do Formulários Google<sup>1</sup> seguindo diretrizes bem estabelecidas descritas em (WOHLIN et al., 2012). Essa estratégia foi escolhida porque o questionário poderia ser aplicado de forma rápida e fácil para coletar dados de indivíduos em locais geograficamente diversos. As perguntas do questionário preocupavam-se em examinar as lacunas de pesquisa de estudos anteriores e apreender as estruturas do questionário desenvolvido anteriormente.

**Parte 1: Perfil do participante.** A primeira parte do questionário consistiu na coleta de dados relacionados às características e opiniões dos participantes. A criação do perfil do participante através desses dados é importante para fazer a seleção dos possíveis usuários do *ModelGame*. Sem esse perfil, participantes com perfil inadequado podem gerar avaliações inconsistentes. Os participantes foram solicitados a fornecer informações mais gerais, como idade, nível de escolaridade e formação acadêmica. Informação sobre o tempo de experiência na docência também foi considerado, incluindo o ensino de modelagem de software e o nível de conhecimento sobre modelos UML.

**Parte 2: Questionário TAM.** A segunda parte abordou questões sobre a usabilidade e aceitação da técnica, com o objetivo de explorar Q3. Para tanto, esta parte do questionário é baseada no modelo de aceitação de tecnologia (TAM) (MARANGUNIĆ; GRANIĆ, 2015). Essa parte continha nove questões, as quais foram respondidas por meio da Escala Likert, incluindo Concordo Totalmente, Concordo Parcialmente, Neutro, Discordo Parcialmente e Discordo Totalmente. As questões formuladas (Q) abordaram diversos tópicos, incluindo facilidade de uso percebida (Q1-3), utilidade percebida (Q4-7), atitude em relação ao uso (Q8) e intenção comportamental de usar (Q9).

#### 4.5.3 Seleção de Participantes

Os participantes foram selecionados com base nos seguintes critérios: instrutores e / ou profissionais que atuam no ensino de modelagem de software em instituições de ensino superior no Brasil. Utilizando esse critério, buscou-se selecionar participantes com formação acadêmica

<sup>1</sup>Questionário: <https://forms.gle/qjaFDpErEtGdLuWw6>

e experiência prática na docência. Este conjunto finito de todos os participantes possíveis representa a população-alvo (KITCHENHAM; PFLEEGER, 2008). Esta população representa aquelas pessoas que podem responder às questões formuladas e às quais se aplicam os resultados do questionário (KITCHENHAM; PFLEEGER, 2008). Ao todo, 19 pessoas (n) responderam ao questionário. Os participantes foram convidados por e-mail a participarem do estudo e cada um deles recebeu previamente a explicação / treinamento sobre o modelo proposto pela pesquisadora e não houve dúvidas, eles poderiam partir para a próxima etapa, que consistia em preencher o questionário TAM. Discutimos o processo experimental na próxima seção.

#### 4.5.4 Processo Experimental

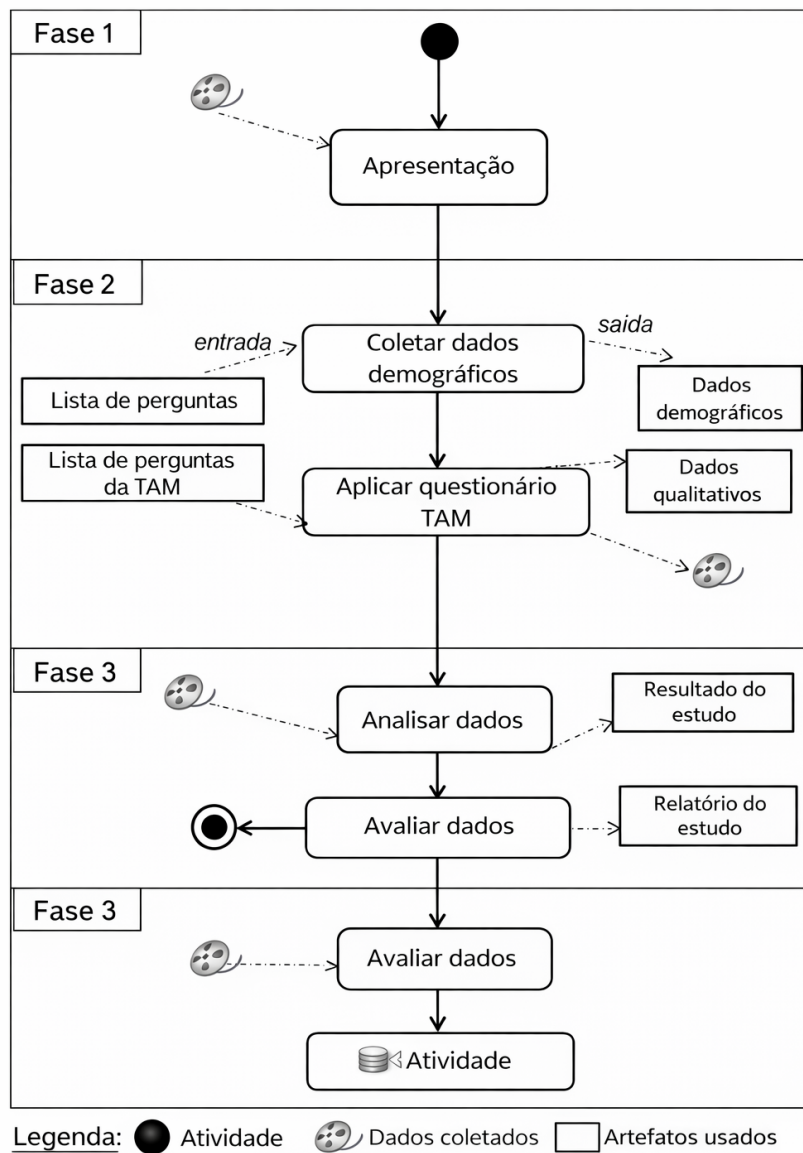
A Figura 12 apresenta o processo experimental utilizado neste estudo, que é composto por três fases discutidas a seguir:

**Fase 1: Apresentação.** Tem uma atividade, apresentação, em que o pesquisador explica aos participantes, por meio de um vídeo, detalhes sobre o modelo de qualidade. Esse processo ocorreu de forma individualizada e padronizada, onde também foi disponibilizado espaço para os participantes esclarecerem eventuais dúvidas sobre a proposta de estudo e modelo, com duração média de 20 minutos.

**Fase 2: Aplicação do questionário TAM.** Possui duas atividades, sendo a primeira a coleta de dados demográficos. Os participantes responderam a uma lista de perguntas (input) para que pudéssemos coletar suas características e opiniões sobre o *ModelGame*. Os dados demográficos coletados (output) passaram a ser o resultado dessa atividade. A segunda atividade foi a de aplicação do questionário TAM (entrada). Os participantes receberam uma lista de perguntas sobre a percepção de facilidade de uso, utilidade percebida, atitudes e intenção de comportamento, em relação ao *ModelGame*. Foram gerados dados qualitativos (output), referentes à usabilidade e aceitação do Modelo na perspectiva dos profissionais que ensinam modelagem de software. Este questionário seguiu as orientações do TAM (MARANGUNIC; GRANIC, 2015).

**Fase 3: Análise e relatório de resultados.** Possui duas atividades. O primeiro, Analisar dados buscou realizar uma análise criteriosa dos dados coletados por meio do questionário e da percepção do pesquisador quanto às dúvidas dos participantes durante a fase de apresentação.

Para isso, os dados coletados foram analisados separadamente, bem como confrontados, visando realizar uma triangulação dos mesmos. Posteriormente, os dados de avaliação, como forma de compreender com mais profundidade o contexto, as percepções dos participantes em relação ao modelo proposto, bem como sua aplicabilidade.

**Figura 12: Processo Experimental**

Fonte: Próprio autor

#### 4.5.5 Análise de Resultado

##### 4.5.5.1 Dados do perfil dos participantes.

A Tabela 5 descreve o perfil a partir dos dados, relatando as características e opiniões dos participantes. Esses dados foram coletados de 18 de maio a 5 de junho de 2021. No total, tivemos 19 participantes. Os participantes têm entre 20 e 49 anos, a maioria deles possui graduação em Ciência da Computação (52,6%), Sistemas de Informação (26,3%) ou Análise de Sistemas (21,1%) e são especialistas (36,8%), mestres (36,8%) e doutores (15,8%). Sobre o tempo de trabalho na docência, a maioria (42,1%) leciona há mais de 8 anos e ministra disciplinas relacionadas à modelagem de software, incluindo engenharia de software, análise de sistemas e projetos de software. Um total de 47,4% possui um nível de conhecimento completo sobre UML e quase metade deles (47,4%) ainda não utilizou a gamificação no ensino de modelagem de software. Portanto, consideramos que, embora pequena, a amostra é adequada para realizar uma avaliação inicial da abordagem proposta.

##### 4.5.5.2 QP2.1: Como os instrutores avaliam o uso de gamificação na modelagem de software?

A Tabela 5 apresenta os dados coletados relacionados à QP formulada. Primeiro, começamos a análise verificando como os instrutores visualizam a gamificação no ensino de modelagem de software. Embora a maioria (47,4%) ainda não tenha utilizado elementos de gamificação (pontuação, desafio, emblema, entre outros) em suas aulas, a maioria (52,6%) concorda totalmente e (42,1%) concorda parcialmente que o uso destes pode contribuir para a qualidade dos modelos desenvolvidos pelos alunos.

Consideramos alto o percentual de instrutores que ainda não utilizaram a gamificação em suas aulas e isso pode estar atrelado a fatores como falta de conhecimento, informação sobre a ferramenta e até mesmo tempo para planejar e incluir essas abordagens (RODRIGUES; SOUZA; FIGUEIREDO, 2018). Apesar de se basearem no contexto de ensino de modelagem de software, estudos anteriores (COSENTINO; GÉRARD; CABOT SAGRERA, 2017a; PORTO et al., 2020; JURGELAITIS; DRUNGILAS; ČEPONIENĖ, 2018; MARÍN, 2022; YOHANNIS, 2016) não contaram com a participação de instrutores e entendemos que esta participação é fundamental para entender as percepções desses profissionais uma vez que eles estarão na vanguarda do uso de gamificação.

O *ModelGame* proposto neste estudo poderia auxiliá-los a inserir a gamificação em suas aulas, de acordo com o projeto de aprendizagem de modelagem de software (YOHANNIS, 2016), partindo do pressuposto de que para isso é necessário desenvolver uma melhor compreensão das tarefas, atividades, habilidades e operações que os diferentes elementos de gamificação podem oferecer e como eles podem corresponder aos resultados de aprendizagem desejados, desenvol-



**Tabela 5:** Dados de perfil dos participantes.

<b>Característica e Opinião (n=19)</b>	<b>Resposta</b>	<b>#</b>	<b>%</b>
Idade	< 20 anos	0	0.0%
	20-29 anos	4	21.1%
	30-39 anos	8	42.1%
	40-49 anos	5	26.3%
	> 49 anos	2	10.5%
Educação	Graduação*	0	0.0%
	Especialização*	7	36.8%
	Mestrado*	7	36.8%
	Doutorado*	3	15.8%
	Outros	2	10.6%
Curso de graduação	Sistemas de informação	5	26.3%
	Ciência da computação	10	52.6%
	Engenharia da computação	0	0.0%
	Análise de sistemas	4	21.1%
	Outros	0	0.0%
Tempo de experiência no ensino	< 2 anos	4	21.1%
	2-4 anos	2	10.5%
	5-6 anos	3	15.8%
	7-8 anos	2	10.5%
	> 8 anos	8	42.1%
Experiência no ensino de modelagem de software	< 2 anos	3	15.8%
	2-4 anos	5	26.3%
	5-6 anos	3	15.8%
	7-8 anos	2	10.5%
	> 8 anos	6	31.6%
Nível de conhecimento sobre modelos UML	Iniciante	2	10.5%
	Júnior	5	26.3%
	Full	9	47.4%
	Sênior	3	15.8%
Já utilizou gamificação no ensino	Sim	9	47.4%
	Não	9	47.4%
	Talvez	1	5.3%
A gamificação pode contribuir para a qualidade dos modelos UML gerados pelos estudantes	Concordo totalmente	10	52.6%
	Concordo parcialmente	8	42.1%
	Neutro	1	5.3%
	Discordo parcialmente	0	0.0%
	Discordo totalmente	0	0.0%

Fonte: Próprio autor

vido uma apresentação mais concreta e motivadora que pode envolver os alunos e facilitar a aprendizagem profunda com UML.

#### 4.5.5.3 QP2.2: Qual é a aceitação do *ModelGame* por instrutores de modelagem de software?

Usando o questionário TAM, foi realizada a avaliação da facilidade de uso, utilidade percebida, atitude e intenção comportamental de usar o Modelo de Qualidade. A Tabela 6 mostra os dados obtidos. Os dados obtidos mostram que ninguém discordou que o *ModelGame* é fácil de usar, aprender e dominar. Pelo contrário, quase 90% dos participantes consideram o modelo fácil de usar (42,1% concorda totalmente e 47,4% concorda parcialmente e 10,5% neutro), aprender (52,6% concorda totalmente e 47,4% concorda parcialmente) e domina (31,6% concorda totalmente, 63,2% concordam parcialmente e 5,3% discordam parcialmente).

Os resultados também são favoráveis considerando a percepção de utilidade. A maioria dos participantes percebeu que o *ModelGame* tornaria mais fácil entender quais elementos de gamificação podem ser usados em cada uma das fases da modelagem usando UML (63,3% concordam totalmente, 26,3% concordam parcialmente e 10,5% neutros), aumentam a produtividade (47,4% totalmente concordam, 42,1% concordam parcialmente e 10,5% neutros), e o uso do modelo de qualidade forneceria um entendimento de como mitigar a incompletude dos diagramas UML (26,3% concordam totalmente 42,1% concordam parcialmente, 26,3% neutros e 5,3% discordam parcialmente). Ainda no aspecto útil, procuramos saber se o modelo de qualidade ajudaria a comparar teorias validadas sobre a inclusão da gamificação no ensino de modelagem de software (68,4% concordam totalmente, 21,1% concordam parcialmente e 10,5% neutros).

Considerando a atitude em relação ao uso, os participantes acreditam que usar o *ModelGame* é uma boa ideia (68,4% concordam totalmente, 26,3% concordam parcialmente e 5,3% neutros), da mesma forma que estão confiantes e usaria o Modelo nas aulas de modelagem de software (52,6% concordam totalmente, 36,8% concordam parcialmente e 10,5% neutros). Essas descobertas mostram o potencial de aceitação por pessoas com perfis semelhantes aos dos participantes. Os resultados são animadores e mostram o potencial de utilização da abordagem proposta no cenário educacional.

## 4.6 Ameaças à Validade

Esta seção discute as possíveis ameaças à validade do estudo.

Validade interna. O principal ponto que afeta a validade interna do estudo diz respeito ao tempo total utilizado para a fase exploratória. Para mitigar essa ameaça, realizamos a gravação em vídeo de um piloto explicando os detalhes operacionais e objetivos do *ModelGame*. Em relação aos métodos utilizados, as ameaças relacionadas à validade interna referem-se a como

**Tabela 6:** Dados coletados relacionados ao questionário TAM.

<b>Constructo e Afirmativa (n=19)</b>	<b>Resposta</b>	<b>#</b>	<b>%</b>
<b>Facilidade de uso percebida</b>			
Achei o modelo de qualidade fácil de usar	Concordo totalmente	8	42.1%
	Concordo parcialmente	9	47.4%
	Neutro	2	10.5%
Achei o modelo de qualidade fácil de aprender	Concordo totalmente	10	52.6%
	Concordo parcialmente	9	47.4%
Achei o modelo de qualidade fácil de dominar	Concordo totalmente	6	31.6%
	Concordo parcialmente	12	63.2%
	Discordo parcialmente	1	5.2%
<b>Utilidade percebida</b>			
O modelo facilitaria a compreensão dos elementos de gamificação	Concordo totalmente	12	63.2%
	Concordo parcialmente	5	26.3%
	Neutro	2	10.5%
Usar o modelo ajudaria a aumentar a produtividade	Concordo totalmente	9	47.4%
	Concordo parcialmente	8	42.1%
	Neutro	2	10.5%
O modelo ajudaria a mitigar a incompletude dos diagramas UML	Concordo totalmente	5	26.3%
	Concordo parcialmente	8	42.1%
	Neutro	5	26.3%
	Discordo parcialmente	1	5.3%
O modelo ajudaria a comparar teorias sobre gamificação	Concordo totalmente	13	68.4%
	Concordo parcialmente	4	21.1%
	Neutro	2	10.5%
<b>Atitude em relação ao uso</b>			
Usar o modelo é uma boa ideia	Concordo totalmente	13	68.4%
	Concordo parcialmente	5	26.3%
	Neutro	1	5.3%
<b>Intenção de uso</b>			
Eu usaria o modelo em aulas de modelagem de software	Concordo totalmente	10	52.6%
	Concordo parcialmente	7	36.8%
	Neutro	2	10.5%

Fonte: Próprio autor

extraímos as percepções das discussões e se elas representam as percepções dos professores sobre o uso do Modelo. Tentamos reduzir essa ameaça aplicando o questionário TAM.

Validade externa. Identificamos ameaças relacionadas à validade externa, como o número de participantes que nunca aplicaram o uso de gamificação. Este estudo limitou-se à 19 participantes (professores) de diversas instituições de ensino, dos quais 9 (47,4%) nunca utilizaram nenhum elemento de gamificação em suas aulas, esse fator pode interferir nos dados, uma vez que o modelo pretende avaliar a qualidade da UML diagramas de atividades gamificadas.

Validade da conclusão. As ameaças relacionadas à validade da conclusão estão relacionadas ao tratamento e ao resultado. Tentamos fazer a redução combinando dados quantitativos e qualitativos por meio de diferentes recursos. Esses dados foram obtidos por meio de áudio e questionários. Analisamos esses dados para responder às questões de pesquisa.

#### **4.7 Considerações Finais do Capítulo**

Este estudo propôs um modelo inicial de qualidade (ModelGame) que serve como um quadro de referência para instrutores realizarem avaliações qualitativas de modelos UML desenvolvidos a partir de atividades gamificadas. A proposta do modelo constitui uma alternativa para solucionar a segunda questão de pesquisa (QP2) desta tese, ao suprir a ausência, na literatura, de um esquema sistemático para avaliar atributos de qualidade de modelos UML e suas métricas associadas. A aplicação de um estudo empírico com 19 participantes permitiu compreender suas percepções sobre a gamificação e a aceitação do modelo proposto, revelando que, embora a maior parte ainda não utilize técnicas gamificadas em suas práticas, reconhecem que tais elementos podem contribuir significativamente para a qualidade dos modelos desenvolvidos por alunos. Esses achados reforçam o potencial da gamificação como estratégia de ensino, favorecendo a aprendizagem em modelagem de software e a produção de modelos UML mais estruturados e consistentes.

Considerando os resultados obtidos, torna-se necessário avançar na investigação dos efeitos da gamificação aplicada diretamente ao processo de modelagem. Assim, este capítulo estabelece as bases conceituais e metodológicas para que possamos aprofundar a análise sobre o impacto da gamificação na qualidade de modelos UML. Dessa forma, o próximo capítulo apresenta um estudo empírico mais abrangente, no qual diferentes formas de aplicação de elementos gamificados são examinadas para responder à terceira questão de pesquisa (QP3). Esse movimento investigativo dá continuidade natural à proposta introduzida aqui, permitindo avaliar, de maneira quantitativa e qualitativa, como o uso da gamificação influencia atributos centrais dos modelos e o desempenho dos participantes em atividades de modelagem.

Por fim, esperasse realizar futuramente uma série de estudos experimentais para analisar cada etapa de aplicação do *ModelGame*, e que este trabalho representa um primeiro passo para melhor apoiar a aplicação de estudos empíricos em modelos de avaliação do uso da gamificação na modelagem de software. Esperamos também que as questões descritas aqui incentivem

outros pesquisadores a estender o estudo a diferentes linguagens de modelagem e metodologias de ensino.

## 5 ESTUDO EMPÍRICO SOBRE O IMPACTO DA GAMIFICAÇÃO NA MODELAGEM DE SOFTWARE COM UML

Linguagens de modelagem, como Unified Modeling Language (UML) (OMG, 2017), oferecem vários modelos, incluindo diagramas de classes e sequência, para representar a estrutura e comportamento de sistemas de software. Esses modelos complementares refletem decisões de design que os desenvolvedores devem implementar mais tarde. No entanto, na prática, esses modelos muitas vezes enfrentam *problemas de inconsistência* (REDER; EGYED, 2013; TORRES; BRAND; SEREBRENIK, 2021; MARCHEZAN et al., 2023b). Essas inconsistências surgem principalmente pela incompatibilidade entre as partes sobrepostas de modelos complementares e a falta de semântica formal para evitar essas contradições. Como resultado, os desenvolvedores precisam investir algum esforço para detectar e resolver essas inconsistências adequadamente; caso contrário, interpretações erradas dos modelos de design podem comprometer a implementação final.

Detectar e resolver inconsistências em modelos UML são tarefas essenciais, mas desafiadoras (MARCHEZAN et al., 2024a, 2023b) que podem afetar a interpretação de decisões de software. A complexidade dos sistemas de software modernos leva a dependências e interações complexas entre diferentes partes dos modelos UML, tornando as inconsistências difíceis de detectar (MARCHEZAN et al., 2023a). Os métodos tradicionais de verificação de consistência muitas vezes não cobrem todas as falhas potenciais, especialmente em grandes sistemas de software (MARCHEZAN et al., 2024b). Isso motiva a necessidade de novas abordagens e ferramentas para apoiar desenvolvedores nesta tarefa crítica.

Uma abordagem que ganha atenção é *gamificação em engenharia de software*, que introduz elementos de jogo como pontos, leaderboards e feedback para apoiar tarefas de desenvolvimento de software (OLGUN et al., 2017; CALL; FOX; SPRINT, 2021). Embora a gamificação tenha demonstrado aumentar o engajamento, seu papel na detecção de inconsistências ainda não foi sistematicamente explorado. Alguns estudos (BUCCHIARONE et al., 2023a; YIGITBAS et al., 2024a) têm usado elementos de gamificação no contexto do campo de engenharia baseado em modelos. No entanto, não está claro se elementos gamificados melhoram a capacidade dos desenvolvedores de detectar inconsistências ou distrair da tarefa de desenvolvimento. A utilidade da gamificação para aliviar a carga de lidar com inconsistências de modelos não tem sido investigada extensiva e sistematicamente. Poderíamos supor que a gamificação ajudaria os desenvolvedores a entender melhor o design antes de implementá-lo. Outros podem postular que o aumento do engajamento e motivação reduziria o esforço para detectar inconsistências ou diminuir interpretações erradas entre modelos de desenho complementares.

No entanto, estes pressupostos não são óbvios. Primeiro, elementos gamificados adicionais podem prejudicar a compreensão do design de software. Segundo, não está claro se o uso de elementos gamificados pode aumentar a taxa de detecção de inconsistências e melhorar a interpretação de modelos UML. Terceiro, os desenvolvedores podem ser distraídos por aspec-

tos lúdicos como competição e recompensas, afetando negativamente a qualidade da detecção de inconsistência. Finalmente, os desenvolvedores podem ainda gastar mais esforços usando modelos gamificados ao examinar todos os objetivos e desafios apresentados, o que poderia superá-los e comprometer a eficácia do processo de verificação. Apesar dessas incertezas, a gamificação tem um potencial significativo para transformar como os desenvolvedores interagem com modelos de software. Incorporar elementos de jogo pode tornar as atividades de modelagem mais envolventes e envolventes, possivelmente resultando em um compromisso mais significativo dos desenvolvedores para a tarefa de detectar e corrigir inconsistências. Além disso, a gamificação pode introduzir uma abordagem mais colaborativa e competitiva para o processo de verificação, incentivando os desenvolvedores a compartilhar conhecimento e soluções. Este aspecto colaborativo pode ser particularmente valioso em grandes equipes de desenvolvimento distribuídas, onde uma comunicação eficaz é crucial para o sucesso do projeto.

Neste capítulo, propomos a ideia de realizar a utilização de técnicas de gamificação para engajar os desenvolvedores na implementação de artefatos de software utilizando UML gerando melhoria nos mesmos, tendo como objetivo responder a QP3 apresentada no Capítulo 1 (Introdução). O capítulo propõe uma série de estudos empíricos, incluindo experimentos controlados e quase-experimentos, os quais seguirão as boas práticas estabelecidas em (WOHLIN et al., 2012) e está organizado da seguinte forma: a Seção 5.1 fornece uma visão geral da técnica de gamificação aplicada no contexto de modelagem de software, com foco em particular no uso de UML. A seção 5.2 descreve a metodologia adotada. A aplicação das atividades são apresentados na Seção 5.3, e a Seção 5.4 detalha como serão aplicadas as entrevistas e o questionário de pesquisa. As considerações finais são descritas na Seção 5.5.

Neste contexto, este artigo relata um experimento controlado para investigar o impacto da gamificação sobre (1) a taxa de detecção de inconsistências, (2) o esforço gasto para detectar essas inconsistências e (3) a taxa de erro de interpretação causada por inconsistências. Comparamos o uso de modelos gamificados com modelos tradicionais em um contexto específico: o uso e compreensão dos desenvolvedores de modelos de projeto UML (ou seja, diagramas de classes UML e sequências) necessários para produzir suas implementações correspondentes. A principal contribuição deste trabalho reside na geração de conhecimento empírico (Seção 5.4) e na proposta de uma agenda de pesquisa e discussão profunda sobre os efeitos da gamificação no contexto da modelagem de software (Seção 5.5).

Os resultados (descritos na seção 5.4), apoiados por testes estatísticos e análise qualitativa, mostram que a gamificação pode aumentar o engajamento do desenvolvedor, mas também pode ter impactos mistos na detecção de inconsistências e modelos de interpretação. Observamos, por exemplo, que a superficialidade da aplicação de elementos gamificados causou principalmente os aspectos negativos da gamificação. Ou seja, quanto mais simples e focado em recompensas imediatas os elementos, menos eficaz é a detecção de inconsistências e maiores são as taxas de erro de interpretação. Observamos também que os sujeitos tenderam a detectar inconsistências mais rapidamente quando os elementos gamificados estavam alinhados com os objetivos de

aprendizagem.

A análise de dados sugere que a gamificação tem um papel complexo na detecção de inconsistências. Em algumas situações, elementos gamificados como pontos e recompensas têm incentivado os desenvolvedores a dedicar mais tempo e esforço para tarefas de verificação, resultando em uma maior taxa de detecção de inconsistência. No entanto, em outros casos, esses elementos distraem a atenção dos desenvolvedores, levando a uma compreensão superficial dos modelos e a uma identificação menos efetiva dos problemas. Esses achados destacam a importância de se projetar cuidadosamente elementos de gamificação para garantir que eles apoiem, não detraem, a verificação de inconsistências. Este estudo empírico inicial beneficia desenvolvedores de software e pesquisadores ao fornecer insights direcionados a dados sobre como os elementos de gamificação influenciam o esforço de detecção de inconsistência, a taxa de detecção e a taxa de interpretação incorreta.

Os achados informam o desenho de ferramentas e metodologias de modelagem mais eficazes, auxiliando os profissionais na otimização de seus processos de verificação. Além disso, os pesquisadores obtêm uma compreensão mais profunda do impacto da gamificação em tarefas cognitivas relacionadas ao design de software. Os resultados são os primeiros a apontar as potenciais vantagens e desvantagens da gamificação na modelagem de software multi-view (OMG, 2017). Entendendo melhor os impactos da gamificação na detecção de inconsistências, esperamos fornecer insights valiosos à comunidade de desenvolvimento de software, que pode orientar a adoção de práticas gamificadas em contextos de modelagem.

O restante do artigo é organizado da seguinte forma. A seção 5.1 descreve os principais conceitos discutidos ao longo do artigo. Seção 5.2 compara o trabalho com outros, destacando as principais diferenças e semelhanças. Seção 5.3 descreve o protocolo de pesquisa adotado. A seção 5.4 discute os resultados do estudo. A seção 5.5 discute os resultados e apresenta uma agenda de pesquisa. A seção 5.6 descreve como ameaças à validade foram mitigadas. Por último, a seção 5.7 apresenta algumas observações finais e trabalhos futuros.

## 5.1 Background

Esta seção apresenta os conceitos essenciais para a compreensão deste trabalho. Seção 5.1.1 foca em inconsistências em modelos UML. Seção 5.1.2 introduz o uso de gamification na modelagem de software.

### 5.1.1 Inconsistências em Modelos UML

*Inconsistências em modelos UML* (OMG, 2017; TORRE et al., 2023; TORRE; LABICHE; GENERO, 2014) ocorrem quando dois ou mais modelos que representam diferentes aspectos de um sistema, como diagramas de classes e sequências, fornecem informações conflitantes ou incompatíveis sobre a estrutura ou comportamento do sistema. Essas inconsistências normal-



mente surgem do desalinhamento de elementos compartilhados entre modelos, como operações de classe ou interações de mensagens, levando a contradições em como o sistema é definido. Por exemplo, um diagrama de classe pode especificar uma operação em falta ou se comportar de forma diferente no diagrama de sequência correspondente. Esta informação contraditória pode levar a interpretações e erros durante a implementação, se não detectado e resolvido.

As regras de consistência têm sido amplamente exploradas e documentadas na literatura. Nesse sentido, ao invés de propor novas inconsistências, utilizamos catálogos de inconsistência previamente publicados e validados na literatura, como (TORRES; BRAND; SEREBRENİK, 2021; TORRE et al., 2020; TORRE; LABICHE; GENERO, 2014; LANGE; CHAUDRON, 2006), garantindo a robustez e confiabilidade das regras empregadas. A Tabela 7 apresenta uma lista de inconsistências exploradas no estudo.

**Tabela 7:** Visão geral das inconsistências utilizadas no estudo

Acrônimo	Descrição
Cm	Múltiplas definições de classe com o mesmo nome
CnSD	Classe não instanciada no Diagrama de Sequência (SD)
ED	Mensagem na direção incorreta
EnN	Mensagem sem nome
EcM	Mensagem sem método associado
CaSD	Classe abstrata instanciada no Diagrama de Sequência (SD)

Estudos empíricos anteriores (MARCHEZAN et al., 2023c; TORRES; BRAND; SEREBRENİK, 2021; LANGE; CHAUDRON, 2006) revelaram que inconsistências em modelos UML podem levar a sérias consequências negativas no processo de desenvolvimento de software. Quando um desenvolvedor lê diagramas com informações conflitantes ou incompatíveis, como discrepâncias entre um diagrama de classe e um diagrama de sequência, aumenta a probabilidade de interpretações incorretas. Essas interpretações erradas podem resultar em desenvolvedores implementando funcionalidades que se alinham de forma diferente do design pretendido, introduzindo defeitos no código. Consequentemente, esses defeitos podem se propagar em todo o sistema de software, levando a comportamentos incorretos, degradação do desempenho e desafios de manutenção. A necessidade de refazer ou depurar esses problemas adiciona complexidade, custo e tempo desnecessários ao ciclo de vida do desenvolvimento, comprometendo a qualidade e confiabilidade do produto de software final.

Nesse sentido, a academia e a indústria precisam investigar os efeitos de inconsistências em modelos UML para entender como essas inconsistências impactam o processo de desenvolvimento de software e a qualidade do código fonte. Modelos inconsistentes podem levar à falta de comunicação entre equipes de desenvolvimento, implementação incorreta dos requisitos do sistema (LANGE; CHAUDRON, 2006) e introdução de defeitos no software. Ao estudar os efeitos das inconsistências, os pesquisadores podem identificar as causas profundas dessas

questões, quantificar a extensão de seu impacto nos esforços de desenvolvimento e confiabilidade do sistema e propor métodos para amenizá-las. Além disso, esta pesquisa é essencial para o desenvolvimento de ferramentas e técnicas automatizadas para detectar, prevenir ou resolver inconsistências, melhorar a qualidade do software, reduzir custos e aumentar a produtividade no desenvolvimento orientado por modelos.

Investigar os efeitos dos elementos de gamificação na detecção de inconsistências no ensino de modelagem de software é essencial por várias razões. Primeiro, a gamificação tem o potencial de melhorar o engajamento e motivação dos alunos, que são fundamentais para uma aprendizagem eficaz em temas complexos como a modelagem de software. Ao incorporar elementos de gamificação como pontos, leaderboards e recompensas, os alunos podem se envolver mais na identificação e resolução de inconsistências em modelos UML, uma habilidade essencial no desenvolvimento orientado por modelos. Estudos anteriores têm destacado os benefícios da gamificação no ensino (YIGITBAS et al., 2024a; CAGNAZZO et al., 2023; GARACCIONE et al., 2024). Em segundo lugar, inconsistências entre modelos UML podem levar a mal-entendidos e erros no design de software, que, quando transferidos para implementação, resultam em defeitos. Entender como ambientes de aprendizagem gamificados impactam a capacidade dos alunos de detectar essas inconsistências ajuda os educadores a otimizar estratégias de ensino e melhorar os resultados de aprendizagem. Por fim, apreender os efeitos de elementos de gamificação pode fornecer informações valiosas sobre se a gamificação pode facilitar uma compreensão mais profunda das complexas inconsistências na modelagem UML. Se demonstrada de forma eficaz, a gamificação poderia ser sistematicamente incorporada à educação em engenharia de software para melhorar a precisão e eficiência do ensino de conceitos fundamentais de modelagem.

### 5.1.2 Gamificação em Modelagem de Software

Retomando o conceito de gamificação, agora no contexto específico da Engenharia de Software, essa abordagem consiste na incorporação estratégica de elementos de jogos, como pontos, emblemas e leaderboards, aos processos de desenvolvimento de software, com o objetivo de apoiar o engajamento e a motivação das equipes (PRIYADI et al., 2022). Nessa perspectiva, tais elementos são utilizados para tornar atividades potencialmente complexas ou repetitivas mais envolventes, explorando mecanismos motivacionais já discutidos anteriormente (DETERDING et al., 2011a; SAPPAILE et al., 2024; SANTOS et al., 2024).

Hoje, não há taxonomia clara e comumente aceita dos elementos do jogo. Em (SHPAKOVA; DÖRFLER; MACBRYDE, 2016), os autores propuseram uma visão unificada das diferentes classificações, que resume as dimensões da gamificação em Componentes, Mecânica e Dinâmica. Componentes são os blocos básicos de construção da gamificação; eles representam os objetos que os usuários veem e interagem com, como emblemas, níveis e pontos. A mecânica define o jogo como um sistema baseado em regras, especificando como tudo se com-

porta e como o jogador pode interagir com o jogo. Dinâmica é o nível superior dos elementos de gamificação. Eles incluem todos os aspectos do jogo que não podem ser implementados e gerenciados diretamente e estão relacionados com as respostas emocionais dos usuários (por exemplo, progressão e exploração). Elementos de gamificação são destinados a atrair atenção, entreter e manter o usuário em uma determinada tarefa. Os elementos mais populares são pontos, emblemas e leaderboards, definidos como a tríade PBL (MORSCHHEUSER et al., 2017).

O sucesso de gamificar um determinado contexto não-gambling depende fortemente do design de gamificação escolhido para essas dimensões. Vários esforços de pesquisa têm focado na identificação das fases que compõem o projeto de gamificação (FU et al., 2024; WEBB, 2013). No entanto, semelhante à taxonomia dos elementos de gamificação, não há fases comumente aceitas; elas variam em número e terminologia utilizadas.

Ao longo dos anos, surgiram alguns estudos para verificar como aplicar a gamificação na modelagem de software (BEECHAM et al., 2008; DUBOIS; TAMBURRELLI, 2013; GARACCIA et al., 2017; DORLING; MCCAFFERY, 2012; TONHÃO et al., 2023). Como esse campo teórico e prático está crescendo rapidamente, é necessário desenvolver estudos quantitativos e qualitativos para apreender seus efeitos nas práticas de modelagem de software, principalmente no contexto da detecção de inconsistências. A inserção de técnicas de gamificação, como feedback, progresso e desafios, na modelagem de software pode ajudar a mitigar as questões de adoção da modelagem UML. Uma investigação empírica sobre a aplicabilidade desta técnica na modelagem de software com UML visando melhorar a qualidade dos modelos ainda está pendente. Isso proporcionaria à comunidade de software conhecimento sobre o retorno sobre o investimento do uso de gamificação na modelagem com UML.

## 5.2 Trabalhos Relacionados

A seleção dos estudos relacionados seguiu duas etapas. Primeiramente, realizamos uma busca inicial em repositórios digitais, incluindo Google Scholar<sup>1</sup> e Scopus<sup>2</sup>, para identificar artigos dos últimos vinte anos sobre o uso da gamificação em modelagem de software com UML. Em segundo lugar, filtramos os artigos selecionados com base em seu alinhamento com os objetivos do estudo. Esses artigos foram então analisados (Seção 5.2.1) e comparados com oportunidades de pesquisa pontuais (Seção 5.2.2).

### 5.2.1 Análise dos Trabalhos Relacionados

**Garaccione et al. (2025) (GARACCIONE et al., 2025a).** Este estudo apresenta o desenvolvimento e avaliação empírica da UMlegende, uma ferramenta educacional gamificada destinada a melhorar o engajamento e a competência dos alunos na modelagem UML. A pla-

<sup>1</sup>Google Scholar: <https://scholar.google.com/>

<sup>2</sup>Scopus: <https://www.scopus.com/>

taforma incorpora elementos comuns de gamificação, como personalização avatar, questlines, sistemas de pontuação, realizações, leaderboards e feedback ao vivo, para melhorar a experiência de aprendizagem. Foi realizado um experimento com 280 estudantes de pós-graduação matriculados em um curso de Sistemas de Informação para medir os efeitos da gamificação na modelagem de produtividade e qualidade. Os resultados indicam um aumento de 8% no tamanho do diagrama e uma melhora de 11% na correção quando mecânica gamificada são empregadas. Além disso, a abordagem foi percebida positivamente pela maioria dos participantes, que relataram maior motivação e satisfação. Embora o estudo contribua com insights valiosos sobre o impacto da gamificação na proficiência de engajamento e modelagem dos alunos, não aborda inconsistências de modelos ou envolve experimentação controlada visando o alinhamento semântico multi-view, o que limita sua aplicabilidade a contextos de modelagem mais complexos.

**Cammarts et al. (2025) (CAMMAERTS; SNOECK, 2025)** Este estudo explora a aplicação de técnicas de gamificação para modelar testes, com base em abordagens existentes para testes de código gamificados. Embora a gamificação tenha mostrado recentemente uma promessa em aumentar o engajamento dos alunos na educação em testes de software, o aumento da Engenharia Modelo-Driven (MDE) permitiu que os alunos não técnicos gerassem software totalmente funcional através de transformações modelo-para-código. No entanto, esses estudantes muitas vezes negligenciam testar seus modelos completamente. Para suprir essa lacuna, os autores propõem uma nova abordagem de teste-modelo gamificado, adaptando e reutilizando o conhecimento a partir de estratégias de teste-código gamificado. O artigo apresenta uma avaliação inicial da solução proposta com foco na facilidade de uso. Ele também destaca os desafios encontrados durante a tradução de gamificação de teste de código para modelo, oferecendo insights e recomendações para futuros esforços na adaptação de ambientes de teste de código gamificados para o domínio teste de modelo.

**Yigitbas et al. (2024) (YIGITBAS et al., 2024a).** Este estudo apresenta GaMoVR, um ambiente virtual de aprendizagem baseado na realidade e gamificado projetado para apoiar os alunos na modelagem UML de aprendizagem. Dado que UML se tornou a linguagem de modelagem de fato adotada em cursos universitários de ciência da computação, GaMoVR foi desenvolvido para melhorar o engajamento, motivação e prazer dos alunos durante o processo de aprendizagem de modelagem. O ambiente oferece uma experiência interativa e imersiva, incorporando tarefas de modelagem em mini-jogos, proporcionando assim uma abordagem lúdica e envolvente à aprendizagem. Além disso, o GaMoVR possui um ambiente de modelagem multi-viewpoint que permite aos alunos explorar como diagramas de classes UML se relacionam com modelos 3D que instanciam os sistemas modelados. O estudo visa abordar as limitações de ferramentas de aprendizagem interativas anteriores, aumentando significativamente a imersão do aprendiz e melhorando os resultados de aprendizagem através da gamificação.

**Bucchiarone et al. (2024) (BUCCHIARONE et al., 2024).** Este estudo apresenta o PolyGloT-UML, um framework flexível projetado para melhorar a aprendizagem de UML, in-

tegrando diversas ferramentas educacionais comumente usadas na comunidade de engenharia orientada por modelos. Reconhecendo que a modelagem é uma atividade fundamental no projeto conceitual e do sistema e que aplicações de aprendizagem UML interativas e gamificadas existentes muitas vezes carecem de adaptabilidade e personalização, PolyGloT-UML tem como objetivo resolver essas lacunas. O framework apoia a criação de experiências de aprendizagem dinâmicas e personalizadas, melhorando assim o engajamento do aluno e os resultados da aprendizagem. Este artigo demo destaca como PolyGloT-UML unifica múltiplas ferramentas em um ambiente educacional coeso e eficaz para ensinar UML e outros conceitos de modelagem.

**Feichas et al. (2023)** (**FEICHAS; SEABRA, 2023**). Este estudo apresenta o uso de uma plataforma web gamificada para estudar modelagem de software com UML. Argumenta que, embora a UML esteja constantemente melhorando e sendo estudada, há evidências de dificuldade em ensinar e aprender o sujeito devido à complexidade de seus conceitos e às dificuldades cognitivas dos aprendizes com abstração. Através das interações, foi possível analisar os comportamentos e engajamento dos alunos participantes da pesquisa. Os dados mostraram que acessaram a plataforma durante e após as horas de aula. Os registros mostraram que os alunos completaram a maioria dos testes de conhecimento disponíveis.

**Almeida et al. (2023)** (**ALMEIDA et al., 2023**). Este trabalho realizou uma pesquisa para fornecer uma visão abrangente dos estudos relatando os efeitos adversos dos elementos de design de jogos e apresentando informações sobre a conscientização dos desenvolvedores sobre esses efeitos e como eles podem ser considerados na prática. Como resultado, identificou-se que, quando bem aplicada, a gamificação pode ter impactos positivos no software educacional. No entanto, o software gamificado também é propenso a gerar efeitos nocivos.

**Marchezan et al. (2023)** (**MARCHEZAN et al., 2023a**). Esta pesquisa apresenta um estudo experimental controlado para avaliar se recomendações automáticas podem ajudar desenvolvedores de software a reparar modelos de projeto inconsistentes de forma mais eficiente. O estudo foi realizado com a participação de 60 desenvolvedores e dividido em dois grupos: um grupo utilizou uma ferramenta que forneceu recomendações automáticas para reparar modelos de projeto inconsistentes, enquanto o outro grupo não teve acesso a essas recomendações. Os participantes realizaram tarefas de reparo do modelo de projeto, e diversos aspectos, como tempo de preenchimento, acurácia e satisfação do usuário, foram mensurados. Os resultados mostraram que o grupo que utilizou a ferramenta de recomendações automáticas completou as tarefas mais rapidamente do que o grupo que não teve acesso a essas recomendações. Além disso, o grupo que utilizou as recomendações foi significativamente melhor quanto à acurácia e satisfação do usuário.

**Alsadoon et al. (2022)** (**ALSADOON; ALKHAWAJAH; SUHAIM, 2022**). Este estudo experimental investiga os efeitos de um ambiente de aprendizagem gamificado no desempenho acadêmico, motivação e satisfação dos alunos. Os autores realizaram um estudo com 116 estudantes de graduação, divididos em dois grupos: um grupo experimental que utilizou um ambiente gamificado para aprender conceitos de programação em C e um grupo controle que

utilizou um ambiente não gamificado. O desempenho acadêmico, a motivação e a satisfação do aluno foram avaliados antes e após o experimento. Os resultados indicaram que os estudantes do grupo experimental apresentaram desempenho acadêmico significativamente melhor do que os do grupo controle. Além disso, estudantes do grupo experimental relataram maior motivação para aprender, maior satisfação com a experiência de aprendizagem e maior intenção de utilizar novamente o ambiente de aprendizagem gamificado.

**Junior et al. (2021)** (**JÚNIOR; FARIAS, 2021**). Este estudo propõe ModelGame, um modelo de qualidade projetado para apoiar a aprendizagem de modelagem de software através da gamificação. Embora a gamificação tenha sido cada vez mais adotada em tarefas de desenvolvimento de software para melhorar o engajamento do desenvolvedor, questões como incompletude e inconsistência em modelos UML permanecem uma preocupação. Os autores conjecturam que a mecânica de gamificação pode melhorar o engajamento do aprendiz e contribuir para abordar esses problemas de qualidade na modelagem UML. Ao contrário do trabalho anterior, que tem negligenciado amplamente a relação entre gamificação e qualidade do modelo UML em contextos educacionais, ModelGame fornece um framework parametrizado para instrutores para avaliar modelos UML criados pelos alunos. Ao integrar atividades gamificadas e oferecer diretrizes com qualidade, a abordagem visa melhorar a qualidade do modelo. Um questionário qualitativo envolvendo 19 instrutores de instituições de ensino superior revelou que 94,7% reconheciam o potencial do ModelGame para melhorar a qualidade do modelo UML e expressavam vontade de adotá-lo em suas práticas de ensino, apesar de 47,4% atualmente não utilizarem nenhuma mecânica de gamificação. Esses achados destacam o papel promissor da gamificação no avanço da educação em modelagem de software.

**Daehli et al. (2021)** (**DÆHLI et al., 2021**). Este estudo apresenta o Learner, uma ferramenta web desenvolvida para apoiar o ensino de habilidades básicas de modelagem de dados. Desenvolvido através de uma colaboração entre a Universidade do Sudeste da Noruega (USN) e Kristiania University College (KUC), o Learner tem sido usado em seis cursos ao longo de quatro anos acadêmicos. A ferramenta permite aos alunos resolver tarefas de modelagem de dificuldade variável ao receber feedback automatizado, incluindo pistas visuais, para apoiar seu processo de aprendizagem. Para aumentar a motivação, o Learner incorpora elementos de gamificação, tais como deduções pontuais para solicitar ajuda e uma tabela de classificação que recompensa os alunos que completam tarefas com assistência mínima. O estudo analisa dados de pesquisa e entrevista e registros de uso de aplicativos para avaliar percepções e desempenho dos alunos. Os resultados indicam o aumento da atividade do aluno em versões mais recentes do Learner, com os alunos relatando que o feedback automatizado ajuda a corrigir erros e que os elementos gamificados contribuem positivamente para o seu engajamento e aprendizagem. A análise também identifica erros e preferências comuns de estudantes para certos tipos de atribuições.

**Jurgelaitis et al. (2018)** (**JURGELAITIS; DRUNGILAS; ČEPONIENĖ, 2018**). Este trabalho investigou como a Gamificação poderia ser inserida em um curso de Modelação de Sis-



temas de Informação cobrindo vários tópicos sobre UML. Como resultado, o curso de modelagem de sistemas gamificados foi implementado no ambiente Moodle, usando plugins adicionais para utilizar os elementos gamificados necessários. O estudo mostrou resultados promissores e obteve aceitação positiva pelos alunos participantes.

**Cosentino et al. (2017) (COSENTINO; GÉRARD; CABOT SAGRERA, 2017b).** Este estudo apresenta uma abordagem baseada em modelos para modelar o aprendizado de forma gamificada. A abordagem incluiu uma nova linguagem para modelar o processo de gamificação e um ambiente onde ele pode ser incorporado em ferramentas de modelagem atuais para permitir que instrutores e alunos projetem e usem um framework de modelagem completo, incluindo elementos de gamificação. Além disso, a abordagem também teve a proposta de subsidiar a coleta e análise de dados de gamificação, facilitando as atividades de monitoramento.

**RW13: Yohannis (2016) (YOHANNIS, 2016).** Este estudo propõe uma abordagem de gamificação que utiliza elementos de design de jogos, como pontos, crachás e leaderboards, para motivar os alunos e criar uma experiência de aprendizagem mais envolvente. Descreve o desenvolvimento de uma plataforma de aprendizagem de modelagem de software chamada "Modeling World" que incorpora recursos de gamificação. A plataforma foi testada em um estudo com 61 estudantes de graduação, e os resultados mostraram que a abordagem de gamificação efetivamente aumentou a motivação e o engajamento dos alunos com o tema. Os estudantes relataram desfrutar da experiência de aprendizagem gamificada e sentiram que isso os ajudou a entender melhor os conceitos.

**Farias et al. (2015) (FARIAS et al., 2015).** Este trabalho apresenta um experimento controlado sobre o esforço necessário para compor modelos de projeto. Os autores realizaram um estudo com 30 participantes, divididos em dois grupos, que foram orientados a criar modelos de design para sistemas de gerenciamento de conteúdo utilizando a linguagem de modelagem UML. Os resultados indicaram que o grupo que utilizou a abordagem top-down levou menos tempo para criar seus modelos e produziu modelos mais completos e coerentes do que aqueles que usaram a abordagem bottom-up. Além disso, o grupo top-down mostrou maior facilidade na compreensão do modelo do colega e identificação de possíveis melhorias.

**Lange et al. (2006) (LANGE; CHAUDRON, 2006).** O estudo apresenta uma investigação experimental dos efeitos de defeitos em modelos UML no processo de desenvolvimento de software. Os autores realizaram um estudo controlado com 42 participantes, divididos em três grupos, que foram instruídos a realizar tarefas de compreensão de modelos UML, identificação de defeitos e correção de defeitos em modelos. Os resultados indicaram que defeitos nos modelos UML afetaram significativamente a efetividade e eficiência das tarefas dos participantes e sua satisfação com o processo de correção de defeitos. Os participantes também relataram que a qualidade do modelo original e a facilidade de compreensão dos modelos influenciaram sua capacidade de identificar e corrigir defeitos.

**Tabela 8:** Resumo dos trabalhos relacionados sobre gamificação e detecção de inconsistências em modelagem de software.

Estudo	Método de Pesquisa	Mecânicas de Gamificação	Nº de Participantes	Variáveis Controladas	Contexto	Contribuição
<i>Este Trabalho</i>	<i>Experimento controlado</i>	<i>Pontos, rankings, avatares, feedback</i>	<i>60</i>	<i>Esforço, Má interpretação, Correção</i>	<i>Ensino de modelagem de software baseada em UML</i>	<i>Conhecimento empírico, agenda de pesquisa</i>
Garaccione et al. (2025)	Experimento controlado	Avatares, missões, pontos, conquistas, rankings, feedback em tempo real	280	Produtividade (tamanho), Correção	Educação em UML	Ferramenta (UM-Legend)
Cammaerts et al. (2025)	Estudo preliminar (não controlado)	Gamificação adaptada de testes de código	Não reportado	Usabilidade	Teste de modelos (MDE)	Processo
Yigitbas et al. (2024)	Estudo exploratório	Mini-games, imersão em VR, gamificação geral	Não reportado	Não mencionado	Educação em UML em ambiente de VR	Ferramenta (GaMoVR)
Bucchiarone et al. (2024)	Demonstração / Proposta de framework	Personalização e adaptação da experiência	Não reportado	Não mencionado	Educação em UML	Ferramenta (PolyGloT-UML)
Feichas et al. (2023)	Estudo observacional	Plataforma gamificada, testes de conhecimento	Não reportado	Engajamento	Educação em UML	Conhecimento empírico
Almeida et al. (2023)	Survey	Discussão sobre efeitos colaterais da gamificação	Não reportado	Não aplicável	Educação em software	Conhecimento empírico
Marchezan et al. (2023)	Experimento controlado	Recomendações automáticas (sem gamificação)	60	Tempo, Acurácia, Satisfação	Detecção e resolução de inconsistências em modelos	Ferramenta/processo
Alsadoon et al. (2022)	Experimento controlado	Sistema de pontos, ambiente gamificado	116	Desempenho acadêmico, Motivação	Programação (C++)	Conhecimento empírico
Junior et al. (2021)	Proposta com avaliação qualitativa	Sistema gamificado, diretrizes de qualidade	19 (professores)	Qualidade do modelo (correção e completude)	Educação em UML	Ferramenta (ModelGame)
Daehli et al. (2021)	Estudo longitudinal	Deduções de pontos, ranking, feedback automatizado	Não reportado	Atividade do estudante, Correção de erros	Modelagem de dados	Ferramenta (LearnER)
Jurgelaitis et al. (2018)	Estudo de implementação	Moodle gamificado (plugins)	Não reportado	Não mencionado	Educação em UML	Ferramenta/processo
Cosentino et al. (2017)	Proposta metodológica	Modelagem de processos de gamificação	Não reportado	Não mencionado	Educação em modelagem	Processo
Yohannis (2016)	Estudo experimental	Pontos, badges, rankings	61	Motivação, Engajamento	Educação em modelagem	Ferramenta (Modelling World)
Farias et al. (2015)	Experimento controlado	Não aplicável (sem gamificação)	30	Tempo, Completeness, Consistência	Criação de modelos UML	Conhecimento empírico
Lange et al. (2006)	Experimento controlado	Não aplicável (sem gamificação)	42	Efetividade, Eficiência, Satisfação	Detecção e correção de defeitos em UML	Conhecimento empírico



### 5.2.2 Análise Comparativa e Oportunidades

Para posicionar rigorosamente este estudo dentro do estado da arte, realizamos uma revisão de literatura estruturada focada na intersecção da gamificação e modelagem de software, com ênfase particular na Linguagem Unificada de Modelação (UML). Embora a gamificação tenha ganhado força no campo da educação em engenharia de software, demonstrando potencial para melhorar a motivação do aprendiz, engajamento e compreensão conceitual, sua aplicação direcionada à modelagem UML, especialmente no contexto da detecção de inconsistência semântica em diagramas multi-view, permanece subexplorada e conceitualmente fragmentada.

A Tabela 8 sintetiza uma selecção de estudos representativos publicados nas últimas duas décadas. Esses trabalhos foram selecionados com base em sua relevância para os temas de intervenções gamificadas, validação empírica e alinhamento com a educação em modelagem. A Tabela 8 compara sistematicamente cada estudo em múltiplas dimensões analíticas, incluindo método de pesquisa, mecânica de gamificação empregada, número de participantes, variáveis controladas, contexto de modelagem e contribuições primárias. Essa matriz comparativa permite identificar pontos fortes metodológicos e limitações conceituais que afetam a validade, replicabilidade e aplicabilidade dos achados relatados na literatura.

A análise revela três tendências notáveis. Primeiro, enquanto vários estudos propõem ferramentas gamificadas ou plataformas para o ensino de UML, poucos são fundamentados em rigorosos projetos experimentais capazes de isolar os efeitos da gamificação em tarefas de modelagem. Em segundo lugar, variáveis controladas como esforço, má interpretação e correção, fundamentais para o estudo da detecção de inconsistência, raramente são abordadas explicitamente. Em terceiro lugar, apenas uma minoria de trabalhos situa suas pesquisas dentro de tarefas que requerem verificação de consistência multi-visão, cenário altamente relevante para práticas de modelagem de nível industrial.

O estudo aborda essas lacunas unicamente combinando um projeto experimental controlado com uma análise comportamental abrangente de alunos envolvidos em tarefas de modelagem UML. Ao incorporar uma camada de gamificação composta por pontos, avatares, leaderboards e mecanismos de feedback, foram avaliados tanto os resultados motivacionais quanto seus efeitos cognitivos não intencionais, particularmente quanto à acurácia da identificação de inconsistência. Dentre a literatura pesquisada, este trabalho é o único que satisfaz todos os critérios de comparação, oferecendo, assim, uma contribuição empírica distinta para o discurso sobre a educação de modelagem gamificada.

Em suma, a síntese estruturada apresentada na Tabela 8 não só destaca a natureza fragmentada dos esforços de pesquisa atuais, mas também corrobora a originalidade e relevância deste trabalho. Ressalta-se a necessidade premente de estudos embasados empiricamente que escrutifiquem não só os recursos motivacionais da gamificação, mas também suas implicações epistêmicas para a compreensão, validação e refinamento de modelos.

## 5.3 Metodologia

Esta seção descreve as decisões chave que moldam o desenho experimental do estudo controlado. A seção 5.3.1 define o objetivo do estudo e formula as questões de investigação. A seção 5.3.2 traduz estas questões em hipóteses sistematicamente declaradas. A Seção 5.3.3 detalha o contexto e os critérios para selecionar os participantes. A seção 5.3.4 identifica as variáveis sob investigação e explica os métodos de quantificação empregados. A seção 5.3.5 apresenta o processo experimental, enquanto a seção 5.3.6 elabora como o estudo foi executado. A seção 5.3.7 descreve as tarefas experimentais específicas realizadas. Por último, a seção 5.3.8 apresenta o pacote de replicação. Os passos metodológicos seguem diretrizes estabelecidas para estudos empíricos, conforme delineado em (WOHLIN et al., 2012), garantindo rigor e reprodutibilidade.

### 5.3.1 Objetivo e Questões de Pesquisa

Este estudo avalia os efeitos dos elementos de gamificação na compreensão dos modelos de desenho da UML em relação a duas variáveis: esforço de detecção de inconsistência, correção e interpretação incorreta. Esses efeitos são investigados a partir de cenários envolvendo atividades gamificadas de modelos UML para que resultados empíricos possam ser gerados. Tendo isso em vista, o objetivo deste estudo é indicado com base no modelo GQM (WOHLIN et al., 2012) da seguinte forma:

**Analisar** os elementos de gamificação  
**para fins de** investigar os seus efeitos  
**No que se diz a respeito de** esforço, correção e interpretação incorreta  
**do ponto de vista dos alunos**  
**no contexto de** modelos de design multivisão.

Assim, focamos em três questões de pesquisa (QP):

- **QP3.1:** O uso de elementos de gamificação impacta a eficiência dos alunos na detecção de inconsistências em modelos UML?
- **QP3.2:** O uso de elementos de gamificação impacta o esforço investido pelos alunos na detecção de inconsistências em modelos UML?
- **QP3.3:** O uso de elementos de gamificação reduz a taxa de interpretação incorreta na modelagem UML em comparação com a modelagem UML sem gamificação?

Um experimento controlado foi projetado para abordar essas três questões de pesquisa, seguindo diretrizes bem estabelecidas da literatura (WERBACH; HUNTER, 2015).

### 5.3.2 Formulação de Hipóteses

**Hipótese 1 (H1): Taxa de detecção ( $\mathbb{DR}$ ).** A primeira pergunta de pesquisa investiga se os desenvolvedores que usam elementos de gamificação produzem uma inconsistência maior (ou menor) *taxa de detecção* em comparação com aqueles que não usam gamificação. Normalmente, os desenvolvedores lutam para detectar inconsistências existentes em modelos multivisão, muitas vezes devido à falta de motivação ou foco ao se envolver em tarefas repetitivas e cognitivamente exigentes. Elementos de gamificação, como pontos, crachás e leaderboards, poderiam aumentar o engajamento e a motivação, incentivando os desenvolvedores a investir mais esforços na detecção de inconsistências. Além disso, mecanismos de feedback gamificados podem ajudar os desenvolvedores a acompanhar seus progressos e realizações, potencialmente melhorando sua atenção aos detalhes e promovendo uma revisão completa dos modelos.

Assim, a intuição é que incorporar elementos de gamificação poderia levar a uma maior taxa de detecção de inconsistência. No entanto, não é de modo algum certo que esta hipótese se mantenha. A introdução de elementos de gamificação pode distrair os desenvolvedores, deslocando seu foco da tarefa para alcançar recompensas ou superar pares. Essa mudança poderia levar a revisões superficiais dos modelos, reduzindo a precisão da detecção de inconsistência. Além disso, os desenvolvedores podem experimentar fadiga ou frustração se a mecânica de gamificação for percebida como excessivamente competitiva ou desalinhada com a natureza da tarefa. Essas hipóteses são resumidas da seguinte forma:

**Hipótese nula 1,  $H1_{null}$ :** A taxa de detecção de inconsistências nos modelos UML gamificados (GAMIF) é igual ou inferior a nos modelos não gamificados (N.GAMIF).

$$H1_{null}: \mathbb{DR}(models)_{GAMIF} \leq \mathbb{DR}(models)_{N.GAMIF}$$

**Hipótese Alternativa 1,  $H1_{-1}$ :** A taxa de detecção de inconsistência em modelos UML gamificados (GAMIF) é maior do que em modelos não gamificados (N.GAMIF).

$$H1_{alt}: \mathbb{DR}(models)_{GAMIF} > \mathbb{DR}(models)_{N.GAMIF}$$

**Segunda hipótese (H2): Esforço de detecção ( $\mathbb{DE}$ ).** A segunda pergunta de pesquisa investiga se os desenvolvedores investem menos (ou mais) esforço para detectar inconsistências ao usar elementos de gamificação em comparação com não usá-los. Elementos de gamificação como rastreamento de progresso, recompensas e leaderboards podem ajudar a manter o foco e a motivação dos desenvolvedores durante tarefas de detecção de inconsistência. Esse aumento do engajamento pode reduzir a fadiga cognitiva tipicamente associada a essas tarefas, permitindo que os desenvolvedores aloquem seu esforço de forma mais eficaz. Consequentemente, o feedback estruturado através da gamificação poderia permitir aos desenvolvedores identificar inconsistências com menos esforço percebido sistematicamente. Essa suposição é sustentada pelos benefícios motivacionais observados em sistemas gamificados, que incentivam os usuários a perseverar por meio de tarefas desafiadoras.

Assim, esperamos que a incorporação de elementos de gamificação reduza o esforço necessário para a detecção da inconsistência, tornando o processo mais agradável e gratificante. No entanto, não está claro se essa intuição se sustenta. A gamificação pode inadvertidamente aumentar o esforço se os desenvolvedores se fixarem excessivamente em obter recompensas ou competir com pares, potencialmente levando a ineficiências ou distrações. Além disso, a sobrecarga cognitiva de compreensão e adaptação à mecânica de gamificação pode superar os benefícios motivacionais, especialmente se os desenvolvedores percebem o sistema como trivial ou não relacionado à sua tarefa primária. Isso leva à segunda hipótese nula e alternativa:

**Hipótese Nula 2,  $H2_{null}$ :** O esforço para detectar inconsistências em modelos gamificados é igual ou superior ao de modelos não gamificados.

$$H2_{null}: \mathbb{DE}(models)_{GAMIF} \geq \mathbb{DE}(models)_{N.GAMIF}$$

**Hipótese Alternativa 2,  $H2_{alt}$ :** O esforço para detectar inconsistências em modelos gamificados é menor do que em modelos não gamificados.

$$H1_{alt}: \mathbb{DE}(models)_{GAMIF} < \mathbb{DE}(models)_{N.GAMIF}$$

**Terceira hipótese (H3): Taxa de interpretação incorreta (MIR).** A terceira pergunta de pesquisa investiga se a taxa de interpretação incorreta dos desenvolvedores é maior (ou menor) ao usar elementos de gamificação ao interpretar modelos de projeto UML com inconsistências. A má interpretação muitas vezes surge de inconsistências que emergem de descompassos entre diagramas que representam múltiplas visões complementares de um sistema de software (LANGE, 2007). Esses descompassos tornam difícil para os desenvolvedores alcançarem uma compreensão compartilhada, levando à produção de código com semântica diferente. Elementos de gamificação, como feedback de progresso e recompensas para interpretação correta, podem ajudar os desenvolvedores a se concentrar na identificação e resolução de inconsistências.

Ao promover o engajamento sustentado e fornecer incentivos estruturados, a gamificação pode reduzir a fadiga cognitiva, melhorar a atenção aos detalhes e mitigar as taxas de interpretação incorreta. Os desenvolvedores podem se sentir mais motivados a colaborar e verificar suas interpretações, promovendo o alinhamento e reduzindo as discrepâncias semânticas. No entanto, esta expectativa pode não se manter. O uso de elementos de gamificação poderia inadvertidamente desviar o foco dos desenvolvedores da tarefa real para a busca de recompensas ou métricas competitivas, potencialmente levando a análise superficial. Além disso, a camada gamificada adicionada pode introduzir distrações ou sobrecarga cognitiva, dificultando a capacidade dos desenvolvedores de interpretar modelos de design inconsistentes com precisão. Isso levaria às seguintes hipóteses nulas e alternativas:

**Hipótese Nula 3,  $H3_{null}$ :** A taxa de interpretação incorreta em modelos gamificados é igual ou superior a em modelos não gamificados.

$$H3_{null}: \text{MIR}(models)_{GAMIF} \leq \text{MIR}(models)_{N.GAMIF}$$

**Hipótese Alternativa 3,  $H3_{alt}$ :** A taxa de interpretação incorreta em modelos gamificados é igual ou superior a em modelos não gamificados.

$$H3_{alt}: \text{MIR}(models)_{GAMIF} > \text{MIR}(models)_{N.GAMIF}$$

### 5.3.3 Seleção dos Participantes

Os participantes do estudo, 60 estudantes de graduação com experiência profissional inicial, foram selecionados seguindo dois critérios primários: sua formação acadêmica e familiaridade com elementos de gamificação. A utilização de estudantes de graduação como participantes é uma simplificação bem estabelecida e justificada da realidade necessária em ambientes laboratoriais (FALESSI et al., 2018; SALMAN; MISIRLI; JURISTO, 2015). Ele fornece um ambiente controlado para coletar insights preliminares, mantendo a validade do projeto experimental.

Todos os participantes estavam matriculados em instituições de ensino do sistema universitário brasileiro, o que enfatiza fundamentos teóricos e práticos em engenharia de software e disciplinas afins. Eles haviam completado mais de 50 horas de curso focado em design de software, modelagem e programação, demonstrando uma base sólida nos fundamentos da engenharia de software. Além disso, os participantes tiveram exposição prévia a elementos de gamificação através de suas rotinas diárias, incluindo contextos acadêmicos e profissionais, o que lhes proporcionou uma compreensão clara da mecânica e dos benefícios da gamificação. Esse contexto mitigava potenciais ameaças à validade, garantindo que os participantes pudessem se envolver significativamente com elementos de gamificação sem curvas de aprendizagem adicionais. Sua familiaridade com esses elementos contribuiu para uma avaliação mais confiável dos resultados do estudo. As tarefas experimentais foram realizadas dentro do ambiente controlado das práticas laboratoriais.

### 5.3.4 Variáveis e Método de Quantificação

A variável *independente* deste estudo está relacionada ao método de detecção de inconsistências entre modelos UML. Neste caso, pode haver dois valores: detecção de inconsistências sem usar gamificação e usando elementos gamificados (pontos, rankings, feedback e emblemas). Investigamos os efeitos dessa variável independente sobre as seguintes variáveis *dependentes*.

**Taxa de Detecção ( $\mathbb{DR}$ ).** Esta variável mede a taxa global em que as inconsistências são detectadas por todos os sujeitos. Especificamente, é calculado como a razão do número de sujeitos que detectam inconsistências em uma pergunta para aqueles que respondem à pergunta sem relatar quaisquer inconsistências. Uma incoerência é considerada detectada quando um sujeito afirma explicitamente que não consegue uma implementação adequada devido a diagramas conflitantes.

**Detecção de Esforço ( $\mathbb{DE}$ ).** Representa a média de tempo (minutos) que os sujeitos gastam para detectar inconsistências em uma pergunta.

**Taxa de Má Interpretação ( $\mathbb{MI}$ ).** Essa variável representa o grau de variação nas respostas, medindo como as respostas concentradas são mais de quatro alternativas possíveis. A quinta alternativa, no entanto, indica a detecção de uma inconsistência. O questionário teve cinco alter-

nativas. Focamos em se variações nas inconsistências detectadas ou não detectadas influenciam na interpretação dos sujeitos dos modelos de desenho da UML. Uma inconsistência não detectada não é inerentemente problemática se todos os sujeitos chegarem à mesma interpretação (LANGE; CHAUDRON, 2006; CICCOTZI; MALAVOLTA; SELIC, 2019). Por exemplo, se todos os 60 sujeitos selecionarem a mesma resposta (por exemplo, alternativa “A”) para uma pergunta, as inconsistências nos diagramas não resultam em interpretações erradas (MIR = 1). Por outro lado, se as respostas estiverem uniformemente distribuídas nas quatro alternativas, as inconsistências causam interpretações errôneas significativas (MIR = 0). Em resumo, a taxa de erro de interpretação é igual a 0 quando as respostas são distribuídas uniformemente e 1 quando estão totalmente concentradas em uma única opção. Essa variável é quantificada utilizando a fórmula fornecida por Lange et al. (LANGE; CHAUDRON, 2006). Equação 1 mede esta variável, onde  $K$  (número de alternativas a uma pergunta),  $k_i$  (o número de vezes que a alternativa  $i$  foi selecionada, onde  $0 \leq i < K$  and for all  $i$ :  $(0 \leq i < K-1: k_i \geq k_{i+1})$ ), and  $N$  (the sum of answers over all alternatives  $N = \sum_{0 \leq i < K} k_i$ ).

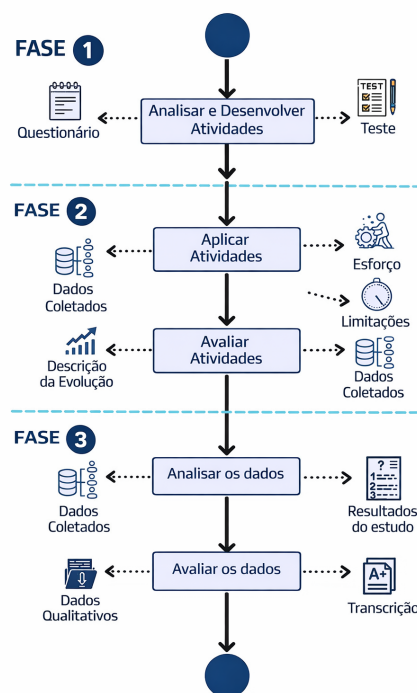
$$MR(k_0, \dots, k_{K-1}) = 1 - 2 \frac{\sum_{0 \leq i < K} k_i i}{N(K-1)} \quad (1)$$

### 5.3.5 Processo Experimental

Figura 16 apresenta o processo experimental fundamentado em metodologias derivadas de estudos empíricos previamente validado (WOHLIN et al., 2012; LANGE, 2007).

**Fase 1: Preparação e planeamento.** Nesta fase, o foco é na concepção e preparação do quadro experimental. Começa pela criação de um questionário, que desempenha um papel crucial na captação de dados de base e identificação de aspectos fundamentais das atividades a serem desenvolvidas. O próximo passo envolve analisar e desenvolver atividades. Isso implica a elaboração de tarefas e experimentos adaptados aos objetivos da pesquisa, garantindo que eles se alinhem às questões exploradas. Essas atividades são iterativamente refinadas e testadas para validar sua relevância e usabilidade. Uma vez concluídas as atividades, o próximo passo crucial é a seleção dos participantes. Isso envolve identificar e recrutar indivíduos cujas características correspondem aos critérios da pesquisa, garantindo uma amostra representativa e válida para o estudo.

**Fase 2: Execução e observação.** Esta fase é dedicada à aplicação e avaliação das atividades projetadas. Inicialmente, as atividades são aplicadas aos participantes selecionados. Durante essa etapa, os dados são coletados enquanto se rastreiam métricas significativas, como o esforço necessário, as limitações enfrentadas durante a execução e quaisquer outras observações notáveis. Após a aplicação, o próximo passo é avaliar as atividades. Isso envolve analisar o desempenho dos participantes e documentar a descrição da evolução, ou seja, como as habilidades, compreensão ou resultados dos participantes evoluem através das tarefas. Esta etapa produz



**Figura 13:** Processo experimental

uma riqueza de dados coletados, o que constitui a base para uma análise mais aprofundada na próxima fase.

**Fase 3: Análise e resultados dos dados.** A fase final centra-se na interpretação dos dados coletados para obter insights significativos. Inicialmente, os dados coletados em fases anteriores são submetidos à análise. Isso inclui métricas quantitativas e observações qualitativas, visando descobrir padrões, correlações ou achados significativos que respondam às questões da pesquisa. Posteriormente, esses dados são avaliados, garantindo que os insights elaborados sejam precisos, relevantes e contextualmente válidos. O resultado inclui resultados do estudo, resumo dos principais achados, e uma transcrição de dados qualitativos, oferecendo documentação detalhada das experiências dos participantes, feedback e resultados. Essa fase é crítica, pois consolida a pesquisa em conclusões acionáveis.

### 5.3.6 Operação

**Fase de Preparação.** Para evitar viés, os participantes não foram informados sobre as questões ou hipóteses da pesquisa. A motivação para sua participação foi ganhar pontos extras para suas notas. No entanto, os resultados do questionário não afetaram suas notas finais. Os profissionais receberam as mesmas questões em formato imprimível. Para garantir sua familiaridade com os conceitos de modelagem utilizados no estudo, todos os sujeitos foram submetidos a uma sessão de treinamento de atualização.

**Fase de Execução.** O experimento foi realizado em dois cursos em diferentes universida-



**Tabela 9:** Características das perguntas utilizadas no estudo.

Tarefas	Tipo de inconsistência	#	Diagrama de classe		Diagrama de sequência	
			Classes	Métodos	Objetos	Mensagens
1	EcM	1	4	7	4	8
2	ED e EcM	2	4	10	4	5
3	Cm e CnSD	2	6	13	4	6
4	EnN, EcM e CaSD	3	5	10	5	9
5	Sem inconsistências	0	4	9	4	6
6	Sem inconsistências	0	3	6	3	4
7	EcM	1	4	7	4	8
8	ED e EcM	2	4	10	4	5
9	Cm e CnSD	2	6	13	4	6
10	EnN, EcM e CaSD	3	5	10	5	9
11	Sem inconsistências	0	4	9	4	6
12	Sem inconsistências	0	3	6	3	4

Nota: Tarefas 1-6 não foram gamificadas, enquanto as tarefas 6-12 foram gamificadas. #: representa a quantidade de incoerências numa determinada tarefa.

des brasileiras. Embora o experimento tenha sido cuidadosamente controlado, ambas as sessões ocorreram em salas de aula sob condições típicas de exame. Cada participante recebeu 12 perguntas e fichas de respostas: cinco gamificadas, cinco não gamificadas e duas questões de controle contendo inconsistências. Se um participante respondeu incorretamente a uma questão de controle, suas respostas foram descartadas, pois isso sugere que eles poderiam estar adivinhando ao invés de fornecer insights válidos. Não houve pressão temporal durante as tarefas, porém os participantes foram supervisionados de perto para garantir o registro preciso do tempo. Estamos confiantes de que o tempo foi gravado corretamente. Para clareza, os participantes foram incentivados a justificar suas respostas. Após completar o experimento, preencheram um questionário que fornece informações sobre sua formação acadêmica e trabalho.

### 5.3.7 Tarefas Experimentais

As tarefas experimentais foram divididas em dois tipos: gamificado e não gamificado, cada um composto por seis questões, quatro com inconsistências e duas sem. As questões 1 a 6 formaram a atividade gamificada, enquanto as questões 7 a 12 constituíram a tarefa não gamificada, conforme detalhado na Tabela 9. Todos os participantes foram obrigados a completar as duas tarefas, respondendo a 12 perguntas. Para dar conta dos potenciais efeitos da ordem, os participantes foram distribuídos aleatoriamente em dois grupos: um grupo completou as tarefas gamificadas primeiro, seguido das tarefas não gamificadas, enquanto o outro grupo completou as tarefas em ordem inversa.

A Figura 14 fornece um exemplo de uma tarefa experimental projetada para ajudar os leitores a entender como as questões experimentais foram estruturadas. Cada tarefa inclui uma descrição da tarefa experimental a ser realizada, acompanhada de um diagrama de classes UML e um diagrama de sequência. Os participantes devem analisar os diagramas e selecionar uma resposta das opções fornecidas. As quatro primeiras alternativas (A, B, C e D) correspondem a interpretações incorretas de código, enquanto a quinta opção (E) indica que os diagramas são inconsistentes, que é a resposta correta.



Suppose you are a developer of a web application. Considering the class and sequence diagram, how would you implement the Cart class?

```

classDiagram
    class StoreFront {
        +confirmOrder() void
    }
    class Cart {
        -descItem : String
        +addItem(descItem : String) void
        +checkout() void
        +processOrder() void
    }
    class Inventory {
        +reserveItem() void
    }
    class Item {
        +getPrice() void
        +getId() void
    }
    StoreFront "1" -- "*" Cart : has
    Cart "1" -- "*" Inventory : - cart
    Cart "1" -- "*" Item : - cart
  
```

```

sequenceDiagram
    participant s as s: StoreFront
    participant c as c: Cart
    participant i as i: Inventory
    participant it as it: Item

    s->>c: 1 addItem(descItem:String) : void
    activate c
    c->>i: 1.1 reserveItem() : void
    activate i
    i->>it: 2 getId() : void
    activate it
    it-->>i: 
    deactivate it
    i-->>c: 
    deactivate i
    c->>s: 3 checkout() : void
    activate s
    s->>c: 3.1.1 confirmOrder() : void
    activate c
    c->>i: 3.1 processOrder() : void
    activate i
    i->>it: 4 getPrice() : void
    activate it
    it-->>i: 
    deactivate it
    i-->>c: 
    deactivate i
    c->>s: 3.2 placeItemInOrder()
    deactivate c
    s-->>s: 
    deactivate s
  
```

```

class Cart {
    void addItem(descItem) {
        i.reserveItem()
        c.processOrder()
    }

    void checkout() {
        s.confirmOrder()
        // Do Something
        i.placeItemInOrder()
    }
}

```

```

class Cart {
    void addItem(descItem) {
        i.reserveItem()
        c.processOrder()
    }

    void checkout() {
        i.confirmOrder()
        // Do Something
        s.placeItemInOrder()
    }
}

```

```

class Cart {
    void addItem(descItem) {
        i.reserveItem()
    }

    void checkout() {
        c.processOrder()
        // Do Something
        i.confirmOrder()
        s.placeItemInOrder()
    }
}

```

```

class Cart {
    void addItem(descItem) {
        i.reserveItem()
    }

    void checkout() {
        c.processOrder()
        // Do Something
        s.confirmOrder()
        i.placeItemInOrder()
    }
}

```

```

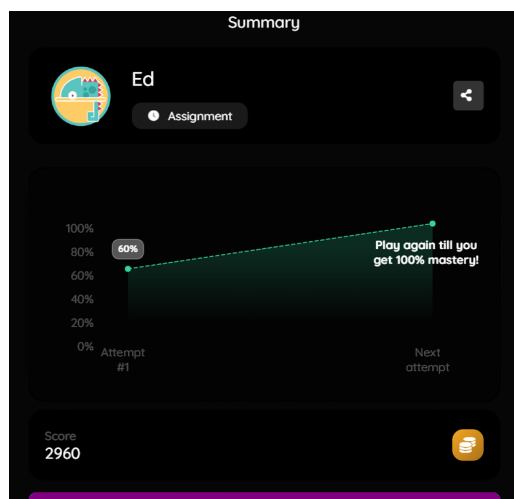
// No interpretation
// can be made
// because there
// is a problem in
// the model

```

**Figura 14:** Tela de perguntas gamificadas

**Tarefa gamificada.** Nas tarefas gamificadas, cada participante tinha até 15 minutos para responder a cada pergunta, com 90 minutos alocados para completar todas as seis questões do desafio. O quiz incorporou elementos de gamificação, como feedback imediato, pontuação, classificação e avatares para melhorar o engajamento. Cada cenário apresentou um contexto específico projetado para introduzir problemas específicos de domínio para os participantes resolverem. Além disso, os participantes puderam visualizar seus pontos totais e ranking geral, como ilustrado na Figura 15.

**Tarefas não gamificadas.** As tarefas não gamificadas seguiram a mesma estrutura e estilo que as tarefas gamificadas, garantindo consistência no desenho e apresentação das atividades experimentais. Os participantes foram apresentados com seis questões, cada uma acompanhada de uma descrição da tarefa, um diagrama de classe UML e um diagrama de sequência, exigindo que interpretassem e escolhessem a resposta mais adequada. No entanto, ao contrário das tarefas gamificadas, a versão não gamificada excluiu todos os elementos de gamificação, como feedback imediato, pontuação, classificação e avatares. Essa ausência de gamificação garantiu que os participantes realizassem as tarefas exclusivamente com base na compreensão dos diagramas e descrições das tarefas, sem os fatores motivacionais adicionais fornecidos na versão gamificada.



**Figura 15:** Tela do dashboard usado no estudo.

### 5.3.8 Pacote de Replicação

Para garantir transparência, replicabilidade e adesão aos princípios da ciência aberta, fornecemos um pacote de replicação acessível ao público no Google Colab<sup>3</sup>. Este pacote consolida todos os artefatos essenciais utilizados ao longo do estudo, permitindo aos pesquisadores reproduzir, verificar ou ampliar resultados. O pacote inclui:

- **Dados experimentais:** Conjuntos de dados brutos e pré-processados derivados do estudo, abrangendo todas as tarefas e variáveis experimentais, ou seja, taxa de detecção, esforço de detecção e taxa de interpretação incorreta;
- **Scripts Python:** Código-fonte escrito em Python para preparação de dados, análise estatística e teste de hipóteses, com base em bibliotecas SciPy e NumPy;
- **Notebook executável:** Um Google Colab Notebook integrando dados, procedimentos estatísticos e saídas visuais, permitindo replicação passo a passo em um ambiente interativo;
- **Documentação:** Descrição das estruturas de dados, definições de variáveis e instruções de execução para apoiar a validação e reutilização independentes;
- **Tabelas de resumo:** Resultados das análises quantitativas, incluindo estatística descritiva e resultados de testes utilizados nos achados relatados no estudo.

## 5.4 Resultados

Esta seção relata os resultados sobre os efeitos da gamificação no desempenho dos participantes durante tarefas de detecção de inconsistência em modelos UML. A Seção 5.4.1 aborda

<sup>3</sup>Pacote de replicação: <https://tinyurl.com/5k7nh6jf>

a QP3.1, que examina se a gamificação influencia a *taxa de detecção*, definida como a proporção de inconsistências corretamente identificadas pelos participantes. A Seção 5.4.2 explora a QP3.2, focando na *taxa de esforço de detecção*, que investiga o processo de detecção relacionando o tempo gasto com o número de detecções precisas. A Seção 5.4.3 investiga a QP3.3, que diz respeito à *taxa de interpretação*, refletindo o nível de variação nas respostas sobre quatro alternativas possíveis, ou seja, casos em que os participantes incorretamente rotularam cenários inconsistentes como consistentes. Para cada questão de pesquisa, foram analisados os dados coletados por meio de estatística descritiva para explorar a distribuição e as tendências centrais das respostas. Testamos também as hipóteses formuladas para determinar se as diferenças observadas foram estatisticamente significativas. A Tabela 10 apresenta os valores médios dos dados coletados para cada tarefa, considerando todas as variáveis de estudo analisadas no experimento. Tabela 11 mostra a análise estatística dos dados.

**Tabela 10:** Conjunto de dados coletados nas tarefas experimentais.

Método	Tarefa	Detecção	Detecção	Interpretação incorreta
		Taxa	Taxa de esforço	Taxa
Sem gamificação	Tarefa 1	0.58	3.15	0.65
	Tarefa 2	0.63	3.01	0.64
	Tarefa 3	0.55	3.21	0.53
	Tarefa 4	0.70	3.57	0.74
	Tarefa 5	0.53	3.41	0.63
	Tarefa 6	0.45	3.01	0.35
Gamificação	Tarefa 7	0.63	2.71	0.66
	Tarefa 8	0.40	2.41	0.56
	Tarefa 9	0.65	2.73	0.70
	Tarefa 10	0.78	2.83	0.76
	Tarefa 11	0.47	2.56	0.61
	Tarefa 12	0.18	2.99	0.56

*Nota:* A taxa de detecção e a taxa de interpretação incorreta são escalonadas entre 0 e 1, representando valores normalizados. O esforço de detecção é medido em minutos.

#### 5.4.1 QP3.1: Taxa de Gamificação e Detecção

*Estatísticas Descritivas.* A QP3.1 explora se a gamificação influencia a capacidade dos participantes de detectar inconsistências em modelos UML. Ao contrário da hipótese inicial, os dados sugerem que *gamificação pode prejudicar ligeiramente, ao invés de melhorar, o desempenho de detecção de inconsistência*. Como mostrado na Tabela 11, a taxa média de detecção na condição gamificada foi de 0,518, em comparação com 0,573 na configuração não gamificada, ou seja, uma redução de 9,6%. A mediana também diminuiu de 0,565 (não jogo) para 0,550 (jogo), reforçando esta tendência. Embora as diferenças sejam modestas, elas favorecem consistentemente a abordagem não gamificada. Esse achado desafia o pressuposto de que a gamificação aumenta inerentemente o desempenho da tarefa. Ressalta que o aumento do enga-

jamento nem sempre se traduz em melhores resultados. O resultado sugere que a gamificação pode introduzir distrações cognitivas ou alterar o foco de maneiras que dificultam uma análise minuciosa. Assim, enquanto a gamificação pode impulsionar a motivação, seu impacto na efetividade da tarefa requer uma cuidadosa consideração e uma investigação empírica mais aprofundada. Na etapa seguinte, examinamos se as diferenças observadas são estatisticamente significativas o suficiente para confirmar ou rejeitar a hipótese nula ( $H_1$ ).

*Teste de Hipótese.* Para avaliar se a gamificação impacta significativamente a taxa de detecção de inconsistência, realizamos testes de hipótese fundamentados nas características distribucionais dos dados. O teste de normalidade de Shapiro-Wilk indicou que os dados da taxa de detecção para as condições gamificadas e não gamificadas seguiram uma distribuição normal ( $p > 0,05$ ), justificando o uso de um teste paramétrico. Aplicou-se um teste  $t$  pareado para avaliar a diferença nas taxas de detecção entre as duas condições. Embora a análise tenha revelado maior taxa média de detecção no grupo não gamificado (média = 0,573) em comparação ao grupo gamificado (média = 0,518), a diferença não foi estatisticamente significativa ( $p = 0,779$ ). O teste não paramétrico de Wilcoxon sinal-rank apresentou resultado consistente ( $p = 0,718$ ), reforçando a conclusão de que a diferença observada carece de significância estatística. Além disso, o tamanho do efeito, medido pelo  $d$  de Cohen, foi de 0,33, indicando efeito de pequeno a médio. Embora isso sugira uma leve tendência favorável à condição não gamificada, as evidências são insuficientes para rejeitar a hipótese nula. Nas condições testadas, *a adoção de elementos de gamificação não aumentou significativamente a capacidade dos participantes de detectar inconsistências em modelos UML*. Esses achados se alinham com estudos anteriores que relatam efeitos mistos ou inconclusivos da gamificação no desempenho de tarefas em contextos de engenharia de software (LI; HEW; DU, 2024). Ao corroborar resultados anteriores, *o estudo reforça a ideia de que, embora a gamificação possa aumentar o engajamento, sua influência em tarefas cognitivas-intensivas (por exemplo, detecção de inconsistência) permanece limitada ou dependente do contexto*. Outras investigações devem explorar estratégias alternativas de gamificação, estruturas de tarefas ou adaptações específicas de domínio para entender melhor onde e como a gamificação pode ser eficaz.

**Resumo da QP3.1:** Foi realizado teste  $t$  pareado para comparar as taxas de detecção entre os modelos com e sem elementos de gamificação. Os resultados mostraram que a taxa de detecção foi menor para os modelos gamificados ( $M = 0,518$ ,  $DP = 0,214$ ) em comparação aos modelos não gamificados ( $M = 0,573$ ,  $DP = 0,086$ ), mas essa diferença não foi estatisticamente significativa,  $t(1) = 0,297$ ,  $p = 0,779$ , Cohen's  $d = 0,33$ . Isso sugere que o uso de elementos de gamificação não levou a uma melhora significativa na taxa de detecção de inconsistências em modelos UML.

**Interpretação dos resultados da QP3.1.** O resultado inesperado – que a gamificação não melhorou significativamente a taxa de detecção de inconsistências – pode ser explicado por barreiras motivacionais identificadas em estudos anteriores. Duas questões centrais provavelmente influenciaram o desempenho: a falta de competência percebida e a falta de autonomia

**Tabela 11:** Resultados dos testes estatísticos.

Variável	Método	Principal	SD	Min	25th	Med	75th	Max	% Diff	Wilcoxon p-value	Paired t-test p-value
DR	Sem gamificação	0.573	0.086	0.45	0.535	0.565	0.618	0.7	9.60%	0.718	0.779
	Game	0.518	0.214	0.18	0.418	0.55	0.645	0.78			
DE	Sem gamificação	3.227	0.224	3.01	3.045	3.18	3.36	3.57	16.18%	<b>0.015</b>	<b>0.003</b>
	Game	2.705	0.203	2.41	2.598	2.72	2.805	2.99			
MR	Sem gamificação	0.59	0.135	0.35	0.555	0.635	0.648	0.74	8.10%	0.281	0.157
	Game	0.642	0.08	0.56	0.573	0.635	0.69	0.76			

Nota: Valores de p em negrito são estatisticamente significativos (p-value < 0.05); SD = Desvio-padrão; Med = Mediana.

percebida. O trabalho anterior (FACEY-SHAW et al., 2020) relatou que esses problemas geralmente surgem quando as tarefas são mal calibradas para os níveis de habilidade dos usuários ou quando os elementos de gamificação não são claros. Isto é, tarefas que são muito fáceis ou difíceis de engajar significativamente os alunos. Se os participantes do estudo acharam as tarefas gamificadas incontestáveis ou confusas, seu foco e engajamento podem ter diminuído. A falta de familiaridade com as regras de gamificação, ou o desânimo causado por quadros públicos de liderança que destacam o subdesempenho, também pode reduzir a motivação (XI; HAMARI, 2019; ZARRAONANDIA et al., 2015). Além disso, muitas vezes os alunos sentem-se menos investidos quando lhes falta autonomia, como a escolha de como abordar uma tarefa. Estudos anteriores (XI; HAMARI, 2019; FERNANDEZ-RIO; ZUMAJO-FLORES; FLORES-AGUILAR, 2022) mostram que oferecer escolha e auto-expressão aumenta a motivação intrínseca. Se presentes no delineamento experimental, esses fatores podem ser responsáveis pela menor taxa de detecção observada na condição gamificada. Portanto, *os achados não contradizem a literatura, mas apoiam seu apelo por estratégias de gamificação mais nuances e centradas no aprendiz em contextos cognitivamente exigentes.*

#### 5.4.2 QP3.2: Taxa de Esforço de Gamificação e Detecção

*Estatísticas Descritivas.* A QP3.2 examina se a gamificação influencia o esforço cognitivo necessário para detectar inconsistências. Os resultados (Tabela 11) sugerem que *participantes usando o método gamificado demonstrou menor esforço de detecção do que aqueles usando o método não gamificado.* A taxa média de esforço de detecção caiu de 3,22 na condição não gamificada para 2,70 na condição gamificada, uma redução de 16,18%. Os valores medianos espelharam essa tendência, diminuindo de 3,18 para 2,72, refletindo uma melhora de 14,47%. Essas reduções sugerem que a gamificação pode contribuir para uma execução mais eficiente da tarefa. Uma explicação plausível é que os elementos do jogo estruturaram a atenção e fomentaram o engajamento, permitindo que os participantes focassem em pontos relevantes e aplicassem estratégias de detecção adequadas. Esses resultados sustentam a hipótese de que ambientes gamificados podem potencializar a economia cognitiva, diminuindo o esforço externo, potencialmente tornando a detecção de inconsistência em modelos UML mais gerenciável e menos tributável cognitivamente. Entretanto, *este ganho observado em eficiência não se traduziu em*

*uma maior taxa de detecção, sugerindo uma complexa relação entre esforço e desempenho.* Esta desconexão convida a uma nova investigação: Pode o reduzido esforço cognitivo coexistir com desfechos não melhorados, e em que condições pode surgir tal paradoxo? Este achado contraintuitivo abre uma linha original de investigação sobre como a gamificação forma o que os alunos fazem e como eles alocam e regulam os recursos cognitivos durante a resolução de problemas.

*Teste de Hipótese.* Foi realizado um teste de hipótese para avaliar a significância estatística. O teste de normalidade de Shapiro-Wilk confirmou que os dados atenderam à suposição de normalidade ( $p > 0,05$ ), justificando o uso de uma análise paramétrica. Um teste t pareado revelou *uma redução estatisticamente significativa no esforço quando a gamificação foi aplicada* ( $t = -7,12$ ,  $p = 0,003$ ). Esse achado foi ainda corroborado por um teste não paramétrico de Wilcoxon ( $p = 0,015$ ), dando robustez ao resultado. O  $d$  de Cohen calculado de 2,44 indica um tamanho de efeito muito grande, sugerindo que a gamificação reduz substancialmente os recursos cognitivos necessários para detectar inconsistências em modelos UML. Este resultado inicial apoia a afirmação de que a gamificação pode aumentar a eficiência cognitiva em tarefas de detecção de inconsistência. Estudos anteriores (BUCCHIARONE et al., 2023a; PAULA PORTO et al., 2021; MARÍN et al., 2018; PARIZI, 2016; PRAUSE; NONNEN; VINKOVITS, 2012) relataram de forma semelhante que mecanismos de gamificação podem diminuir a dificuldade de percepção da tarefa e reduzir a fadiga mental durante as atividades de engenharia de software. O estudo corrobora esses achados e os amplia ao demonstrar que tais benefícios persistem mesmo em tarefas baseadas em modelos como a detecção de inconsistência. Além disso, quando considerado ao lado do achado anterior de que as taxas de detecção não aumentaram significativamente sob gamificação, essa notável redução no esforço levanta uma importante questão teórica: Em que condições a gamificação reduz simultaneamente o esforço e mantém ou melhora a precisão? Isso pode abrir uma nova avenida de pesquisa sobre como carga cognitiva, design motivacional e resultados de desempenho interagem em contextos de engenharia orientados por modelos.

**Resumo da QP3.2:** Foi realizado um teste t pareado para comparar as taxas de esforço de detecção entre os modelos com e sem elementos de gamificação. Os resultados indicaram que a taxa de esforço de detecção foi significativamente menor para os modelos gamificados ( $M = 2,705$ ,  $DP = 0,203$ ) em comparação com os modelos não gamificados ( $M = 3,227$ ,  $DP = 0,224$ ),  $t(5) = -7,12$ ,  $p = 0,003$ , Cohen's  $d = 2,44$ . Este grande tamanho de efeito sugere que a introdução de elementos de gamificação reduziu grandemente o esforço necessário para detectar inconsistências em modelos UML.

**Interpretação dos resultados do QP3.2.** A redução do esforço de detecção observada no estudo pode ser parcialmente explicada pela dinâmica motivacional promovida pela gamificação. Conforme postulado pela Teoria da Autodeterminação de Deci e Ryan (DECI; RYAN, 2004, 2000), a motivação intrínseca (ou seja, o engajamento em uma atividade para sua satisfação inerente) desempenha um papel central na sustentação do esforço cognitivo e aprendizagem

profunda. Ambientes gamificados, por design, muitas vezes introduzem elementos como desafios, feedback e progressão de objetivos que podem estimular esta unidade intrínseca (RYAN; DECI, 2020). Quando os participantes percebem a tarefa como agradável ou significativa, eles são mais propensos a persistir com atenção focada e diminuição da fadiga cognitiva, resultando em menor esforço relatado (CERASOLI; NICKLIN; FORD, 2014; VANSTEENKISTE; LENS; DECI, 2006). Isso contrasta com comportamentos extrinsecamente motivados, que, embora potencialmente efetivos em curto prazo, estão associados ao engajamento em nível de superfície e maior percepção da dificuldade de tarefa (CLANTON HARPINE; CLANTON HARPINE, 2015; LEE et al., 2010). Os achados sugerem que *os elementos de gamificação podem ter aumentado a motivação intrínseca dos participantes, reduzindo assim a carga de trabalho mental necessária para detectar inconsistências*. Essa interpretação se alinha com pesquisas anteriores demonstrando que os aprendizes intrinsecamente motivados são mais resilientes, eficientes e mentalmente ágeis ao resolver tarefas cognitivamente exigentes (RYAN; DECI, 2000a; DECI; RYAN, 2008). Assim, *a diminuição observada no esforço de detecção não é apenas um resultado comportamental, mas pode refletir uma mudança motivacional mais profunda induzida pela configuração gamificada*.

#### 5.4.3 QP3.3: Gamificação e Taxa de má interpretação

**Estatísticas Descritivas.** A QP3.3 investiga se a gamificação afeta a taxa de interpretação incorreta (RM). Os scores de RM variam de 0 a 1, onde 1 denota interpretação perfeita, e 0 reflete um completo mal-entendido, ou seja, seleção uniforme entre alternativas. A média da RM para modelos não gamificados foi de 0,59, enquanto que os modelos gamificados alcançaram uma média ligeiramente maior de 0,64, um aumento de 8,10%. Apesar dessa aparente melhora, os valores medianos foram idênticos (0,635), sugerindo uma mudança limitada na tendência central. Essas figuras sugerem um aprimoramento marginal na precisão interpretativa devido à gamificação. Contudo, dada a estreita diferença, deve-se questionar se o efeito é praticamente significativo ou estatisticamente significativo. A expectativa inicial era de que a gamificação levaria a uma redução mensurável da taxa de interpretação incorreta, reforçando a ideia de que elementos envolventes poderiam potencializar o foco e a compreensão dos participantes. No entanto, essa expectativa não foi totalmente apoiada. Embora os modelos gamificados tenham apresentado uma RM média ligeiramente maior, a magnitude da melhora foi marginal, e as medianas idênticas indicam uma mudança negligenciável na distribuição geral. Isso sugere que *incluindo elementos de gamificação por si só não reduz substancialmente a interpretação errada*. Em vez disso, os participantes que trabalham com modelos gamificados ainda podem lutar para alcançar uma compreensão significativamente mais clara, levando a resultados comparáveis aos observados em configurações não gamificadas.

**Teste de Hipótese.** Para examinar o impacto da gamificação na taxa de interpretação incorreta, aplicamos um teste t pareado. Os testes de Shapiro-Wilk confirmaram a normalidade das



distribuições gamificadas e não gamificadas ( $p = 0,322$  e  $p = 0,545$ , respectivamente), justificando o uso da análise paramétrica. O teste  $t$  obteve  $p$ -valor de  $0,157$  ( $p > 0,05$ ), fornecendo evidências insuficientes para rejeitar a hipótese nula. O tamanho do efeito, medido pelo  $d$  de Cohen =  $-0,4649$ , aponta para uma magnitude pequena a média, com o sinal negativo indicando que o grupo não gamificado superou ligeiramente o gamificado em relação à menor interpretação incorreta.

Este resultado sugere que, nas condições testadas, *a adoção de elementos de gamificação não leva a uma melhoria estatisticamente significativa na redução de interpretações erradas*. Esses achados são reforçados pelo teste não paramétrico de Wilcoxon ( $p = 0,281$ ), que também não indica diferença significativa. Assim, a hipótese nula ( $H_{3_{null}}$ ) não pode ser rejeitada. Os achados se alinham a estudos prévios que têm suscitado preocupações quanto à eficácia inconsistente da gamificação no aprimoramento dos resultados cognitivos, particularmente em tarefas envolvendo compreensão e interpretação (GARACCIONE et al., 2025b; ZENG et al., 2024; LI; MA; SHI, 2023). Esses estudos sugerem que, embora a gamificação possa melhorar o engajamento, seus benefícios não se traduzem consistentemente em melhor precisão ou compreensão em tarefas complexas de tomada de decisão. O estudo corrobora essas observações e chama a atenção para a importância de fatores contextuais e específicos para determinar a eficácia de intervenções gamificadas.

**Resumo da QP3.3:** Foi realizado um teste  $t$  pareado para comparar as taxas de erro de interpretação entre modelos com e sem elementos de gamificação. Os resultados indicaram que a taxa de erro de interpretação foi ligeiramente maior para os modelos gamificados ( $M = 0,642$ ,  $DP = 0,080$ ) em comparação com os modelos não gamificados ( $M = 0,590$ ,  $DP = 0,135$ ). Entretanto, essa diferença não foi estatisticamente significativa,  $t(1) = -1,125$ ,  $p = 0,157$  (uma cauda). O tamanho do efeito, medido pelo  $d$  de Cohen ( $d = -0,4649$ ), sugere um efeito pequeno a médio, indicando que a gamificação aumentou ligeiramente as taxas de interpretação incorreta, embora a significância prática desse achado seja limitada dada a falta de significância estatística. Esse resultado sugere que a incorporação de elementos de gamificação não reduziu significativamente as taxas de interpretação incorreta na detecção de inconsistências em modelos UML.

**Interpretação dos Resultados da QP3.3.** A ausência de melhora estatisticamente significativa na taxa de erro de interpretação pode estar nas demandas cognitivo-emocionais duplas impostas por ambientes gamificados. Embora a gamificação tenha como objetivo aumentar o engajamento e a motivação dos usuários, estudos têm demonstrado que esse aumento no engajamento não necessariamente se traduz em melhor desempenho cognitivo, especialmente em tarefas que requerem alta precisão, como identificar inconsistências em modelos de software. A integração dos elementos do jogo pode introduzir estímulos adicionais que, embora motivacionais, competem por recursos cognitivos essenciais ao raciocínio analítico e à tomada de decisão focada (HAMARI; HASSAN; DIAS, 2018). Essa sobrecarga cognitiva pode interferir na capacidade dos participantes de interpretar com precisão a semântica do modelo, levando



ao aumento da interpretação incorreta ou, na melhor das hipóteses, a efeitos neutros sobre a acurácia.

Além disso, a mecânica de gamificação mal alinhada (ou seja, aqueles que enfatizam a velocidade ou a conclusão da recompensa sobre a compreensão profunda) pode inadvertidamente incentivar estratégias de processamento superficial (LANDERS, 2019), diminuindo o investimento atencional dos usuários na compreensão de inconsistências de grãos finos. Os achados iniciais se alinham a esse corpo de evidências, sugerindo que *enquanto a gamificação pode alterar a experiência da tarefa, ela não aumenta inerentemente, e pode mesmo dificultar, a acurácia interpretativa quando a tarefa demanda o processamento cognitivo focado em alto esforço*. Esses resultados ecoam as preocupações levantadas por (LUMSDEN et al., 2016), que enfatizaram a necessidade de um desenho pensativo dos sistemas gamificados para evitar minar os objetivos cognitivos que pretendem apoiar.

## 5.5 Discussão

Esta seção discute as implicações dos achados, interpreta os resultados à luz de trabalhos anteriores e destaca limitações e direções para futuras pesquisas. Tabela 12 e Tabela 13 procuram relacionar os achados empíricos e aprendizado com implicações para pesquisa e desenho de ferramentas.

### 5.5.1 Usando a Teoria da Autodeterminação

Considerando a QP3.1, os achados podem ser melhor compreendidos por meio da lente da Teoria da Autodeterminação (RYAN; DECI, 2020), que identifica autonomia, competência e parentesco como necessidades psicológicas essenciais para fomentar a motivação intrínseca. Embora a gamificação seja muitas vezes assumida para apoiar essas necessidades, o projeto experimental pode ter as prejudicado involuntariamente. Por exemplo, se as tarefas na condição gamificada fossem percebidas como demasiado complicadas ou insuficientemente explicadas, o senso de competência dos participantes poderia ter sido diminuído. Da mesma forma, os leaderboards públicos podem ter reduzido a percepção de autonomia e aumentado a ansiedade entre indivíduos com desempenho inferior, frustrando seu impulso intrínseco. Estes efeitos são bem documentados na literatura de gamificação educacional (por exemplo, (XI; HAMARI, 2019)). Diante dessas interpretações, as menores taxas de detecção observadas refletem não uma falha de gamificação, mas um desalinhamento entre as características gamificadas e as necessidades psicológicas centrais à teoria da autodeterminação. Os projetos futuros devem visar explicitamente satisfazer autonomia, competência e parentesco, especialmente em tarefas cognitivas exigentes como a detecção de inconsistência.

Para investigar ainda mais essas explicações motivacionais, estudos futuros devem operacionalizar os componentes centrais da Teoria da Autodeterminação (TAD) (RYAN; DECI, 2020),

Tabela 12: Resultados empíricos e suas implicações na pesquisa

Achado empírico	Implicação para a Pesquisa	Referências de Apoio
Redução significativa no esforço percebido (p = 0.022)	Investigar como mecânicas específicas de jogo (ex.: feedback, recompensas) reduzem a carga de trabalho percebida em tarefas de MBE	(LI; HEW; DU 2024), (XU; XU; XING 2024), (LUMSDEN et al. 2016), (ZENG et al. 2024), (MULLINS; SABHERWAL 2020)
Nenhuma melhora significativa na taxa de detecção (p = 0.094)	Examinar como a gamificação impacta a alocação de atenção e a sensibilidade a erros em tarefas complexas de modelagem	(GARACCIONE et al. 2025b), (BUCCHIARONE et al. 2023b), (BUCCHIARONE et al. 2023a), (LANDERS 2019), (LUMSDEN et al. 2016)
Nenhum efeito significativo na taxa de má interpretação (p = 0.264)	Estudar a carga cognitiva e os padrões de má interpretação em contextos gamificados vs. não gamificados	(LI et al. 2025), (ROMEO et al. 2025), (MULLINS; SABHERWAL 2020), (LUMSDEN et al. 2016), (PINTRICH 2000)
Evidência de benefícios motivacionais, mas sem ganho na precisão das tarefas	Distinguir entre engajamento e resultados de aprendizagem; avaliar estratégias de aprendizagem profunda vs. superficial	(LI; HEW; DU 2024), (LI; MA; SHI 2023), (BEKK et al. 2022), (PINTRICH 2000)
Potencial para engajamento superficial via pontos e recompensas	Desenvolver modelos de gamificação adaptativos que se alinhem à expertise e ao esforço cognitivo	(HAMARI; HASSAN; DIAS 2018), (BUCCHIARONE et al. 2023a), (KLARE et al. 2021), (KIENZLE et al. 2024)
Risco de desalinhamento entre mecânicas de jogo e modelagem semântica	Projetar mecânicas de jogo sensíveis ao contexto que incentivem precisão semântica e profundidade de modelagem	(BUCCHIARONE et al. 2023b), (BUCCHIARONE et al. 2023a), (GARACCIONE et al. 2025a), (CAGNAZZO et al. 2023)

ou seja, autonomia, competência e relação, dentro de um projeto experimental mais robusto e realista (RYAN; DECI, 2020; LI; HEW; DU, 2024). A autonomia pode ser suportada dando aos alunos escolhas significativas, como selecionar o nível de dificuldade das tarefas, personalizar elementos de gamificação (por exemplo, avatares ou recompensas), ou escolher entre diferentes desafios de detecção de inconsistência. A competência pode ser aumentada através do alinhamento da dificuldade de tarefa com os níveis de habilidade dos participantes usando mecanismos adaptativos ou estruturas de desafio progressivo e oferecendo andaimes através de feedback construtivo. Por exemplo, as tarefas podem ser organizadas em níveis crescentes de complexidade (mais controlados dos que os que foram realizados), desde inconsistências sintáticas até semânticas, com orientações de desempenho-contingente fornecidas ao longo do caminho.

Relacionamento pode ser incentivado pela implementação de modos de jogo baseados em equipe ou elementos narrativos que situam os alunos dentro de um contexto colaborativo de resolução de problemas, afastando o foco da competição individual. Os leaderboards relativos ou objetivos de equipe compartilhados podem reforçar a conexão social, minimizando o impacto negativo da comparação social. Ao manipular sistematicamente essas variáveis, investigações empíricas futuras poderão elucidar se a ausência de efeitos significativos observada neste estudo decorre de um desenho de gamificação pouco eficaz, e não de uma possível ineficácia da gamificação em si. Tal delineamento também permitiria uma compreensão mais refinada das interações entre recursos motivacionais e demandas cognitivas em tarefas complexas de engenharia de software, como a detecção de inconsistências.

### 5.5.2 Equilibrando Engajamento e Precisão

Este estudo fornece informações úteis sobre os efeitos da gamificação sobre o esforço investido pelos participantes para interpretar e entender modelos UML. Os resultados demonstram um impacto particular: *enquanto a gamificação melhorou a compreensão dos participantes dos modelos UML em 8,10% e reduziu o esforço necessário para interpretá-los em 16.18%, reduziu simultaneamente a taxa de correção das respostas em 9,60%*. Essa dicotomia ressalta a necessidade de equilibrar cuidadosamente os benefícios da gamificação com seu potencial de diminuir a precisão da tarefa. Técnicas de gamificação, como pontos, rankings, feedback e crachás, ajudaram a envolver os participantes e aliviar a carga cognitiva. No entanto, eles podem desviar o foco da tarefa primária de modelagem precisa e análise de contexto. Foi analisado que os participantes podem se distrair excessivamente por recompensas externas, como pontuação e classificação, muitas vezes em detrimento de compreensão mais profunda e pensamento crítico. Com base em estudo inicial, esse desalinhamento entre os objetivos da gamificação e os objetivos da modelagem eficaz de software pode evidenciar uma limitação crítica dessas técnicas em atividades de alto risco ou cognitivamente exigentes.

Uma análise do comportamento dos participantes poderia revelar que os elementos gami-

ficados, particularmente a imediatismo de feedback e pontuação visível, criaram um senso de urgência. Embora essa urgência tenha motivado os participantes a completar rapidamente as tarefas, também reduziu o tempo gasto refletindo sobre as complexidades das inconsistências utilizadas no estudo. Esse fenômeno se alinha aos achados da psicologia educacional e cognitiva, onde a dependência excessiva de recompensas pode prejudicar a motivação intrínseca e levar ao engajamento superficial da tarefa. Estudos anteriores (CHEN, 2023, 2017) já relataram que a dependência excessiva em recompensas pode levar a uma diminuição de longo prazo na motivação dos estudantes devido ao efeito de superjustificação, onde a motivação intrínseca diminui à medida que os estudantes associam seu interesse em atividades com recompensas externas, em vez de satisfação pessoal ou engajamento. Assim, o foco excessivo em elementos gamificados pareceu dificultar a capacidade dos participantes de interpretar cenários complexos com precisão, sugerindo que a gamificação pode não ser adequada para tarefas que requerem maior aprofundamento.

Curiosamente, as tarefas não gamificadas provocaram maiores níveis de engajamento e concentração, pois os participantes estavam menos distraídos com estímulos externos. Sem gamificação, os participantes demonstraram uma compreensão mais clara dos contextos apresentados, o que levou a uma maior taxa de acertos. No entanto, isso veio ao custo do aumento do esforço cognitivo, pois os participantes necessitavam de orientações mais estruturadas fornecidas por elementos gamificados. Esse achado sugere que, embora a gamificação possa reduzir o esforço, pode comprometer a profundidade e a precisão do desempenho da tarefa. Os resultados também apontam para um desafio mais amplo em ambientes de aprendizagem e trabalho gamificados: *quando a gamificação recompensa realizações triviais em vez de fomentar a criatividade e resolução de problemas, ela corre o risco de reforçar o engajamento em nível de superfície em vez de aprendizagem mais profunda*. Para tarefas de modelagem de software, que requerem pensamento estratégico e interpretação precisa de requisitos e problemas de domínio, as potenciais desvantagens da gamificação tornam-se mais pronunciadas.

Resumindo, *enquanto a gamificação promete melhorar o engajamento e reduzir o esforço cognitivo, sua aplicação no domínio da modelagem multivisão UML deve ser abordada com cautela*. O excesso de ênfase em recompensas e elementos competitivos pode diminuir a qualidade dos resultados, particularmente em tarefas que exigem pensamento crítico e precisão. Esses achados enfatizam a importância de projetar sistemas gamificados que equilibrem o engajamento e a precisão em tarefas de desenvolvimento baseadas em modelos, promovendo uma compreensão mais profunda das tarefas específicas de domínio sem sacrificar o desempenho.

### 5.5.3 Sobre o Impacto da Gamificação nas Taxas de Má Interpretação

Os achados indicam que a gamificação não reduziu significativamente a taxa de erro de interpretação em relação aos modelos não gamificados. Embora tenha sido observada melhora marginal na estatística descritiva, o teste t pareado e o teste de Wilcoxon confirmaram que es-

**Tabela 13:** Achados empíricos e implicações para pesquisa e para o design de ferramentas.

Achado Empírico	Implicação para a Pesquisa	Implicação para o Design de Ferramentas
Os benefícios da gamificação variam conforme a complexidade da tarefa e a experiência do usuário.	Explorar estratégias de gamificação diferenciadas que associem elementos de jogo à carga cognitiva dos sub-tarefas (ex.: feedback para questões sintáticas vs. mecanismos baseados em narrativa para questões semânticas).	Implementar componentes de gamificação modulares e adaptáveis a tipos de tarefa e perfis de usuário.
Usuários podem sofrer sobrecarga cognitiva devido a mecanismos de pontuação em ambientes gamificados.	Investigar abordagens de gamificação adaptativa que ajustem estímulos com base no comportamento e na carga cognitiva do usuário.	Projetar sistemas de pontuação e feedback que se adaptem em tempo real para evitar sobrecarregar os usuários, especialmente iniciantes.
O estado cognitivo (ex.: fadiga, atenção) afeta o desempenho durante tarefas de modelagem.	Utilizar dados biométricos (ex.: microestados de EEG) para estudar e modelar a influência dos estados cognitivos durante atividades de design.	Desenvolver ferramentas de modelagem sensíveis a EEG que ajustem a intensidade do feedback ou a dificuldade da tarefa em tempo real com base no estado do usuário.
Usuários exibem padrões estruturados de interação ao resolver inconsistências.	Classificar e modelar esses padrões de interação para compreender melhor a cognição e o esforço durante correções de modelos.	Instrumentar ambientes de modelagem para detectar esses padrões e acionar assistência ou estímulos motivacionais conforme necessário.
O engajamento com a gamificação é distinto dos padrões de resolução analítica de problemas.	Examinar como diferentes tipos de interação influenciam os resultados de modelagem e a experiência do usuário.	Projetar interfaces que distingam interações motivacionais de fluxos de trabalho analíticos para personalizar o suporte.

Nota: EEG = eletroencefalograma; a carga cognitiva e os padrões de interação são inferidos por meio de observação e sinais biométricos.

sas diferenças não foram estatisticamente significativas. Esse desfecho se alinha com pesquisas anteriores sugerindo que a efetividade da gamificação não é universalmente garantida, e pode depender de como os indivíduos se engajam cognitivamente e emocionalmente com os elementos do jogo (HAMARI; HASSAN; DIAS, 2018; LANDERS, 2019). Uma possível explicação é que a carga cognitiva introduzida pela mecânica do jogo pode desviar a atenção de informações relevantes para a tarefa, particularmente quando os indivíduos estão extrinsecamente motivados ou não estão familiarizados com o domínio (LUMSDEN et al., 2016). Estudos recentes defendem uma compreensão mais fina da gamificação através de uma lente cognitivo-emocional, destacando que diferenças individuais na orientação de objetivos podem moderar a eficácia de ambientes gamificados (PINTRICH, 2000). Por exemplo, indivíduos com uma forte orientação do objetivo de desempenho podem experimentar frustração ou desengajamento quando expostos a feedback negativo dentro de tarefas gamificadas, enquanto aqueles com uma orientação do objetivo de domínio podem responder com maior persistência e esforço cognitivo (MULLINS; SABHERWAL, 2020). Essas dinâmicas motivacionais podem ter influenciado a consistência das interpretações entre os participantes do estudo, atenuando os benefícios esperados da gamificação. O trabalho futuro deve considerar a incorporação de variáveis de nível individual – como orientação motivacional, estilos cognitivos e familiaridade com gamificação – como potenciais moderadores para entender melhor como e quando a gamificação impacta a precisão da compreensão do modelo.

#### 5.5.4 Implicações para a Pesquisa sobre Gamificação em Engenharia Baseada em Modelos

Este estudo avança na compreensão de como a gamificação influencia as tarefas baseadas em modelos, examinando empiricamente seus efeitos sobre o esforço cognitivo, a precisão da tarefa e a interpretação. Os resultados apontam implicações para a comunidade de pesquisa de Engenharia Baseada em Modelos (MBE), particularmente para aqueles que trabalham na interseção de gamificação, modelagem de software e esforço cognitivo em ambientes educacionais e apoiados por ferramentas.

**Gamificação pode reduzir o custo cognitivo, mas nem sempre melhorar a precisão.** Como relatado anteriormente (Seção 5.4.2), a gamificação pode reduzir o esforço necessário para detectar inconsistências, como suportado por uma diminuição estatisticamente significativa no esforço percebido ( $p = 0,022$ ). Esse achado está alinhado com evidências meta-analíticas prévias indicando que elementos de jogo, ou seja, particularmente aqueles que visam construtos motivacionais como autonomia e engajamento, podem diminuir a dificuldade de percepção de tarefa e carga cognitiva (por exemplo, (LI; HEW; DU, 2024; XU; XU; XING, 2024; ZENG et al., 2024)). No domínio do MBE, isso sugere que a gamificação pode servir como um mecanismo para melhorar a experiência do usuário sem sacrificar a conclusão da tarefa. Ferramentas como Umlegend (CAGNAZZO et al., 2023) e PapyGame (BUCCHIARONE et al., 2023a,b) exemplificam tais projetos, onde recompensas, feedback e rastreamento de progresso podem melhorar o engajamento.

No entanto, os dados também revelam que a gamificação não produziu melhorias estatisticamente significativas na taxa de detecção de inconsistência ( $p = 0,094$ ) ou reduções na taxa de interpretação incorreta ( $p = 0,264$ ). Estes resultados reforçam a necessidade de confundir pontos de vista excessivamente otimistas sobre a gamificação em tarefas técnicas. Embora a motivação aumentada possa facilitar a persistência, ela não traduz automaticamente para melhor precisão ou compreensão mais profunda (GARACCIONE et al., 2025b; LI; MA; SHI, 2023; LANDERS, 2019). Isso reflete a cautela de Landers (LANDERS, 2019) que *"mal executada a gamificação pode obscurecer seu verdadeiro valor criando engajamento sem aprender"*.

Os pesquisadores devem reconhecer que, embora a gamificação possa efetivamente reduzir o custo cognitivo percebido da detecção de inconsistência em tarefas de desenvolvimento baseadas em modelos, ela não necessariamente aumenta a precisão ou profundidade de interpretação. Pesquisas futuras devem focar em desestabilizar ganhos motivacionais de real melhoria cognitiva para projetar estratégias de gamificação que vão além do engajamento e apoiar precisão semântica na modelagem. Para resumir esta implicação, *gamificação pode envolver mas nem sempre enriquecer*.

**Motivação sem domínio: Riscos de envolvimento superficial.** A gamificação pode promover um comportamento orientado para objetivos, mas nem todos os objetivos levam ao domínio. De acordo com Pintrich (PINTRICH, 2000), quando os alunos focam em objetivos

extrínsecos (como coletar pontos ou bater pares), eles muitas vezes “*engage em processamento de nível de superfície e evitar tarefas complexas.*” Pesquisas mostram que quando os alunos são motivados externamente (por exemplo, por pontos, emblemas, ou leaderboards), eles podem adotar estratégias superficiais para “vencer o jogo” em vez de envolver profundamente com a semântica dos diagramas UML (PINTRICH, 2000; MARÍN et al., 2018; MULLINS; SABHERWAL, 2020). Essa compensação cognitivo-emocional tem sido observada em ambientes de programação gamificados (TASADDUQ et al., 2021), e também parece estender-se à modelagem conceitual.

No contexto da modelagem UML, tais estratégias podem incentivar inspeções de diagramas rápidas, mas rasas, que perdem inconsistências sutis. Isto é particularmente preocupante em contextos educacionais onde os alunos ainda estão adquirindo a profundidade semântica necessária para interpretar relações estruturais e semânticas em diagramas UML (LI et al., 2025; ROMEO et al., 2025). Os resultados se alinham a essa preocupação. Os participantes podem ter se sentido mais “em controle” ou “motivado”, mas esses sentimentos não produziram melhorias significativas na precisão analítica. Lumsden et al. (LUMSDEN et al., 2016) encontraram padrões semelhantes em estudos de avaliação cognitiva, onde intervenções gamificadas aumentaram as taxas de conclusão, mas nem sempre melhoraram a acurácia. Portanto, os pesquisadores devem distinguir entre esforço sentido e cognição efetiva. Os esforços podem diminuir, mas sem ganhos de compreensão ou identificação correta, o benefício pedagógico é limitado.

Acreditasse que a comunidade de pesquisa deve ser cautelosa com as estratégias de gamificação que enfatizam recompensas extrínsecas, como pontos, emblemas e leaderboards. Isso pode incentivar os usuários a se concentrarem na realização da tarefa e não na correção semântica, levando ao engajamento superficial e a uma maior probabilidade de ignorar inconsistências sutis. Futuras investigações devem explorar projetos de gamificação que a compreensão de modelos de andaimes, como feedback adaptativo baseado em erros semânticos, pistas de visualização que destacam violações de dependência ou missões que exigem resolução de inconsistências de visão cruzada. Esses mecanismos podem promover um engajamento cognitivo mais profundo com a semântica do modelo e a correção estrutural, o que é fundamental para o desenvolvimento eficaz baseado em modelos e detecção precisa de inconsistência.

**Uma chamada para gamificação adaptativa e cognitiva-consciente em ferramentas MBE.** Uma terceira implicação da pesquisa é a necessidade de estratégias de gamificação mais direcionadas em ferramentas de modelagem. Os achados atuais sugerem que nem todas as tarefas se beneficiam igualmente da gamificação e que seu sucesso depende de integração pensativa com resultados de aprendizagem específicos do domínio. Os pesquisadores devem explorar diferentes estratégias de gamificação, onde elementos de jogo são mapeados para a complexidade cognitiva das subtarefas, por exemplo, usando feedback para questões sintáticas e progressão baseada em narrativas para inconsistências semânticas. O trabalho anterior já traz ações nessa direção (por exemplo, (GARACCIONE; COPPOLA; ARDITO, 2024; GARACCIONE et al., 2025a)). Para ir além do engajamento, os pesquisadores devem projetar estratégias de gamifi-



cação que se adaptem à expertise do usuário e carga de trabalho cognitiva. Um estudo sobre ambientes MBE gamificados (BUCCHIARONE et al., 2023b) mostrou que cenários de jogo cuidadosamente projetados ajudaram os alunos “*manter-se engajado sem perder a correção da modelagem.*” No entanto, mesmo nesse contexto, alguns alunos foram sobrecarregados por mecanismos de pontuação, particularmente quando não tinham conhecimento prévio de modelagem.

Além disso, os resultados exigem pesquisas sobre abordagens de gamificação adaptativa que adaptem estímulos baseados no comportamento ou expertise do usuário em tempo real, conforme proposto na literatura sobre tecnologias motivacionais (HAMARI; HASSAN; DIAS, 2018). Isso sugere que a gamificação adaptativa, ou seja, a adaptação da mecânica do jogo baseada no desempenho do usuário e complexidade da tarefa, pode oferecer um tradeoff melhor. Hamari et al. (HAMARI; HASSAN; DIAS, 2018) argumentam que a gamificação é mais eficaz quando “*alinhada com objetivos do usuário, necessidades psicológicas e características da tarefa.*”

**Para ambientes de modelagem de microestado de EEG.** Os achados sugerem duas implicações fundamentais para as ferramentas de modelagem UML: desenvolver ambientes de modelagem com conhecimento de microestado de EEG e identificar padrões de interação de desencadeamento de adaptação em interações gamificadas e centradas em tarefas. Primeiro, a integração de dados biométricos (MENZEN; FARIAS; BISCHOFF, 2021), particularmente microestados de EEG (NAGABHUSHAN KALBURGI et al., 2024), pode permitir inferências em tempo real sobre os estados cognitivos dos usuários, como atenção sustentada ou fadiga mental. Com base em estudos prévios que abordam ambientes de modelagem cognitivo-consciente (KLARE et al., 2021; KIENZLE et al., 2024), tais ferramentas poderiam ajustar dinamicamente a interface gamificada, por exemplo, modulando intensidade de feedback ou nível de desafio baseado em sinais neurofisiológicos. Isso personalizaria a experiência de modelagem, reduzindo a sobrecarga cognitiva, preservando o engajamento.

Segundo, padrões de interação em ferramentas de modelagem podem ser classificados ao longo de dois eixos: padrões emergentes do engajamento do usuário com elementos de gamificação e aqueles observados durante o processo de detecção de inconsistência do núcleo. Enquanto o primeiro pode desencadear pistas motivacionais adaptativas, este último revela a estrutura cognitiva de como os usuários navegam e resolvem inconsistências de modelagem. Por exemplo, em ambientes de desenvolvimento baseados em modelos como Papyrus<sup>4</sup> ou Enterprise Architect<sup>5</sup>, os usuários geralmente seguem uma sequência recorrente ao abordar inconsistências de diagramas de classe: (1) selecionando um elemento de modelo inconsistente (por exemplo, uma classe faltando um atributo necessário); (2) navegando para elementos relacionados através de links de dependência ou visualizações de rastreabilidade; (3) editando o elemento para resolver a inconsistência; e (4) revalidando o modelo usando mecanismos de verificação

<sup>4</sup>Eclipse Papyrus: <https://eclipse.dev/papyrus/>

<sup>5</sup>Enterprise Architect: <https://sparxsystems.com/>



suportados por ferramentas. Esse padrão ilustra uma alça cognitivamente exigente de percepção, raciocínio e verificação, muitas vezes exigindo transições fluidas entre a estrutura sintática e a correção semântica ([KIENZLE et al., 2024; KLARE et al., 2021]). Compreender e apoiar ambos os tipos de padrões (i.e., motivacional e analítico) são essenciais para projetar ambientes de modelagem adaptativos e cognitivamente sintonizados.

Finalmente, a gamificação em ferramentas de modelagem UML deve ser adaptada à complexidade da tarefa e à expertise do usuário, com estratégias adaptativas que alinham os elementos do jogo às demandas cognitivas e ao desempenho em tempo real. Além disso, identificar padrões de interação motivacionais e analíticos por meio de pistas de engajamento e comportamentos de resolução de inconsistência pode informar o projeto de ambientes de modelagem cognitivos e adaptativos EEG que equilibram o engajamento com precisão.

### 5.5.5 Desafios e Direções para Pesquisas Futuras

Esta seção descreve os principais desafios e questões abertas considerando o uso da gamificação na modelagem de software. A Tabela 14 apresenta uma agenda de pesquisa que consolida esses desafios e propõe direções para futuras investigações.

**Documentação incorreta no desenvolvimento de software.** A documentação continua sendo uma das práticas mais negligenciadas no desenvolvimento de software, muitas vezes percebidas pelos desenvolvedores como tediosas, demoradas, e oferecendo pouco valor imediato ([BOOCH, 2025; ROMEO et al., 2025]). Essa negligência levou estudos recentes a explorar a gamificação como estratégia motivacional para estimular tarefas de documentação, particularmente sob a forma de comentários de código e matrizes de rastreabilidade entre casos de código e teste ([PARIZI, 2016; PARIZI; KASEM; ABDULLAH, 2015; SUKALE; PFAFF, 2014; PRAUSE; NONNEN; VINKOVITS, 2012]). No entanto, enquanto a gamificação tem mostrado promessa em apoiar atividades de codificação, como depuração, criação de casos de teste, e comentando ([BUCCHIARONE et al., 2023b; CAGNAZZO et al., 2023; MARÍN et al., 2018]), seu potencial permanece subexplorado em outras atividades de desenvolvimento crítico.

**Gamificando atividades de fase de projeto.** Outro desafio de pesquisa reside na gamificação de atividades de fase de projeto, nomeadamente a modelagem de software, documentação arquitetônica, o uso de padrões de projeto e a rastreabilidade das decisões de projeto. Apesar da centralidade da UML em engenharia orientada por modelos ([LI et al., 2025; ROMEO et al., 2025; JACOBSON; BOOCH, 2021; UNHELKAR, 2017]), tarefas como manter a consistência entre diagramas UML (por exemplo, diagramas de classe e sequência) e documentar a racionalidade arquitetônica são muitas vezes descartadas. Por exemplo, as alterações na arquitetura do sistema podem ser parcialmente refletidas em diagramas de classe, enquanto representações comportamentais correspondentes em diagramas de sequência são deixadas desatualizadas, levando a inconsistências semânticas que degradam a coerência do projeto ao longo do tempo ([KLARE et al., 2021]).

**Tabela 14:** Agenda de pesquisa sobre gamificação no ensino de modelagem conceitual

Problema (P)	Questão de Pesquisa (QP)	Objetivo (O)	Contribuição (C)
<p><b>P1:</b> Baixo engajamento dos estudantes em atividades de modelagem ((GARACCIONE et al. [2025b], (LI et al. [2025], (GARACCIONE et al. [2025a], (PAULA PORTO et al. [2021]))</p>	<p><b>QP1:</b> Como a gamificação influencia o engajamento na aprendizagem de diagramas de classes UML?</p> <p><b>QP2:</b> Quais elementos de jogo são mais eficazes em modelagem conceitual?</p>	<p><b>O1:</b> Avaliar os efeitos motivacionais da gamificação em tarefas de modelagem UML.</p> <p><b>O2:</b> Identificar elementos de jogo impactantes para melhorar o engajamento dos estudantes.</p>	<p><b>C1:</b> Evidências empíricas sobre melhoria motivacional ((GARACCIONE et al. [2025b], (LI; HEW; DU [2024], (XU; XU; XING [2024], (LI; HEW; DU [2024], (XU; XU; XING [2024])).</p> <p><b>C2:</b> Padrões de design para gamificar atividades de modelagem ((BUCCHIARONE et al. [2023b], (CAGNAZZO et al. [2023])).</p>
<p><b>P2:</b> Dificuldade em reter conceitos complexos de modelagem ((UNHELKAR [2017], (JACOBSON; BOOCH [2021], (GARACCIONE et al. [2025a], (KIENZLE et al. [2024]))</p>	<p><b>QP1:</b> Ambientes de aprendizagem gamificados podem melhorar a retenção de longo prazo de conceitos de modelagem?</p> <p><b>QP2:</b> Como diferentes estratégias instrucionais gamificadas afetam a retenção cognitiva?</p>	<p><b>O1:</b> Comparar a retenção de aprendizagem entre métodos gamificados e tradicionais.</p> <p><b>O2:</b> Projetar mecânicas de jogo focadas em retenção.</p>	<p><b>C1:</b> Percepções sobre o impacto cognitivo da gamificação ((LUMSDEN et al. [2016])).</p> <p><b>C2:</b> Diretrizes para retenção de longo prazo via modelagem gamificada ((GARACCIONE et al. [2025a], (CAGNAZZO et al. [2023])).</p>
<p><b>P3:</b> Desalinhamento entre objetivos pedagógicos e design de jogos ((LI; MA; SHI [2023], (LANDERS [2019], (PAULA PORTO et al. [2021], (PINTRICH [2000]))</p>	<p><b>QP1:</b> Como a gamificação pode ser alinhada aos objetivos de aprendizagem em modelagem conceitual?</p> <p><b>QP2:</b> Quais princípios de design garantem validade educacional em ferramentas gamificadas?</p>	<p><b>O1:</b> Identificar problemas de design que prejudicam resultados de aprendizagem.</p> <p><b>O2:</b> Desenvolver um framework que alinhe mecânicas de jogo com objetivos pedagógicos.</p>	<p><b>C1:</b> Framework para gamificação orientada à aprendizagem ((LI; MA; SHI [2023], (LANDERS [2019])).</p> <p><b>C2:</b> Rúbrica de avaliação para design de jogos educacionais ((PAULA PORTO et al. [2021], (MARIN et al. [2018])).</p>
<p><b>P4:</b> Evidências empíricas limitadas sobre resultados de aprendizagem ((GARACCIONE et al. [2025b], (LI; HEW; DU [2024], (XU; XU; XING [2024], (ZENG et al. [2024], (MARIN et al. [2018]))</p>	<p><b>QP1:</b> Qual é o impacto empírico da gamificação na aquisição de habilidades de modelagem?</p> <p><b>QP2:</b> Como ambientes gamificados afetam a transferência de conhecimento para tarefas reais?</p>	<p><b>O1:</b> Conduzir experimentos controlados comparando habilidades de modelagem.</p> <p><b>O2:</b> Examinar a transferibilidade de habilidades aprendidas via gamificação.</p>	<p><b>C1:</b> Conjuntos de dados e benchmarks experimentais ((GARACCIONE et al. [2025b], (MARIN et al. [2018])).</p> <p><b>C2:</b> Efeitos mensuráveis nos resultados de aprendizagem dos estudantes ((ZENG et al. [2024], (XU; XU; XING [2024])).</p>
<p><b>P5:</b> Escalabilidade e sustentabilidade de ferramentas gamificadas ((BUCCHIARONE et al. [2023b], (BUCCHIARONE et al. [2023a], (KIENZLE et al. [2024], (CAGNAZZO et al. [2023]))</p>	<p><b>QP1:</b> Quais padrões arquiteturais suportam gamificação escalável em ferramentas de modelagem?</p> <p><b>QP2:</b> Como sistemas gamificados podem ser mantidos em cursos diversos?</p>	<p><b>O1:</b> Projetar arquiteturas escaláveis de ferramentas gamificadas.</p> <p><b>O2:</b> Fornecer diretrizes para integração sustentável em currículos.</p>	<p><b>C1:</b> Toolkits e plugins para reutilização ((BUCCHIARONE et al. [2023a], (KIENZLE et al. [2024], (CAGNAZZO et al. [2023])).</p> <p><b>C2:</b> Boas práticas para adoção sustentável em programas de computação.</p>
<p><b>P6:</b> Efeitos colaterais da gamificação: superjustificação ou desmotivação ((BEKK et al. [2022], (MULLINS; SABHERWAL [2020], (LANDERS [2019]))</p>	<p><b>QP1:</b> Quais efeitos não intencionais surgem de ambientes de modelagem gamificados mal projetados?</p> <p><b>QP2:</b> Como mitigar os efeitos cognitivos e emocionais da gamificação?</p>	<p><b>O1:</b> Identificar e analisar efeitos negativos da gamificação.</p> <p><b>O2:</b> Propor estratégias de mitigação usando design emocional-cognitivo.</p>	<p><b>C1:</b> Tipologia de efeitos colaterais na gamificação educacional ((BEKK et al. [2022], (MULLINS; SABHERWAL [2020])).</p> <p><b>C2:</b> Diretrizes para evitar armadilhas da gamificação ((LANDERS [2019])).</p>
<p><b>P7:</b> Heterogeneidade nos perfis e preferências dos estudantes ((HAMARI; HASSAN; DIAS [2018], (HAMARI; HASSAN; DIAS [2018], (PINTRICH [2000]))</p>	<p><b>QP1:</b> Como diferenças individuais (ex.: estratégias de aprendizagem, orientação motivacional) afetam os resultados da gamificação?</p> <p><b>QP2:</b> Como ambientes gamificados podem ser adaptados a diferentes tipos de aprendizes?</p>	<p><b>O1:</b> Classificar perfis de aprendizes em ambientes de modelagem gamificados.</p> <p><b>O2:</b> Permitir recursos gamificados adaptativos ajustados às necessidades dos usuários.</p>	<p><b>C1:</b> Modelos de gamificação adaptativa ao aprendiz ((HAMARI; HASSAN; DIAS [2018], (PINTRICH [2000])).</p> <p><b>C2:</b> Técnicas de personalização para educação em modelagem ((HAMARI; HASSAN; DIAS [2018])).</p>

**Consequências da documentação de projeto inconsistente.** As implicações de tais inconsistências no mundo real são tangíveis. Em sistemas de grande escala, a documentação de projeto não mantida ou desalinhada pode causar deriva arquitetônica, aumentar o tempo de integração para novos desenvolvedores e impedir a evolução do sistema. Incorporar elementos de jogo, como mecânica de progressão, escores ou desafios de detecção de inconsistência, poderia estimular o engajamento sustentado em atividades de documentação que são tradicionalmente subvalorizadas.

**Papel negligenciado da fundamentação arquitetural** Além disso, a documentação de nível de projeto, como a justificativa para selecionar estilos arquitetônicos ou aplicar padrões de projeto específicos, é crucial para manter a integridade do sistema a longo prazo e apoiar a tomada de decisão informada durante a evolução do sistema. No entanto, essa forma de documentação raramente é gamificada ou suportada por ferramentas interativas, apesar de sua importância conhecida em ambientes colaborativos e tarefas de manutenção (BOOCH, 2025; ROMEO et al., 2025; KLARE et al., 2021). A introdução de incentivos gamificados, como o reconhecimento pela integralidade ou consistência de melhorias arquitetônicas, pode ajudar a promover essa forma crítica, mas muitas vezes negligenciada de documentação.

**Oportunidades de gamificação na reengenharia de software.** Um domínio igualmente negligenciado é a reengenharia de software. Ao contrário do refatoramento, que tem sido o foco de vários estudos de gamificação, a reengenharia, especialmente envolvendo recuperação de modelos ou engenharia reversa de sistemas legados, requer documentação intensiva e rastreabilidade para ponte de estados atuais e históricos do sistema. A gamificação poderia suportar a redução da carga cognitiva e orientar os desenvolvedores através de tarefas de modelagem progressiva como reconstruir artefatos de design perdidos ou alinhar modelos recuperados com o comportamento do sistema pretendido (PAULA PORTO et al., 2021; KLARE et al., 2021).

**Chamada para uma agenda de pesquisa de gamificação.** Estas lacunas definem um grande desafio de investigação (Tabela 14): *Investigar se a gamificação pode estimular a documentação consistente, precisa e oportuna de artefatos de projeto de software, particularmente modelos UML e decisões arquitetônicas, mitigando assim as consequências a longo prazo da degradação arquitetônica.* Se bem sucedido, esta linha de investigação poderia redefinir como as equipes abordam a documentação de software, transformando-a de uma obrigação relutante em uma parte envolvente e integrante do ciclo de vida de desenvolvimento. Isso poderia suportar melhor integridade de projeto e manutenção e garantir a rastreabilidade da lógica de projeto, um fator muitas vezes perdido em contextos de entrega ágil e contínua.

### 5.5.6 Limitações do Estudo

Uma das principais limitações do estudo é que não distinguiu entre motivação intrínseca e extrínseca na explicação do esforço reduzido. Embora elementos gamificados possam ter despertado interesse genuíno, eles agiram como incentivos externos através de pontos, crachás

ou rastreamento de progresso. Pesquisas mostram que, embora a motivação extrínseca seja efetiva em curto prazo, pode levar à aprendizagem superficial e dependência de recompensas (CLANTON HARPINE; CLANTON HARPINE, 2015; LEE et al., 2010). Como Zichermann e Cunningham (ZICHERMANN; CUNNINGHAM, 2011) advertiu, “*uma vez que você começar a dar a alguém uma recompensa, você tem que mantê-la nesse loop recompensa para sempre*”. Sem separar esses motoristas motivacionais, os achados permanecem abertos à interpretação. Estudos futuros devem abordar isso incorporando instrumentos validados para mensurar o tipo motivacional. Isso esclareceria se o esforço reduzido decorre de um engajamento mais profundo com a tarefa ou apenas de uma resposta a estímulos externos.

## 5.6 Ameaças à Validade

Esta seção aborda ameaças à validade dos achados e delinea as ações realizadas para mitigá-los, seguindo diretrizes estabelecidas em engenharia de software empírico (WOHLIN et al., 2012). Foi desenhado cuidadosamente o estudo para garantir a validade das conclusões internas, externas, de construção e estatísticas. A Tabela 15 resume as ameaças à validade abordadas e suas estratégias de mitigação.

**Validade interna.** Refere-se a se os efeitos observados podem ser causalmente atribuídos ao tratamento e não a outros fatores (WOHLIN et al., 2012). O estudo atende a esse critério. Primeiramente, foi respeitada a precedência temporal, ou seja, elementos de gamificação foram introduzidos antes da mensuração dos resultados de desempenho. Em segundo lugar, observou-se uma covariável consistente: à medida que foram aplicados elementos de gamificação, seguiram-se mudanças na taxa de detecção e erro de interpretação. Em terceiro lugar, controlamos as potenciais variáveis de confusão, garantindo que todos os participantes receberam as instruções exatas, trabalharam em condições semelhantes e completaram as mesmas tarefas. Não foram detectadas influências externas, corroborando a conclusão de que o tratamento foi a causa dos efeitos observados.

**Validade externa.** Diz respeito à generalização dos resultados para outras configurações, populações ou tempos (WOHLIN et al., 2012). Três ações reforçaram este aspecto. Primeiro, as tarefas experimentais foram baseadas em fragmentos UML extraídos de sistemas de software do mundo real, aumentando a validade ecológica. Em segundo lugar, embora os participantes fossem estudantes universitários, pesquisas anteriores mostram que estudantes podem servir como representantes válidos de profissionais de software novatos, especialmente em contextos como startups e pequenas empresas (FALESSI et al., 2018). Em terceiro lugar, a configuração experimental controlada mimetiza ambientes de aprendizagem utilizados na educação de engenharia de software, aumentando a relevância dos achados para cenários pedagógicos e de treinamento.

**Validade de Construção.** Garante que as medidas operacionais refletem os construtos teóricos investigados. Todas as variáveis dependentes foram derivadas de construtos validados em

**Tabela 15:** Resumo das ameaças à validade e das estratégias de mitigação

Tipo	Descrição	Estratégia de mitigação
Validade Interna	Garante que os efeitos observados estejam causalmente vinculados ao tratamento.	<ul style="list-style-type: none"> <li>✓ Garantiu precedência temporal (tratamento antes do resultado)</li> <li>✓ Controlou variáveis de confusão</li> <li>✓ Padronizou tarefas e instruções entre os participantes</li> </ul>
Validade Externa	Determina se os achados se generalizam para outros contextos e populações.	<ul style="list-style-type: none"> <li>✓ Usou fragmentos UML de sistemas reais</li> <li>✓ Considerou estudantes como representantes válidos de novatos</li> <li>✓ Replicou cenários realistas de aprendizagem com relevância prática</li> </ul>
Validade de Construção	Avalia se os construtos foram operacionalizados de forma precisa	<ul style="list-style-type: none"> <li>✓ Utilizou métricas validadas</li> <li>✓ Tarefas revisadas por especialistas para garantir alinhamento com a teoria</li> </ul>
Validade de Conclusão Estatística	Examina a correção da inferência estatística	<ul style="list-style-type: none"> <li>✓ Verificou pressupostos usando testes de Shapiro-Wilk</li> <li>✓ Utilizou t-test pareado e teste de Wilcoxon de forma apropriada</li> <li>✓ Garantiu homogeneidade dos participantes em formação e experiência</li> </ul>

pesquisas anteriores sobre qualidade do modelo UML e detecção de inconsistência (LANGE, 2007). O esforço de detecção e a taxa de erro de interpretação foram medidos utilizando protocolos estabelecidos e instrumentação. Peritos de domínio revisaram as tarefas para confirmar seu alinhamento com os construtos pretendidos. Esse rigor garante que as medidas reflitam com precisão as dimensões cognitivas e de desempenho de interesse.

**Validade de conclusão estatística.** Diz respeito à adequação e robustez dos métodos estatísticos utilizados (WOHLIN et al., 2012). Verificamos todos os pressupostos necessários para os testes estatísticos. A normalidade foi avaliada pelo teste de Shapiro-Wilk, e alternativas não paramétricas (teste de Wilcoxon) foram utilizadas quando os pressupostos foram violados. Optou-se por testes pareados com base no desenho de sujeitos internos. Também foi garantida a homogeneidade da amostra selecionando participantes com formação acadêmica comparável e experiência prévia em UML. Dado o tamanho da amostra e o tamanho do efeito observado, o poder estatístico foi suficiente, minimizando o risco de erros tipo I e tipo II.

## 5.7 Considerações Finais do Capítulo

Este estudo analisou o uso da gamificação para impulsionar a motivação e o engajamento no ensino de modelagem de software com UML. O objetivo foi avaliar empiricamente como os elementos de gamificação afetam a capacidade dos alunos de detectar inconsistências em modelos UML. Uma experiência controlada envolvendo 60 estudantes universitários combinava atividades tradicionais de aprendizagem com tarefas gamificadas, incluindo pontos, leaderboards, avatars e feedback. Utilizando uma metodologia estruturada, avaliou os efeitos duplos da gamificação sobre os resultados de engajamento e aprendizagem na educação em engenharia de software.

Os resultados indicam que a gamificação elevou de forma significativa o nível de engajamento dos estudantes e reduziu o esforço cognitivo empregado na detecção de inconsistências. Entretanto, os dados também revelaram uma compensação relevante: os participantes demonstraram maior dificuldade em analisar criticamente e selecionar implementações corretas, o que resultou em uma menor taxa de identificação de inconsistências. Essa tensão sugere que, em-

bora a gamificação possa potencializar a motivação e a participação, projetos mal calibrados podem comprometer processos de raciocínio analítico, sobretudo em atividades de maior complexidade, como aquelas envolvidas na modelagem de software.

Trabalhos futuros devem refinar as estratégias de gamificação para alcançar um equilíbrio entre engajamento e efetividade educacional. As pesquisas devem investigar como elementos específicos de jogo interagem com processos cognitivos para promover o pensamento analítico. O avanço de ferramentas de neurociência, como dispositivos de EEG vestíveis, oferece uma nova e promissora fronteira. Estudos futuros podem explorar como padrões de ondas cerebrais revelam os mecanismos cognitivos envolvidos na detecção de inconsistências e avaliar o impacto neural da gamificação. Pesquisas longitudinais com diferentes populações estudantis e variados contextos educacionais aprofundarão essas compreensões. Em conjunto, esses esforços conduzirão ao desenvolvimento de um arcabouço robusto para integrar a gamificação ao ensino de modelagem de software, maximizando a motivação e a efetividade da aprendizagem.

## 6 GAMIFICAÇÃO NO ENSINO DE ENGENHARIA DE SOFTWARE: UM ESTUDO DE CASO SOBRE SUA APLICAÇÃO NO DESENVOLVIMENTO ÁGIL E NA MODELAGEM UML

O objetivo deste capítulo é apresentar e analisar os resultados do estudo de caso conduzido após a implementação do plano de gamificação na disciplina de Engenharia de Software, examinando seus efeitos sobre o comportamento, o desempenho técnico e o engajamento das equipes. Este capítulo responde diretamente às três questões de pesquisa formuladas no estudo: (QP3.4) em que medida a gamificação contribui para a melhoria da execução dos elementos do processo Scrum; (QP3.5) se a intervenção gamificada impacta a qualidade e a consistência dos artefatos produzidos ao longo das sprints; e (QP3.6) se a gamificação influencia a evolução das notas atribuídas às equipes durante o semestre. A partir dessas questões, o capítulo discute evidências quantitativas e qualitativas que permitem compreender como a gamificação moldou práticas de colaboração, aderência ao processo ágil e qualidade técnica dos entregáveis.

A modelagem de software utilizando a *Unified Modeling Language* (UML) é reconhecida como uma etapa crítica no ensino e prática da engenharia de software, desempenhando papel essencial na documentação e comunicação das decisões arquiteturais e de design. Contudo, apesar de sua ampla adoção acadêmica e profissional, os estudantes frequentemente encontram dificuldades em aplicar consistentemente os conceitos teóricos dessa linguagem em cenários práticos, particularmente em contextos que envolvem processos ágeis como o Scrum.

Inconsistências em modelos UML frequentemente surgem de sobreposições conceituais mal alinhadas, falta de clareza semântica entre diagramas complementares e insuficiência no entendimento das dependências existentes entre diferentes visões do sistema. Tais inconsistências, quando não adequadamente detectadas e corrigidas, elevam os riscos de implementação incorreta das funcionalidades previstas, causando retrabalho e impactando negativamente a qualidade do software desenvolvido.

Nesse cenário desafiador, a gamificação emerge como uma estratégia pedagógica promissora, introduzindo elementos típicos de jogos, como pontos, rankings, badges e feedback contínuo, com o objetivo de aumentar o engajamento e a motivação dos estudantes em atividades de modelagem. Embora pesquisas recentes tenham explorado amplamente o potencial da gamificação em contextos diversos de engenharia de software, poucos estudos investigaram empiricamente os efeitos da gamificação especificamente sobre práticas ágeis, como Scrum, na qualidade dos artefatos técnicos produzidos e no desempenho acadêmico das equipes.

A problemática central que orienta este estudo reside, portanto, na lacuna entre o engajamento promovido por estratégias gamificadas e o desenvolvimento efetivo das competências técnicas exigidas na modelagem de software em contextos ágeis. Em outras palavras, questiona-se em que medida a gamificação é capaz de melhorar não apenas a motivação e a participação dos estudantes, mas também a consistência, a precisão e a qualidade dos artefatos UML gerados durante o processo de aprendizagem.

Além disso, permanece incerto em que medida os elementos gamificados realmente potencializam o aprendizado técnico ou, contrariamente, se sua introdução pode acarretar efeitos adversos, tais como superficialidade nas análises realizadas ou desvio do foco técnico em função da competição excessiva. Frente a tais lacunas, este capítulo relata um estudo empírico realizado em contexto educacional real, com o intuito de avaliar sistematicamente os efeitos da gamificação sobre três aspectos principais: (1) a aderência das equipes ao processo Scrum, incluindo a realização efetiva de suas cerimônias e papéis, (2) a qualidade técnica dos artefatos gerados ao final de cada sprint, particularmente diagramas UML e protótipos, e (3) as notas obtidas pelas equipes nas avaliações intermediárias e finais. Especificamente, foi conduzido um estudo longitudinal comparando cenários antes e após a implementação da gamificação, permitindo uma análise robusta das mudanças decorrentes da intervenção pedagógica.

Os resultados apresentados neste capítulo fornecem contribuições relevantes para o avanço do conhecimento sobre o papel da gamificação em disciplinas práticas de Engenharia de Software. Ao analisar os impactos em profundidade, tanto no comportamento quanto nos resultados técnicos das equipes, o estudo possibilita uma compreensão mais precisa sobre as condições sob as quais a gamificação pode ser efetivamente adotada em processos ágeis no ensino superior. Além disso, o capítulo oferece recomendações claras para docentes e pesquisadores interessados em integrar elementos gamificados às suas práticas pedagógicas, destacando cuidados essenciais para maximizar benefícios motivacionais sem comprometer o rigor técnico exigido pelas disciplinas da área.

Assim, este capítulo foi organizado da seguinte forma: a Seção 6.1 aborda os conceitos fundamentais que embasam o estudo, incluindo a aplicação de gamificação em contextos de Engenharia de Software e as práticas do framework Scrum. A Seção 6.2 descreve detalhadamente a metodologia adotada, incluindo o objetivo, as questões de pesquisa formuladas, o delineamento metodológico, os participantes e o contexto de aplicação, as intervenções gamificadas realizadas, a coleta e análise dos dados, bem como uma análise comparativa robusta dos cenários antes e após a gamificação. Em seguida, a Seção 6.3 apresenta os resultados obtidos por meio das análises qualitativas e quantitativas realizadas, interpretando-os à luz das questões de pesquisa. Já a Seção 6.4 aborda as ameaças à validade do estudo, enquanto a Seção 6.5 apresenta as conclusões gerais e sugestões para trabalhos futuros.

## **6.1 Fundamentação Teórica**

Esta seção reúne os fundamentos conceituais e empíricos que sustentam o presente estudo, articulando quatro eixos complementares. Inicialmente, na Subseção 6.2.1, são examinadas as inconsistências em modelos UML, suas origens e classificações, bem como os efeitos dessas incongruências sobre a qualidade dos artefatos e o desempenho no desenvolvimento de software.

Em seguida, a Subseção 6.2.2 aborda a gamificação no ensino de Engenharia de Software, discutindo a conceituação, os elementos estruturantes, os componentes, as mecânicas e dinâ-



micas e os resultados contrastantes observados na literatura quanto ao seu potencial motivacional e pedagógico. A Subseção 6.2.3 descreve a condução da busca bibliográfica e analisa as principais limitações metodológicas e empíricas observadas nos estudos anteriores, destacando lacunas quanto à duração das investigações, à heterogeneidade das amostras e à ausência de correlações robustas entre elementos gamificados e desempenho técnico.

Por fim, a Subseção 6.2.4 apresenta a justificativa para a adoção de um estudo de caso aprofundado, enfatizando sua adequação para investigar fenômenos educacionais complexos em contextos reais e sua contribuição potencial para o avanço das práticas pedagógicas em Engenharia de Software. Ao integrar esses quatro eixos, buscou-se estabelecer a base analítica necessária para compreender as relações entre inconsistências em modelagem UML, elementos de gamificação e resultados de aprendizagem em ambientes educacionais mediados por metodologias ágeis.

### 6.1.1 Inconsistências em Modelos UML

Inconsistências em modelos UML surgem quando dois ou mais diagramas, como os diagramas de classe e sequência, fornecem informações conflitantes ou incompatíveis sobre a estrutura ou comportamento de um sistema. Estas inconsistências geralmente decorrem de desalinhamentos entre elementos compartilhados, como operações de classes ou interações de mensagens, resultando em contradições que podem dificultar a interpretação correta do projeto durante sua implementação. Por exemplo, um diagrama de classe pode definir operações que não estão presentes ou são contraditórias em relação ao comportamento descrito em um diagrama de sequência correspondente (GARACCIONE et al., 2025a; TORRE et al., 2018; OMG, 2017).

A literatura documenta extensivamente diversos tipos de inconsistências, frequentemente categorizados em inconsistências sintáticas e semânticas, cada uma afetando diferentes aspectos do processo de desenvolvimento de software (MARCHEZAN et al., 2023a; REDER; EGYED, 2013). Inconsistências sintáticas envolvem erros na utilização das notações UML, enquanto inconsistências semânticas referem-se a contradições no significado ou comportamento esperado entre os modelos. Inconsistências pragmáticas, por sua vez, estão associadas ao uso inadequado ou interpretação errônea dos modelos por diferentes stakeholders, resultando em erros e retrabalho significativos durante a implementação (MARCHEZAN et al., 2024a).

Pesquisas empíricas anteriores têm demonstrado que essas inconsistências podem causar sérios prejuízos ao processo de desenvolvimento, incluindo aumento da complexidade, erros de comunicação entre equipes, falhas na implementação das funcionalidades previstas e redução geral na qualidade do código-fonte produzido (LANGE; CHAUDRON; MUSKENS, 2006; FARIAS et al., 2015). Considerando esses impactos, torna-se crucial investigar métodos eficazes para identificação e resolução dessas inconsistências, especialmente em contextos educacionais, onde a habilidade de interpretar e validar modelos de software corretamente é fundamental para

a formação profissional dos estudantes (FEICHAS; SEABRA, 2023).

### 6.1.2 Gamificação no Ensino de Engenharia de Software

A gamificação é definida como a aplicação estratégica de elementos típicos de jogos, tais como pontos, badges, rankings e desafios, em contextos não lúdicos com o objetivo de aumentar o engajamento, motivação e desempenho dos participantes (WERBACH; HUNTER, 2012; YIGITBAS et al., 2024b). No contexto específico da Engenharia de Software, a gamificação tem sido explorada como uma abordagem pedagógica inovadora, potencializando a participação ativa dos estudantes e melhorando sua experiência de aprendizagem em tarefas tradicionalmente consideradas complexas ou repetitivas, como modelagem UML e práticas ágeis (COSENTINO; GÉRARD; CABOT SAGRERA, 2017b; JURGELAITIS; DRUNGILAS; ČEPONIENĖ, 2018).

Werbach e Hunter (WERBACH; HUNTER, 2015, 2012) propõem uma categorização amplamente adotada dos elementos gamificados em três níveis principais: componentes, mecânicas e dinâmicas. Os componentes são elementos básicos com os quais os estudantes interagem diretamente, como pontos, badges e rankings. As mecânicas correspondem às regras que definem como os componentes são utilizados para motivar e orientar os participantes durante as atividades. Finalmente, as dinâmicas envolvem os aspectos psicológicos e emocionais mais profundos da experiência gamificada, como sensação de conquista, progressão e competição saudável (YOHANNIS, 2016).

Embora diversos estudos apontem benefícios significativos na utilização da gamificação em contextos educacionais, incluindo maior motivação, participação ativa e colaboração entre estudantes, outros estudos identificam riscos potenciais associados ao uso inadequado ou superficial dos elementos gamificados. Esses riscos incluem a superficialidade nas análises, desvio de foco das tarefas técnicas e aumento de competição excessiva que pode prejudicar a profundidade analítica e compreensão dos conceitos técnicos (ALMEIDA et al., 2023; ALSADOON; ALKHAWAJAH; SUHAIM, 2022; BUCCHIARONE et al., 2023a).

Dada essa dualidade, é essencial investigar detalhadamente como a gamificação pode ser estruturada de maneira a maximizar benefícios educacionais, minimizando seus potenciais efeitos adversos. Compreender a interação entre elementos gamificados e processos cognitivos envolvidos na identificação de inconsistências em modelos UML constitui uma lacuna importante na literatura atual, justificando a relevância e oportunidade da presente investigação.

### 6.1.3 Limitações dos Estudos Anteriores

Para caracterizar o estado da arte sobre gamificação aplicada ao ensino de Engenharia de Software, foi conduzida uma busca bibliográfica exploratória nas bases Scopus, IEEE Xplore, ACM Digital Library e ScienceDirect, abrangendo o período de 2010 a 2024. Utilizou-se a string de busca “gamification” OR “game-based learning” AND “software engineering” OR

“UML” OR “agile” OR “Scrum”. Foram incluídos estudos empíricos revisados por pares que abordavam intervenções gamificadas em contextos educacionais da área, sendo excluídos trabalhos teóricos ou revisões narrativas sem análise de dados.

Após a triagem inicial, cinco estudos foram selecionados por sua relevância, representatividade e contribuição à compreensão dos efeitos da gamificação em práticas técnicas e colaborativas. Hamari, Koivisto e Sarsa (2014) (HAMARI; KOIVISTO; SARSA, 2014) realizaram uma meta-análise sobre gamificação em múltiplos contextos educacionais, identificando evidências consistentes de aumento de engajamento, mas destacando a ausência de padronização nos métodos de avaliação, o que dificulta a comparação entre estudos.

Farias et al. (2015) (FARIAS et al., 2015), por sua vez, examinaram inconsistências em modelos UML produzidos por estudantes e identificaram deficiências conceituais persistentes, sugerindo a necessidade de estratégias pedagógicas que favoreçam feedback imediato e validação cruzada entre artefatos. Já García et al. (2017) (GARCIA et al., 2017) avaliaram o uso de badges e rankings em disciplinas de modelagem e práticas ágeis, verificando ganhos motivacionais pontuais, porém sem impacto sustentado sobre a qualidade técnica dos artefatos.

Em complemento, Morschheuser et al. (2017) (MORSCHHEUSER et al., 2017) investigaram a gamificação em atividades colaborativas de desenvolvimento de software e observaram que a competição excessiva pode reduzir a cooperação e a profundidade das análises técnicas. Por fim, Koivisto e Hamari (2019) (KOIVISTO; HAMARI, 2019) destacaram a escassez de estudos longitudinais e a dificuldade em medir efeitos duradouros da gamificação sobre aprendizagem complexa e competências técnicas.

Assim, verifica-se que, embora pesquisas anteriores tenham explorado amplamente os efeitos da gamificação em contextos educacionais e especificamente no ensino da Engenharia de Software, algumas limitações importantes ainda persistem nesses estudos, comprometendo parcialmente a compreensão plena dos impactos dessa estratégia pedagógica. Uma primeira limitação está relacionada à amplitude metodológica limitada observada em grande parte dos estudos prévios. Muitas investigações se concentram predominantemente em abordagens qualitativas ou estudos exploratórios de curta duração, restringindo a generalização dos achados para contextos educacionais mais amplos ou situações mais complexas (Easterbrook et al., 2008) (EASTERBROOK et al., 2008).

Além disso, frequentemente esses estudos não consideram rigorosamente as variações entre diferentes perfis de estudantes ou equipes, potencialmente negligenciando fatores importantes, como experiências prévias, níveis diferenciados de motivação e habilidades técnicas individuais. Outra limitação identificada é a ausência de investigações detalhadas sobre os possíveis efeitos adversos da gamificação, como a competição excessiva, a superficialidade na análise dos problemas e o desvio de foco técnico em função de recompensas imediatas (BISTA et al., 2014; MORSCHHEUSER et al., 2017; BUCCHIARONE et al., 2023a).

Poucos estudos investigam de forma aprofundada como elementos gamificados influenciam diretamente a execução de práticas ágeis, a qualidade dos artefatos técnicos e o engajamento

dos estudantes — dimensões essenciais para o ensino de Engenharia de Software. Essa lacuna dificulta compreender em que condições a gamificação contribui para reduzir inconsistências em modelos UML, melhorar a aderência ao processo Scrum e ampliar a participação nas atividades práticas. Tais limitações estão diretamente relacionadas ao foco do presente estudo, que busca examinar, de forma empírica, como a intervenção gamificada afeta essas variáveis em um contexto educacional real.

Adicionalmente, verifica-se uma escassez de estudos longitudinais robustos que analisem o impacto sustentado da gamificação ao longo do tempo. A maioria das pesquisas existentes examina resultados imediatos ou de curto prazo, não abordando suficientemente como o engajamento, motivação e desempenho podem evoluir com o uso contínuo e prolongado dos elementos gamificados em disciplinas técnicas complexas, como modelagem UML e desenvolvimento ágil de software (KOIVISTO; HAMARI, 2019; HAMARI; KOIVISTO; SARSA, 2014).

Por fim, alguns estudos carecem de análises empíricas que correlacionem diretamente os elementos gamificados específicos com os resultados obtidos em termos de aprendizagem técnica e desempenho acadêmico. Frequentemente, elementos gamificados são introduzidos sem uma fundamentação clara de sua seleção ou sem uma análise sistemática dos mecanismos psicológicos e educacionais que explicam seu sucesso ou fracasso em contextos específicos (YOHANNIS, 2016).

Essas constatações revelam oportunidades de pesquisa relevantes: a necessidade de estudos empíricos de maior duração, metodologicamente diversificados e focados na relação entre engajamento e qualidade técnica; a integração de métricas objetivas de desempenho em modelagem UML e a análise das interações entre elementos gamificados e processos cognitivos complexos. Essas lacunas fundamentam a justificativa para o estudo de caso apresentado na subseção seguinte.

#### 6.1.4 Justificativa Para o Estudo de Caso

Considerando as limitações identificadas na literatura e a escassez de estudos empíricos que articulem gamificação, práticas ágeis e modelagem UML em contextos educacionais reais, a realização de um estudo de caso aprofundado se justifica por diversas razões que emergem das lacunas identificadas na literatura existente e da necessidade prática de aprimorar estratégias educacionais na Engenharia de Software. Primeiramente, estudos de caso proporcionam uma investigação detalhada e contextualizada, permitindo observar e analisar profundamente fenômenos complexos em ambientes reais, algo difícil de ser alcançado através de métodos puramente experimentais ou quantitativos (HOLLWECK, 2015; RUNESON; HÖST, 2009). Este formato é particularmente adequado para investigar os impactos detalhados da gamificação sobre práticas educacionais, já que proporciona uma compreensão rica sobre como diferentes elementos gamificados interagem e influenciam as equipes em contextos reais e dinâmicos.

Em segundo lugar, o estudo de caso permite explorar profundamente tanto os benefícios

quanto as potenciais desvantagens da gamificação, especialmente no contexto específico do ensino prático em Engenharia de Software. Ao fornecer insights detalhados sobre as experiências dos estudantes ao longo do tempo, é possível entender melhor como elementos específicos da gamificação afetam a motivação, o comportamento colaborativo e a qualidade técnica dos artefatos gerados. Assim, a abordagem metodológica adotada neste estudo possibilita uma avaliação equilibrada e crítica dos efeitos práticos e pedagógicos da gamificação, indo além das avaliações superficiais e imediatas frequentemente relatadas em estudos anteriores.

Além disso, estudos de caso são particularmente adequados para lidar com a complexidade inerente ao ensino de Engenharia de Software, que frequentemente envolve interações multifacetadas entre aspectos técnicos, comportamentais e psicológicos dos estudantes. A metodologia adotada permite uma análise integrada dessas dimensões, considerando simultaneamente a aderência às práticas ágeis (Scrum), a qualidade técnica dos artefatos produzidos (como diagramas UML) e as percepções subjetivas dos estudantes sobre o impacto da gamificação. Dessa maneira, o estudo oferece uma visão holística e profundamente embasada, capaz de informar práticas educacionais futuras de forma mais precisa e aplicável.

Por fim, o estudo de caso se justifica pela sua contribuição direta ao corpo de conhecimento na área, fornecendo dados empíricos robustos que podem embasar futuras pesquisas e práticas pedagógicas. Os resultados obtidos podem não apenas confirmar ou desafiar achados prévios na literatura, mas também oferecer novas perspectivas e recomendações específicas para melhorar a eficácia educacional em disciplinas técnicas. Ao focar em um contexto real e longitudinal, o estudo amplia significativamente o entendimento científico e prático sobre a implementação e eficácia da gamificação, beneficiando diretamente tanto educadores quanto pesquisadores interessados em metodologias inovadoras e efetivas para o ensino de Engenharia de Software.

## 6.2 Metodologia

O presente estudo adota um delineamento de estudo de caso educacional único, de caráter exploratório e de abordagem mista (qualitativa e quantitativa), fundamentado nas diretrizes clássicas de Runeson e Höst (2009) (RUNESON; HÖST, 2009) para pesquisas empíricas em Engenharia de Software e nas proposições de Yin (2014) sobre estudos de caso em contextos educacionais complexos. Essa estratégia metodológica é adequada para investigar fenômenos contemporâneos em seu ambiente real, possibilitando examinar interações entre variáveis técnicas, cognitivas e motivacionais de maneira integrada.

O delineamento metodológico está alinhado ao objetivo central de investigar os efeitos da introdução de um plano de gamificação sobre o comportamento, o desempenho e a percepção de estudantes de graduação em uma disciplina prática de Engenharia de Software, com foco em desenvolvimento ágil e modelagem UML. O estudo foi conduzido ao longo de um semestre letivo completo, em uma instituição privada de ensino superior no Brasil, envolvendo estudantes de cursos superiores em tecnologia. Os participantes foram organizados em equipes autogerenciadas.

ciadas e realizaram entregas iterativas conforme os princípios do framework Scrum, abrangendo desde a elicitación de requisitos até a prototipação funcional de sistemas fictícios.

### 6.2.1 Objetivo e Questões de Pesquisa

As questões de pesquisa que orientam este estudo foram formuladas com base nas evidências obtidas em um estudo experimental anterior, conduzido com 60 estudantes, em ambiente controlado. Naquele experimento, analisamos a relação entre gamificação e esforço cognitivo na detecção de inconsistências em modelos UML, identificando tanto benefícios quanto limitações da abordagem gamificada. Os resultados serviram como ponto de partida para o desenho do presente estudo de caso, que amplia o escopo da investigação para além da análise individual de modelos, incluindo também variáveis como colaboração, completude de artefatos e práticas reais de desenvolvimento ágil. As seguintes questões norteiam a investigação:

- **QP3.4:** A gamificação contribui para a melhoria na execução dos elementos do processo Scrum pelas equipes (ex.: cerimônias, papéis e planejamento)?
- **QP3.5:** A gamificação impacta positivamente a qualidade dos resultados entregues ao final de cada sprint (ex.: artefatos, modelagem UML, protótipos)?
- **QP3.6:** A gamificação influencia de forma significativa a evolução das notas atribuídas às equipes ao longo das sprints?

### 6.2.2 Hipóteses

A formulação das hipóteses decorre diretamente das questões de pesquisa apresentadas na subseção anterior e fundamenta-se em evidências empíricas reportadas em estudos prévios sobre gamificação aplicada ao ensino de Engenharia de Software (HAMARI; KOIVISTO; SARSA, 2014; MORSCHHEUSER et al., 2017; JÚNIOR; FARIAS, 2021; BUCCHIARONE et al., 2023a). As hipóteses foram estruturadas em pares nulos e alternativos, de modo a permitir a verificação estatística de possíveis efeitos da gamificação sobre três dimensões principais do aprendizado: o processo, a qualidade técnica e o desempenho global. Essa formulação segue o princípio de que o impacto da intervenção pedagógica deve ser analisado tanto em termos comportamentais quanto produtivos, integrando métricas objetivas e percepções qualitativas.

- $H1_{null}$ : A gamificação não afeta a execução dos elementos do processo Scrum.
- $H1_{alt}$ : A gamificação melhora a execução dos elementos do processo Scrum.
- $H1_{null}$ : A gamificação não afeta a execução dos elementos do processo Scrum.
- $H1_{alt}$ : A gamificação melhora a execução dos elementos do processo Scrum.

- $H2_{null}$ : A gamificação não afeta a qualidade dos artefatos entregues.
- $H2_{alt}$ : A gamificação melhora a qualidade dos artefatos entregues (ex.: diagramas UML, protótipos).
- $H3_{null}$ : A gamificação não afeta a evolução das notas nas sprints.
- $H3_{alt}$ : A gamificação contribui para o aumento progressivo das notas ao longo das sprints.

Essas hipóteses articulam três níveis complementares de análise: (i) processual, ao considerar a adesão ao método Scrum; (ii) técnica, ao examinar a consistência e completude dos artefatos produzidos, e (iii) avaliativa, ao observar a evolução das notas atribuídas às equipes. Essa estrutura possibilita uma triangulação entre resultados quantitativos e qualitativos, oferecendo uma visão integrada do impacto da gamificação no desempenho e na motivação dos estudantes em contextos educacionais reais.

### 6.2.3 Participantes e Contexto

**Contexto.** A presente investigação empírica foi conduzida em uma instituição de ensino superior privada no Brasil, com tradição na formação de profissionais da área de computação e reconhecida por integrar metodologias ativas ao currículo. O estudo foi inserido em uma disciplina prática cuja proposta pedagógica está fortemente alinhada à formação por competências, com ênfase no desenvolvimento de soluções computacionais orientadas por frameworks ágeis. Essa configuração ofereceu um contexto autêntico e controlado para avaliar os efeitos da gamificação sobre práticas colaborativas, engajamento dos estudantes e qualidade dos artefatos técnicos. A aplicação se deu ao longo de um semestre letivo, viabilizando a observação longitudinal dos fenômenos em questão, conforme sintetizado na Tabela 16.

Participaram do estudo 60 estudantes de cursos superiores tecnológicos e de bacharelado da área de computação, como Análise e Desenvolvimento de Sistemas, Engenharia de Software e Sistemas de Informação. Os participantes foram distribuídos em nove equipes autogerenciadas, compostas de maneira a preservar a diversidade de perfis e competências. A composição das equipes seguiu os princípios fundamentais do framework Scrum, assegurando a vivência plena dos papéis de Product Owner, Scrum Master e membros do Time de Desenvolvimento. Esse arranjo organizacional favoreceu a análise do comportamento coletivo sob diferentes níveis de autonomia, responsabilidade e interação social, aspectos críticos quando se busca compreender o impacto da gamificação em contextos educacionais complexos.

Cada equipe participou de três sprints completos, com duração aproximada de três semanas cada, respeitando a estrutura iterativo-incremental do Scrum. Durante esses ciclos, os estudantes foram desafiados a aplicar os conceitos teóricos em atividades práticas de análise, concepção e construção de sistemas, com base em problemas fictícios representativos de demandas reais do setor de tecnologia. As sprints envolveram todas as cerimônias formais do Scrum, incluindo

<b>Tabela 16:</b> Caracterização dos participantes e do contexto do estudo					
<b>Curso</b>	<b>N.º de participantes</b>	<b>Tipo de curso</b>	<b>Duração da disciplina</b>	<b>N.º de equipes</b>	<b>Duração dos sprints</b>
Análise e Desenvolvimento de Sistemas	20	Tecnólogo	16 semanas	3	3 semanas cada
Engenharia de Software	25	Bacharelado	16 semanas	3	3 semanas cada
Sistemas de Informação	15	Bacharelado	16 semanas	3	3 semanas cada

Fonte: elaborado pelo pesquisador (2025).



planejamento, reuniões diárias, revisões e retrospectivas, além de entregas obrigatórias ao fim de cada iteração. Essa organização permitiu avaliar não apenas o progresso técnico das equipes, mas também a maturidade no uso de metodologias ágeis e o impacto motivacional decorrente da introdução de mecanismos de gamificação.

Os artefatos desenvolvidos ao longo das sprints incluíram documentos e produtos essenciais para o ciclo de desenvolvimento ágil, tais como personas, product backlog priorizado, histórias de usuário refinadas, protótipos de interface, quadros de tarefas (taskboards), e uma variedade de diagramas UML, incluindo casos de uso, classes e sequência. A avaliação das entregas considerou critérios como completude, consistência semântica, clareza estrutural e aderência às boas práticas da Engenharia de Software. Esses parâmetros foram fundamentais para a análise comparativa entre os cenários antes e após a implementação da gamificação, permitindo a triangulação dos dados obtidos em observações, notas das sprints e métricas de aderência ao processo ágil.

#### 6.2.4 Variáveis e Critérios de Análise

As variáveis analisadas foram definidas de modo a refletir diretamente as dimensões centrais investigadas neste estudo: (i) processo, relacionado à execução das práticas do framework Scrum; (ii) artefatos técnicos, referentes à qualidade, completude e consistência dos diagramas UML e demais entregáveis; e (iii) desempenho, representado pelas notas atribuídas às equipes ao longo das sprints. Desse modo, é possível observar os efeitos da introdução da gamificação sobre o comportamento das equipes, a qualidade dos artefatos técnicos e os resultados avaliativos. A relação entre as variáveis foi delineada de acordo com o modelo teórico proposto por (RUNESON; HÖST, 2009), que recomenda a distinção entre variáveis independentes (intervenções) e dependentes (resultados observáveis).

Variável independente: introdução do plano de gamificação no processo de desenvolvimento da disciplina.

Variáveis dependentes: execução do processo Scrum: frequência e completude das cerimônias, definição de papéis, reuniões de planejamento e retrospectiva; qualidade dos artefatos: completude dos diagramas UML, coerência entre artefatos, clareza dos protótipos; Nota Sprint: média ponderada atribuída a cada entrega, considerando critérios técnicos e comportamentais.

As métricas de cada variável foram consolidadas em planilhas de acompanhamento, permitindo o cálculo de médias, desvios-padrão e variações percentuais entre os cenários com e sem gamificação, conforme sintetizado na tabela a seguir, buscando assegurar a comparabilidade dos resultados e reforçar a validade construtiva da análise.

**Tabela 17:** Variáveis, critérios e métricas de análise

Tipo de variável	Dimensão avaliada	Indicadores	Forma de mensuração	Escala	Fonte de dados
Independente	Introdução da gamificação	Implementação do plano (pontos, badges, rankings, trocas)	Comparação pré/pós-intervenção	Binária (antes/-depois)	Registros da disciplina
Dependente	Execução do processo Scrum	Frequência e completude das cerimônias; definição de papéis	Rubrica avaliativa do-cente	0–10	Observações e documentos
Dependente	Qualidade dos artefatos	Completude, coerência e clareza dos diagramas UML e protótipos	Rubrica técnica	0–10	Artefatos entregues
Dependente	Nota Sprint	Média ponderada de desempenho técnico e comportamental	Cálculo automatizado em planilha	0–10	Sistema institucional

### 6.2.5 Desenho do Estudo e Fases do Processo

O delineamento do estudo foi estruturado em três fases sequenciais: planejamento, execução e análise, conforme o modelo de estudos de caso (RUNESON; HÖST, 2009; HOLLWECK, 2015). Essa organização buscou assegurar a rastreabilidade entre as decisões metodológicas e os resultados obtidos, permitindo a observação longitudinal dos efeitos da gamificação no contexto da disciplina. A Figura 1 apresenta uma visão geral do fluxo metodológico adotado.

**Fase 1 – Preparação e planejamento:** A primeira fase consistiu na concepção e estruturação do plano de gamificação que distingue componentes, mecânicas e dinâmicas de jogo (WERBACH; HUNTER, 2012). Nessa etapa, foram definidas as regras do sistema gamificado, incluindo a atribuição de pontos, badges, rankings e mecanismos de trocas. Também foram elaborados os critérios de avaliação das equipes e seu alinhamento aos objetivos pedagógicos da disciplina, assegurando coerência entre motivação extrínseca e aprendizagem significativa. Foram ainda desenvolvidas as planilhas e os formulários para registro de pontuação, entregas e feedback qualitativo, bem como o protocolo de análise comparativa entre os cenários pré e pós-gamificação, com métricas de completude, diversidade e qualidade dos modelos UML. Essa preparação garantiu a consistência dos instrumentos de coleta e a possibilidade de replicação do estudo.

**Fase 2 – Execução da disciplina:** A segunda fase correspondeu à implementação do plano de gamificação ao longo de 12 semanas, distribuídas em 3 sprints completos. Durante esse período, as equipes interagiram continuamente com o sistema gamificado, registrando suas entregas técnicas, pontuações e conquistas simbólicas. Foram coletados todos os artefatos produzidos (diagramas UML, protótipos, histórias de usuário) e aplicou-se um questionário reflexivo ao término da disciplina, com o objetivo de capturar percepções subjetivas sobre engajamento e



Fonte: Próprio autor

**Figura 16:** Processo experimental

aprendizagem. Para apoiar a execução e garantir transparência no acompanhamento do desempenho, foi desenvolvido um ambiente de monitoramento composto por painéis de conquistas e quadros de classificação (leaderboards). Esses instrumentos, ilustrados nas Figuras 2 e 3, permitiram que cada equipe visualizasse sua evolução em tempo real, comparasse seu progresso com o de outras equipes e ajustasse suas estratégias ao longo dos sprints. A disponibilização pública desses dados reforçou a autorregulação, estimulou a competição saudável e fortaleceu a ligação entre esforço, resultado e reconhecimento, um dos princípios centrais da eficácia da gamificação em contextos educacionais.



**Figura 17:** Painel de conquistas utilizado para acompanhamento em tempo real do progresso

As Figuras 17 e 23 exemplificam a interface utilizada pelos estudantes para visualizar pon-

**Figura 18:** Leaderboard exibido para incentivar a competição saudável e reforçar o feedback contínuo durante os sprints



Fonte: Próprio autor

tos, badges e rankings acumulados em cada sprint. A presença constante desses indicadores favoreceu a adoção de práticas colaborativas, ao mesmo tempo que possibilitou intervenções imediatas do docente em casos de queda de engajamento. Esse acompanhamento contínuo fortaleceu a ligação entre esforço, resultado e reconhecimento, cujo elemento é central para a eficácia da gamificação em contextos educacionais.

**Fase 3 – Análise dos dados:** A terceira fase envolveu a análise quantitativa e qualitativa integrada dos dados, com foco em identificar padrões e relações entre as variáveis de interesse. Inicialmente, foi realizada uma síntese descritiva das pontuações por equipe, seguida de testes estatísticos (t de Student e correlação de Spearman) para examinar a relação entre pontuação, entregas técnicas e badges. Em seguida, aplicou-se a análise qualitativa temática, (BRAUN; CLARKE, 2006), para codificar as respostas dos questionários e as observações do docente, identificando evidências de motivação, colaboração e engajamento.

Os resultados foram organizados em uma matriz interpretativa que integrou dados quantitativos e qualitativos, relacionando-os diretamente às três questões de pesquisa (Q3.4, Q3.5 e Q3.6). Essa triangulação metodológica buscou fortalecer a validade interna do estudo e ampliar a compreensão dos efeitos da gamificação em um contexto educacional autêntico e dinâmico.

## 6.2.6 Materiais Utilizados e Replicabilidade

A replicação do estudo foi apoiada por um conjunto de materiais que asseguram transparência e reprodutibilidade metodológica. Foram utilizados: planilha de pontuação automatizada, com critérios definidos por categoria (processo, resultado e badges); rubricas de avaliação por sprint; instrumentos de coleta (formulários e questionários abertos); roteiro detalhado de aplicação do plano de gamificação com regras de progressão e troca; scripts de análise estatística desenvolvidos em R, disponibilizados em repositório privado, abrangendo etapas de limpeza, normalização e testes comparativos; templates padronizados dos diagramas UML utilizados nas entregas e manual de orientação para docentes interessados em replicar a experiência, contendo prazos, requisitos de infraestrutura e recomendações de adaptação. Todos os materiais estão armazenados em ambiente digital controlado, acessível mediante solicitação, de modo a garantir a reprodutibilidade do experimento e a comparabilidade dos resultados em futuras investigações.

## 6.2.7 Análise dos Dados

A coleta de dados utilizou três fontes principais de evidência: (i) artefatos produzidos pelas equipes, como personas, histórias de usuário, diagramas UML e protótipos; (ii) dados do sistema de gamificação, incluindo pontuação acumulada, conquistas e uso de trocas; e (iii) registros qualitativos, compostos por notas de campo do professor, observações durante as sprints e respostas a um questionário reflexivo ao final da disciplina. A análise dos dados foi conduzida em duas etapas complementares:

**Análise quantitativa descritiva e inferencial:** foi elaborada uma síntese dos dados de pontuação, número de entregas e *badges* por equipe, com posterior aplicação de testes estatísticos. A correlação de Spearman foi utilizada para identificar associações entre a pontuação total e duas variáveis-chave: número de entregas técnicas e número de *badges* conquistados. Um teste *t* de Student foi também conduzido para comparar o número médio de entregas entre equipes com pontuação acima e abaixo da mediana.

**Análise quantitativa descritiva e inferencial:** os dados textuais dos relatos dos estudantes e observações do docente foram submetidos a um processo de codificação temática (Braun e Clarke, 2006) (BRAUN; CLARKE, 2006), com o objetivo de identificar padrões de comportamento, percepções sobre motivação, colaboração e dificuldades na modelagem. Essa análise buscou compreender os significados atribuídos pelos participantes à experiência gamificada, complementando as evidências numéricas com interpretações contextuais e fenomenológicas.

### 6.2.8 Considerações Éticas

A condução deste estudo seguiu rigorosamente os princípios éticos aplicáveis à pesquisa com seres humanos, conforme estabelecido pelas diretrizes nacionais e institucionais. Antes da coleta de dados, todos os participantes foram informados sobre os objetivos da pesquisa, os procedimentos envolvidos, os potenciais benefícios e riscos, bem como sobre a confidencialidade e o anonimato garantidos durante todas as etapas do estudo. A participação foi inteiramente voluntária, sendo solicitada a assinatura do Termo de Consentimento Livre e Esclarecido (TCLE) por parte dos alunos, garantindo assim sua concordância formal e consciente para participação no experimento (MICHEL-CROSATO; BIAZEVIC et al., 2024).

Além disso, todas as atividades de coleta de dados — como o acompanhamento das interações nas cerimônias Scrum, a avaliação dos artefatos produzidos e a análise das notas obtidas — foram conduzidas de maneira a preservar a identidade dos envolvidos. Nenhuma informação sensível foi divulgada, e os resultados apresentados neste capítulo foram tratados de forma agregada, com foco nas médias de desempenho das equipes e não em dados individuais. Essa abordagem visa proteger a privacidade dos estudantes, assegurando que nenhum deles pudesse ser identificado direta ou indiretamente a partir das análises ou das tabelas publicadas.

## 6.3 Resultados e Discussão

Esta seção organiza-se em três movimentos: (i) análise geral de desempenho, (ii) comparação “antes vs. depois” e (iii) análises por questão de pesquisa — QP3.4, QP3.5 e QP3.6. A análise foi conduzida a partir da triangulação entre evidências quantitativas (pontuação acumulada, diversidade de entregas, conquistas gamificadas) e qualitativas (relatos dos estudantes, observações docentes e artefatos produzidos). Os resultados são interpretados à luz da literatura especializada, com o objetivo de elucidar os efeitos da gamificação em atividades de desenvol-

**Tabela 18:** Pontuação total obtida por cada equipe participante do estudo de caso

<b>Equipe</b>	<b>Pontuação Total</b>
Time 03 - CodeCrafters	1235
Time 01 - Dev\$ Team	1205
Time 02 - DeVemos e Não Pagamos	1130
Time 02 - UniDev Team	1010
Time 01 - Devs! Team	965
Time 02 - Dévis Team	930
Time 01 - Pandora Box Team	855
Time 03 - Gofteh Team	855
Time 04 - DevSinos	855
Time 05 - UniDev Team	855

vimento ágil e modelagem UML. A Tabela 18 resume o desempenho agregado por equipe, ao passo que a Figura 23 visualiza a dispersão dessas pontuações.

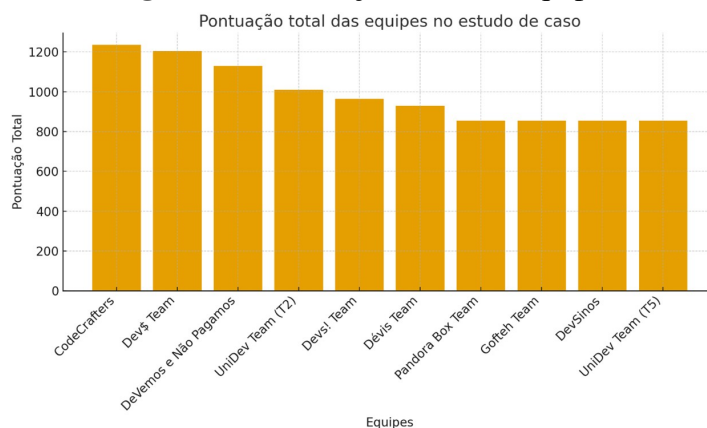
Na sequência, por meio da análise qualitativa temática, etapa em que os relatos dos estudantes e observações do professor foram codificados por meio de análise temática buscando identificar padrões de comportamento, percepções sobre motivação e colaboração, e efeitos percebidos sobre a aprendizagem (BRAUN; CLARKE, 2006). As respostas ao questionário aberto e as notas de campo foram analisadas de maneira independente por dois avaliadores, com discussão posterior para resolver divergências de codificação, assegurando maior consistência interpretativa. Essas análises foram posteriormente integradas em uma matriz interpretativa, relacionando os achados empíricos às três questões de pesquisa.

A Tabela 18 apresenta a pontuação total acumulada por cada equipe participante ao longo da disciplina. Observa-se que as equipes *CodeCrafters* e *Dev\$ Team* destacaram-se com os maiores valores, indicando uma forte aderência às práticas propostas no plano de gamificação. Essa distribuição de pontuação sugere não apenas um maior volume de entregas técnicas, mas também um engajamento mais estratégico com os elementos gamificados, como trocas de pontos, conquista de badges e uso tático do feedback formativo.

Além disso, a variação entre os desempenhos evidencia a heterogeneidade no modo como as equipes lidaram com os estímulos externos e com a própria organização interna. Equipes com pontuações intermediárias, como *UniDev Team* e *Devs! Team*, demonstraram regularidade nas entregas, embora com menor destaque em conquistas simbólicas. Já as equipes que permaneceram na faixa mínima da pontuação, ainda que tenham completado os artefatos principais, apresentaram menor frequência de interação com o sistema gamificado.

Esses dados sustentam as análises qualitativas discutidas nas seções seguintes, especialmente no que tange aos padrões de colaboração e ao planejamento estratégico adotado pelas equipes. A análise conjunta dos dados quantitativos e qualitativos permite uma compreensão mais robusta sobre como os elementos de gamificação impactaram o comportamento coletivo dos estudantes. É importante ressaltar que a pontuação acumulada não deve ser interpretada de forma isolada como métrica de desempenho absoluto, mas sim como um indicador do engaja-



**Figura 19:** Pontuação total das equipes

Fonte: Próprio autor

mento com a lógica gamificada da disciplina. A Figura 19, apresentada a seguir, complementa a Tabela 18, permitindo identificar de forma intuitiva as diferenças de desempenho entre as equipes.

Complementando a interpretação da Figura 19, observa-se que a distribuição das pontuações evidencia não apenas diferenças quantitativas entre as equipes, mas também distintos padrões de resposta ao ambiente gamificado. Equipes posicionadas nos extremos do gráfico, como CodeCrafters e Dev\$ Team, demonstraram capacidade consistente de converter esforço técnico em conquistas dentro do sistema, enquanto aquelas com valores mais baixos, ainda que tenham cumprido as entregas mínimas, apresentaram menor interação com os mecanismos de trocas, badges e feedback imediato.

Essa heterogeneidade reforça a importância de considerar fatores qualitativos, como estratégias de organização interna, comunicação entre membros e percepção de valor da gamificação, para compreender plenamente o impacto pedagógico da intervenção, evitando interpretações simplistas baseadas apenas na soma final de pontos. Além disso, a variação observada sugere que o engajamento em ambientes gamificados não decorre exclusivamente do design dos elementos de jogo, mas também da maturidade das equipes na gestão de processos ágeis e da capacidade de atribuir sentido pedagógico às recompensas simbólicas.

Nesse sentido, alguns estudos ((HAMARI; KOIVISTO; SARSA, 2014; MORSCHHEUSER et al., 2017)) destacam que o efeito motivacional da gamificação é altamente dependente da percepção de autonomia, competência e pertencimento, cujas dimensões psicológicas podem ser amplificadas ou inibidas conforme o contexto social e cognitivo do grupo. No caso analisado, tais dimensões parecem ter influenciado o modo como cada equipe articulou os elementos gamificados ao ciclo de desenvolvimento, revelando que o desempenho técnico e o engajamento não são necessariamente convergentes.

### 6.3.1 Análise Comparativa dos Cenários Antes e Após o Uso da Gamificação

Para assegurar rastreabilidade, esta subseção apresenta a comparação “antes vs. depois” por métrica e por equipe, remetendo à Tabela 19 e à Figura 19 para síntese visual das médias. A análise comparativa dos cenários antes e após o uso da gamificação revelou diferenças significativas em três dimensões principais: aderência ao processo Scrum, qualidade dos artefatos entregues e desempenho global nas entregas intermediárias. No cenário anterior à gamificação, os dados sugerem que as equipes enfrentavam dificuldades em manter o ritmo das cerimônias, como reuniões de planejamento e retrospectiva, além de apresentarem inconsistências frequentes nos artefatos, especialmente nos diagramas UML. Esses resultados reforçam achados da literatura que apontam para a baixa motivação dos estudantes em atividades técnicas contínuas sem estímulos extrínsecos (FEICHAS; SEABRA, 2023; ALMEIDA et al., 2023).

Com a introdução da gamificação, observou-se um aumento expressivo no engajamento das equipes com o processo ágil. As cerimônias passaram a ser conduzidas com maior regularidade e completude, os papéis foram desempenhados com mais clareza e as tarefas distribuídas de forma mais equitativa. A pontuação atribuída com base em critérios objetivos contribuiu para a responsabilização individual e coletiva das equipes. Além disso, os artefatos passaram a refletir uma melhoria técnica notável, com maior coesão entre os diagramas e uma maior preocupação com a consistência semântica. Tais ganhos sugerem que a gamificação, quando bem estruturada, pode funcionar como um catalisador para a interiorização de boas práticas de modelagem e colaboração (JURGELAITIS; DRUNGILAS; ČEPONIENĖ, 2018; COSENTINO; GÉRARD; CABOT SAGRERA, 2017b).

Em termos quantitativos, a comparação entre os cenários mostra um avanço considerável nas médias de desempenho, como ilustrado nas Tabelas 19 e 21. A métrica de Processo Scrum apresentou evolução média superior a 50%, enquanto as notas atribuídas aos artefatos e ao desempenho geral das equipes (Nota Sprint) também demonstraram crescimento consistente. Esses dados fornecem evidência empírica de que a gamificação pode ser uma ferramenta eficaz para mitigar desafios comuns em disciplinas práticas de Engenharia de Software, especialmente no que se refere à participação ativa, coesão de equipe e profundidade técnica na elaboração dos artefatos. Assim, os achados desta análise reforçam a pertinência de estratégias pedagógicas que integrem elementos motivacionais ao ensino de metodologias ágeis e modelagem orientada a objetos.

Além dos indicadores quantitativos, a triangulação com dados qualitativos obtidos por meio de questionários e observações do professor revelou relatos de maior senso de pertencimento, colaboração e satisfação dos estudantes. Essa convergência entre evidências indica que a melhoria não se limitou ao desempenho técnico, mas alcançou dimensões atitudinais essenciais para o aprendizado em Engenharia de Software, favorecendo a coesão social das equipes e a autonomia na tomada de decisão.

### 6.3.1.1 Cenário Antes da Gamificação

Antes da introdução da gamificação, as equipes frequentemente enfrentavam desafios significativos relacionados ao engajamento nas cerimônias do Scrum, baixa qualidade técnica nas entregas intermediárias e superficialidade na elaboração dos artefatos técnicos. As notas médias obtidas eram indicativas de dificuldades estruturais no processo de ensino-aprendizagem, refletindo engajamento limitado e deficiências metodológicas. A Tabela 19 sumariza esses desafios, representando a média das métricas avaliadas.

Além disso, a análise qualitativa identificou comportamentos de desmotivação, como atrasos recorrentes e baixa participação em discussões de modelagem, corroborando a literatura que associa a ausência de incentivos extrínsecos à queda de engajamento em atividades de alta complexidade cognitiva. Esses indícios sugerem que, antes da introdução de estímulos estruturados, a participação dos estudantes era predominantemente reativa e orientada por exigências avaliativas, em vez de refletir envolvimento autêntico com o processo de aprendizagem.

Os registros de observação apontaram que, durante a fase pré-gamificação, as reuniões de planejamento e retrospectiva eram frequentemente abreviadas ou realizadas apenas de maneira formal, sem discussão substantiva sobre riscos ou melhorias. Essa limitação reduziu a oportunidade de reflexão coletiva, um elemento essencial do método Scrum para promover o aprendizado contínuo (SCHWABER; SUTHERLAND, 2020). Em diversas equipes, papéis como Product Owner e Scrum Master eram executados de modo rotativo, mas sem clareza de responsabilidades, ocasionando sobreposição de tarefas e lacunas na comunicação.

No que diz respeito à produção dos artefatos, a ausência de estímulos competitivos ou recompensas percebidas resultou em diagramas UML incompletos ou com inconsistências semânticas recorrentes. Foram identificados erros de cardinalidade, omissões de relacionamentos entre classes e baixa padronização na nomenclatura, problemas esses que a literatura associa à falta de feedback imediato e de mecanismos de reforço positivo (LANGE; CHAUDRON; MUSKENS, 2006). Em entrevistas, alguns participantes relataram que a execução das atividades era vista como mera obrigação avaliativa, sem conexão clara com a aplicação prática dos conceitos de modelagem.

Do ponto de vista quantitativo, a Tabela 19 evidencia médias modestas em todas as métricas, com destaque para o Processo Scrum (média geral de 5,84), indicando dificuldade em manter cadência regular de entregas e revisão de requisitos. A proximidade entre as médias de “Resultado dos Artefatos” (5,86) e “Nota Sprint” (6,48) sugere que a baixa qualidade técnica impactou diretamente o desempenho global das equipes. Em termos qualitativos, observou-se ainda que a distribuição desigual de esforço entre membros favoreceu a concentração de tarefas em poucos participantes, reduzindo oportunidades de aprendizado colaborativo. Essa dinâmica reforça achados de estudos prévios sobre a importância de mecanismos de motivação extrínseca para sustentar a participação em atividades que exigem alto esforço cognitivo (MORSCHHEUSER et al., 2017).

Como pode ser observado na Tabela 19, os valores refletem dificuldades generalizadas das equipes na execução disciplinada dos processos ágeis e na geração consistente de artefatos técnicos de alta qualidade. Esses resultados indicam que, sem intervenções capazes de promover engajamento e feedback contínuo, a aprendizagem baseada em metodologias ágeis tende a reproduzir padrões de baixa interação e entrega superficial, limitando o potencial de desenvolvimento das competências esperadas em Engenharia de Software.

### 6.3.1.2 Cenário Após a Gamificação

Após a aplicação da intervenção gamificada, as equipes apresentaram melhorias notáveis em todas as métricas avaliadas. Observou-se adesão significativa às cerimônias do Scrum, engajamento intensificado na elaboração de artefatos técnicos e melhorias expressivas nas notas das entregas intermediárias. Estes resultados estão sintetizados na Tabela 19.

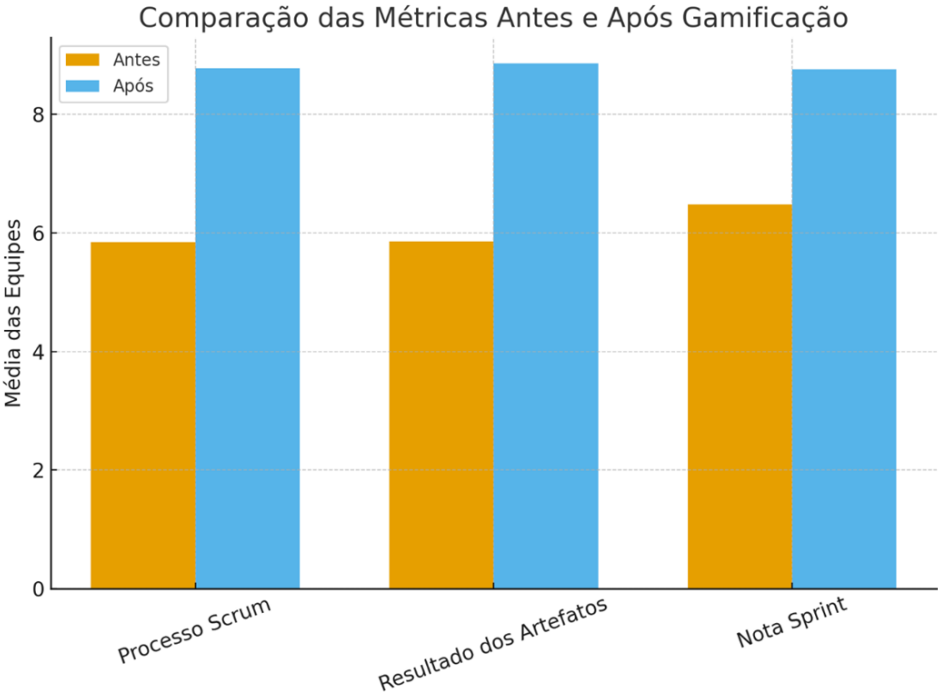
Os registros dos painéis de conquistas e leaderboards indicaram ainda uma dinâmica de competição saudável entre as equipes, que relatavam acompanhar as atualizações semanais para ajustar estratégias e aumentar sua pontuação, favorecendo a autorregulação do aprendizado.

Complementarmente, os resultados sugerem que a visibilidade contínua do progresso, proporcionada pelos painéis de conquistas, funcionou como reforço positivo, incentivando comportamentos alinhados às boas práticas de desenvolvimento e modelagem. A análise dos dados da Tabela 5 revela claramente a eficácia da estratégia gamificada para catalisar melhorias substanciais na adesão às práticas ágeis, na qualidade técnica das entregas e no desempenho global das equipes.

Observou-se que o acompanhamento em tempo real do desempenho coletivo estimulou a autogestão e favoreceu a responsabilização individual, elementos essenciais para a consolidação de equipes maduras em contextos educacionais baseados em metodologias ágeis. Essa evolução indica que a gamificação atuou não apenas como ferramenta de incentivo, mas também como mecanismo regulador do comportamento colaborativo e da consistência das práticas técnicas. Para tornar mais evidente essas melhorias, a Figura 20 apresenta a comparação das médias gerais das equipes nas três métricas avaliadas antes e após a aplicação da gamificação.

Para complementar a análise gráfica, a Tabela 20 apresenta a comparação direta entre os resultados obtidos antes e após a aplicação da gamificação, discriminados por equipe e por indicador. As diferenças absolutas e percentuais evidenciam o ganho médio superior a 50% em todos os critérios avaliados, com destaque para a qualidade dos artefatos e a execução do processo Scrum.

Figura 20: Comparação das métricas antes e após a gamificação



Fonte: Próprio autor

Tabela 19: Comparação antes e depois da gamificação (por equipe e indicador)

Equipe	Processo Scrum				Artefatos				Nota Sprint			
	Antes	Depois	Dif. Abs.	Dif. (%)	Antes	Depois	Dif. Abs.	Dif. (%)	Antes	Depois	Dif. Abs.	Dif. (%)
Pandora Box Team	5,75	9,03	+3,28	+57,0%	6,90	9,18	+2,28	+33,0%	6,73	8,09	+1,36	+20,2%
DeVemos e Não Pagamos	6,20	8,79	+2,59	+41,8%	5,31	8,13	+2,82	+53,1%	6,16	8,47	+2,31	+37,5%
CodeCrafters	5,12	8,94	+3,82	+74,6%	6,73	9,57	+2,84	+42,2%	6,60	9,20	+2,60	+39,4%
Dev\$ Team	6,42	9,21	+2,79	+43,4%	5,04	9,68	+4,64	+92,1%	6,97	9,46	+2,49	+35,7%
Dévis Team	6,66	8,11	+1,45	+21,8%	5,42	8,19	+2,77	+51,1%	6,18	8,33	+2,15	+34,8%
Gofteh Team	5,37	8,15	+2,78	+51,8%	5,61	8,91	+3,30	+58,8%	6,52	8,82	+2,30	+35,3%
DevSinos	5,86	8,83	+2,97	+50,7%	5,58	8,33	+2,75	+49,3%	6,61	8,78	+2,17	+32,8%
UniDev Team	5,28	8,54	+3,26	+61,7%	5,58	9,01	+3,43	+61,4%	6,37	8,65	+2,28	+35,8%
Devs! Team	5,91	9,42	+3,51	+59,4%	6,57	8,75	+2,18	+33,2%	6,20	9,03	+2,83	+45,6%
Média Geral	5,84	8,78	+2,94	+50,3%	5,86	8,86	+3,00	+51,2%	6,48	8,76	+2,28	+35,2%

Fonte: elaborado pelo pesquisador com base nos dados da pesquisa (2025).

Observa-se que, embora todas as equipes tenham apresentado evolução positiva, as magnitudes das diferenças variaram, indicando a influência de fatores, como maturidade técnica e coesão do grupo. Para aprofundar essa interpretação, apresenta-se, a seguir, a Tabela 7, que sintetiza a estatística descritiva das variáveis principais, considerando todo o conjunto de equipes.

**Tabela 20:** Estatística descritiva das variáveis principais (n = 9 equipes)

Variável	n	Mínimo	Q1	Média	Mediana	Desvio Padrão
Processo Scrum (antes)	9	5,12	5,37	5,84	5,86	0,48
Processo Scrum (depois)	9	8,11	8,54	8,78	8,83	0,40
Artefatos (antes)	9	5,04	5,42	5,86	5,61	0,58
Artefatos (depois)	9	8,13	8,75	8,86	8,91	0,54
Nota Sprint (antes)	9	6,16	6,18	6,48	6,52	0,30
Nota Sprint (depois)	9	8,09	8,47	8,76	8,78	0,39

Fonte: elaborado pelo pesquisador com base nos dados da pesquisa (2025).

Para complementar as análises quantitativas, a Tabela 21 apresenta as medidas descritivas da variável Motivação (1–5), coletada individualmente antes e após a intervenção gamificada. Essa métrica permite observar se o aumento no desempenho técnico e processual foi acompanhado por mudanças na disposição subjetiva dos participantes em relação à atividade.

**Tabela 21:** Estatística descritiva da Motivação (1–5), antes e após a gamificação (n = 30)

Variável	n	Mínimo	Q1	Média	Mediana	Desvio-padrão
Motivação (antes)	30	2	2	2,87	3,00	0,73
Motivação (depois)	30	4	4	4,50	4,50	0,51
Diferença (depois e antes)	30	1	1	1,63	2,00	0,49

Fonte: elaborado pelo pesquisador com base nos dados da pesquisa (2025).

Os resultados de motivação revelam aumento médio de 1,63 ponto na escala de 1 a 5, o que representa crescimento de aproximadamente 57% em relação à média inicial (2,87 - 4,50). O teste t pareado ( $t(29) = 18,25$ ;  $p < 0,001$ ) confirmou diferença estatisticamente significativa, com tamanho de efeito elevado ( $d = 3,33$ ). Esses achados sugerem que a gamificação exerceu influência positiva sobre a motivação autorreferida, reforçando as evidências de engajamento observadas nas métricas de processo e artefatos.

Os valores da Tabela 21, com médias e medianas superiores no período pós-gamificação e baixa dispersão entre as equipes, indicam, assim, a homogeneidade nos ganhos obtidos. Esses resultados fortalecem a evidência empírica de que a intervenção gamificada produziu melhorias estatisticamente relevantes nas dimensões avaliadas do processo de ensino-aprendizagem.

Dessa forma, a análise comparativa realizada fornece evidências empíricas sobre o impacto positivo da gamificação no ensino prático de Engenharia de Software, especialmente em contextos envolvendo metodologias ágeis e modelagem UML. Os dados indicam que a introdução estruturada de elementos gamificados (pontos, rankings e recompensas simbólicas) exerce influência positiva não apenas sobre a participação ativa e disciplinada das equipes, mas também sobre a qualidade técnica dos artefatos gerados.

Os resultados sugerem que a gamificação pode ser utilizada como estratégia educacional efetiva para mitigar problemas recorrentes de engajamento, melhorar a organização coletiva e, potencialmente, aprimorar a aprendizagem técnica e analítica exigida na formação em Engenharia de Software. Contudo, para maximizar tais benefícios, recomenda-se o desenho cuidadoso de elementos gamificados que valorizem explicitamente tanto a qualidade conceitual quanto a profundidade analítica das entregas produzidas pelas equipes.

A elevação simultânea das médias de Processo Scrum, Resultado dos Artefatos e Nota Sprint, aliada aos relatos de maior motivação e senso de pertencimento, demonstra que a intervenção não produziu apenas ganhos pontuais, mas favoreceu a consolidação de práticas de planejamento, revisão e comunicação centrais para o desenvolvimento de software em ambientes colaborativos. Além de evidenciar melhoria técnica, os resultados apontam que a visibilidade contínua do progresso, materializada nos painéis de conquistas e nos rankings, atuou como elemento regulador do comportamento coletivo, induzindo as equipes a manterem cadência nas cerimônias do Scrum, distribuírem responsabilidades e revisarem artefatos de forma mais rigorosa.

Essa dinâmica sugere que a gamificação, quando concebida como parte integrante da metodologia de ensino, e não como adição meramente lúdica, pode reforçar competências de organização e autoavaliação que extrapolam o escopo imediato da disciplina. Ao mesmo tempo, os achados destacam a necessidade de calibrar cuidadosamente os componentes da gamificação para evitar que a busca por recompensas simbólicas se sobreponha aos objetivos conceituais da atividade, garantindo que a estratégia mantenha o equilíbrio entre estímulos extrínsecos e promoção de compreensão técnica profunda.

### 6.3.2 QP3.4: A Gamificação Contribui Para o Aumento da Completude e Diversidade dos Artefatos Produzidos?

Nesta subseção, referenciam-se diretamente as evidências de completude/diversidade mostradas nas Tabelas 18, 19 e 20 e a relação com a pontuação em [23](#). A análise das entregas realizadas ao longo do semestre revelou um padrão consistente de aumento na quantidade e variedade de artefatos produzidos. Todas as nove equipes participantes entregaram os principais artefatos previstos no plano pedagógico da disciplina, como personas, histórias de usuário, diagramas UML (casos de uso, classes e sequência), documentos de arquitetura, protótipos navegáveis, backlogs priorizados e taskboards atualizados. Esse comportamento representa um avanço em relação a turmas anteriores não submetidas à intervenção gamificada.

Esse padrão é consistente com os ganhos de desempenho observados no comparativo ‘antes vs. depois’ (ver Tabelas 18, 19 e 20) e com a baixa dispersão entre equipes no período pós-gamificação (ver Tabela 21), reforçando que o aumento de completude e diversidade não se restringiu a casos isolados.

Além da completude, observou-se uma maior diversidade de artefatos entre as equipes, o



que sugere uma mobilização mais ampla das competências exigidas na disciplina. A estrutura gamificada, com pontos associados à produção de entregáveis diversos, funcionou como catalisador para que os estudantes se engajassem com diferentes dimensões do projeto, não se limitando a tarefas mais operacionais ou de baixa complexidade. Isso confirma o potencial da gamificação como instrumento de mediação pedagógica, ao tornar visíveis os objetivos da aprendizagem por meio de metas tangíveis.

O depoimento de participantes reforça essa interpretação. Em particular, a função da matriz de pontuação como instrumento de planejamento foi amplamente destacada. Estudantes relataram que utilizavam os valores atribuídos a cada tarefa como critério de priorização durante as sprints, permitindo maior autonomia na tomada de decisões e na distribuição de responsabilidades. Esse aspecto reforça a importância de sistemas de recompensa bem estruturados como dispositivos de orientação para a aprendizagem ativa.

A **análise estatística descritiva** confirma esses achados qualitativos. As equipes com maiores pontuações também apresentaram maior volume de entregas, embora a correlação entre pontuação total e número de artefatos não tenha sido estatisticamente significativa ( $\rho = -0.098$ ,  $p = 0.787$ ). Tal resultado sugere que outros fatores—como engajamento subjetivo, dinâmica interna e qualidade da mediação—mediaram o efeito da gamificação sobre a produtividade das equipes.

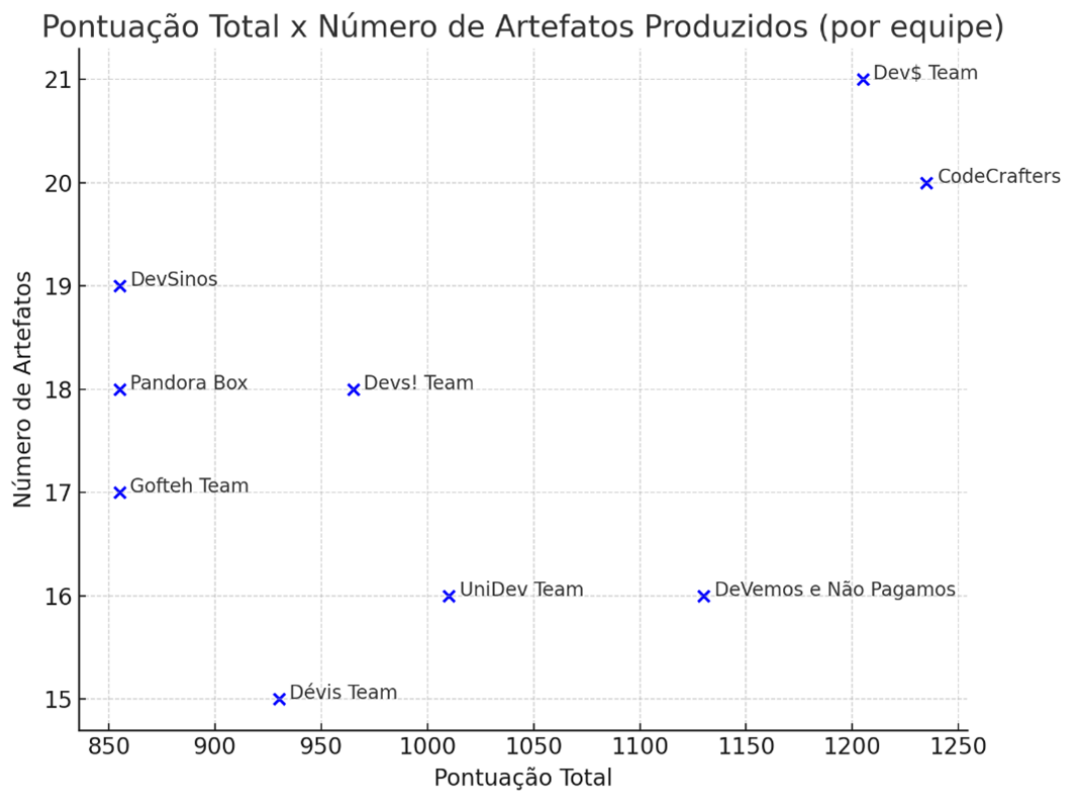
A Figura 21, apresentada a seguir, ilustra a relação entre a pontuação total obtida pelas equipes e o número de artefatos entregues ao final do semestre, permitindo visualizar, de forma intuitiva, a ausência de correlação linear entre as variáveis. A dispersão dos pontos reforça a ideia de que a diversidade e a completude das entregas dependem de múltiplas condições, para além do acúmulo de pontos no sistema gamificado.

Observa-se que equipes como CodeCrafters e Dev\$ Team, embora tenham atingido as maiores pontuações, não diferiram substancialmente em quantidade de artefatos quando comparadas a grupos com desempenho intermediário. Essa constatação indica que a gamificação atuou mais fortemente na regularidade das práticas e na variedade das entregas do que no simples aumento de volume, reforçando o papel da estratégia como mecanismo de orientação pedagógica, e não apenas de recompensa quantitativa. Como evidência, as medianas de ‘Processo Scrum’ e ‘Artefatos’ no pós-intervenção (ver Tabela 21) permanecem superiores às médias do período anterior, sugerindo melhoria sistemática e não apenas deslocamentos em equipes de topo.

Finalmente, é importante destacar que a completude e a diversidade não devem ser vistas apenas como indicadores de volume, mas também como expressões da compreensão ampliada do processo de engenharia de software. A produção de múltiplos artefatos demonstra uma internalização mais consistente das fases do desenvolvimento, reforçando a hipótese de que a gamificação pode funcionar como dispositivo estruturante da experiência formativa dos estudantes. Essa interpretação dialoga diretamente com os estudos que relacionam a motivação extrínseca à ampliação do esforço cognitivo em atividades de alta complexidade (RYAN; DECI, 2000b; HAMARI; KOIVISTO; SARSA, 2014). Do mesmo modo, a maior completude dos diagramas



**Figura 21:** Gráfico de dispersão entre pontuação total e número de artefatos



Fonte: Próprio autor

UML e a produção de múltiplos artefatos respondem à literatura que enfatiza a importância do feedback imediato e da competição saudável para a redução de inconsistências semânticas em modelos de software (LI; HEW; DU, 2024; ALMEIDA et al., 2023; LANGE; CHAUDRON; MUSKENS, 2006; EGYED, 2010; PETRE, 2013).

### 6.3.3 QP3.5: A Gamificação Melhora o Engajamento e a Colaboração Entre os Membros das Equipes?

Nesta subseção, referenciam-se os indicadores de engajamento e participação coletiva em conexão com o uso do leaderboard (20). A investigação dos efeitos da gamificação sobre o engajamento e a colaboração revelou indicativos consistentes de fortalecimento da dinâmica coletiva ao longo do semestre. Observações em sala, registros no ambiente virtual e relatos dos estudantes apontam para uma maior distribuição de responsabilidades, participação ativa nas entregas e comunicação entre os membros das equipes. Esse fenômeno parece estar relacionado à estrutura de incentivos simbólicos e à visibilidade pública do desempenho das equipes no sistema gamificado.

A rotação do papel de Scrum Master em todas as equipes funcionou como catalisador para a partilha de liderança, promovendo uma experiência mais equilibrada de autonomia e corresponsabilidade. Essa prática foi incentivada pelo plano de gamificação, que atribuía pontos adicionais à diversidade de papéis exercidos ao longo do projeto. Essa dimensão colabora para o desenvolvimento de competências transversais, como organização, mediação de conflitos e liderança situacional, fundamentais no contexto da Engenharia de Software.

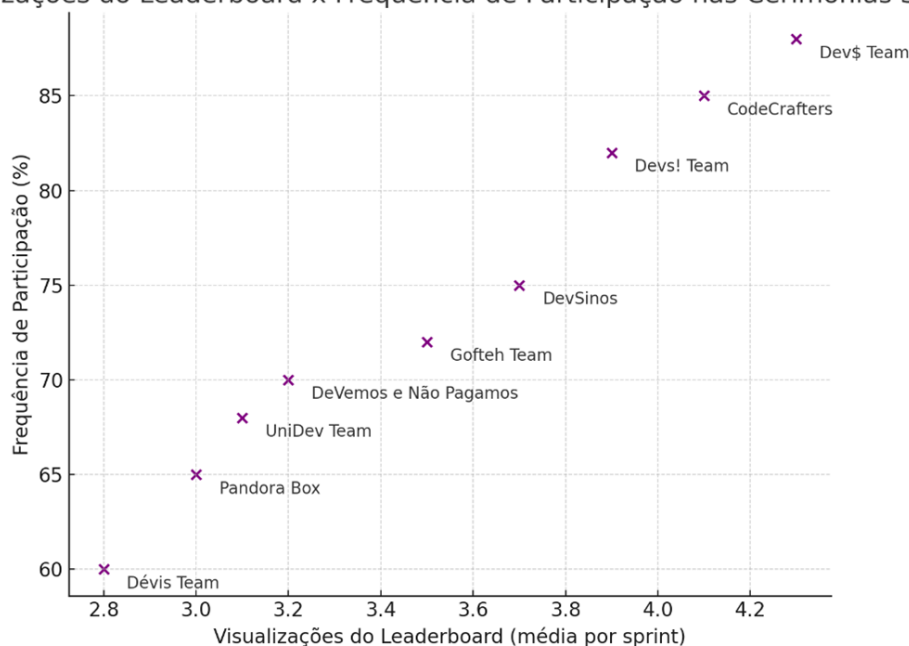
Outro fator relevante foi a utilização do quadro de líderes, visualizado em média 3,6 vezes por sprint por cada equipe. Esse indicador sugere não apenas interesse competitivo, mas também monitoramento constante do próprio desempenho e da evolução das demais equipes. A literatura aponta que essa visibilidade coletiva pode gerar sentimentos de pertencimento e incentivo à autoavaliação, o que se confirmou em diversos relatos dos estudantes.

As conquistas simbólicas (badges), como "Líder de Participação" e "Pioneiro do Planejamento", também tiveram papel importante na dinâmica das equipes. As equipes que obtiveram mais badges apresentaram maior regularidade nas entregas e melhor organização interna. Esses dados reforçam o entendimento de que recompensas simbólicas, quando bem alinhadas às práticas pedagógicas, tem potencial para incentivar o comportamento colaborativo. Adicionalmente, a variação positiva na Motivação (1–5) após a intervenção (ver Tabela 21) corrobora o aumento de engajamento autorreferido, alinhando indicadores subjetivos e comportamentais.

Por fim, é relevante notar que, apesar da presença de elementos competitivos, o estudo não identificou efeitos negativos sobre a cooperação entre os times. Ao contrário, os relatos indicam que o progresso de outras equipes era visto com entusiasmo, e não como uma ameaça. Esse resultado decorre possivelmente do cuidado em desenhar uma gamificação centrada em objetivos coletivos, com foco em reconhecimento e progresso compartilhado, e não apenas em

**Figura 22:** Relação entre visualizações do leaderboard e frequência de participação nas cerimônias Scrum

Visualizações do Leaderboard x Frequência de Participação nas Cerimônias Scrum



Fonte: Próprio autor

hierarquização de desempenho.

A análise quantitativa reforça esses achados qualitativos. O cruzamento entre o número médio de visualizações do quadro de líderes por *sprint* e a frequência de participação nas cerimônias *Scrum* mostra tendência positiva, sugerindo que equipes mais atentas ao ranking também registraram maior assiduidade e distribuição de falas durante as reuniões. Embora a correlação ( $\rho = 0.21$ ,  $p > 0.05$ ) não tenha alcançado significância estatística, a Figura 20 demonstra que os grupos que interagiram mais intensamente com o sistema gamificado apresentaram níveis superiores de participação coletiva, confirmando a importância da visibilidade contínua como mecanismo de autorregulação.

Esses resultados dialogam com os trabalhos que destacam a interação entre motivação extrínseca e satisfação de necessidades psicológicas básicas — autonomia, competência e pertencimento — como motor para o engajamento sustentado (GARACCIONE et al., 2025a; RYAN; DECI, 2000b; HAMARI; KOIVISTO; SARSA, 2014). A rotação de papéis e a ênfase em badges de colaboração refletem a categoria de dinâmicas, na qual a sensação de progresso coletivo e reconhecimento social fortalece a participação e a coesão do grupo (WERBACH; HUNTER, 2012). Ressalta-se que a menor dispersão nas métricas de processo e produto no pós-intervenção (Tabela 21) indica convergência de comportamento entre equipes, compatível com maior coordenação e colaboração.

Em conjunto, a combinação de incentivos simbólicos, feedback imediato e mecanismos de transparência demonstrou ser capaz de promover comportamentos colaborativos duradouros,

superando o risco de competição desagregadora, frequentemente apontado pela literatura crítica à gamificação (DETERDING et al., 2011a; ALVES, 2015; MARAND et al., 2025; GARACCI-ONE et al., 2025a; CAGNAZZO et al., 2023). Além disso, o fortalecimento da cooperação e da corresponsabilidade entre os estudantes evidencia que a gamificação, quando pedagogicamente orientada, pode funcionar como vetor de internalização de valores profissionais essenciais à prática da Engenharia de Software, como a interdependência construtiva, a comunicação efetiva e o compromisso com a qualidade dos artefatos produzidos.

#### 6.3.4 QP3.6: A Gamificação Afeta a Profundidade Analítica da Modelagem UML?

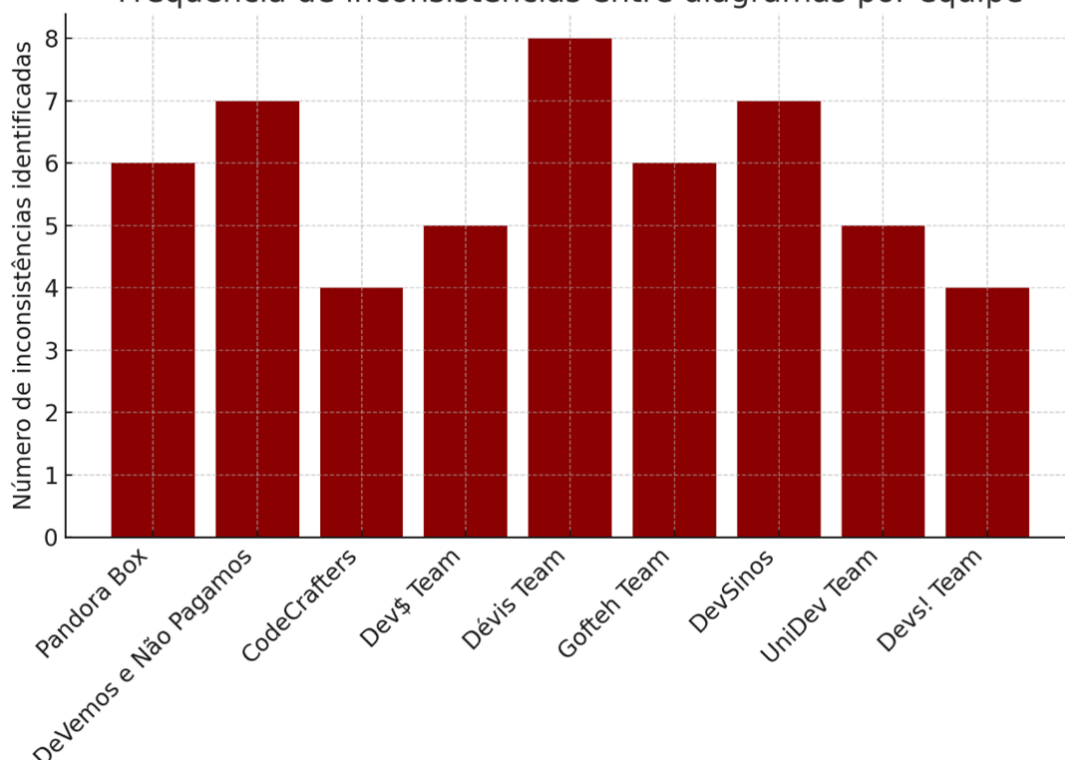
Nesta subseção, a ênfase recai sobre a consistência semântica entre os diagramas e a necessidade de incentivos específicos para a validação cruzada (Figura 20). A terceira questão de pesquisa concentrou-se em um aspecto crítico da formação em Engenharia de Software: a capacidade de produzir modelos conceitualmente consistentes, coesos e alinhados entre si. Embora os dados apontem avanço na quantidade e no formalismo dos diagramas UML, também revelam fragilidades persistentes na consistência semântica entre os modelos. A análise detalhada dos artefatos revelou que cinco das nove equipes apresentaram inconsistências entre diagramas de casos de uso e diagramas de sequência, como operações não modeladas, fluxos contraditórios e nomenclaturas incoerentes.

Além disso, apenas três equipes realizaram validação cruzada de forma sistemática, indicando que a articulação entre artefatos não foi devidamente internalizada como critério de qualidade. Os relatos dos estudantes evidenciam essa lacuna. Várias equipes reconheceram que o foco em acumular pontos levou à execução apressada de algumas tarefas, sem a devida atenção à coerência conceitual. Como expresso por um dos participantes: “A gente queria bater a pontuação, então às vezes fazia correndo para entregar” (Equipe 2). Esse depoimento ilustra como a ausência de critérios específicos relacionados à consistência intermodelos pode enfraquecer a profundidade analítica da aprendizagem.

Tais resultados estão em consonância com estudos anteriores, que apontam para uma ambivalência nos efeitos da gamificação: ao mesmo tempo em que amplia o engajamento, pode induzir abordagens mais instrumentais, especialmente quando os elementos gamificados não são diretamente vinculados à qualidade conceitual das tarefas (ALMEIDA et al., 2023; PORTO et al., 2020; FEICHAS; SEABRA, 2023; TONHÃO et al., 2023). A eficácia da gamificação, portanto, depende de seu alinhamento com os objetivos cognitivos mais complexos. Importa notar que a melhora agregada em Artefatos (Tabelas 19 e 20) não se traduziu automaticamente em consistência semântica entre diagramas (Figura 23), o que diferencia quantidade/variedade de entregas de profundidade analítica.

Uma possível solução consiste em incorporar badges e pontos que reconheçam especificamente a validação cruzada e a coerência semântica entre modelos. Tais mecanismos poderiam estimular a revisão entre pares e o uso de checklists conceituais antes das entregas. Além disso,

**Figura 23:** Frequência de inconsistências entre diagramas por equipe  
Frequência de inconsistências entre diagramas por equipe



Fonte: Próprio autor

a criação de desafios gamificados voltados à identificação de inconsistências poderia transformar a análise crítica em uma competência valorizada dentro do sistema de recompensa.

Em síntese, os achados indicam que a gamificação promove avanços em termos de volume e engajamento, mas apresenta limitações quanto à promoção da consistência semântica e da profundidade analítica, a menos que seus elementos sejam especificamente desenhados para esse fim. A revisão do sistema de pontuação com foco em qualidade conceitual pode ser um caminho promissor para equilibrar quantidade e profundidade na aprendizagem de modelagem UML.

A Figura 20, apresentada a seguir, ilustra a distribuição de inconsistências identificadas por equipe, permitindo visualizar a heterogeneidade na capacidade de manter coerência entre diagramas. Observa-se que equipes com maior pontuação global não foram necessariamente as que apresentaram menor número de inconsistências, reforçando que o acúmulo de pontos não garante, por si só, maior profundidade analítica.

Essa evidência dialoga diretamente com a fundamentação teórica discutida, especialmente com os trabalhos que destacam a necessidade de feedback imediato e validações intermodelos para reduzir contradições semânticas (MARCHEZAN et al., 2023a; CHAUDRON; HEIJSTEK; NUGROHO, 2012; BUCCHIARONE et al., 2021). Ao mesmo tempo, confirma sobre a importância de alinhar elementos extrínsecos de motivação aos objetivos cognitivos mais complexos,

evitando que a busca por recompensas leve a estratégias superficiais de cumprimento de tarefas (HAMARI; KOIVISTO; SARSA, 2014; HAMARI; HASSAN; DIAS, 2018; GARACCIONE et al., 2025a)

Assim, embora a gamificação tenha favorecido o engajamento e o aumento do número de artefatos, a ausência de incentivos específicos para a consistência semântica limitou o avanço na dimensão qualitativa da modelagem, reforçando a recomendação de incorporar métricas e badges orientados à validação cruzada em futuros desenhos pedagógicos. Essa constatação sugere que o impacto da gamificação é condicionado pela natureza dos critérios avaliativos que a sustentam, uma vez que, quando o sistema privilegia apenas quantidade e velocidade, há risco de reduzir a profundidade analítica do processo, contudo, quando o foco se desloca para a precisão conceitual e a coerência entre diagramas, a gamificação pode se tornar um instrumento eficaz de refinamento técnico e cognitivo.

## 6.4 Ameaças à Validade

Estudos de caso estão sujeitos a diversas ameaças à validade que devem ser discutidas de forma transparente (RUNESON; HÖST, 2009). Abaixo estão as principais ameaças identificadas e as estratégias adotadas para mitigá-las:

A validade interna refere-se à possibilidade de que fatores não controlados, além da intervenção gamificada, tenham influenciado os resultados, como o perfil dos estudantes, a experiência prévia com metodologias ágeis ou o estilo de ensino. Para mitigar essa ameaça, utilizou-se triangulação de dados, incluindo artefatos produzidos, observações em sala e relatos dos participantes, e procedeu-se à análise comparativa entre equipes, o que permitiu identificar padrões consistentes e reduzir interferências individuais. Além disso, buscou-se manter estabilidade do ambiente instrucional ao longo do semestre, com registro sistemático das intervenções docentes, de modo a assegurar rastreabilidade e consistência das condições experimentais.

A validade de construto (ou de construção) diz respeito ao risco de viés na forma como os fenômenos foram operacionalizados, particularmente conceitos como “engajamento” e “qualidade da modelagem”. Para reduzir esse risco, foram utilizadas rubricas previamente validadas para a avaliação dos artefatos e conduzida análise temática sistemática dos dados qualitativos (BRAUN; CLARKE, 2006). As rubricas e critérios de pontuação foram revisados por pares e explicitados no plano de gamificação, reforçando a correspondência entre os indicadores empíricos e os construtos teóricos. Essa estratégia visou garantir coerência semântica e mensuração consistente dos fenômenos observados.

A validade externa refere-se à capacidade de generalização dos resultados. Como o estudo foi conduzido em um único curso de graduação e em uma única instituição, a extrapolação deve ser feita com cautela. Para minimizar esse risco, descreveu-se detalhadamente o contexto da disciplina, o plano de gamificação e os instrumentos de coleta e análise de dados, de forma a possibilitar replicações em cenários análogos. Adicionalmente, especificaram-se a unidade

de análise (equipes), a duração e o calendário das sprints, de modo a permitir comparações e replicações literais e teóricas (HOLLWECK, 2015).

A validade de conclusão envolve o risco de interpretações incorretas ao relacionar evidências quantitativas e qualitativas. Para reduzir essa ameaça, foram apresentadas estatísticas descritivas e gráficos de apoio que ilustram tendências sem recorrer a inferências causais não suportadas pelos dados. Quando aplicável, reportaram-se tamanhos de efeito e significância, complementados pela triangulação entre resultados numéricos e relatos qualitativos, de forma a sustentar as inferências com múltiplas fontes de evidência.

Reconhece-se, ainda, que a proximidade do pesquisador com o contexto da disciplina pode introduzir vieses de observação. Para mitigar essa possibilidade, adotou-se a triangulação de fontes de evidência e a anonimização dos dados antes da análise qualitativa. Manteve-se, também, distinção clara entre os papéis de pesquisador e docente, restringindo as interpretações às evidências empíricas e submetendo os resultados à revisão por pares.

Essa estratégia assegura maior objetividade analítica e transparência no processo interpretativo, preservando a confiabilidade dos achados e a validade interna do estudo. Em síntese, as salvaguardas adotadas são coerentes com as recomendações metodológicas clássicas para estudos de caso (RUNESON; HÖST, 2009; HOLLWECK, 2015) e devem ser consideradas ao interpretar o alcance dos resultados obtidos.

## 6.5 Considerações Finais do Capítulo

Este estudo de caso investigou os efeitos da aplicação de um plano de gamificação em uma disciplina prática de Engenharia de Software, com foco na modelagem UML e nas práticas ágeis. A análise empírica, baseada em múltiplas fontes de evidência, indicou que a gamificação contribuiu significativamente para o aumento da completude dos artefatos, para o engajamento das equipes e para a organização das entregas. Rankings, pontos, badges e feedback contínuo funcionaram como catalisadores de motivação e planejamento, estruturando o fluxo de trabalho dos estudantes em sprints semanais.

No entanto, os resultados também evidenciam que o desenho da gamificação influencia diretamente a natureza da aprendizagem promovida. Enquanto houve ganho na quantidade e regularidade das entregas, a análise revelou lacunas na profundidade técnica, especialmente na coerência entre os diagramas UML. A ausência de incentivos específicos voltados à validação cruzada e à consistência semântica limitou o desenvolvimento de competências analíticas mais sofisticadas. Isso corrobora a literatura que aponta para a ambivalência dos efeitos da gamificação em tarefas cognitivamente complexas (FARIAS et al., 2013; ALMEIDA et al., 2023; SANTOS et al., 2024; TONHÃO et al., 2023).

Esses achados sugerem que o uso da gamificação em Engenharia de Software deve ser cuidadosamente alinhado com os objetivos pedagógicos e os princípios da engenharia disciplinada. Gamificar não pode significar apenas tornar o processo mais lúdico ou competitivo, mas, so-



bretudo, reforçar aquilo que é pedagogicamente essencial: o desenvolvimento da autonomia, da atenção ao detalhe e da responsabilidade técnica.

A análise integrada dos dados confirma que a combinação entre feedback imediato, transparência dos resultados e incentivos simbólicos potencializa a motivação extrínseca, mas não substitui a necessidade de estratégias voltadas à construção de conhecimento conceitual profundo. Essa constatação dialoga com os modelos de motivação e sobre o alinhamento entre mecânicas de jogo e objetivos de aprendizagem (RYAN; DECI, 2000b;a; WERBACH; HUNTER, 2015).

Como direções futuras, propomos: a introdução de desafios de consistência entre artefatos, com pontos e badges específicos para validação cruzada; o uso de revisões entre equipes como parte da mecânica gamificada, estimulando a leitura crítica de artefatos alheios; o desenvolvimento de dashboards inteligentes que retroalimentem os times com indicadores de qualidade técnica dos modelos; a realização de estudos longitudinais que avaliem o impacto da gamificação em ciclos formativos completos, permitindo observar a evolução do engajamento e da proficiência técnica ao longo de múltiplos semestres; o uso de revisões entre equipes como parte da mecânica gamificada, estimulando a leitura crítica de artefatos alheios; e a condução de experimentos comparativos entre diferentes tipos de gamificação (ex: competitiva vs. colaborativa) em contextos similares.

Essas iniciativas podem contribuir para refinar o equilíbrio entre quantidade, qualidade e profundidade da aprendizagem, oferecendo evidências mais robustas para orientar futuras práticas pedagógicas em cursos de Engenharia de Software. Destaca-se ainda que os resultados deste estudo estão sendo organizados para dar origem a manuscritos a serem submetidos a periódicos e eventos científicos nacionais e internacionais nas áreas de Engenharia de Software e Educação em Computação, concentrando em três eixos principais: (1) a análise empírica do impacto da gamificação na qualidade de artefatos UML; (2) a avaliação de estratégias gamificadas para o ensino de metodologias ágeis e (3) a proposição de um modelo replicável de gamificação aplicado a contextos educacionais técnicos.

Ademais, a ampliação da investigação para contextos institucionais diversos e a integração com ferramentas de análise automatizada de código e modelagem podem fortalecer a generalização dos achados e consolidar um referencial empírico mais abrangente sobre a efetividade da gamificação no ensino técnico e superior.



## 7 CONCLUSÕES E TRABALHOS FUTUROS

A presente tese buscou compreender, por meio de uma abordagem empírica multiestudo, como a gamificação pode apoiar a aprendizagem e a prática de modelagem de software baseada em UML, bem como investigar seu impacto sobre a qualidade dos modelos produzidos. Os estudos conduzidos revelaram evidências convergentes de que, embora a UML permaneça amplamente reconhecida na indústria e no meio acadêmico, sua adoção prática ainda é limitada por fatores como falta de tempo, ausência de padronização no ensino, dificuldades de avaliação dos modelos e barreiras culturais que restringem sua utilização sistemática. Nesse contexto, a gamificação emergiu como uma alternativa promissora para mitigar esses desafios, ampliando o engajamento dos aprendizes, promovendo maior consistência nas atividades de modelagem e oferecendo mecanismos motivacionais capazes de sustentar processos de aprendizagem mais profundos.

A partir dessas lacunas, esta tese apresentou três contribuições principais. A primeira consistiu em mapear o estado da prática da modelagem UML na indústria, revelando fatores que influenciam sua adoção e evidenciando a necessidade de novas estratégias de ensino e apoio ao uso da linguagem. A segunda contribuição foi o desenvolvimento do ModelGame, um modelo de qualidade gamificado que sistematiza atributos e métricas para avaliação de modelos UML em atividades educacionais, suprimindo uma lacuna identificada na literatura e oferecendo um referencial para instrutores avaliadores. Por fim, a terceira contribuição está associada à execução de experimento controlado e estudo de caso que demonstraram, de forma quantitativa e qualitativa, que elementos gamificados podem influenciar positivamente a qualidade dos modelos, promovendo maior completude, consistência e clareza na elaboração dos artefatos, além de ampliar a motivação e o envolvimento dos participantes nos processos de modelagem.

Apesar das contribuições alcançadas, algumas limitações devem ser reconhecidas. Os estudos envolveram amostras específicas — estudantes e equipes de desenvolvimento — o que pode restringir a generalização dos resultados para organizações de larga escala ou contextos industriais altamente regulados. Além disso, os modelos gamificados foram aplicados em cenários controlados de ensino; portanto, seu impacto em projetos reais de engenharia de software ainda requer investigação mais aprofundada. Essas limitações, contudo, não reduzem o valor dos achados, mas apontam caminhos frutíferos para pesquisas futuras.

Com base nessas considerações, diversos desdobramentos podem ser explorados. Estudos futuros podem ampliar o uso do ModelGame em ambientes industriais, investigando sua efetividade em equipes profissionais, pipelines ágeis, desenvolvimento colaborativo e integração contínua. É igualmente promissor investigar como diferentes perfis de aprendizes respondem a elementos gamificados, explorando fatores motivacionais, cognitivos e comportamentais que influenciam a produção de modelos UML. Outra direção relevante consiste em expandir o modelo proposto para outras linguagens de modelagem e metodologias contemporâneas, examinando seu potencial em áreas como BPMN, SysML, Domain-Driven Design e Arquitetura de Soft-

ware. Por fim, a evolução de tecnologias emergentes — como inteligência artificial generativa, análise automatizada de modelos e plataformas adaptativas de aprendizagem — abre novas possibilidades para integrar gamificação, avaliação contínua e suporte inteligente à modelagem de software.

Em síntese, esta tese representa um passo importante para o entendimento do papel da gamificação na prática e no ensino da modelagem de software. Ao articular investigação empírica rigorosa, desenvolvimento de modelos aplicáveis e análise sistemática dos efeitos da gamificação na qualidade dos artefatos, este trabalho contribui para o avanço da Engenharia de Software baseada em evidências e para a consolidação de abordagens inovadoras que fortalecem o aprendizado, a motivação e a capacidade analítica de futuros profissionais da área. Espera-se que os resultados aqui apresentados inspirem novas pesquisas e práticas que ampliem a integração entre gamificação e modelagem, promovendo ambientes de aprendizagem mais eficazes, dinâmicos e alinhados às demandas contemporâneas da Engenharia de Software.

## 7.1 Artigos Publicados

- *A Survey on the Use of UML in the Brazilian Industry* foi apresentado na trilha principal (pesquisa) do 35th Brazilian Symposium on Software Engineering (SBES 2021), sendo eleito um dos cinco melhores papers (JÚNIOR; FARIAS; SILVA, 2021).
- *ModelGame: A Quality Model for Gamified Software Modeling Learning* foi apresentado no 15th Brazilian Symposium on Software Components, Architectures, and Reuse (SBCARS 2021) (JÚNIOR; FARIAS, 2021).
- *On The Use of UML in The Brazilian Industry: A Survey* foi publicado no *Journal of Software Engineering Research and Development* (2022) (JÚNIOR; FARIAS; SILVA, 2022).
- *The Impact of Gamification on Detecting Inconsistencies in UML Class and Sequence Diagrams: A Controlled Experiment in an Educational Setting* foi submetido ao *International Journal on Software and Systems Modeling (SoSyM)* e está em fase de correções (2025).

## REFERÊNCIAS

- ADCOCK, R.; ALEF, E.; AMATO, B.; ARDIS, M.; BERNSTEIN, L.; BOEHM, B.; BOURQUE, P.; BRACKETT, J.; CANTOR, M.; CASSEL, L. et al. **Curriculum guidelines for graduate degree programs in software engineering**. [S.l.]: ACM, 2009.
- AKDUR, D.; SAY, B.; DEMIRÖRS, O. Modeling cultures of the embedded software industry: feedback from the field. **Software and Systems Modeling**, [S.l.], v. 20, n. 2, p. 447–467, 2021.
- ALMEIDA, C.; KALINOWSKI, M.; UCHÔA, A.; FEIJÓ, B. Negative effects of gamification in education software: systematic mapping and practitioner perceptions. **Information and Software Technology**, [S.l.], v. 156, p. 107142, 2023.
- ALSADOON, E.; ALKHAWAJAH, A.; SUHAIM, A. B. Effects of a gamified learning environment on students' achievement, motivations, and satisfaction. **Heliyon**, [S.l.], v. 8, n. 8, p. e10249, 2022.
- ALVES, F. **Gamification**: como criar experiências de aprendizagem engajadoras. [S.l.]: DVS editora, 2015.
- ARDIS, M.; BUDGEN, D.; HISLOP, G. W.; OFFUTT, J.; SEBERN, M.; VISSER, W. SE 2014: curriculum guidelines for undergraduate degree programs in software engineering. **Computer**, [S.l.], v. 48, n. 11, p. 106–109, 2015.
- BEECHAM, S.; BADDOO, N.; HALL, T.; ROBINSON, H.; SHARP, H. Motivation in Software Engineering: a systematic literature review. **Information and software technology**, [S.l.], v. 50, n. 9-10, p. 860–878, 2008.
- BEKK, M.; EPPMANN, R.; KLEIN, K.; VÖLCKNER, F. All that glitters is not gold: an investigation into the undesired effects of gamification and how to mitigate them through gamification design. **International Journal of Research in Marketing**, [S.l.], v. 39, n. 4, p. 1059–1081, 2022.
- BISTA, S. K.; NEPAL, S.; PARIS, C. Engagement and cooperation in social networks: do benefits and rewards help? In: IEEE 11TH INTERNATIONAL CONFERENCE ON TRUST, SECURITY AND PRIVACY IN COMPUTING AND COMMUNICATIONS, 2012., 2012. **Anais...** [S.l.: s.n.], 2012. p. 1405–1410.
- BISTA, S. K.; NEPAL, S.; PARIS, C.; COLINEAU, N. Gamification for online communities: a case study for delivering government services. **international Journal of Cooperative information Systems**, [S.l.], v. 23, n. 02, p. 1441002, 2014.
- BLAHA, M.; RUMBAUGH, J. **Modelagem e projetos baseados em objetos com UML 2**. [S.l.]: Elsevier, 2006.
- BOOCH, G. UML in action. **Communications of the ACM**, [S.l.], v. 42, n. 10, p. 26–28, 1999.
- BOOCH, G. **The unified modeling language user guide**. [S.l.]: Pearson Education India, 2005.

BOOCH, G. Object-Oriented Development, Revisited. **IEEE Transactions on Software Engineering**, [S.l.], n. 01, p. 1–4, 2025.

BOOCH, G.; MAKSIMCHUK, R. A.; ENGLE, M. W.; YOUNG, B. J.; CONNALLAN, J.; HOUSTON, K. A. Object-oriented analysis and design with applications. **ACM SIGSOFT software engineering notes**, [S.l.], v. 33, n. 5, p. 29–29, 2008.

BRAUN, V.; CLARKE, V. Using thematic analysis in psychology. **Qualitative research in psychology**, [S.l.], v. 3, n. 2, p. 77–101, 2006.

BUCCHIARONE, A.; CICCIOZZI, F.; LAMBERS, L.; PIERANTONIO, A.; TICHY, M.; TISI, M.; WORTMANN, A.; ZAYTSEV, V. What Is the Future of Modeling? **IEEE software**, [S.l.], v. 38, n. 2, p. 119–127, 2021.

BUCCHIARONE, A.; GUIDOLIN, T.; FASOL, L.; SCHIAVO, G.; KIENZLE, J.; GERARD, S.; NÉGRIER, D.; MARTORELLA, T. PolyGloT-UML: a gamified framework for enhancing uml learning paths. In: OF THE ACM/IEEE 27TH INTERNATIONAL CONFERENCE ON MODEL DRIVEN ENGINEERING LANGUAGES AND SYSTEMS, 2024. **Proceedings...** [S.l.: s.n.], 2024. p. 31–35.

BUCCHIARONE, A.; SAVARY-LEBLANC, M.; LE PALLEC, X.; BRUEL, J.-M.; CICHETTI, A.; CABOT, J.; GÉRARD, S. Gamifying model-based engineering: the papygame tool. **Science of Computer Programming**, [S.l.], v. 230, p. 102974, 2023.

BUCCHIARONE, A.; SAVARY-LEBLANC, M.; LE PALLEC, X.; CICHETTI, A.; GÉRARD, S.; BASSANELLI, S.; GINI, F.; MARCONI, A. Gamifying model-based engineering: the papygame experience. **Software and Systems Modeling**, [S.l.], v. 22, n. 4, p. 1369–1389, 2023.

CAGNAZZO, C.; GARACCIONE, G.; COPPOLA, R.; ARDITO, L.; TORCHIANO, M. UMLegend: a gamified learning tool for conceptual modeling with uml class diagrams. In: INTERNATIONAL WORKSHOP ON GAMIFICATION IN SOFTWARE DEVELOPMENT, VERIFICATION, AND VALIDATION, 2., 2023. **Proceedings...** [S.l.: s.n.], 2023. p. 2–5.

CALL, M. W.; FOX, E.; SPRINT, G. Gamifying software engineering tools to motivate computer science students to start and finish programming assignments earlier. **IEEE Transactions on Education**, [S.l.], v. 64, n. 4, p. 423–431, 2021.

CAMMAERTS, F.; SNOECK, M. Towards Leveraging Gamified Code-Testing for Effective Model. In: ADVANCES IN CONCEPTUAL MODELING, 2025. **Anais...** [S.l.: s.n.], 2025. v. 14932, p. 233.

CARVALHO, L. d. F. S. et al. BLOOM BLAST Object Oriented Management: uma solução integrada para gerenciamento dos resultados do blast por meio de um paradigma orientado a objetos. , [S.l.], 2003.

CERASOLI, C. P.; NICKLIN, J. M.; FORD, M. T. Intrinsic motivation and extrinsic incentives jointly predict performance: a 40-year meta-analysis. **Psychological bulletin**, [S.l.], v. 140, n. 4, p. 980, 2014.

CHAUDRON, M. R.; HEIJSTEK, W.; NUGROHO, A. How effective is UML modeling? **Software & Systems Modeling**, [S.l.], v. 11, n. 4, p. 571–580, 2012.

CHEN, L. Continuous delivery: huge benefits, but challenges too. **IEEE Software**, [S.l.], v. 32, n. 2, p. 50–54, 2015.

CHEN, L. Continuous delivery: overcoming adoption challenges. **Journal of Systems and Software**, [S.l.], v. 128, p. 72–86, 2017.

CHEN, Z. The influence of school's reward systems on students' development. **Journal of Education, Humanities and Social Sciences**, [S.l.], v. 8, p. 1822–1827, 2023.

CICCHETTI, A.; CICCOCCHI, F.; CARLSON, J. Software Evolution Management: industrial practices. In: ME@ MODELS, 2016. **Anais...** [S.l.: s.n.], 2016. p. 8–13.

CICCOCCHI, F.; MALAVOLTA, I.; SELIC, B. Execution of UML models: a systematic review of research and practice. **Software & Systems Modeling**, [S.l.], v. 18, n. 3, p. 2313–2360, 2019.

CLANTON HARPINE, E.; CLANTON HARPINE, E. Is intrinsic motivation better than extrinsic motivation? **Group-centered prevention in mental health: Theory, training, and practice**, [S.l.], p. 87–107, 2015.

COSENTINO, V.; GÉRARD, S.; CABOT SAGRERA, J. A model-based approach to gamify the learning of modeling. In: 2017. **Anais...** [S.l.: s.n.], 2017.

COSENTINO, V.; GÉRARD, S.; CABOT SAGRERA, J. A model-based approach to gamify the learning of modeling. In: 2017. **Anais...** [S.l.: s.n.], 2017.

DÆHLI, O.; KRISTOFFERSEN, B.; LAUVÅS JR, P.; SANDNES, T. Exploring Feedback and Gamification in a Data Modeling Learning Tool. **Electronic Journal of e-Learning**, [S.l.], v. 19, n. 6, p. 559–574, 2021.

DECI, E. L.; RYAN, R. M. The "what" and "why" of goal pursuits: human needs and the self-determination of behavior. **Psychological inquiry**, [S.l.], v. 11, n. 4, p. 227–268, 2000.

DECI, E. L.; RYAN, R. M. **Handbook of self-determination research**. [S.l.]: University Rochester Press, 2004.

DECI, E. L.; RYAN, R. M. Self-determination theory: a macrotheory of human motivation, development, and health. **Canadian psychology/Psychologie canadienne**, [S.l.], v. 49, n. 3, p. 182, 2008.

DESTEFANIS, G.; ORTU, M.; COUNSELL, S.; SWIFT, S.; MARCHESI, M.; TONELLI, R. Software development: do good manners matter? **PeerJ Computer Science**, [S.l.], v. 2, p. e73, 2016.

DETERDING, S. Gamification Absolved. **Gamification Research Network**. Online verfügbar unter <http://gamification-research.org/2014/08/gamification-absolved-2/>, zuletzt geprüft am, [S.l.], v. 6, p. 2014, 2014.

DETERDING, S.; DIXON, D.; KHALED, R.; NACKE, L. From game design elements to gamefulness: defining "gamification". In: INT. ACADEMIC MINDTREK CONFERENCE: ENVISIONING FUTURE MEDIA ENVIRONMENTS, 15., 2011. **Anais...** [S.l.: s.n.], 2011. p. 9–15.

DETERDING, S.; SICART, M.; NACKE, L.; O'HARA, K.; DIXON, D. Gamification. using game-design elements in non-gaming contexts. In: **CHI'11 extended abstracts on human factors in computing systems**. [S.l.: s.n.], 2011. p. 2425–2428.

DORI, D. Why significant UML change is unlikely. **Communications of the ACM**, [S.l.], v. 45, n. 11, p. 82–85, 2002.

DORLING, A.; MCCAFFERY, F. The gamification of SPICE. In: INTERNATIONAL CONFERENCE ON SOFTWARE PROCESS IMPROVEMENT AND CAPABILITY DETERMINATION, 2012. **Anais...** [S.l.: s.n.], 2012. p. 295–301.

DUBOIS, D. J.; TAMBURRELLI, G. Understanding gamification mechanisms for software development. In: JOINT MEETING ON FOUNDATIONS OF SOFTWARE ENGINEERING, 2013., 2013. **Proceedings...** [S.l.: s.n.], 2013. p. 659–662.

DZIDEK, W. J.; ARISHOLM, E.; BRIAND, L. C. A realistic empirical evaluation of the costs and benefits of UML in software maintenance. **IEEE Transactions on software engineering**, [S.l.], v. 34, n. 3, p. 407–432, 2008.

EASTERBROOK, S.; SINGER, J.; STOREY, M.-A.; DAMIAN, D. Selecting empirical methods for software engineering research. In: **Guide to advanced empirical software engineering**. [S.l.]: Springer, 2008. p. 285–311.

EGYED, A. Automatically detecting and tracking inconsistencies in software design models. **IEEE Transactions on Software Engineering**, [S.l.], v. 37, n. 2, p. 188–204, 2010.

ELAZHARY, O.; WERNER, C.; LI, Z. S.; LOWLIND, D.; ERNST, N. A.; STOREY, M.-A. Uncovering the benefits and challenges of continuous integration practices. **IEEE Transactions on Software Engineering**, [S.l.], 2021.

FACEY-SHAW, L.; SPECHT, M.; ROSMALEN, P. van; BARTLEY-BRYAN, J. Do badges affect intrinsic motivation in introductory programming students? **Simulation & Gaming**, [S.l.], v. 51, n. 1, p. 33–54, 2020.

FALESSI, D.; JURISTO, N.; WOHLIN, C.; TURHAN, B.; MÜNCH, J.; JEDLITSCHKA, A.; OIVO, M. Empirical software engineering experts on the use of students and professionals in experiments. **Empirical Software Engineering**, [S.l.], v. 23, p. 452–489, 2018.

FARIAS, K. Empirical evaluation of effort on composing design models. **arXiv preprint arXiv:1610.09012**, [S.l.], 2016.

FARIAS, K.; GARCIA, A.; LUCENA, C. Evaluating the impact of aspects on inconsistency detection effort: a controlled experiment. In: INTERNATIONAL CONFERENCE ON MODEL DRIVEN ENGINEERING LANGUAGES AND SYSTEMS, 2012. **Anais...** [S.l.: s.n.], 2012. p. 219–234.

FARIAS, K.; GARCIA, A.; LUCENA, C.; GONZAGA JR, L.; COSTA, C. A. da; ROSA RIGHI, R. da; BASSO, F.; OLIVEIRA, T. Towards a quality model for model composition effort. In: ANNUAL ACM SYMPOSIUM ON APPLIED COMPUTING, 29., 2014. **Proceedings...** [S.l.: s.n.], 2014. p. 1181–1183.

FARIAS, K.; GARCIA, A.; WHITTLE, J.; FLACH GARCIA CHAVEZ, C. von; LUCENA, C. Evaluating the effort of composing design models: a controlled experiment. **Software & Systems Modeling**, [S.l.], v. 14, n. 4, p. 1349–1365, 2015.

FARIAS, K.; GARCIA, A.; WHITTLE, J.; LUCENA, C. Analyzing the effort of composing design models of large-scale software in industrial case studies. In: INTERNATIONAL CONFERENCE ON MODEL DRIVEN ENGINEERING LANGUAGES AND SYSTEMS, 2013. **Anais...** [S.l.: s.n.], 2013. p. 639–655.

FARIAS, K.; GONÇALES, L.; BISCHOFF, V.; SILVA, B. C. da; GUIMARÃES, E. T.; NOGLE, J. On the UML use in the Brazilian industry: a state of the practice survey (s). In: SEKE, 2018. **Anais...** [S.l.: s.n.], 2018. p. 372–371.

FARIAS, K.; OLIVEIRA CAVALCANTE, T. de; JOSÉ GONÇALES, L.; BISCHOFF, V. UML2Merge: a uml extension for model merging. **IET Software**, [S.l.], v. 13, n. 6, p. 575–586, 2019.

FEICHAS, F. A.; SEABRA, R. D. Evaluation of Perception of Use of a Gamified Platform from the Student Perspective: an approach for studying unified modeling language. **Informatics in Education**, [S.l.], 2023.

FERNANDEZ-RIO, J.; ZUMAJO-FLORES, M.; FLORES-AGUILAR, G. Motivation, basic psychological needs and intention to be physically active after a gamified intervention programme. **European Physical Education Review**, [S.l.], v. 28, n. 2, p. 432–445, 2022.

FERNÁNDEZ-SAEZ, A.; GENERO, M.; NELSON, J.; POELS, G.; PIATTINI, M. A systematic literature review on the quality of UML models. **J. Data. Manage**, [S.l.], v. 22, n. 3, p. 46–70, 2012.

FERNÁNDEZ-SÁEZ, A. M.; CAIVANO, D.; GENERO, M.; CHAUDRON, M. R. On the use of UML documentation in software maintenance: results from a survey in industry. In: ACM/IEEE 18TH INT. CONF. ON MODEL DRIVEN ENGINEERING LANGUAGES AND SYSTEMS (MODELS), 2015., 2015. **Anais...** [S.l.: s.n.], 2015. p. 292–301.

FERNÁNDEZ-SÁEZ, A. M.; CHAUDRON, M. R.; GENERO, M. An industrial case study on the use of UML in software maintenance and its perceived benefits and hurdles. **Empirical Software Engineering**, [S.l.], v. 23, n. 6, p. 3281–3345, 2018.

FITZGERALD, B.; STOL, K.-J. Continuous software engineering: a roadmap and agenda. **Journal of Systems and Software**, [S.l.], v. 123, p. 176–189, 2017.

FORRESTER, J. W. **World dynamics**. [S.l.]: Wright-Allen Press, 1971.

FORWARD, A.; BADREDDIN, O.; LETHBRIDGE, T. C. Perceptions of software modeling: a survey of software practitioners. In: MDD (C2M: EEMDD), 5., 2010. **Anais...** [S.l.: s.n.], 2010.

FORWARD, A.; LETHBRIDGE, T. C. Problems and opportunities for model-centric versus code-centric software development: a survey of software professionals. In: MODELS IN SOFTWARE ENGINEERING, 2008., 2008. **Proceedings...** [S.l.: s.n.], 2008. p. 27–32.

FOWLER, M.; SCOTT, K. **UML distilled**: a brief guide to the standard object. [S.l.]: Addison-Wesley Professional, 2003.

FRANÇA, A. C. C. A theory of motivation and satisfaction of software engineers. , [S.l.], 2014.

- FRANCE, R.; RUMPE, B. Model-driven development of complex software: a research roadmap. In: FUTURE OF SOFTWARE ENGINEERING (FOSE'07), ICSE'07, 2007. **Anais...** [S.l.: s.n.], 2007. p. 37–54.
- FU, K.; LIU, Z.; REN, X.; ZHANG, S. 7. Design and research of educational mode in context of teaching gamification. **Entertainment Computing**, [S.l.], 2024.
- GARACCIONE, G.; COPPOLA, R.; ARDITO, L. Gamifying business process modeling education: a longitudinal study. In: INTERNATIONAL CONFERENCE ON EVALUATION AND ASSESSMENT IN SOFTWARE ENGINEERING, 28., 2024. **Proceedings...** [S.l.: s.n.], 2024. p. 580–589.
- GARACCIONE, G.; COPPOLA, R.; ARDITO, L.; TORCHIANO, M. Gamification of business process modeling education: an experimental analysis. **Software and Systems Modeling**, [S.l.], v. 23, n. 6, p. 1569–1594, 2024.
- GARACCIONE, G.; COPPOLA, R.; ARDITO, L.; TORCHIANO, M. Gamification of conceptual modeling education with UML class diagrams: an experimental analysis. **Software and Systems Modeling**, [S.l.], p. 1–32, 2025.
- GARACCIONE, G.; COPPOLA, R.; ARDITO, L.; TORCHIANO, M. Gamification of conceptual modeling education: an analysis of productivity and students' perception. **Software Quality Journal**, [S.l.], v. 33, n. 1, p. 1–19, 2025.
- GARCIA, F.; PEDREIRA, O.; PIATTINI, M.; CERDEIRA-PENA, A.; PENABAD, M. A framework for gamification in software engineering. **Journal of Systems and Software**, [S.l.], v. 132, p. 21–40, 2017.
- GHAJ, A. Facts and Fallacies of Software Engineering. **Software Quality Professional**, [S.l.], v. 5, n. 3, p. 38, 2003.
- GIRARDI, D.; LANUBILE, F.; NOVIELLI, N.; SEREBRENIK, A. Emotions and Perceived Productivity of Software Developers at the Workplace. **IEEE Transactions on Software Engineering**, [S.l.], p. 1–1, 2021.
- GONÇALES, L. J.; FARIAS, K.; BISCHOFF, V. Towards a hybrid approach to measure similarity between UML models. In: XV BRAZILIAN SYMPOSIUM ON INFORMATION SYSTEMS, 2019. **Proceedings...** [S.l.: s.n.], 2019. p. 1–8.
- GORSCHKE, T.; TEMPERO, E.; ANGELIS, L. On the use of software design models in software development practice: an empirical investigation. **Journal of Systems and Software**, [S.l.], v. 95, p. 176–193, 2014.
- HAMARI, J.; HASSAN, L.; DIAS, A. Gamification, quantified-self or social networking? Matching users' goals with motivational technology. **User Modeling and User-Adapted Interaction**, [S.l.], v. 28, n. 1, p. 35–74, 2018.
- HAMARI, J.; KOIVISTO, J.; SARSA, H. Does gamification work?—a literature review of empirical studies on gamification. In: HAWAII INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES, 2014., 2014. **Anais...** [S.l.: s.n.], 2014. p. 3025–3034.



HELDAL, R.; PELLICCIONE, P.; ELIASSON, U.; LANTZ, J.; DEREHAG, J.; WHITTLE, J. Descriptive vs prescriptive models in industry. In: OF THE ACM/IEEE 19TH INTERNATIONAL CONFERENCE ON MODEL DRIVEN ENGINEERING LANGUAGES AND SYSTEMS, 2016. **Proceedings...** [S.l.: s.n.], 2016. p. 216–226.

HERNÁNDEZ, L.; MUÑOZ, M.; MEJIA, J.; PEÑA, A. Gamification in software engineering teamworks: a systematic literature review. In: INTERNATIONAL CONFERENCE ON SOFTWARE PROCESS IMPROVEMENT (CIMPS), 2016., 2016. **Anais...** [S.l.: s.n.], 2016. p. 1–8.

HO-QUANG, T.; HEBIG, R.; ROBLES, G.; CHAUDRON, M. R.; FERNANDEZ, M. A. Practices and perceptions of UML use in open source projects. In: ICSE: SOFTWARE ENGINEERING IN PRACTICE TRACK, 39., 2017. **Anais...** [S.l.: s.n.], 2017. p. 203–212.

HOLLWECK, T. Robert K. Yin.(2014). Case Study Research Design and Methods . **Canadian Journal of Program Evaluation**, [S.l.], v. 30, n. 1, p. 108–110, 2015.

HUGOS, M. **Enterprise games**: using game mechanics to build a better business. [S.l.]: "O'Reilly Media, Inc.", 2012.

HUOTARI, K.; HAMARI, J. A definition for gamification: anchoring gamification in the service marketing literature. **Electronic Markets**, [S.l.], v. 27, n. 1, p. 21–31, 2017.

HUTCHINSON, J.; ROUNCEFIELD, M.; WHITTLE, J. Model-driven engineering practices in industry. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, 33., 2011. **Proceedings...** [S.l.: s.n.], 2011. p. 633–642.

HUTCHINSON, J.; WHITTLE, J.; ROUNCEFIELD, M.; KRISTOFFERSEN, S. Empirical assessment of MDE in industry. In: OF THE 33RD INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, 2011. **Proceedings...** [S.l.: s.n.], 2011. p. 471–480.

JACKSON, D. Alloy: a language and tool for exploring software designs. **Commun. ACM**, New York, NY, USA, v. 62, n. 9, p. 66–76, aug 2019.

JACOBSON, I.; BOOCH, G.; RUMBAUGH, J. The unified process. **Ieee Software**, [S.l.], v. 16, n. 3, p. 96, 1999.

JACOBSON, L.; BOOCH, J. R. G. **The Unified Modeling Language Reference Manual**. [S.l.]: Addison-Wesley, 2021.

JÚNIOR, E.; FARIAS, K. ModelGame: a quality model for gamified software modeling learning. In: BRAZILIAN SYMPOSIUM ON SOFTWARE COMPONENTS, ARCHITECTURES, AND REUSE, 15., 2021. **Anais...** [S.l.: s.n.], 2021. p. 100–109.

JÚNIOR, E.; FARIAS, K.; SILVA, B. A Survey on the Use of UML in the Brazilian Industry. In: BRAZILIAN SYMPOSIUM ON SOFTWARE ENGINEERING, 2021. **Anais...** [S.l.: s.n.], 2021. p. 275–284.

JÚNIOR, E. W.; FARIAS, K.; SILVA, B. da. On the use of uml in the brazilian industry: a survey. **Journal of Software Engineering Research and Development**, [S.l.], v. 10, p. 10–1, 2022.

JURGELAITIS, M.; DRUNGILAS, V.; ČEPONIENĖ, L. Gamified moodle course for teaching UML. **Baltic journal of modern computing**, [S.l.], v. 6, n. 2, p. 119–127, 2018.

KHELLADI, D. E.; KRETSCHMER, R.; EGYED, A. Detecting and exploring side effects when repairing model inconsistencies. In: ACM INT. CONF. ON SOFTWARE LANGUAGE ENGINEERING, 12., 2019. **Anais...** [S.l.: s.n.], 2019. p. 113–126.

KIENZLE, J.; ZSCHALER, S.; BARNETT, W.; SAĞLAM, T.; BUCCHIARONE, A.; ABRAHÃO, S.; SYRIANI, E.; KOLOVOS, D.; LETHBRIDGE, T.; MUSTAFIZ, S. et al. Requirements for modelling tools for teaching. **Software and Systems Modeling**, [S.l.], v. 23, n. 5, p. 1055–1073, 2024.

KITCHENHAM, B. A.; PFLEEGER, S. L. Personal opinion surveys. In: **Guide to advanced empirical software engineering**. [S.l.]: Springer, 2008. p. 63–92.

KLARE, H.; KRAMER, M.; LANGHAMMER, M.; WERLE, D.; BURGER, E.; REUSSNER, R. Enabling consistency in view-based system development—the vitruvius approach. **Journal of Systems and Software**, [S.l.], v. 171, p. 110815, 2021.

KLOCK, A. C. T.; CARVALHO, M. F. de; ROSA, B. E.; GASPARINI, I. Análise das técnicas de Gamificação em Ambientes Virtuais de Aprendizagem. **RENOTE**, [S.l.], v. 12, n. 2, 2014.

KOBRYN, C. Will UML 2.0 be agile or awkward? **Communications of the ACM**, [S.l.], v. 45, n. 1, p. 107–110, 2002.

KOIVISTO, J.; HAMARI, J. The rise of motivational information systems: a review of gamification research. **International Journal of Information Management**, [S.l.], v. 45, p. 191–210, 2019.

KRETSCHMER, R.; KHELLADI, D. E.; LOPEZ-HERREJON, R. E.; EGYED, A. Consistent change propagation within models. **Software and Systems Modeling**, [S.l.], v. 20, n. 2, p. 539–555, 2021.

KUHN, A.; MURPHY, G. C.; THOMPSON, C. A. An exploratory study of forces and frictions affecting large-scale model-driven development. In: INT. CONF. ON MODEL DRIVEN ENGINEERING LANGUAGES AND SYSTEMS, 2012. **Anais...** [S.l.: s.n.], 2012. p. 352–367.

LANDERS, R. N. Gamification misunderstood: how badly executed and rhetorical gamification obscures its transformative potential. **Journal of Management inquiry**, [S.l.], v. 28, n. 2, p. 137–140, 2019.

LANGE, C. F.; CHAUDRON, M. R. Effects of defects in UML models: an experimental investigation. In: SOFTWARE ENGINEERING, 28., 2006. **Proceedings...** [S.l.: s.n.], 2006. p. 401–411.

LANGE, C. F.; CHAUDRON, M. R.; MUSKENS, J. In practice: uml software architecture and design description. **IEEE Software**, [S.l.], v. 23, n. 2, p. 40–46, 2006.

LANGE, C. F. J. Assessing and Improving the Quality of Modeling: a series of empirical studies about the uml. , [S.l.], 2007.

LARMAN, C.; UTILIZANDO, U. **padrões**: uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo. [S.l.]: Bookman, 2007.

LAUKKANEN, E.; ITKONEN, J.; LASSENIUS, C. Problems, causes and solutions when adopting continuous delivery—A systematic literature review. **Information and Software Technology**, [S.l.], v. 82, p. 55–79, 2017.

LEE, J. Q.; MCINERNEY, D. M.; LIEM, G. A. D.; ORTIGA, Y. P. The relationship between future goals and achievement goal orientations: an intrinsic–extrinsic motivation perspective. **Contemporary educational psychology**, [S.l.], v. 35, n. 4, p. 264–279, 2010.

LI, L.; HEW, K. F.; DU, J. Gamification enhances student intrinsic motivation, perceptions of autonomy and relatedness, but minimal impact on competency: a meta-analysis and systematic review. **Educational technology research and development**, [S.l.], v. 72, n. 2, p. 765–796, 2024.

LI, M.; MA, S.; SHI, Y. Examining the effectiveness of gamification as a tool promoting teaching and learning in educational settings: a meta-analysis. **Frontiers in Psychology**, [S.l.], v. 14, p. 1253549, 2023.

LI, Z.; MCGOWAN, A.; LIU, Y.; HAMOU-LHADJ, A. UML crisis! An educational perspective. In: ACM INTERNATIONAL CONFERENCE ON THE FOUNDATIONS OF SOFTWARE ENGINEERING, 33., 2025, Trondheim, Norway. **Anais...** [S.l.: s.n.], 2025.

LIEBEL, G.; MARKO, N.; TICHY, M.; LEITNER, A.; HANSSON, J. Model-based engineering in the embedded systems domain: an industrial survey on the state-of-practice. **Software & Systems Modeling**, [S.l.], v. 17, n. 1, p. 91–113, 2018.

LINDLAND, O. I.; SINDRE, G.; SOLVBERG, A. Understanding quality in conceptual modeling. **IEEE software**, [S.l.], v. 11, n. 2, p. 42–49, 1994.

LIU, D.; SANTHANAM, R.; WEBSTER, J. Toward Meaningful Engagement: a framework for design and research of gamified information systems. **MIS quarterly**, [S.l.], v. 41, n. 4, 2017.

LUMSDEN, J.; EDWARDS, E. A.; LAWRENCE, N. S.; COYLE, D.; MUNAFÒ, M. R. et al. Gamification of cognitive assessment and cognitive training: a systematic review of applications and efficacy. **JMIR serious games**, [S.l.], v. 4, n. 2, p. e5888, 2016.

MACHUCA-VILLEGAS, L.; GASCA-HURTADO, G. P.; TAMAYO, L. M. R.; PUENTE, S. M. Elementos de Gamificación en el Contexto de Ingeniería de Software. **Revista Ibérica de Sistemas e Tecnologías de Informação**, [S.l.], n. E27, p. 718–732, 2020.

MARAND, E. A.; SHEIKHAHMADI, A.; CHALLENGER, M.; MORADI, P.; KHALILIPOUR, A. Recommender Systems for Unified Modeling Language and Vice Versa—A Systematic Literature Review. **IEEE Access**, [S.l.], v. 13, p. 23426–23460, 2025.

MARANGUNIĆ, N.; GRANIĆ, A. Technology acceptance model: a literature review from 1986 to 2013. **Universal access in the information society**, [S.l.], v. 14, n. 1, p. 81–95, 2015.

MARCHEZAN, L.; ASSUNÇÃO, W. K.; HERAC, E.; KEPLINGER, F.; EGYED, A.; LAUWERYS, C. Fulfilling industrial needs for consistency among engineering artifacts. In: IEEE/ACM 45TH INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING: SOFTWARE ENGINEERING IN PRACTICE (ICSE-SEIP), 2023., 2023. **Anais...** [S.l.: s.n.], 2023. p. 246–257.

MARCHEZAN, L.; ASSUNÇÃO, W. K.; HERAC, E.; SHAFIQ, S.; EGYED, A. Exploring Dependencies Among Inconsistencies to Enhance the Consistency Maintenance of Models. In: IEEE INTERNATIONAL CONFERENCE ON SOFTWARE ANALYSIS, EVOLUTION AND REENGINEERING (SANER), 2024., 2024. **Anais...** [S.l.: s.n.], 2024. p. 147–158.

MARCHEZAN, L.; ASSUNÇÃO, W. K.; MICHELON, G. K.; EGYED, A. Do Developers Benefit from Recommendations when Repairing Inconsistent Design Models? a Controlled Experiment. , [S.l.], 2023.

MARCHEZAN, L.; HOMOLKA, M.; BLOKHIN, A.; ASSUNÇÃO, W. K.; HERAC, E.; EGYED, A. A Tool for Collaborative Consistency Checking During Modeling. In: ACM/IEEE 27TH INTERNATIONAL CONFERENCE ON MODEL DRIVEN ENGINEERING LANGUAGES AND SYSTEMS, 2024. **Proceedings...** [S.l.: s.n.], 2024. p. 655–659.

MARCHEZAN, L.; KRETSCHMER, R.; ASSUNÇÃO, W. K.; REDER, A.; EGYED, A. Generating repairs for inconsistent models. **Software and Systems Modeling**, [S.l.], v. 22, n. 1, p. 297–329, 2023.

MARCZEWSKI, A. **Gamification**: a simple introduction. [S.l.]: Andrzej Marczewski, 2013.

MARÍN, B. Lessons Learned About Gamification in Software Engineering Education. In: **Latin American Women and Research Contributions to the IT Field**. [S.l.]: IGI Global, 2021. p. 174–197.

MARÍN, B. Lessons Learned About Gamification in Software Engineering Education. In: **Research Anthology on Developments in Gamification and Game-Based Learning**. [S.l.]: IGI Global, 2022. p. 1473–1496.

MARÍN, B.; FREZ, J.; CRUZ-LEMUS, J.; GENERO, M. An empirical investigation on the benefits of gamification in programming courses. **ACM Transactions on Computing Education (TOCE)**, [S.l.], v. 19, n. 1, p. 1–22, 2018.

MASON, R.; RENNIE, F. **Elearning**: the key concepts. [S.l.]: Routledge, 2006.

MENZEN, J. P.; FARIAS, K.; BISCHOFF, V. Using biometric data in software engineering: a systematic mapping study. **Behaviour & Information Technology**, [S.l.], v. 40, n. 9, p. 880–902, 2021.

MICHEL-CROSATO, E.; BIAZEVIC, M. et al. **Ética e pesquisa**: novas diretrizes no contexto brasileiro. [S.l.]: FOU SP, 2024.

MONTEIRO, R. H. B.; ALMEIDA SOUZA, M. R. de; OLIVEIRA, S. R. B.; SANTOS PORTELA, C. dos; CRISTO LOBATO, C. E. de. The Diversity of Gamification Evaluation in the Software Engineering Education and Industry: trends, comparisons and gaps. In: IEEE/ACM 43RD INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING: SOFTWARE ENGINEERING EDUCATION AND TRAINING (ICSE-SEET), 2021., 2021. **Anais...** [S.l.: s.n.], 2021. p. 154–164.

MORA, A.; RIERA, D.; GONZALEZ, C.; ARNEDO-MORENO, J. A literature review of gamification design frameworks. In: INTERNATIONAL CONFERENCE ON GAMES AND VIRTUAL WORLDS FOR SERIOUS APPLICATIONS (VS-GAMES), 2015., 2015. **Anais...** [S.l.: s.n.], 2015. p. 1–8.

- MORSCHHEUSER, B.; HAMARI, J.; WERDER, K.; ABE, J. How to gamify? A method for designing gamification. In: HAWAII INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES 2017, 50., 2017. **Proceedings...** [S.l.: s.n.], 2017.
- MULLINS, J. K.; SABHERWAL, R. Gamification: a cognitive-emotional view. **Journal of Business Research**, [S.l.], v. 106, p. 304–314, 2020.
- NAGABHUSHAN KALBURGI, S.; KLEINERT, T.; ARYAN, D.; NASH, K.; SCHILLER, B.; KOENIG, T. MICROSTATELAB: the eeglab toolbox for resting-state microstate analysis. **Brain topography**, [S.l.], v. 37, n. 4, p. 621–645, 2024.
- OLGUN, S.; YILMAZ, M.; CLARKE, P. M.; O'CONNOR, R. V. A systematic investigation into the use of game elements in the context of software business landscapes: a systematic literature review. In: SOFTWARE PROCESS IMPROVEMENT AND CAPABILITY DETERMINATION: 17TH INTERNATIONAL CONFERENCE, SPICE 2017, PALMA DE MALLORCA, SPAIN, OCTOBER 4–5, 2017, PROCEEDINGS, 2017. **Anais...** [S.l.: s.n.], 2017. p. 384–398.
- OLIVEIRA, K.; GARCIA, A.; WHITTLE, J. On the quantitative assessment of class model compositions: an exploratory study. **1th ESMDE at MODELS**, [S.l.], 2008.
- OMG. **UML**: infrastructure specification. <https://www.omg.org/spec/UML/2.5.1/PDF>.
- OUHBI, S.; POMBO, N. Software engineering education: challenges and perspectives. In: IEEE GLOBAL ENGINEERING EDUCATION CONFERENCE (EDUCON), 2020., 2020. **Anais...** [S.l.: s.n.], 2020. p. 202–209.
- OZKAYA, M.; ERATA, F. A survey on the practical use of UML for different software architecture viewpoints. **Information and Software Technology**, [S.l.], v. 121, p. 106275, 2020.
- PAHARIA, R. **Loyalty 3.0**. [S.l.]: McGraw-Hill, 2013. 304 p.
- PARIZI, R. M. On the gamification of human-centric traceability tasks in software testing and coding. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING RESEARCH, MANAGEMENT AND APPLICATIONS (SERA), 14., 2016. **Anais...** [S.l.: s.n.], 2016. p. 193–200.
- PARIZI, R. M.; KASEM, A.; ABDULLAH, A. Towards gamification in software traceability: between test and code artifacts. In: INTERNATIONAL JOINT CONFERENCE ON SOFTWARE TECHNOLOGIES (ICSOFT), 2015., 2015. **Anais...** [S.l.: s.n.], 2015. v. 1, p. 1–8.
- PAULA PORTO, D. de; JESUS, G. M. de; FERRARI, F. C.; FABBRI, S. C. P. F. Initiatives and challenges of using gamification in software engineering: a systematic mapping. **Journal of Systems and Software**, [S.l.], v. 173, p. 110870, 2021.
- PEDREIRA, O.; GARCÍA, F.; BRISABOA, N.; PIATTINI, M. Gamification in software engineering—A systematic mapping. **Information and software technology**, [S.l.], v. 57, p. 157–168, 2015.
- PETRE, M. UML in practice. In: 2013 35TH INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING (ICSE), 2013. **Anais...** [S.l.: s.n.], 2013. p. 722–731.

- PETRE, M. No shit or Oh, shit!: responses to observations on the use of uml in professional practice. **Software & Systems Modeling**, [S.l.], v. 13, n. 4, p. 1225–1235, 2014.
- PINTRICH, P. R. **The role of goal orientation in self-regulated learning**. [S.l.]: Elsevier, 2000. 451–502 p.
- PORTO, D.; JESUS, G.; FERRARI, F.; FABBRI, S. Initiatives and Challenges of Using Gamification in Software Engineering: a systematic mapping. **arXiv preprint arXiv:2011.07115**, [S.l.], 2020.
- PRAUSE, C. R.; NONNEN, J.; VINKOVITS, M. A Field Experiment on Gamification of Code Quality in Agile Development. In: WORKSHOP PSYCHOLOGY OF PROGRAMMING INTEREST GROUP, 2012. **Anais...** [S.l.: s.n.], 2012. v. 2012, p. 17.
- PRESSMAN, R. S. **Software engineering: a practitioner's approach**. [S.l.]: Palgrave macmillan, 2005.
- PRIYADI, O.; RAMADHAN, I.; SENSUSE, D. I.; SURYONO, R. R.; KAUTSARINA. Gamification in Software Development: systematic literature review. In: INTERNATIONAL CONFERENCE ON NETWORKING, INTELLIGENT SYSTEMS AND SECURITY, 2022. **Anais...** [S.l.: s.n.], 2022. p. 386–398.
- REDER, A.; EGYED, A. Determining the cause of a design model inconsistency. **IEEE Transac. on Software Engineering**, [S.l.], v. 39, n. 11, p. 1531–1548, 2013.
- REN, W.; BARRETT, S.; DAS, S. Toward Gamification to Software Engineering and Contribution of Software Engineer. In: INT. CONF. ON MANAGEMENT ENGINEERING, SOFTWARE ENGINEERING AND SERVICE SCIENCES, 4., 2020. **Anais...** [S.l.: s.n.], 2020. p. 1–5.
- ROBLES, G.; HO-QUANG, T.; HEBIG, R.; CHAUDRON, M. R.; FERNANDEZ, M. A. An extensive dataset of UML models in GitHub. In: IEEE/ACM 14TH INTERNATIONAL CONFERENCE ON MINING SOFTWARE REPOSITORIES (MSR), 2017., 2017. **Anais...** [S.l.: s.n.], 2017. p. 519–522.
- RODRIGUES, P.; SOUZA, M.; FIGUEIREDO, E. Games and gamification in software engineering education: a survey with educators. In: IEEE FRONTIERS IN EDUCATION CONFERENCE (FIE), 2018., 2018. **Anais...** [S.l.: s.n.], 2018. p. 1–9.
- ROMEO, J.; RAGLIANTI, M.; NAGY, C.; LANZA, M. UML is Back. Or is it? Investigating the Past, Present, and Future of UML in Open Source Software. In: IEEE/ACM 47TH INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING (ICSE), 2025., 2025. **Anais...** [S.l.: s.n.], 2025. p. 692–692.
- RUMBAUGH, J.; BLAHA, M.; PREMERLANI, W.; EDDY, F.; LORENSEN, W. **Modelagem e projetos baseados em objetos**. [S.l.]: Campus Rio de Janeiro, 1994. v. 8.
- RUMBAUGH, J.; BLAHA, M.; PREMERLANI, W.; EDDY, F.; LORENSEN, W. E. et al. **Object-oriented modeling and design**. [S.l.]: Prentice-hall Englewood Cliffs, NJ, 1991. v. 199, n. 1.
- RUNESON, P.; HÖST, M. Guidelines for conducting and reporting case study research in software engineering. **Empirical software engineering**, [S.l.], v. 14, n. 2, p. 131–164, 2009.

- RYAN, R.; DECI, E. L. La Teoría de la Autodeterminación y la Facilitación de la Motivación Intrínseca, el Desarrollo Social, y el Bienestar. **American psychologist**, [S.l.], v. 55, n. 1, p. 68–78, 2000.
- RYAN, R. M.; DECI, E. L. Intrinsic and extrinsic motivations: classic definitions and new directions. **Contemporary educational psychology**, [S.l.], v. 25, n. 1, p. 54–67, 2000.
- RYAN, R. M.; DECI, E. L. Intrinsic and extrinsic motivation from a self-determination theory perspective: definitions, theory, practices, and future directions. **Contemporary educational psychology**, [S.l.], v. 61, p. 101860, 2020.
- SALMAN, I.; MISIRLI, A. T.; JURISTO, N. Are students representatives of professionals in software engineering experiments? In: IEEE/ACM 37TH IEEE INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, 2015., 2015. **Anais...** [S.l.: s.n.], 2015. v. 1, p. 666–676.
- SANTOS, I.; FELIZARDO, K. R.; GEROSA, M. A.; STEINMACHER, I. Game Elements to Engage Students Learning the Open Source Software Contribution Process. In: IEEE SYMPOSIUM ON VISUAL LANGUAGES AND HUMAN-CENTRIC COMPUTING (VL/HCC), 2024., 2024. **Anais...** [S.l.: s.n.], 2024. p. 59–70.
- SAPPAILE, B. I.; XU, S.; OCI, M.; XAVIER, M.; HALIM, C. The Influence of Gamification Techniques on Students' Learning Performance and Motivation in Learning: an experimental study. **Journal Neosantara Hybrid Learning**, [S.l.], v. 2, n. 1, p. 394–408, 2024.
- SCANNIELLO, G.; GRAVINO, C.; GENERO, M.; CRUZ-LEMUS, J.; TORTORA, G. On the impact of UML analysis models on source-code comprehensibility and modifiability. **ACM TOSEM**, [S.l.], v. 23, n. 2, p. 1–26, 2014.
- SCHWABER, K.; SUTHERLAND, J. **Przewodnik po Scrumie**. 2020.
- SEABORN, K.; FELS, D. I. Gamification in theory and action: a survey. **International Journal of human-computer studies**, [S.l.], v. 74, p. 14–31, 2015.
- SHARMA, V. S.; MEHRA, R.; KAULGUD, V.; PODDER, S. An immersive future for software engineering: avenues and approaches. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING: NEW IDEAS AND EMERGING RESULTS, 40., 2018. **Proceedings...** [S.l.: s.n.], 2018. p. 105–108.
- SHPAKOVA, A.; DÖRFLER, V.; MACBRYDE, J. The role (s) of gamification in knowledge management. In: EURAM 2016: 16TH ANNUAL CONFERENCE OF THE EUROPEAN ACADEMY OF MANAGEMENT, 2016. **Anais...** [S.l.: s.n.], 2016. p. 1–40.
- SICART, M. Defining game mechanics. **Game Studies**, [S.l.], v. 8, n. 2, p. 1–14, 2008.
- STÖRRLE, H. How are Conceptual Models used in Industrial Software Development? A Descriptive Survey. In: INT. CONF. ON EVALUATION AND ASSESSMENT IN SOFTWARE ENGINEERING, 21., 2017. **Anais...** [S.l.: s.n.], 2017. p. 160–169.
- SUKALE, R.; PFAFF, M. S. QuoDocs: improving developer engagement in software documentation through gamification. In: **CHI'14 Extended Abstracts on Human Factors in Computing Systems**. [S.l.: s.n.], 2014. p. 1531–1536.

TASADDUQ, M.; KHAN, M. S.; NAWAB, R. M.; JAMAL, M. H.; CHAUDHRY, M. T. Exploring the effects of gamification on students with rote learning background while learning computer programming. **Computer Applications in Engineering Education**, [S.l.], v. 29, n. 6, p. 1871–1891, 2021.

THOMAS, D. MDA: revenge of the modelers or uml utopia? **IEEE software**, [S.l.], v. 21, n. 3, p. 15–17, 2004.

TONHÃO, S.; SHIGENAGA, M.; HERCULANI, J.; MEDEIROS, A.; AMARAL, A.; SILVA, W.; COLANZI, T.; STEINMACHER, I. Gamification in software engineering education: a tertiary study. In: XXXVII BRAZILIAN SYMPOSIUM ON SOFTWARE ENGINEERING, 2023. **Proceedings...** [S.l.: s.n.], 2023. p. 358–367.

TORCHIANO, M.; TOMASSETTI, F.; RICCA, F.; TISO, A.; REGGIO, G. Relevance, benefits, and problems of software modelling and model driven techniques—A survey in the Italian industry. **Journal of Systems and Software**, [S.l.], v. 86, n. 8, p. 2110–2126, 2013.

TORRE, D.; GENERO, M.; LABICHE, Y.; ELAASAR, M. How consistency is handled in model-driven software engineering and UML: an expert opinion survey. **Software Quality Journal**, [S.l.], v. 31, n. 1, p. 1–54, 2023.

TORRE, D.; LABICHE, Y.; GENERO, M. UML consistency rules: a systematic mapping study. In: INTERNATIONAL CONFERENCE ON EVALUATION AND ASSESSMENT IN SOFTWARE ENGINEERING, 18., 2014. **Proceedings...** [S.l.: s.n.], 2014. p. 1–10.

TORRE, D.; LABICHE, Y.; GENERO, M.; ELAASAR, M. A systematic identification of consistency rules for UML diagrams. **Journal of Systems and Software**, [S.l.], v. 144, p. 121–142, 2018.

TORRE, D.; LABICHE, Y.; GENERO, M.; ELAASAR, M.; MENGHI, C. UML consistency rules: a case study with open-source uml models. In: INTERNATIONAL CONFERENCE ON FORMAL METHODS IN SOFTWARE ENGINEERING, 8., 2020. **Proceedings...** [S.l.: s.n.], 2020. p. 130–140.

TORRES, W.; BRAND, M. G. Van den; SEREBRENIK, A. A systematic literature review of cross-domain model consistency checking by model management tools. **Software and Systems Modeling**, [S.l.], v. 20, p. 897–916, 2021.

UNHELKAR, B. **Software engineering with UML**. [S.l.]: Auerbach Publications, 2017.

USMAN, M.; FELDERER, M.; UNTERKALMSTEINER, M.; KLOTINS, E.; MENDEZ, D.; ALÉGROTH, E. Compliance Requirements in Large-Scale Software Development: an industrial case study. In: INT. CONF. ON PRODUCT-FOCUSED SOFTWARE PROCESS IMPROVEMENT, 2020. **Anais...** [S.l.: s.n.], 2020. p. 385–401.

VANSTEENKISTE, M.; LENS, W.; DECI, E. L. Intrinsic versus extrinsic goal contents in self-determination theory: another look at the quality of academic motivation. **Educational psychologist**, [S.l.], v. 41, n. 1, p. 19–31, 2006.

WEBB, E. N. Gamification: when it works, when it doesn't. In: INTERNATIONAL CONFERENCE OF DESIGN, USER EXPERIENCE, AND USABILITY, 2013. **Anais...** [S.l.: s.n.], 2013. p. 608–614.



WERBACH, K.; HUNTER, D. **For the win**: how game thinking can revolutionize your business. [S.l.]: Wharton digital press, 2012.

WERBACH, K.; HUNTER, D. **The gamification toolkit**: dynamics, mechanics, and components for the win. [S.l.]: University of Pennsylvania Press, 2015.

WHITLEY JR, B. E.; KITE, M. E. **Principles of research in behavioral science**. [S.l.]: Routledge, 2012.

WOHLIN, C.; RUNESON, P.; HÖST, M.; OHLSSON, M. C.; REGNELL, B.; WESSLÉN, A. **Experimentation in software engineering**. [S.l.]: Springer Science & Business Media, 2012.

XI, N.; HAMARI, J. Does gamification satisfy needs? A study on the relationship between gamification features and intrinsic need satisfaction. **International Journal of Information Management**, [S.l.], v. 46, p. 210–221, 2019.

XU, W.; XU, G.-R.; XING, Q.-W. The impact of different combinations of game elements for gamified learning in higher education on student learning outcomes: a multilevel meta-analysis. **Studies in Higher Education**, [S.l.], p. 1–28, 2024.

YIGITBAS, E.; SCHMIDT, M.; BUCCHIARONE, A.; BASSANELLI, S.; ENGELS, G. Gamification-and Virtual Reality-Based Learning Environment for UML Class Diagram Modeling. In: IEEE GLOBAL ENGINEERING EDUCATION CONFERENCE (EDUCON), 2024., 2024. **Anais...** [S.l.: s.n.], 2024. p. 1–10.

YIGITBAS, E.; SCHMIDT, M.; BUCCHIARONE, A.; GOTTSCHALK, S.; ENGELS, G. Gamovr: gamification-based uml learning environment in virtual reality. **Science of Computer Programming**, [S.l.], v. 231, p. 103029, 2024.

YOHANNIS, A. Gamification of Software Modelling Learning. In: DS@ MODELS, 2016. **Anais...** [S.l.: s.n.], 2016.

ZARRAONANDIA, T.; DIAZ, P.; AEDO, I.; RUIZ, M. R. Designing educational games through a conceptual model based on rules and scenarios. **Multimedia Tools and Applications**, [S.l.], v. 74, p. 4535–4559, 2015.

ZENG, J.; SUN, D.; LOOI, C.-K.; FAN, A. C. W. Exploring the impact of gamification on students' academic performance: a comprehensive meta-analysis of studies from the year 2008 to 2023. **British Journal of Educational Technology**, [S.l.], v. 55, n. 6, p. 2478–2502, 2024.

ZICHERMAN, G.; CUNNINGHAM, C. **Gamification by design**. Sebastopol. [S.l.]: CA: O'Reilly Media, 2011.

ZICHERMANN, G.; CUNNINGHAM, C. **Gamification by design**: implementing game mechanics in web and mobile apps. 1st. ed. [S.l.]: O'Reilly Media, 2011.

**APÊNDICE A - ARTIGOS PUBLICADOS**

JÚNIOR, Ed; FARIAS, Kleinner; SILVA, Bruno. A Survey on the Use of UML in the Brazilian Industry. In: Brazilian Symposium on Software Engineering. 2021. p. 275-284.

JÚNIOR, Ed; FARIAS, Kleinner. ModelGame: A Quality Model for Gamified Software Modeling Learning. In: 15th Brazilian Symposium on Software Components, Architectures, and Reuse. 2021. p. 100-109.

Júnior, Ed Wilson, Kleinner Farias, and Bruno da Silva. "On the use of uml in the brazilian industry: A survey". Journal of Software Engineering Research and Development 10 (2022): 10-1.