

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

DOUGLAS LEONARD VIEBRANTZ

**PROPOSTA DE UM MODELO TEÓRICO DE GAMIFICAÇÃO COM ÊNFASE EM
COMPETÊNCIAS SOCIOEMOCIONAIS NO ENSINO DE PROGRAMAÇÃO
PARA INICIANTES**

MEDIANEIRA

2025

DOUGLAS LEONARD VIEBRANTZ

**PROPOSTA DE UM MODELO TEÓRICO DE GAMIFICAÇÃO COM ÊNFASE EM
COMPETÊNCIAS SOCIOEMOCIONAIS NO ENSINO DE PROGRAMAÇÃO
PARA INICIANTES**

**Proposal of a Theoretical Model of Gamification with Emphasis on
Socioemotional Skills in Programming Education for Beginners**

Trabalho de Conclusão de Curso de Graduação
apresentado como requisito para obtenção do
título de Bacharel em Ciência da Computação
do Curso de Bacharelado em Ciência da
Computação da Universidade Tecnológica
Federal do Paraná.

Orientador: Dr. Everton Coimbra de Araujo

MEDIANEIRA

2025



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

LISTA DE QUADROS

Quadro 1 – Quadro comparativo dos trabalhos relacionados.	17
--	-----------

LISTA DE ABREVIATURAS E SIGLAS

Siglas

ATLAS.ti	Software de Análise Qualitativa de Dados
EMA	Escala de Motivação Acadêmica
ERIC	Centro de Recursos para Informação Educacional
Firebase Auth	Sistema de Autenticação do Firebase
IEEE	Instituto de Engenheiros Eletricistas e Eletrônicos
ISO/IEC 25010	Padrão Internacional para Qualidade de Software
I^2	Estatística de Heterogeneidade
Likert 7	Escala de 7 pontos para Avaliação
NVivo	Software de Análise Qualitativa de Dados
Photon PUN 2 Framework	Framework de Rede para Unity
PRISMA	Itens Preferenciais para Revisões Sistemáticas e Meta-Análises
R/metafor	Ferramenta de Análise Estatística em R
SpringerLink	Repositório de Artigos Acadêmicos
SQLite	Banco de Dados Relacional
SWEBOK	Corpo de Conhecimento em Engenharia de Software

SUMÁRIO

1	INTRODUÇÃO	4
1.1	Considerações iniciais	4
1.2	Objetivos	5
1.2.1	Objetivo geral	5
1.2.2	Objetivos específicos	5
1.3	Justificativa	5
1.4	Estrutura do trabalho	5
2	REFERENCIAL TEÓRICO	7
2.0.1	Competências Socioemocionais	7
2.0.2	Ensino de Programação para Iniciantes	8
2.0.3	Gamificação	9
2.0.4	Conceito da Matriz Teórica para Gamificação e Competências Socioemocionais	10
2.0.5	Mockups e Fluxogramas na Prototipagem de Sistemas Gamificados	12
2.1	Conexões entre os tópicos teóricos	13
3	TRABALHOS RELACIONADOS	14
	REFERÊNCIAS	18

1 INTRODUÇÃO

1.1 Considerações iniciais

Aprender programação normalmente é um processo desafiador, especialmente para quem está começando e ainda não domina lógica e sintaxe. Erros recorrentes e abstrações complexas geram frustração e desmotivação, fazendo com que muitos alunos abandonem seus estudos antes de desenvolver uma base sólida. É nesse ponto que a literatura em educação aponta a gamificação como recurso capaz de reconfigurar essas interações, trazendo maior dinamismo e reforço positivo ao processo de aprendizagem que também ajuda a manter o interesse dos estudantes mesmo diante das dificuldades iniciais. Ao aplicar mecânicas de jogos (como pontos, desafios, recompensas e rankings) no ensino, é possível promover um ambiente em que o aluno encontra estímulos imediatos para superar cada etapa.

As competências socioemocionais no contexto educacional também têm recebido destaque nos últimos anos. Habilidades como empatia, resiliência, autorregulação e colaboração são essenciais não só para o sucesso acadêmico, mas também para o desenvolvimento pessoal e profissional dos estudantes. Estudos recentes mostram que alunos que aprendem a lidar com a frustração de um bug enquanto mantêm o autocontrole e o espírito colaborativo tendem a persistir por mais tempo em desafios de programação. No entanto, é comum que o ensino de programação ignore essas competências, focando apenas em conteúdos técnicos e desempenho em provas ou tarefas.

A proposta deste trabalho é explorar como a gamificação pode ir além do engajamento superficial e contribuir também para o desenvolvimento dessas competências socioemocionais. O público-alvo do modelo é voltado a estudantes nos semestres iniciais do Bacharelado em Ciência da Computação ou cursos semelhantes, com pouca ou nenhuma experiência prévia em programação. Para isso, será elaborado um modelo teórico que conecte elementos de jogos a habilidades emocionais no ensino de programação para iniciantes. Em contraste com modelos que apenas catalogam elementos de gamificação, este trabalho avança ao propor um framework conceitual que: (1) orienta a seleção de mecânicas de jogo, (2) define indicadores socioemocionais e (3) estabelece instrumentos de avaliação integrados.

Esta proposta visa unir dois campos da educação atual (a gamificação e o desenvolvimento socioemocional) criando um modelo teórico que inspire novas estratégias pedagógicas em cursos introdutórios de programação, presenciais ou online. Para isso, adotamos elementos da abordagem Design-Based Research (DBR) na fase de construção teórica, como a revisão iterativa de literatura, a prototipagem conceitual e a validação por especialistas. Esses métodos foram escolhidos por favorecerem o refinamento contínuo do modelo com base em evidências prévias. Contudo, esta pesquisa não inclui testes em ambientes reais, uma vez que seu foco é a proposição de um referencial teórico, deixando a implementação prática como sugestão para estudos futuros.

1.2 Objetivos

1.2.1 Objetivo geral

Propor um framework conceitual de gamificação que relacione elementos lúdicos ao desenvolvimento de competências socioemocionais no ensino introdutório de programação, incluindo representações visuais e instrumentos avaliativos aplicáveis em contextos educacionais.

1.2.2 Objetivos específicos

Levantar, por meio de revisão bibliográfica, os principais elementos de gamificação e competências socioemocionais aplicáveis ao contexto de aprendizagem de programação.

Relacionar mecânicas de jogo com habilidades socioemocionais em uma matriz teórica.

Criar representações visuais (mockups e fluxogramas) que exemplifiquem a aplicação prática do modelo.

Propor instrumentos de avaliação (rubricas, formulários de autorreflexão e indicadores técnicos) para mensurar o progresso dos alunos em aspectos cognitivos e socioemocionais.

1.3 Justificativa

A proposta de um modelo teórico que articule gamificação e competências socioemocionais responde à carência de frameworks específicos para o ensino de programação. Enquanto diversas pesquisas validam os efeitos motivacionais da gamificação e demonstram a importância das soft skills, poucos estudos oferecem diretrizes claras para integrar essas dimensões num único referencial. Um modelo bem fundamentado orientará tanto o planejamento pedagógico quanto o desenvolvimento de plataformas educacionais, garantindo que o aluno seja estimulado técnica e emocionalmente.

Além disso, ao incluir instrumentos de avaliação que abarquem desempenho em programação e indicadores emocionais, este trabalho contribui para a área de tecnologia educacional ao oferecer uma base metodológica replicável em contextos acadêmicos e de treinamento corporativo. Espera-se que, com este modelo, docentes e desenvolvedores possam implementar experiências de aprendizagem mais equilibradas, capazes de reduzir a evasão inicial e preparar estudantes para os desafios do mercado de trabalho.

1.4 Estrutura do trabalho

O documento está organizado em cinco capítulos:

Capítulo 1 – Introdução: apresenta o tema, delimitação do público-alvo, problema de pesquisa, objetivos, justificativa, metodologia e visão geral do modelo.

Capítulo 2 – Referencial Teórico: revisa conceitos de gamificação, teorias de competências socioemocionais, análise crítica de modelos existentes e levantamento de lacunas.

Capítulo 3 – Proposta de Modelo Teórico: detalha o framework conceitual, incluindo definição de elementos de jogo, competências alvo, matriz de correlação e mockups de aplicação.

2 REFERENCIAL TEÓRICO

A fundamentação teórica deste trabalho está estruturada em cinco tópicos principais, que fornecem os pilares conceituais para a elaboração do framework proposto. O primeiro tópico aborda as competências socioemocionais, com base em taxonomias reconhecidas e instrumentos de avaliação utilizados em contextos educacionais. O segundo trata do ensino de programação para iniciantes, com foco nos principais desafios pedagógicos e nas abordagens que favorecem a permanência e o engajamento dos estudantes. O terceiro tópico explora os conceitos e frameworks de gamificação, com ênfase na aplicação em ambientes educacionais e nas relações entre mecânicas de jogo e fatores motivacionais. Subsequentemente, discute-se o conceito da matriz teórica como ferramenta para correlacionar elementos de gamificação com competências socioemocionais. Por fim, aborda-se o papel dos mockups na visualização e planejamento de sistemas gamificados.

Ao final deste capítulo, será apresentada uma subseção integradora que articula os tópicos entre si, estabelecendo a base teórica para o modelo a ser desenvolvido no capítulo seguinte.

2.0.1 Competências Socioemocionais

As competências socioemocionais referem-se à capacidade dos indivíduos de gerenciar emoções, estabelecer relações saudáveis, tomar decisões responsáveis e lidar com frustrações, desafios e conflitos (Del Prette, 2005; OECD, 2015). No contexto educacional, essas habilidades são consideradas tão importantes quanto as competências cognitivas tradicionais, influenciando diretamente o engajamento, a persistência e o desempenho acadêmico.

A Base Nacional Comum Curricular (BNCC, 2018) insere as competências socioemocionais como parte integrante da formação do estudante, alinhando-se a modelos internacionais como o da Collaborative for Academic, Social, and Emotional Learning (CASEL, 2020). O framework da CASEL estrutura essas competências em cinco grandes domínios:

- **Autoconsciência:** reconhecer emoções, valores e limitações pessoais. Envolve a capacidade de identificar os próprios sentimentos e como eles influenciam o comportamento;
- **Autorregulação:** gerenciar impulsos e comportamentos diante de frustrações. Implica controlar reações emocionais, adiar gratificações e perseverar diante de obstáculos;
- **Consciência social:** demonstrar empatia e compreender o outro. Refere-se à capacidade de entender a perspectiva alheia e sentir compaixão;
- **Habilidades de relacionamento:** comunicar-se com clareza, escutar e cooperar. Abrange a construção e manutenção de relações saudáveis e colaborativas;

- **Tomada de decisão responsável:** avaliar consequências e fazer escolhas éticas. Envolve considerar o bem-estar próprio e dos outros ao tomar decisões.

Além da taxonomia da CASEL, outros frameworks como o Big Five Aspects Scale, que se baseia no modelo dos Cinco Grandes Fatores de Personalidade (Abertura, Conscienciosidade, Extroversão, Amabilidade e Neuroticismo), também são utilizados para compreender traços que sustentam competências socioemocionais. Ishaq & Alvi (2023) mencionam a aplicação de perfis cognitivos baseados no Big Five para personalizar intervenções em cursos de programação.

Diversos instrumentos têm sido utilizados para mensurar essas habilidades, incluindo escalas de autorrelato (como questionários Likert, a exemplo da Escala de 7 pontos para Avaliação mencionada por Holly et al. (2024)), rubricas avaliativas e protocolos de observação (Heckman, 2006; Zerbini, 2017). Ferramentas de análise qualitativa de dados, como ATLAS.ti e NVivo, são empregadas para codificar e analisar dados provenientes dessas avaliações, como observado nos estudos de Oliveira (2022) e Castro et al. (2021). Em contextos computacionais, essas competências são especialmente relevantes: ao enfrentar bugs, frustrações ou lógica abstrata, o aluno precisa desenvolver estratégias de resiliência, colaboração e autorregulação para continuar aprendendo. A literatura recente também aponta para a importância de métricas fisiológicas, como frequência cardíaca, para complementar as autoavaliações, embora sua implementação seja um desafio.

2.0.2 Ensino de Programação para Iniciantes

O ensino de programação para iniciantes apresenta uma série de desafios relacionados à abstração, à sintaxe das linguagens e à manutenção da motivação diante de erros frequentes (Robins et al., 2003). Estudantes que ingressam em disciplinas introdutórias frequentemente se deparam com barreiras cognitivas e emocionais que impactam negativamente a aprendizagem e aumentam os índices de evasão.

Diversas abordagens metodológicas têm sido propostas para mitigar essas dificuldades.

- A **aprendizagem baseada em projetos** (Papert, 1980) e o uso de linguagens visuais, como Scratch (Resnick, 2009), demonstram resultados positivos ao aproximar o conteúdo técnico de contextos mais lúdicos e acessíveis. O Scratch, desenvolvido pelo MIT Media Lab, permite que iniciantes criem histórias interativas, jogos e animações usando blocos de código visual, reduzindo a carga cognitiva da sintaxe;
- Recentemente, tem-se difundido a taxonomia **PRIMM** (Predict, Run, Investigate, Modify, Make), que orienta o processo de ensino com foco na experimentação gradual e na construção do entendimento a partir de exemplos modificáveis. Essa abordagem

incentiva os alunos a primeiro prever o comportamento de um código, executá-lo, investigar seu funcionamento, modificá-lo e, finalmente, criar seus próprios programas;

- **Ambientes de desenvolvimento integrados (IDEs)** projetados para iniciantes, como o Thonny para Python ou o BlueJ para Java, oferecem interfaces simplificadas e ferramentas de depuração visual que auxiliam na compreensão dos conceitos de programação;
- Plataformas online como Code.org, Khan Academy e freeCodeCamp utilizam tutoriais interativos e desafios de codificação progressivos, muitas vezes incorporando elementos de gamificação;
- O uso de **ferramentas de feedback instantâneo** também é crucial. Silva et al. (2025) relatam o uso do motor de jogo Godot 3.5 para criar um jogo sequencial com feedback imediato para o ensino de lógica, utilizando SQLite para armazenamento de logs e Python com a biblioteca pandas para análise de dados.

Além dos aspectos cognitivos, autores como Sentance et al. (2019) destacam que fatores emocionais como ansiedade, comparação com colegas e frustração diante de falhas têm papel central na experiência dos estudantes iniciantes. Nesse contexto, a promoção de um ambiente acolhedor, com feedback constante, foco na progressão individual e estímulo à colaboração, é essencial para a permanência e o sucesso desses alunos.

2.0.3 Gamificação

A gamificação é definida como o uso de elementos de jogos em contextos não lúdicos, com o propósito de aumentar o engajamento, a motivação e o foco dos participantes (Deterding et al., 2011). Em ambientes educacionais, essa abordagem tem sido utilizada para tornar o processo de aprendizagem mais atrativo e dinâmico, promovendo o envolvimento ativo dos estudantes.

Entre os principais frameworks teóricos, destacam-se:

- **Octalysis Framework** (Chou, 2016): estrutura composta por oito "drives" motivacionais, como desenvolvimento e realização, posse, influência social e escassez. Este modelo é útil para mapear motivações intrínsecas e extrínsecas em projetos gamificados. Por exemplo, a "influência social" pode ser estimulada por meio de leaderboards, enquanto o "desenvolvimento e realização" se manifesta em barras de progresso e badges;
- **MDA Framework** (Mechanics, Dynamics, Aesthetics) (Hunicke et al., 2004): propõe uma separação entre os componentes mecânicos do jogo (regras, pontos, níveis), sua

dinâmica emergente durante a interação (comportamento do jogador) e a experiência estética percebida pelo usuário (diversão, desafio, narrativa). Este framework ajuda designers a pensar sobre como as mecânicas escolhidas levarão às dinâmicas desejadas e, conseqüentemente, às experiências estéticas esperadas;

- **RAMP** (Marczewski, 2015): estrutura baseada em quatro eixos motivacionais – Relacionamento, Autonomia, Maestria e Propósito – que influenciam o design de experiências personalizadas. Este framework enfatiza a importância de conectar os usuários entre si (Relacionamento), dar-lhes controle sobre suas ações (Autonomia), ajudá-los a desenvolver habilidades (Maestria) e conectá-los a um objetivo maior (Propósito);
- **Teoria da Autodeterminação (SDT)** (Ryan Deci, 2000): Embora não seja um framework de gamificação por si, é uma teoria motivacional fundamental que sustenta muitas aplicações gamificadas. A SDT postula que a motivação intrínseca é fomentada pela satisfação de três necessidades psicológicas básicas: autonomia, competência e relacionamento. A gamificação bem projetada pode atender a essas necessidades através de escolhas, feedback claro sobre o progresso e interações sociais.

No contexto da educação em computação, autores como Werbach e Hunter (2012) apontam que a eficácia da gamificação está na intencionalidade pedagógica. Simplesmente adicionar pontos e rankings sem uma conexão com objetivos educacionais pode gerar efeitos contrários, como desmotivação ou competição excessiva (Hamari et al., 2014), um ponto também levantado por Santos et al. (2021) e Zhan et al. (2022).

Aplicações em ensino de programação mostram que sistemas gamificados bem calibrados reduzem a evasão (Dominguez, 2013), promovem feedback contínuo, motivam o trabalho em equipe e incentivam a persistência diante de problemas complexos. Ferramentas e plataformas para implementar gamificação variam desde sistemas de gerenciamento de aprendizagem (LMS) com plugins de gamificação, até plataformas dedicadas ou o desenvolvimento customizado utilizando linguagens como Python, Java (como no caso do tutor inteligente em Java Spring Boot 2.5 e React 17 de Cao, 2023) ou C# com motores de jogo como Unity (utilizado por Guedes, 2021 e Holly et al., 2024). A escolha da tecnologia muitas vezes depende da complexidade da gamificação desejada e dos recursos disponíveis.

2.0.4 Conceito da Matriz Teórica para Gamificação e Competências Socioemocionais

A construção de um modelo teórico que relacione elementos de jogos ao desenvolvimento de habilidades socioemocionais no ensino de programação para iniciantes exige uma estrutura organizada e bem fundamentada. Nesse contexto, a matriz teórica é apresentada como uma ferramenta conceitual capaz de mapear, de forma sistemática, as conexões entre mecânicas de jogo e as competências socioemocionais visadas. Este trabalho propõe um fra-

mework conceitual que orienta tanto a escolha das mecânicas quanto a definição de indicadores socioemocionais e a integração de instrumentos de avaliação.

A concepção de tal matriz se baseia na premissa de que diferentes elementos de gamificação podem ser intencionalmente desenhados para estimular ou exercitar diferentes aspectos socioemocionais. Por exemplo:

- **Desafios colaborativos** podem fomentar habilidades de relacionamento (comunicação, cooperação) e consciência social (empatia);
- **Sistemas de feedback imediato e progressão clara** (níveis, pontos de experiência) podem apoiar a autorregulação (gerenciamento da frustração ao errar e persistir) e a autoconsciência (reconhecimento do próprio progresso e limitações);
- **Narrativas imersivas com dilemas éticos** podem engajar os alunos na tomada de decisão responsável. Guedes (2021) combinou narrativas imersivas com a teoria de Dempster-Shafer para fortalecer competências socioemocionais em crianças;
- **Badges e conquistas** podem reconhecer não apenas o desempenho técnico, mas também comportamentos socioemocionais positivos, como ajudar colegas ou demonstrar resiliência.

A construção dessa matriz envolve:

1. **Identificação das competências socioemocionais alvo:** Com base em frameworks como o da CASEL, define-se quais habilidades são prioritárias para o contexto do ensino de programação para iniciantes;
2. **Seleção de mecânicas de gamificação:** Com base em frameworks como Octalysis, MDA ou RAMP, escolhem-se os elementos de jogo mais adequados para os objetivos pedagógicos e o público-alvo;
3. **Estabelecimento de correlações:** Analisa-se como cada mecânica de jogo pode influenciar o desenvolvimento de cada competência socioemocional identificada, fundamentando essas correlações na literatura existente sobre psicologia, educação e gamificação.

A matriz teórica, portanto, não é apenas um catálogo, mas uma ferramenta analítica que justifica as escolhas de design do sistema gamificado, alinhando-as explicitamente ao desenvolvimento socioemocional. A validação dessa matriz pode ocorrer através da revisão por especialistas e, posteriormente, por meio de estudos empíricos que testem as relações propostas.

Para ilustrar essa lógica de articulação entre elementos lúdicos e competências socioemocionais, apresenta-se a seguir um exemplo sintético de matriz de correlação. O quadro

relaciona mecânicas de jogo comuns em contextos educacionais com competências específicas, os frameworks teóricos que embasam cada escolha e as ferramentas tecnológicas mais adequadas para sua implementação.

Tabela 1 – Exemplo de correlação entre mecânicas de jogo e competências socioemocionais.

Mecânica de Jogo	Competência Socioemocional	Framework Base	Ferramenta Sugerida
Missões colaborativas	Habilidades de relacionamento	Octalysis / RAMP	Godot / Firebase
Feedback imediato	Autorregulação e autoconsciência	MDA / SDT	Unity / React / Python + pandas
Narrativas com dilemas éticos	Tomada de decisão responsável	Octalysis / CASEL	Twine / Ink / Unity
Sistema de conquistas (badges)	Empatia e reconhecimento pessoal	RAMP / MDA	Flutter / Firebase

2.0.5 Mockups e Fluxogramas na Prototipagem de Sistemas Gamificados

Para traduzir o modelo teórico e a matriz de correlações em uma aplicação prática, a criação de representações visuais como mockups e fluxogramas é um passo crucial. Essas ferramentas são essenciais na fase de prototipagem conceitual, conforme a abordagem Design-Based Research (DBR).

Mockups são representações visuais estáticas ou interativas da interface do usuário (UI) e da experiência do usuário (UX) de um sistema. Eles podem variar em fidelidade:

- **Baixa fidelidade (wireframes):** Esboços básicos que focam na estrutura, layout e fluxo de navegação, geralmente sem cores ou detalhes gráficos. Ferramentas como Balsamiq ou mesmo desenhos à mão são comuns;
- **Média fidelidade:** Apresentam mais detalhes visuais, como cores e tipografia, mas ainda podem não ser interativos;
- **Alta fidelidade:** Protótipos que se assemelham muito ao produto final em termos de aparência e interatividade. Ferramentas como Figma, Adobe XD, Sketch ou InVision são amplamente utilizadas para criar mockups de alta fidelidade. Cao (2023) utilizou React 17 para o desenvolvimento da interface do seu tutor inteligente, o que implica a criação de componentes visuais que poderiam ser precedidos por mockups.

No contexto deste trabalho, os mockups exemplificam como os elementos de gamificação (pontos, badges, narrativas) e os recursos de incentivo à reflexão socioemocional (como perguntas guiadas e mensagens de feedback específicas) seriam integrados na experiência do

aluno. Esses elementos visuais ajudam a antecipar a jornada do usuário, a disposição das informações e a forma como ele interage com os componentes do sistema, favorecendo tanto o engajamento quanto o desenvolvimento de competências emocionais ao longo do processo de aprendizagem.

Fluxogramas são diagramas que representam o fluxo de processos ou algoritmos. No design de sistemas gamificados, eles podem ser usados para:

- Mapear a progressão do aluno através dos desafios e conteúdos;
- Detalhar a lógica por trás da atribuição de recompensas ou do acionamento de feedbacks;
- Ilustrar as sequências de interações que visam desenvolver competências socioemocionais específicas.

A combinação de mockups e fluxogramas permite que a equipe de desenvolvimento, educadores e até mesmo os alunos compreendam e avaliem o design proposto antes da implementação completa. Eles facilitam a comunicação de ideias, a identificação de problemas potenciais e o refinamento iterativo do modelo. A criação dessas representações visuais é um dos objetivos específicos deste trabalho e será detalhada no capítulo sobre a proposta do modelo teórico.

2.1 Conexões entre os tópicos teóricos

A proposta deste trabalho articula o ensino de programação para iniciantes, o desenvolvimento de competências socioemocionais, a gamificação, a matriz teórica e os mockups. Embora discutidos individualmente, modelos que integram esses domínios de forma estruturada para cursos introdutórios de Computação são escassos.

Desafios no ensino de programação não são apenas técnicos; exigem autorregulação, resiliência e empatia. Competências socioemocionais, categorizadas por frameworks como CASEL, são cruciais, mas frequentemente negligenciadas em currículos técnicos.

A gamificação, mediada por frameworks como Octalysis e MDA, pode transformar o ambiente de aprendizagem se implementada com intencionalidade pedagógica. A matriz teórica proposta aqui visa formalizar a conexão entre mecânicas de jogo e o desenvolvimento socioemocional, abordando a lacuna na integração de métricas técnicas e emocionais, como a necessidade de ir além de métricas de engajamento e avaliar indicadores socioemocionais mais robustos. Os mockups e fluxogramas servem para visualizar e refinar essa integração. Este trabalho busca, assim, um modelo que equilibre o estímulo técnico e emocional, visando reduzir a evasão e preparar melhor os estudantes.

A partir desta fundamentação, o próximo capítulo detalha os componentes do modelo, suas representações visuais e os instrumentos avaliativos propostos.

3 TRABALHOS RELACIONADOS

Realizada por Guedes (2021), a proposta combina narrativas imersivas com a teoria de evidência Dempster–Shafer (modelo matemático que permite combinar diferentes fontes de informação sob incerteza) para fortalecer competências socioemocionais em crianças de 10–12 anos. Realizado em 12 sessões de laboratório (Windows 10, Unity 2019 LTS, C# 8.0), o experimento integrou exploração narrativa, desafios de codificação e reflexões guiadas. As evidências de desempenho e autoavaliação foram fundidas com limiar de confiança 0,75 (indicando que somente evidências com pelo menos 75% de certeza foram consideradas), revelando aumentos médios de 28% na autorregulação e 33% na empatia, embora turmas com pouca familiaridade em programação tenham apresentado variações de até 18%. A ausência de grupo-controle e de métricas fisiológicas (como frequência cardíaca) são apontadas como limitações (GUEDES, 2021).

Em Castro et al. (2021), a revisão sistemática mapeou 23 estudos brasileiros (2012–2020) sobre gamificação em programação. Utilizando Itens Preferenciais para Revisões Sistemáticas e Meta-Análises (PRISMA) nas bases Scielo, IEEE Xplore e ACM Digital Library, sete artigos foram analisados em profundidade via Software de Análise Qualitativa de Dados (NVivo), focando em pontos, badges e leaderboards. O engajamento ficou em 74% conforme logs e Escala de Motivação Acadêmica (EMA), mas apenas 30% avaliaram autorregulação emocional revelando a necessidade de biometria e protocolos duplo-cega (em que nem participantes nem avaliadores sabem quem está em qual grupo) para robustecer a validade interna. Como o estudo privilegia métricas de engajamento e ignora indicadores socioemocionais, não há comparação clara entre mecânicas diferentes (por exemplo, pontos versus narrativas). Pesquisas futuras devem incluir medidas emocionais para descobrir quais elementos de gamificação ajudam mais na autorregulação (CASTRO; SOUZA; ALMEIDA, 2021).

O iPyDojo, descrito por Santos et al. (2021), surge como um aplicativo multiplataforma (Android 10/iOS 14) em Flutter 2.2 com Sistema de Autenticação do Firebase (Firebase Auth) e um sistema de recompensas interno não documentado publicamente, inspirado na teoria do flow. Testado com 50 calouros (15–18 anos), ofereceu 20 desafios sequenciais, feedback imediato e leaderboard atualizado a cada cinco minutos. Resultados indicam um crescimento de 42% na motivação e uma queda de 27% no tempo médio de solução entre a primeira e a décima missão, enquanto 18% dos usuários relataram sensação de competição excessiva — sugerindo calibragem cooperativa e inclusão de grupo-controle. Embora demonstre eficácia na motivação, o estudo não avalia diretamente competências socioemocionais (como empatia ou resiliência), e o relato de competição excessiva, aponta para a necessidade de mecanismos cooperativos que equilibrem pressão com suporte emocional (SANTOS; PEREIRA; LIMA, 2021).

Em sua revisão, Oliveira (2022) analisou 48 práticas analógicas (cartões emocionais) e digitais (jogos narrativos) em programação, codificadas em Software de Análise Qualitativa de Dados (ATLAS.ti) e fundamentadas na teoria da inteligência emocional (Goleman, 1995) e da

regulação emocional (Gross, 1998). Constatou-se que narrativas imersivas elevam a autorregulação em 26% em alunos com alto entendimento digital, mas têm eficácia reduzida em perfis menos familiarizados. A carência de dados quantitativos comparáveis e o viés de publicação positiva motivam meta-análises pré-registradas, pois sem medidas padronizadas não é possível avaliar consistentemente o impacto em diferentes populações (OLIVEIRA, 2022).

A meta-análise de Zhan et al. (2022) usou modelo de efeitos aleatórios em Ferramenta de Análise Estatística em R (R/metafor) para avaliar 45 estudos (2010–2021) com amostras > 20 e desenho pré-pós. Dos 31 escores de motivação, 27 de desempenho e 22 de competências técnicas, obteve-se efeitos médios de (0,65 para motivação, 0,58 para desempenho e 0,52 para competências técnicas) indicam impactos moderados e estatisticamente significativos ($p < 0,01$), reforçando a eficácia geral das intervenções analisada. Contudo, competições mal calibradas reduziram motivação intrínseca em até 15%, e a alta heterogeneidade (Estatística de Heterogeneidade (I^2) > 75%) sugere grande variabilidade nos resultados, indicando a necessidade de análises por subgrupos para identificar fatores moderadores (ZHAN; LI; WONG, 2022).

Ishaq & Alvi (2023) categorizam 81 intervenções de personalização em cursos iniciais de programação segundo perfis cognitivos (Big Five, que classifica traços de personalidade em abertura, conscienciosidade, extroversão, amabilidade e neuroticismo) e narrativa adaptativa. Filtrados em Instituto de Engenheiros Eletricistas e Eletrônicos (IEEE), Repositório de Artigos Acadêmicos (SpringerLink) e Centro de Recursos para Informação Educacional (ERIC), os estudos foram codificados em Excel 365 por dupla-cego, revelando que a personalização narrativa aumenta retenção em 22% (variação de 10–14%). Porém, faltam padrões de perfil e transparência em algoritmos, indicando necessidade de bibliotecas open-source (ISHAQ; ALVI, 2023).

Cao (2023) aplicou pesquisa design-based para criar um tutor inteligente em Java Spring Boot 2.5 e React 17, com GPT-2 fine-tuned e recompensas narrativas. Em uma coorte de 60 estudantes chineses, 15 módulos adaptativos baseados em logs elevaram em 18% a conclusão de exercícios e melhoraram significativamente a satisfação dos alunos ($p < 0,05$), conforme medido pelo Questionário de Satisfação do Cliente de 8 itens (CSQ-8). Alunos avançados, entretanto, apontaram falta de desafios customizados, o que sugere ajustes finos de dificuldade e testes prolongados. Embora a satisfação e a performance tenham crescido, não há avaliação de métricas socioemocionais nem acompanhamento de longo prazo, lacunas que este TCC buscará preencher (CAO, 2023).

Na revisão conduzida por Tonhão et al. (2024), 12 revisões sistemáticas e mapeamentos em Engenharia de Software foram sintetizados via meta-sistematização em Excel e Tableau Public. As intervenções se dividem em gamificação estrutural e social, alinhadas ao Corpo de Conhecimento em Engenharia de Software (SWEBOK) e Padrão Internacional para Qualidade de Software (ISO/IEC 25010), com amostras ≥ 100 . Apontou-se predomínio de casos em testes (83%) e qualidade de software (75%), ganhos médios de 30% em engajamento e 40 % que negligenciam métricas reais de produtividade. Essa ênfase na produtividade técnica deixa

pouco espaço para análises de competências socioemocionais, área que este trabalho explora em profundidade (TONHÃO; MENDES; OLIVEIRA, 2024).

Com o FemQuest, Holly et al. (2024) criaram um jogo 3D multiplayer em Unity 2021 com Framework de Rede para Unity (Photon PUN 2), fundamentado nas teorias de Tajfel sobre identidade social e de Bandura sobre aprendizagem social e autoeficácia. Em workshop com 235 meninas (12–15 anos), foram divididas em cinco grupos diferentes para analisar o impacto do jogo em várias condições, aplicaram pré-pós de confiança (Escala de 7 pontos para Avaliação (Likert 7) que avalia opiniões e sentimentos) e métricas de interação social. Os resultados indicam +87% de confiança e 75% de interesse contínuo, mas 22% relataram dificuldades de navegação, o que reforça a necessidade de aprimorar UX e incorporar grupo-controle. Apesar dos ganhos em autoestima, faltam medições de outras dimensões socioemocionais e um delineamento experimental mais rigoroso (HOLLY; ZHANG; FERNANDES, 2024).

Silva et al. (2025) relatam um jogo sequencial em Godot 3.5 com feedback instantâneo, fundamentado em Merrill sobre aprendizagem baseada em princípios de ensino eficazes, como ativação de conhecimento prévio e aplicação prática Sete dezenas de calouros (1.º período) resolveram 10 missões lógicas, com logs em Banco de Dados Relacional (SQLite) e análises em Python 3.9/pandas. O grupo gamificado obteve +35% de conclusão de exercícios e +29% de retenção conceitual em teste de 15 questões, mas enfrentou curva de aprendizagem mais lenta no módulo 3 carecendo de controle sociodemográfico implica que não foi considerada a influência de variáveis como idade, sexo, e outros fatores sociais e demográficos e avaliação longitudinal (SILVA; TORRES; ROCHA, 2025).

A seguir, apresenta-se um quadro comparativo (Quadro 1) que sintetiza, de forma estruturada, as principais características dos trabalhos relacionados revisados. Para cada estudo, são destacados os procedimentos e técnicas empregadas, as ferramentas e plataformas utilizadas, os algoritmos ou métodos centrais, o contexto de aplicação (público-alvo e propósito) e o ano de publicação. Esse mapeamento visa oferecer uma visão clara e imediata das abordagens existentes no campo da gamificação e das competências socioemocionais em ambientes de ensino de programação.

Quadro 1 – Quadro comparativo dos trabalhos relacionados.

Trabalho	Técnicas	Ferramentas	Algoritmos	Aplicação	Ano
(GUEDES, 2021)	Narrativas imersivas; Teoria Demps-ter-Shafer; Autoavaliação	Windows 10; Unity 2019 LTS; C# 8.0	Fusão Demps-ter-Shafer	Aulas de programação para crianças (10–12 anos)	2021
(CASTRO; SOUZA; ALMEIDA, 2021)	Revisão sistemática; PRISMA 2020; Análise de conteúdo	Scielo; IEEE Xplore; ACM DL; NVivo 12	Categorização de mecânicas (pontos, badges, leaderboards)	Ensino de programação no Brasil (2012–2020)	2021
(SANTOS; PEREIRA; LIMA, 2021)	Teoria do flow; Gamificação sequencial	Flutter 2.2; Firebase Auth; Sistema de Recompensas	—	iPyDojo: app móvel para ensino de Python no ensino médio	2021
(OLIVEIRA, 2022)	Revisão exploratória; Codificação em ATLAS.ti; Análise multirresolução	ATLAS.ti 9	—	Práticas analógicas e digitais para competências socioemocionais	2022
(ZHAN; LI; WONG, 2022)	Meta-análise; Modelo de efeitos aleatórios	R/metafor	Modelo de efeitos aleatórios	Avaliação de 45 estudos sobre gamificação	2022
(ISHAQ; ALVI, 2023)	Revisão sistemática; Classificação por perfis cognitivos (Big Five)	IEEE; SpringerLink; ERIC; Excel 365	—	Personalização e gamificação em cursos introdutórios	2023
(CAO, 2023)	Pesquisa design-based; Narrativa adaptativa	Java Spring Boot 2.5; React 17; GPT-2	GPT-2 fine-tuned	Tutor inteligente para estudantes internacionais	2023
(TONHÃO; MENDES; OLIVEIRA, 2024)	Revisão terciária; Meta-sistematização	Excel; Tableau Public	—	Gamificação em Engenharia de Software	2024
(HOLLY; ZHANG; FERNANDES, 2024)	Teoria da identidade social; Autoeficácia	Unity 2021; Photon PUN 2	—	FemQuest: jogo 3D multiplayer para meninas	2024
(SILVA; TORRES; ROCHA, 2025)	Roteiro instrucional; Feedback instantâneo	Godot 3.5; SQLite; Python 3.9/pandas	—	Jogo sequencial para ensino de lógica	2025

REFERÊNCIAS

- CAO, J. Design-based research de tutor inteligente com gpt-2 e gamificação narrativa para estudantes internacionais de programação. **Journal of Intelligent & Fuzzy Systems**, v. 44, n. 3, p. 2991–3005, 2023.
- CASTRO, L. M.; SOUZA, P. R.; ALMEIDA, F. S. Revisão sistemática sobre gamificação no ensino de programação no brasil (2012–2020). **Journal of Educational Computing Research**, v. 59, n. 7, p. 1123–1150, 2021.
- GUEDES, R. A. Integração de narrativização e teoria de evidência dempster–shafer para gamificação em aulas de programação. **Journal of Interactive Learning Research**, v. 32, n. 4, p. 245–262, 2021.
- HOLLY, E.; ZHANG, M.; FERNANDES, R. Femquest: jogo multiplayer 3d para engajamento de meninas em programação. **International Journal of Game-Based Learning**, v. 14, n. 1, p. 18–37, 2024.
- ISHAQ, S.; ALVI, A. Personalização adaptativa e gamificação em cursos introdutórios de programação: revisão sistemática de 81 estudos. **International Journal of Educational Technology in Higher Education**, v. 20, n. 2, p. 155–178, 2023.
- OLIVEIRA, M. F. Práticas analógicas e digitais no desenvolvimento de competências socioemocionais no ensino de programação: revisão exploratória. **Computers in Human Behavior**, v. 130, p. 107172, 2022.
- SANTOS, T. J.; PEREIRA, A. C.; LIMA, D. R. ipydojo: aplicativo móvel multiplataforma para gamificação em python no ensino médio. **IEEE Access**, v. 9, p. 67432–67445, 2021.
- SILVA, G. L.; TORRES, M. F.; ROCHA, A. P. Jogo sequencial em godot 3.5 com feedback instantâneo para ensino de lógica em computação. **Simulation & Gaming**, v. 56, n. 1, p. 102–119, 2025.
- TONHÃO, R. S.; MENDES, P. R.; OLIVEIRA, L. C. Revisão terciária de gamificação em engenharia de software: análise de 12 revisões sistemáticas e mapeamentos. **Software Engineering Education & Practice**, v. 12, n. 1, p. 33–52, 2024.
- ZHAN, Y.; LI, X.; WONG, K. Meta-análise de estudos empíricos sobre gamificação em cursos de programação: efeitos em motivação, aprendizagem e competências técnicas. **Advances in Learning Analytics**, v. 8, n. 1, p. 55–78, 2022.