

# XPCoDeHub: Um modelo de sistema gamificado para o aprendizado de programação através de trilhas de conhecimento

Gustavo Steinmetz, Luís Guilherme Eich, Gustavo Lazarotto Schroeder,  
Rosemary Francisco, Jorge Luis Victória Barbosa

Universidade do Vale do Rio dos Sinos (UNISINOS)  
Av. Unisinos, 950 – Cristo Rei, São Leopoldo – RS, 93022-000 – Brazil

sbie@gustavostz.com, luisguilherme@unisinos.br,  
gschroeder@edu.unisinos.br, {rosemaryf, jbarbosa}@unisinos.br

**Abstract.** *Computer programming is a strategic skill, but difficulties in learning it often influence student engagement. In this scenario, gamification emerges as an alternative for stimulating programming learning. This article proposes the XPCoDeHub, a gamified system model for learning programming through knowledge trails. The knowledge trails are created through a combination of gamification elements, such as objectives, levels, and progression, and allow to guide and motivate students in their learning process. A usability evaluation was conducted through the TAM model, involving 10 users, and indicated high rates of ease of use (96%) and perceived usefulness (94%).*

**Resumo.** *A programação de computadores é uma habilidade estratégica, mas as dificuldades para o seu aprendizado costumam influenciar o engajamento dos alunos. Nesse cenário, a gamificação surge como uma alternativa que permite estimular o aprendizado de programação. Este artigo propõe o XPCoDeHub, um modelo de sistema gamificado para o aprendizado de programação através de trilhas de conhecimento. As trilhas são criadas por meio de uma combinação de elementos de gamificação, como objetivos, níveis e progressão, e possibilitam guiar e motivar os alunos em seu processo de aprendizado. Uma avaliação de usabilidade foi conduzida através do modelo TAM, envolvendo 10 usuários, e indicou altos índices de facilidade de uso (96%) e utilidade percebida (94%).*

## 1. Introdução

Segundo o relatório *The Future of Jobs Report 2020* do Fórum Econômico Mundial<sup>1</sup>, a programação de computadores está entre uma das 10 mais importantes habilidades para os futuros trabalhadores. Além disso, atualmente há uma demanda significativa por programadores altamente qualificados na indústria de TIC [Rahman et al. 2022]. No entanto, aprender a programar não é trivial, sendo indicado na literatura como algo desafiador. Estes desafios no aprendizado podem levar a perda de interesse e engajamento por parte dos alunos [Rouhani et al. 2022].

<sup>1</sup>[https://www3.weforum.org/docs/WEF\\_Future\\_of\\_Jobs\\_2020.pdf](https://www3.weforum.org/docs/WEF_Future_of_Jobs_2020.pdf)

Como os alunos vêm de diferentes formações acadêmicas, muitos deles acham difícil aprender programação [Mekouar 2022]. Essas dificuldades despertam sentimentos de ansiedade [Fuchs and Wolff 2016] e influenciam negativamente o engajamento e a motivação dos alunos no processo de aprendizado. O desafio se torna ainda maior quando o ensino de programação é no ambiente *online* [Mekouar 2022]. Cursos *online* de ensino a programação possuem altas taxas de desistência e baixo engajamento [Hagedorn et al. 2022], e a conclusão dos cursos varia entre 10% e 20% [Lehtonen et al. 2015, Miller et al. 2014]

Neste contexto, abordagens como a *gamificação* podem ser adotadas para auxiliar na motivação e engajamento dos alunos durante o processo de aprendizado de programação [Palaniappan and Noor 2022, Jusas et al. 2022]. A *gamificação* consiste no uso de elementos do *design* de jogos em outros contextos [Deterding et al. 2011]. Presente em vários domínios, o conceito de *gamificação* tem ganhando cada vez mais destaque na área educacional [Kučak and Kučak 2022], devido a sua capacidade de ampliar a motivação, o interesse e o desempenho do aluno no processo de aprendizado [Bachtiar et al. 2018, Paiva et al. 2022].

Diante deste cenário, esse artigo apresenta um modelo de sistema *gamificado*, o XPCodeHub, cujo propósito é auxiliar no aprendizado de programação através de trilhas de conhecimento. Aprender programação exige um esforço significativo dos aprendizes, visto que demanda um tempo considerável de prática para o aprendizado [Rodrigues et al. 2022]. Desta forma, o modelo propõe trilhas de conhecimento, construídas através da combinação de elementos de *gamificação* considerados essenciais para o engajamento e motivação dos estudantes [Toda et al. 2019], como: objetivos, níveis e progressão. Além disso, observou-se na literatura a falta de um sistema *gamificado* como o proposto, dedicado ao aprendizado teórico e prático da programação e que guiasse o aluno em seu processo de aprendizado.

O restante do artigo está estruturado em 5 seções. A seção 2 apresenta os trabalhos relacionados. A seção 3 aborda os detalhes do modelo proposto, como sua arquitetura e ontologia. A seção 4 descreve a implementação e a avaliação do XPCodeHub. Por fim, a seção 5 apresenta conclusões e sugestões de trabalhos futuros.

## 2. Trabalhos Relacionados

Durante o desenvolvimento deste trabalho, um mapeamento sistemático da literatura foi realizado com o intuito de analisar a aplicação da *gamificação* no processo de aprendizado de programação de computadores. Devido a limitações de extensão, esta seção apresenta a análise de trabalhos relevantes para o escopo deste artigo. Para esta seleção, utilizou-se como critério, estudos que aplicaram *gamificação* no suporte ao processo de aprendizado de programação de computadores, publicados recentemente. A Tabela 1 apresenta a análise destes trabalhos e uma comparação com o modelo proposto neste artigo.

O trabalho de Jusas et al. (2022) usou *gamificação* com o intuito de aumentar o engajamento dos alunos em um curso de programação orientada a objetos. O trabalho também visou impulsionar o aprendizado e a aquisição de habilidades por meio de um comportamento mais sustentável dos alunos, como a prática constante de programação. O trabalho utilizou quatro elementos de *gamificação* com a plataforma *Moodle* e *plugins* adicionais para conter pontos de experiência e conteúdo interativo. Os elementos

**Tabela 1. Análise e comparação dos trabalhos com o modelo proposto**

Critérios	Jusas et al. (2022)	Palaniappan e Noor (2022)	Soepriyanto et al. (2022)	Rodrigues et al. (2022)	Paiva et al. (2022)	XPCoDeHub
Uso da gamificação	Aumentar o engajamento dos alunos	Aumentar o engajamento e motivação em cursos <i>online</i>	Aumentar o engajamento e motivação em cursos <i>online</i>	Aumentar o desempenho acadêmico dos alunos	Aumentar o engajamento e diminuir a evasão dos alunos	Aumentar o engajamento e motivação durante o aprendizado
Elementos de gamificação utilizados	Pontos de experiência, conteúdo interativo, colaboração com equipe local e global	Emblemas, distintivos, pontos e quadro de liderança	Desafios, regras, emblemas, distintivos, pontos e tabelas de classificação	Narrativas, Avatar, Desafios, Progressão, Pontos, Recompensas, Competição	Desafios, missões, duelos, tabelas de classificação, pontuações, níveis, distintivos	Desafios, pontuações, trilhas de conhecimento (objetivos, níveis e progressão), tabelas de classificação, habilidades de RPG, feedback imediato e <i>boosts</i>
Plataforma utilizada	Moodle com dois plugins adicionais para os pontos de experiência e conteúdo interativo	Moodle versão 3.1	Moodle e Whatsapp	CodeBench, um juiz online criado por um dos autores	Plataforma de código aberto e propósito geral para criação de atividades gamificadas	Modelo de sistema gamificado e um protótipo dedicado para o aprendizado de programação
Público-alvo	Alunos aprendendo programação	Alunos aprendendo programação em curso <i>online</i>	Alunos aprendendo programação em curso <i>online</i>	Alunos aprendendo programação	Professores preparando aulas gamificadas	Alunos aprendendo programação
Conteúdo de aprendizado	POO (Programação Orientada a Objetos)	Introdução a Programação em Python	Introdução a Delphi - POO Visual (Programação Visual Orientada a Objetos)	Introdução a programação	Programação de computadores no geral	Conteúdos de programação com base em trilhas de conhecimento
Metodologia	Quasi-Experimento com 27 alunos de uma turma e comparação dos resultados com turmas passadas	Experimento com 29 estudantes	Estudo com 101 alunos e avaliação por meio de questionário	Quasi-Experimento com 399 alunos e comparação dos resultados com turmas passadas	Pesquisa online com 8 professores de programação e 15 estudantes de graduação com um questionário de <i>Usability Metric for User Experience</i>	Avaliação por meio de dois cenários realizados com 5 pessoas com conhecimentos em programação e 5 pessoas sem experiência
Principal resultado	Aumento nas notas dos alunos e redução no número de alunos solicitando segunda tentativa no exame final	Aumento do desempenho dos alunos no autoaprendizado em um ambiente de aprendizado <i>online</i>	O uso dos tipos de jogadores não influenciaram nos resultados de aprendizado dos alunos	Os alunos que praticaram na versão gamificada apresentaram desempenho acadêmico superior	Os professores avaliaram positivamente a ferramenta	A avaliação de usabilidade foi conduzida através do modelo TAM, e indicam altos índices de facilidade de uso (96%) e utilidade percebida (94%)

de *gamificação* utilizados foram pontos de experiência, conteúdo interativo, equipe local (pequenos grupos de alunos que trabalham juntos para um objetivo comum) e trabalho em time global (grandes grupos de alunos que colaboram e ajudam reciprocamente). Entre os resultados, observou-se um aumento significativo nas notas dos alunos.

O trabalho de Palaniappan e Noor (2022) também tinha como propósito aumentar o engajamento e motivação dos alunos, porém em um curso *online* de programação em Python. O curso em foco se caracteriza como um MOOC (*Massive Online Open Course*), em que o aluno é responsável pelo seu próprio percurso e aprendizado. O trabalho utilizou os elementos clássicos de *gamificação*, como o PBL (*Points, Badges and Leaderboards*), com a versão 3.1 da plataforma *Moodle*. O resultado indicou um aumento do desempenho dos alunos no ambiente de aprendizado *online*.

Soepriyanto et al. (2022) investigaram o efeito dos tipos de jogadores [Bartle 1996] no engajamento e motivação dos alunos, em um curso *online* de introdução a linguagem de programação Delphi. O trabalho usou os elementos clássicos de *gamificação*, como o PBL (*Points, Badges and Leaderboards*), além de desafios e regras. O *Moodle* e o WhatsApp foram usados como plataforma de aprendizagem. O estudo foi realizado com 101 estudantes e apesar de ter sido identificado um aumento do desempenho dos alunos no ambiente de aprendizado *online*, não se observou diferença significativa entre os tipos de jogadores nos resultados de aprendizado dos alunos.

Rodrigues et al. (2022) realizaram um estudo onde elementos de *gamificação* foram adicionados a uma ferramenta de juiz *online*. Uma ferramenta de juiz *online* tem foco em competição e resolução de desafios de programação. No estudo foram utilizados elementos como narrativas com base em um mundo fictício, avatares para os alunos escolherem, desafios, progressão, pontos e recompensas. As recompensas eram baseadas na troca de pontos por armas a serem utilizadas no mundo fictício. Os resultados indicaram que os alunos que praticaram na versão gamificada apresentaram desempenho acadêmico superior aos que praticavam a mesma quantidade na versão não gamificada.

Por fim, o trabalho de Paiva et. al (2022) apresenta uma plataforma que aplica a *Gamified Education Interoperability Language* (GEdIL) [Swacha et al. 2020] para a elaboração de uma interface de criação de atividades *gamificadas*. A plataforma permite que professores criem exercícios de programação e depois adicionem camadas de *gamificação* com a GEdIL. Tais elementos incluem desafios, missões, duelos, tabelas de classificação, pontuações, níveis, *badges*, entre outros. Os professores avaliaram positivamente a ferramenta em todas as dimensões consideradas.

A Tabela 1 permite observar uma lacuna existente e uma oportunidade de desenvolvimento de um modelo de sistema gamificado, dedicado ao aprendizado teórico e prático da programação, que guie o aluno em seu processo de aprendizado. Para tanto, o modelo XPCodeHub usa trilhas de conhecimento, que combina elementos como objetivos, níveis e progressão, e permite indicar a evolução do aluno através de uma árvore de habilidades conforme abordado em jogos RPG. Além disso, no caso dos desafios de programação, o modelo recorre a ferramentas do cotidiano de um programador, tornando a experiência de aprender programação mais semelhante à realidade.

### 3. Modelo XPCodeHub

Esta seção apresenta o XPCodeHub, um modelo *gamificado* para auxiliar no aprendizado de programação. Trata-se de um modelo de tecnologia no ensino-aprendizagem de disciplinas de computação [Bispo Jr et al. 2020].

#### 3.1. Visão Geral

O nome XPCodeHub diz respeito às três principais partes do modelo: “XP”, “Code” e “Hub”. “XP” representa a experiência do usuário, que pode ser observada pela interface do usuário com o sistema gamificado. O modelo utiliza os seguintes elementos de gamificação: pontuação, *boosts*, desafios, objetivos, níveis, progressão, tabela de classificação, *feedback* instantâneo e mecanismos de RPG como habilidades e árvores de talentos. As habilidades evoluem conforme atividades são realizadas pelo usuário. Por exemplo, se for feita uma atividade da linguagem de programação “JAVA”, o usuário terá um incremento da habilidade “JAVA”. Além disso, as árvores de talentos são trilhas de desenvolvimento a serem seguidas. O uso de trilhas de conhecimento permite ao aluno construir seu conhecimento progressivamente e ter uma papel de condutor na sua jornada de conhecimento [Elshani and Nuçi 2021]. No caso deste modelo, por exemplo, caso o aluno escolha a trilha de *back end* será necessário primeiro passar pelo módulo de “Lógica de Programação” para depois poder fazer o de “Orientação a Objetos”.

“Code” representa a prática contínua de codificação. Ao expor o aluno a prática constante, o aluno se sente cada vez mais seguro ao programar [Rodrigues et al. 2022]. Além disso, o modelo utiliza ferramentas que permitem o aluno executar e avaliar o código implementado antes de submetê-lo para a avaliação pelo instrutor. Após o envio do código são executados testes unitários, retornando ao aluno um *feedback* instantâneo. “Hub”, por fim, representa a integração de todas essas ferramentas em um único lugar, permitindo a interação social dos alunos através dos elementos do sistema.

#### 3.2. Arquitetura

O modelo XPCodeHub foi projetado usando a modelagem técnica de arquitetura da SAP (TAM, do inglês, *Standard for Technical Architecture Modeling*)<sup>2</sup>. A Figura 1 apresenta arquitetura do XPCodeHub composta por atores (Aluno e Instrutor), acessos, bloco (Servidor Aplicação XPCodeHub) e componentes. Os componentes do modelo aparecem na parte interna do bloco Servidor Aplicação XPCodeHub.

A arquitetura é composta de dois componentes, quatro módulos e um banco de dados. Na parte superior da Figura 1 é ilustrada a comunicação entre os atores e o *front end*, o qual é acessado via navegador. O *front end* se comunica com a API REST (do inglês, *Application Programming Interface Representational State Transfer*) através do protocolo de rede HTTPS. Tal comunicação REST é feita através de mensagens JSON (*JavaScript Object Notation*).

No componente da API REST existem 4 módulos:

1. **Autenticação Usuário:** este módulo é responsável pela segurança e autenticação dos usuários, alunos e instrutores;

<sup>2</sup><http://www.fmc-modeling.org/fmc-and-tam>

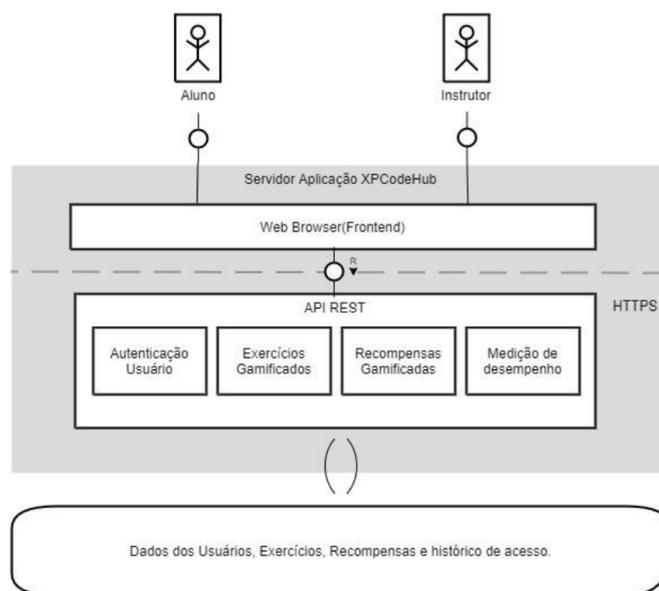


Figura 1. Arquitetura do XPCodeHub utilizando padrão TAM/SAP

2. **Exercícios Gamificados:** este módulo é responsável por todos os exercícios e pelas consultas no banco dados para trazer os dados referentes as perguntas e suas respostas. Em caso de exercícios envolvendo código este módulo executa os testes validando se tal exercício foi feito corretamente e retorna um *feedback* instantâneo para o aluno;
3. **Recompensas Gamificadas:** este módulo recompensa o aluno por meio de bonificações. Isso ocorre toda vez que o aluno faz alguma atividade ou realiza algum objetivo;
4. **Medição de desempenho:** este módulo analisa o progresso das pontuações dos alunos, fazendo um paralelo com o histórico de acessos, assim conseguindo medir quais alunos estão mais engajados e comprometidos com o aprendizado de programação.

A estrutura do banco de dados do XPCodeHub foi organizada conforme a Ontologia apresentada na próxima seção.

### 3.3. Ontologia

Arquiteturas sensíveis a trilhas de conhecimento precisam armazenar os contextos observados para uso posterior. Para representar o conhecimento destes contextos, pode-se usar uma ontologia, com o intuito de apresentar as entidades envolvidas, bem como sua relação de forma semântica [Munir and Anjum 2018]. A Figura 2 ilustra uma representação de conhecimento do domínio de dados de contexto utilizada pelo XPCodeHub. De uma perspectiva ontológica [Guarino et al. 2009], a representação proposta é uma conceituação, na qual os conceitos-chave utilizados pelo XPCodeHub são representados.

A entidade *Instrutor* representa a pessoa responsável por prestar auxílio aos alunos e avaliá-los. A entidade *Aluno* representa a pessoa com o objetivo de aprender programação. Tanto *Aluno* quanto *Instrutor* herdam características da entidade *Usuário*, que representa os dados usados para acessar o sistema.

As entidades que compõem o módulo de recompensas são *Conquista* e *Pontuação*. A entidade *Habilidade* representa a habilidade do aluno em certa linguagem de programação, que pode ser incrementada através de exercícios. A entidade *Exercício* representa as possíveis atividades a serem feitas para aprender programação, sendo as mesmas perguntas ou exercícios. A entidade *Pergunta*, por sua vez, pode ser utilizada como uma pergunta de múltipla escolha ou uma questão de verdadeiro ou falso. Já a entidade *Programação* é responsável pelos exercícios de programação. Por fim a entidade *TrilhaDesenvolvimento* representa a trilha de atividades que o aluno irá seguir (*front end*, *mobile* ou *back end*).

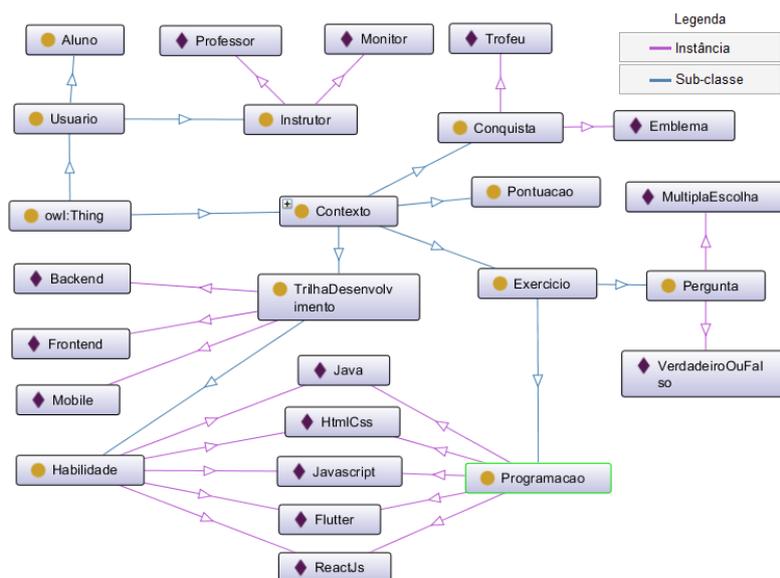


Figura 2. Ontologia do XPCodeHub

#### 4. Implementação e avaliação do modelo

As subseções a seguir apresentam a implementação do protótipo do modelo, bem como a avaliação realizada através de um questionário baseado no modelo TAM [Davis 1989, Marangunić and Granić 2015], por 10 usuários divididos em dois grupos, pessoas com e sem experiência prévia na área de programação.

##### 4.1. Protótipo

O *back end* foi implementado em um servidor Apache Tomcat, na linguagem Java 11 com o framework Spring Boot<sup>3</sup> e gerenciador de dependências Maven<sup>4</sup>. As comunicações assíncronas foram implementadas com o *middleware* RabbitMQ<sup>5</sup>. As comunicações síncronas oriundas das requisições do *front end* para o *back end* foram implementadas em arquitetura REST. Por se tratar de uma aplicação *stateless* foi utilizado JWT (*JSON Web Token*) para a comunicação entre as partes e criptografia BCrypt. O banco de dados utilizado foi PostgreSQL<sup>6</sup> 14.3. O *front end* foi construído em formato *Single Page*

<sup>3</sup><https://www.spring.io/projects/spring-boot>

<sup>4</sup><https://www.maven.apache.org/>

<sup>5</sup><https://www.rabbitmq.com/>

<sup>6</sup><https://www.postgresql.org/>

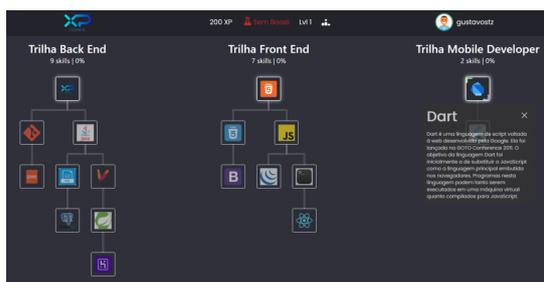


Figura 3. Página Inicial

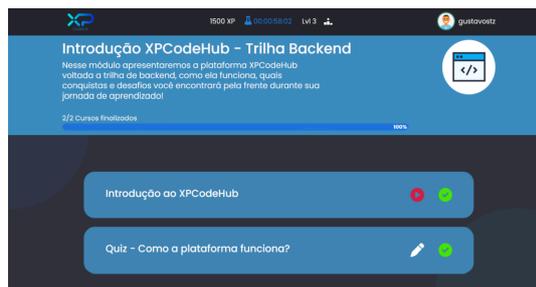


Figura 4. Módulo de Introdução

Application, usando o framework ReactJs<sup>7</sup>.

A Figura 3 mostra a tela inicial do protótipo, com três trilhas de aprendizado (*Back end*, *Front End* e *Mobile*). A Figura 4 apresenta o módulo de introdução ao XPCoDeHub com uma videoaula sobre o funcionamento da plataforma e *quiz* com *feedback* imediato, com perguntas sobre o módulo. A pontuação do aluno depende de seu desempenho no *quiz*. Caso acerte todas as perguntas o aluno ganhará 100 pontos, entre 100% e 60% receberá 50 pontos. Abaixo de 60% é necessário que o aluno refaça o *quiz*.

A Figura 4 ilustra um tempo de *boost* na barra superior da tela. Enquanto o *boost* estiver ativo, as pontuações recebidas serão em dobro. Para o usuário conseguir tal *boost* basta completar todos os tópicos de uma habilidade que ele ganhará uma hora de *boost* e 500 pontos de recompensa. O nível do aluno é calculado pela equação  $\sqrt{\frac{(x*2)}{1000}} + 1$ , com a qual é mais fácil evoluir de nível no início e mais difícil conforme o aluno avança, representando o elemento de progressão da *gamificação*.

#### 4.2. Avaliação com usuários

A avaliação por meio de cenários é uma abordagem utilizada pela comunidade científica [Satyanarayanan 2001, Koulouri et al. 2021]. Foram criados dois cenários para a avaliação do XPCoDeHub. O primeiro abrange usuários que não tenham experiência prévia em programação. O cenário 2 abrange profissionais com experiência em programação de computadores, mas que estão buscando expandir suas habilidades.

Para a aplicação dos cenários foi solicitado a dez participantes que realizassem o seu respectivo cenário na plataforma XPCoDeHub. Dentre os dez participantes, cinco são profissionais da área de programação e cinco são pessoas sem experiência na área. O roteiro seguido para a aplicação dos cenários foi: apresentação da proposta do trabalho; solicitação da execução do cenário, ressaltando que na primeira habilidade, “Introdução ao XPCoDeHub”, existe uma videoaula explicando como a plataforma funciona.

Após a execução do cenário foi aplicado um questionário elaborado com base no modelo de aceitação de tecnologia (TAM), proposto por Davis (1989) e revisado por Marangunic e Granic (2015). O modelo TAM considera como critérios para aceitação a facilidade de uso percebida e a utilidade percebida. As opções de resposta para cada afirmação seguiram o padrão da escala Likert de 5 pontos. A Tabela 2 apresenta as afirmações, onde os itens de 1 a 5 consideram a facilidade de uso e as questões de 6 a 10 focam na utilidade percebida.

<sup>7</sup><https://www.reactjs.org/>

**Tabela 2. Afirmções usadas no questionário de avaliação**

Item	Afirmção
1	A interface é intuitiva e permite navegar com facilidade através dela.
2	Não houve dificuldade na utilização da plataforma XPCodeHub.
3	A <i>gamificação</i> contribui positivamente na usabilidade da plataforma.
4	A estrutura da trilha de conhecimento em forma de árvore de habilidades (estilo RPG) faz o usuário ter uma clareza maior sobre sua jornada de aprendizado.
5	O feedback instantâneo dos resultados das atividades contribui para uma maior clareza do progresso do usuário.
6	O sistema de pontuação e níveis motiva o uso da plataforma para tentar conseguir mais pontos e avançar mais níveis e consequentemente motivando o aluno a aprender mais.
7	A tabela de classificação motiva o aluno a conseguir mais pontos para ficar entre os melhores e consequentemente motivando o aluno a aprender mais.
8	O Boost de pontuação motiva o aluno a seguir utilizando a plataforma por mais tempo para conseguir mais pontos.
9	A utilização de ferramentas de desenvolvimento para os desafios de programação contribui para um ambiente de aprendizado mais realista, por utilizar ferramentas do cotidiano de um programador.
10	A plataforma ajuda no aprendizado de programação.

Também foram realizadas quatro perguntas abertas, de modo a compreender melhor as opiniões dos participantes a respeito do protótipo. As perguntas questionavam sobre os pontos positivos e negativos, se a pessoa acredita que a utilização desta ferramenta pode impulsionar a aprendizagem de programação, além da percepção das pessoas sobre os elementos de *gamificação*.

### 4.3. Resultados

Os resultados do questionário apresentados nas Figuras 5 e 6 evidenciam uma forte aceitação nos dois cenários. Em relação à facilidade de uso (afirmações 1 a 5), as afirmações de 3 a 5 tiveram uma unanimidade de concordância. A afirmação com menos aceitação total foi a que se relacionava à dificuldades no uso da plataforma XPCodeHub, principalmente por aqueles que não tinham experiência na área de programação.

Em relação à utilidade percebida (afirmações 6 a 10), nota-se que a afirmação relativa à ajuda prestada pelo XPCodeHub no aprendizado teve concordância unânime. O modelo é visto de forma positiva no quesito utilidade percebida para o aprendizado de programação. Nota-se também que há uma unanimidade na afirmação 9 por parte dos participantes com experiência em programação, entretanto isso não se reflete por parte daqueles sem experiência. Isso pode ser justificado pelo fato de que quem não tem tal experiência acaba tendo mais dificuldades para utilizar ferramentas do cotidiano de um programador. Tal justificativa também pode ser aplicada para a afirmação 2, pois na pergunta de viés qualitativo sobre pontos positivos e negativos, o ponto negativo mais recorrente dos participantes sem experiência foi a dificuldade na utilização do Git<sup>8</sup> para conseguir fazer alguns desafios da plataforma, aparecendo duas vezes tal citação sobre o Git e uma sobre o GitHub<sup>9</sup>.

Ambos os grupos tiveram a aceitação máxima para a afirmação 4 que se refere a trilha de conhecimento. Por fim, as afirmações 6 e 7, que retratam o sistema de pontuação

<sup>8</sup><https://www.git-scm.com/>

<sup>9</sup><https://www.github.com/>

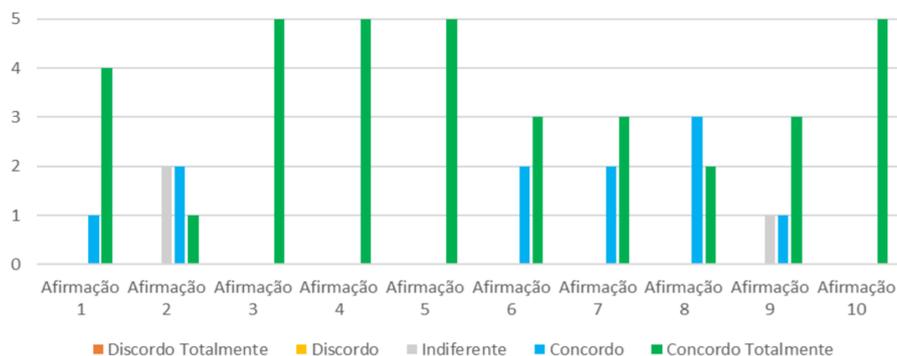


Figura 5. Resultado do Cenário 1

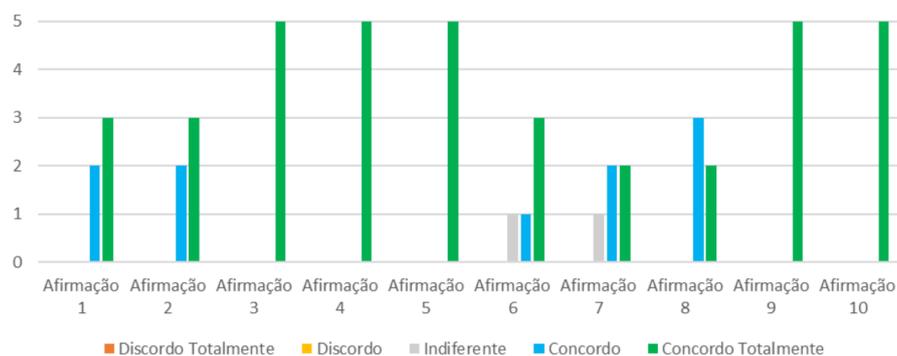


Figura 6. Resultado do Cenário 2

e a tabela de classificação, respectivamente tiveram o menor desempenho, porém, ainda com uma boa aceitação.

## 5. Conclusão

Esse trabalho propôs o XPCodeHub, um modelo de sistema gamificado para o auxílio no aprendizado de programação. O sistema tem como propósito auxiliar tanto pessoas com uma experiência prévia na área, quanto quem não tem nenhum conhecimento em programação, motivando-as no processo de aprendizado por meio da *gamificação*. O modelo utiliza trilhas de conhecimento o que possibilita guiar os alunos e os manter motivados no seu processo de aprendizado. As trilhas foram construídas com base na combinação de elementos essenciais de gamificação para os domínios de aprendizado, como objetivos, níveis e progressão. O modelo endereça a lacuna observada na literatura, de um sistema de gamificação dedicado ao aprendizado teórico e prático de programação.

Assim como nos trabalhos relacionados, o modelo de sistema gamificado teve um resultado positivo em sua avaliação. Inclusive, os participantes da avaliação perguntaram posteriormente se o sistema estaria disponível para o uso, visto que tinham interesse de indicar o XPCodeHub para outras pessoas interessadas em aprender programação. Dentre as limitações deste trabalho destaca-se a avaliação não ter sido realizada com professores de programação. Trabalhos futuros podem incluir a avaliação do protótipo com professores além de um experimento, a fim de possibilitar uma melhor análise do desempenho dos alunos utilizando o XPCodeHub em relação a alunos que não o utilizaram.

## Referências

- Bachtiar, F. A., Pradana, F., Priyambadha, B., and Bastari, D. I. (2018). Coma: Development of gamification-based e-learning. In *2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE)*, pages 1–6.
- Bartle, R. (1996). Hearts, clubs, diamonds, spades: Players who suit muds. *Journal of MUD research*, 1(1):19.
- Bispo Jr, E. L., Raabe, A., Matos, E., Maschio, E., Barbosa, E., Carvalho, L., Bittencourt, R., Duran, R., and Falcão, T. (2020). Tecnologias na educação em computação: Primeiros referenciais. *Revista Brasileira de Informática na Educação*, 28:509–527.
- Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, 13(3):319–340.
- Deterding, S., Khaled, R., Nacke, L. E., Dixon, D., et al. (2011). Gamification: Toward a definition. In *CHI 2011 gamification workshop proceedings*, volume 12, pages 12–15. Vancouver BC, Canada.
- Elshani, L. and Nuçi, K. P. (2021). Constructing a personalized learning path using genetic algorithms approach. *CoRR*, abs/2104.11276.
- Fuchs, M. and Wolff, C. (2016). Improving programming education through gameful, formative feedback. *IEEE Global Engineering Education Conference, EDUCON*, 10-13-April(April):860–867.
- Guarino, N., Oberle, D., and Staab, S. (2009). What is an ontology? In *Handbook on ontologies*, pages 1–17. Springer.
- Hagedorn, C., Betz, E.-S., and Meinel, C. (2022). Storified programming moocs: A case study on learner engagement and perception. In *2022 IEEE Global Engineering Education Conference (EDUCON)*, pages 1144–1153. IEEE.
- Jusas, V., Barisas, D., and Jančiukas, M. (2022). Game elements towards more sustainable learning in object-oriented programming course. *Sustainability*, 14(4).
- Koulouri, T., Macredie, R. D., and Olakitan, D. (2021). Chatbots to support young adults' mental health: an exploratory study of acceptability: Acceptability of mental health chatbots for young adults. *ACM Transactions on Interactive Intelligent Systems*.
- Kučák, D. and Kučák, M. (2022). Gamification in computer programming education—systematic literature review. In *2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO)*, pages 517–520. IEEE.
- Lehtonen, T., Aho, T., Isohanni, E., and Mikkonen, T. (2015). On the role of gamification and localization in an open online learning environment: Javala experiences. In *Proceedings of the 15th Koli Calling Conference on Computing Education Research, Koli Calling '15*, page 50–59, New York, NY, USA. Association for Computing Machinery.
- Marangunić, N. and Granić, A. (2015). Technology acceptance model: A literature review from 1986 to 2013. *Univers. Access Inf. Soc.*, 14(1):81–95.
- Mekouar, L. (2022). The art of teaching programming languages: Challenges and accomplishments. In *2022 IEEE World Engineering Education Conference (EDUNINE)*, pages 1–6. IEEE.

- Miller, H., Haller, P., Rytz, L., and Odersky, M. (2014). Functional programming for all! scaling a mooc for students and professionals alike. In *Companion Proceedings of the 36th International Conference on Software Engineering, ICSE Companion 2014*, page 256–263, New York, NY, USA. Association for Computing Machinery.
- Munir, K. and Anjum, M. S. (2018). The use of ontologies for effective knowledge modelling and information retrieval. *Applied Computing and Informatics*, 14(2):116–126.
- Paiva, J. C., Queirós, R., Leal, J. P., Swacha, J., and Miernik, F. (2022). Managing gamified programming courses with the fgpe platform. *Information*, 13(2).
- Palaniappan, K. and Noor, N. M. (2022). Gamification strategy to support self-directed learning in an online learning environment. *International Journal of Emerging Technologies in Learning (iJET)*, 17(3):104–116.
- Rahman, M. M., Watanobe, Y., Matsumoto, T., Kiran, R. U., and Nakamura, K. (2022). Educational data mining to support programming learning using problem-solving data. *IEEE Access*, 10:26186–26202.
- Rodrigues, L., Pereira, F., Toda, A., Palomino, P., Oliveira, W., Pessoa, M., Carvalho, L., Oliveira, D., Oliveira, E., Cristea, A., et al. (2022). Are they learning or playing? moderator conditions of gamification’s success in programming classrooms. *ACM Transactions on Computing Education (TOCE)*.
- Rouhani, M., Lillebo, M., Farshchian, V., and Divitini, M. (2022). Learning to program: an in-service teachers’ perspective. In *2022 IEEE Global Engineering Education Conference (EDUCON)*, pages 123–132. IEEE.
- Satyanarayanan, M. (2001). Pervasive computing: vision and challenges. *IEEE Personal Communications*, 8(4):10–17.
- Soepriyanto, Y., Fazarini, P. F. A., Maulida, T. R., Nurfahrudianto, A., Afandi, Z., et al. (2022). The effect of player type in learning programming with online gamification activities on student learning outcomes. In *2022 2nd International Conference on Information Technology and Education (ICIT&E)*, pages 382–386. IEEE.
- Swacha, J., Paiva, J. C., Leal, J. P., Queirós, R., Montella, R., and Kosta, S. (2020). Gedil—gamified education interoperability language. *Information*, 11(6).
- Toda, A. M., Oliveira, W., Klock, A. C., Palomino, P. T., Pimenta, M., Gasparini, I., Shi, L., Bittencourt, I., Isotani, S., and Cristea, A. I. (2019). A taxonomy of game elements for gamification in educational contexts: Proposal and evaluation. In *2019 IEEE 19th International Conference on Advanced Learning Technologies (ICALT)*, volume 2161, pages 84–88. IEEE.