

# **JOGLOG – Jogos de Raciocínio Lógico para Alunos do Ensino Fundamental: Um Estudo de Caso Utilizando *Gamification* e Pensamento Computacional**

**Odair Moreira de Souza<sup>1</sup>**

<sup>1</sup>Instituto Federal do Paraná (IFPR)  
Quedas do Iguaçu, PR – Brasil

odair.desouza@ifpr.edu.br

**Abstract.** *Studies demonstrate the importance of teaching computational thinking and programming logic in early school training due to its relevance in the development of problem solving in any area of knowledge. Thus, the teaching-learning process should be more attractive and encouraging, given the complexity of the contents. The aim of this study is to demonstrate the advantages of association of gamification with computational thinking in an extension course that uses logical reasoning games. The course was accomplished with elementary school students and used the tools Plastelina, Lightbot, Code.org and Scratch. The results show the effectiveness of the use of gamification associated with computational thinking in the teaching of programming logic in relation to the smoothing of the learning curve.*

**Resumo.** *Estudos demonstram a importância do ensino do pensamento computacional e da lógica de programação na formação escolar o mais cedo possível devido a sua relevância no desenvolvimento de resoluções de problemas em qualquer área do conhecimento. Sendo assim, o processo de ensino-aprendizagem deve ser mais atrativo e incentivador, dada a complexidade dos conteúdos. O objetivo deste estudo é demonstrar as vantagens da associação da gamification com o pensamento computacional em um curso de extensão que utiliza jogos de raciocínio lógico. O curso foi realizado com alunos do ensino fundamental e utilizou as ferramentas Plastelina, Lightbot, Code.org e Scratch. Os resultados obtidos evidenciam a efetividade do uso da gamification associada com o pensamento computacional no ensino de lógica de programação em relação à suavização da curva de aprendizagem.*

## **1. Introdução**

Atualmente, existe uma ampla gama de pesquisas relacionadas com a definição e aplicação de métodos para tornar o processo de ensino-aprendizagem na área da computação mais interativo e atrativo para os estudantes. Uma das metodologias que se destaca é a *gamification* (ou ludificação) que vem sendo aplicada em sala de aula por meio de jogos.

De acordo com Kapp (2012) *gamification* é uma metodologia que utiliza jogos fora do seu contexto com o propósito de fornecer suporte na resolução de problemas e promover a aprendizagem. O potencial dos jogos no processo de ensino-aprendizagem se deve por sua propriedade de despertar a curiosidade e por promover o envolvimento

do jogador [Fernandes et al. 2018]. Gomes et al. (2015) afirmam que a aprendizagem decorrente do processo de *gamification* é potencializada quando o aluno desenvolve seu próprio jogo, nesta condição o engajamento, a motivação, a criatividade, o desenvolvimento cognitivo e da autonomia são favorecidos.

Segundo Cavalcante et al. (2016) o pensamento computacional é o conjunto de habilidades e atitudes relacionadas na realização de uma ação, onde o objetivo é a resolução de problemas, por meio de recursos, conceitos e ferramentas computacionais. Apesar de ser baseado em fundamentos da computação, não é restrito apenas a esta área, sendo relevante para todas as pessoas, pois é uma forma de trabalhar conceitos de diversos campos, podendo ser aplicado na resolução dos mais variados problemas. Nos próximos anos o pensamento computacional será uma competência fundamental, sendo um requisito para a formação básica dos profissionais de todas as áreas [Cavalcante et al. 2016].

Diversos profissionais e pesquisadores da área de educação e de tecnologia têm defendido o ensino de computação para crianças e adolescentes devido os benefícios do ensino do pensamento computacional. Segundo Wing (2006) o pensamento computacional deveria ser integrado no processo educacional das crianças o mais breve possível, visto que é tão importante quanto a leitura e a resolução de cálculos.

Neste contexto o principal objetivo desse trabalho é evidenciar as vantagens da *gamification* e a sua associação com a introdução do pensamento computacional no ensino fundamental, por meio do curso de extensão *JOGLOG* que trabalha com jogos de raciocínio lógico para computação.

Neste curso aplicou-se a metodologia *gamification* por meio das ferramentas: Plastelina, Lightbot, Code.org, Scratch e linguagem de programação C, em um curso de extensão para alunos do oitavo e nono ano do ensino fundamental dos colégios públicos e privados de Quedas do Iguaçu - PR. O curso de extensão *JOGLOG: Jogos de Raciocínio Lógico para Alunos do Ensino Fundamental* foi conduzido pelo professor orientador e contou com a participação dos alunos do curso técnico em informática integrado ao ensino médio do Instituto Federal do Paraná (IFPR) Campus Quedas do Iguaçu como monitores.

Para realizar a avaliação do aprendizado do pensamento computacional utilizou-se o *framework* de avaliação de competências computacionais definido por Brennan e Resnick (2012).

## 2. Trabalhos Relacionados

Vários estudos estão sendo realizados em relação a *gamification* e ao pensamento computacional visando o seu aprimoramento, visto que são assuntos atuais, importantes para a educação e que estão sendo utilizados no processo de ensino-aprendizagem de várias disciplinas, com alunos de qualquer faixa etária. Neste sentido apresenta-se uma relação de pesquisas que abordam tais temas.

Raposo e Dantas (2016) desenvolveram um jogo denominado "Desafio da Serpente" destinado aos alunos que cursam a disciplina de Introdução à Programação, do curso Bacharelado em Sistemas de Informação e Licenciatura em Ciências da Computação da UFPB. O objetivo principal do jogo é motivar os alunos de programação a criarem uma rotina de estudos e se sentirem motivados a resolver questões de programação, que são selecionadas pelo professor e disponibilizadas em um site web.

O jogo obteve aceitação de 96% dos alunos, que mostraram uma mudança de comportamento em relação a rotina de estudos por meio da grande quantidade de submissões diárias e a realização de atividades na plataforma em finais de semana. Além disso, a melhora na motivação foi notória.

Ramos e Teixeira (2015) realizaram um minicurso voltado para o desenvolvimento de jogos para alunos do ensino médio com a utilização do Scratch. Durante o curso foi abordado os conceitos do Pensamento Computacional por meio da produção de algoritmos e a criação de jogos. Os pesquisadores concluem que mesmo sem os conhecimentos prévios específicos em Ciências da Computação, foi possível produzir objetos digitais usando técnicas de programação.

Kologeski et al. (2016) realizaram um projeto interdisciplinar desenvolvido por estudantes e professores da Universidade Feevale, para desenvolver o raciocínio lógico dos alunos dos últimos anos do ensino fundamental. O projeto abordou desde os fundamentos de lógica de programação até o desenvolvimento de jogos. Foram usadas as plataformas *Code.Org*, o aplicativo *MIT App Inventor*, o *Scratch* e recursos desenvolvidos pelos acadêmicos (os jogos na rede social Teia: Navegática e Pantanal Escolar). Os resultados mostram que houve melhora no desenvolvimento do raciocínio lógico e motivação dos alunos, além disso, os alunos concluintes tem mostrado interesse na continuação dos estudos na área da informática.

Lockwood e Mooney (2018) identificaram através de uma revisão sistemática da literatura que o Scratch é a ferramenta mais usada para ensino de pensamento computacional na educação básica. Sobreira et al. (2018) comprovaram os benefícios pedagógicos e a capacidade de desenvolvimento do raciocínio lógico na solução de problemas do Scratch, após seu uso em uma oficina de jogos digitais.

### 3. Metodologia

O estudo de caso baseia-se no curso de extensão denominado *JogLog*: Jogos de Lógica para Computação, formatado com o objetivo de ensinar lógica de programação e raciocínio lógico para alunos do oitavo e nono ano do ensino fundamental de uma forma mais atraente facilitando o aprendizado, tendo em vista que os alunos iniciantes na área de tecnologia possuem dificuldades para assimilar os conceitos de lógica de programação e acabam desmotivados devido a complexidade.

O curso foi desenvolvido durante 20 encontros, sendo um por semana, totalizando 60 horas, realizados no campus do IFPR – Quedas do Iguaçu no contra turno escolar. Com o intuito de engajar os estudantes do técnico em informática no curso de extensão *JogLog*, elaborou-se um plano de trabalho para monitores voluntários.

Na Figura 1, apresenta-se as principais etapas do curso de extensão *JogLog*. Observa-se que na primeira etapa aconteceu o **levantamento bibliográfico** sobre pensamento computacional, *gamification*, ferramentas digitais para o ensino de lógica de programação e estudos referentes aos métodos de avaliação do processo de ensino-aprendizagem para o contexto do pensamento computacional.

Em seguida realizou-se o **treinamento dos monitores** com o objetivo de realizar a capacitação dos estudantes do curso técnico em informática integrado ao ensino médio, para atuarem e conduzirem as atividades do curso sob a orientação e o acompa-



Figura 1. Visão geral do curso de extensão *JogLog*.

nhamento do professor orientador. Nesta formação trabalhou-se a elaboração de problemas e a resolução dos desafios de lógica com uma abordagem de ensino-aprendizagem aplicada por meio de plataformas digitais de ensino de raciocínio lógico e programação, tais como, Plastelina, LightBot, Code.org e Scratch. Estas ferramentas digitais foram selecionadas na etapa de levantamento bibliográfico, são todas livres e *online* e amplamente utilizadas no ensino básico de ciência da computação e programação de computadores. Além disso, o treinamento abordou questões e princípios de didática, para os monitores terem a capacidade de aplicar as atividades com o máximo de clareza e objetividade.

Em cada encontro do curso de extensão *JogLog*, os monitores previamente escalados, assumiam as apresentações das atividades logo após a introdução da aula realizada pelo professor. Estabeleceu-se que após cada participação na aula, o monitor deveria descrever o seu relato de experiência no diário de bordo, descrevendo as atividades propostas e conduzida por ele, citando quais foram as dificuldades, os pontos positivos e os pontos negativos identificados.

Na Figura 2, apresenta-se a descrição da aplicação das principais fases do **desenvolvimento do curso** de extensão *JogLog* utilizando *gamification* e o pensamento computacional:

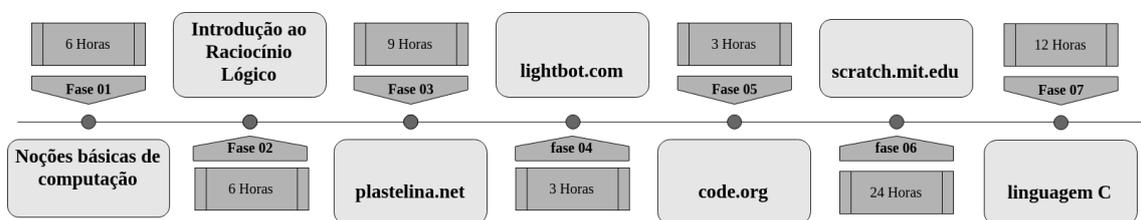


Figura 2. Fases do Curso de Extensão *JogLog*.

**Noções Básicas de Computação:** Neste módulo utilizou-se como base para algumas atividades o livro *Computer Science Unplugged*, de autoria de Tim Bell (2011), o qual traz atividades e dinâmicas para serem desenvolvidas em equipe, para fixação dos conceitos básicos de computação, tais como, numeração binária, representação de informação e transmissão de dados.

**Introdução ao Raciocínio Lógico:** No desenvolvimento dessa atividade abordou-

se a criação de algoritmos do cotidiano, tais como, algoritmo para fazer um miojo, para trocar um pneu, para trocar uma lâmpada, para resolver o desafio do lobo, ovelha e repolho, entre outros. A solução era realizada de maneira manuscrita e revisada pelo professor antes de avançar para a próxima atividade.

**Plastelina.net:** Nesse conjunto de atividades realizou-se os desafios disponibilizados no site, onde cada jogo possui um conjunto de regras e dinâmicas diferentes que exercitam o raciocínio lógico envolvendo temáticas conhecidas, tais como: (i) Travessia de um rio com um lobo, uma ovelha e um repolho; (ii) Travessia de um rio com padres e canibal; (iii) Disposição de 8 rainhas em um tabuleiro de xadrez; (iv) Percorrer todas as casas de um tabuleiro de xadrez considerando o caminhar da peça cavalo; e, (v) Obter um volume específico com recipientes de diferentes capacidades.

**lightbot.com:** Com essa ferramenta foram abordados diferentes níveis de complexidade relacionados à programação de computadores, tais como, estruturas sequenciais, condicionais e repetição. Ela é baseada na movimentação de um pequeno robô, onde cada nível possui um conjunto de atividades nas quais o usuário é informado de seu objetivo e das operações de movimentação disponíveis. Além disso, a ferramenta estimula a utilização de funções/métodos e inclusive recursividade.

**code.org:** Essa plataforma foi utilizada como objetivo de introduzir a construção de algoritmos por meio de blocos de comandos. Utilizou-se o desafio *Star Wars: Construindo uma Galáxia com Código*<sup>1</sup>. A resolução desse desafio consiste em um algoritmo sequencial utilizando blocos de movimentação, adição de objetos, colisão de objetos e reprodução de sons.

**scratch.mit.edu:** O Scratch é uma plataforma digital de ensino de algoritmos criada no *Massachusetts Institute of Technology (MIT)* em 2009 e liderada pelo Prof. Dr. Mitchel Resnick. Utilizou-se essa plataforma com o propósito de propiciar suporte ao ensino dos principais conceitos de programação por meio da representação de comandos em blocos lógicos, itens de imagem e som para o desenvolvimento de histórias interativas, animações e a criação de jogos, além de possibilidade de compartilhar as criações online [Resnick et al. 2009].

Nesta fase os monitores e o professor realizavam as exposições de no máximo 15 minutos e em seguida os alunos tinham um período, dentro da própria aula, para desenvolverem um conjunto de atividades de fixação dos conteúdos. Após a conclusão dos conteúdos fundamentais estabeleceu-se uma atividade de criação livre, trata-se de um projeto que poderia ser a criação de um jogo, uma animação ou uma história interativa, essa atividade foi realizada em 4 encontros. Além de desenvolverem o projeto, os alunos tiveram que realizar uma apresentação de aproximadamente 10 minutos para a turma e compartilhar a sua criação na plataforma scratch para os demais alunos. Com base nesta atividade realizou-se avaliação do pensamento computacional e da evolução do alunos nas fases da abordagem *gamification* utilizada.

**Linguagem de programação C:** Nesta fase do curso realizou-se a introdução dos conteúdos de linguagem de programação, tais como, (i) declaração de variáveis e constantes; (ii) comandos de entrada e saída; (iii) operadores aritméticos, relacionais e lógicos; (iv) comandos condicionais; e, (v) estruturas de repetição. Aplicou-se um con-

<sup>1</sup>Disponível em <https://code.org/starwars>

junto de exemplos práticos de programação e durante as aulas desenvolveu-se exercícios para fixação dos conteúdos. Ao final, os exercícios eram enviados para revisão do professor/orientador.

Para a **avaliação do curso** de extensão *JogLog* utilizou-se o *framework* definido por Brennan e Resnick (2012) para estudar e avaliar o desenvolvimento do pensamento computacional. Esse *framework* foi estabelecido com base no desenvolvimento longitudinal dos criadores no scratch, na participação de oficinas e portfólios de projetos durante anos, para compreender as práticas de criador-em-ação envolvendo três dimensões de competências, descritas a seguir.

- **Conceitos computacionais:** são os conceitos que os projetistas empregam ao programar, Brennan e Resnick (2012) identificaram sete conceitos que são amplamente úteis em projetos de Scratch e podem ser aplicados no contexto de programação.
  - Sequência: é uma determinada série de instruções a serem executadas pelo computador que especifica o comportamento ou ação a ser produzida de maneira sequencial;
  - Repetição: é a execução repetida de uma sequência de instruções de maneira controlada;
  - Eventos: são os componentes de interação que executam quando uma determinada ação específica acontece, acionando um conjunto de instruções;
  - Paralelismo: são sequências de instruções que são executadas ao mesmo tempo, otimizando a realização de uma determinada atividade;
  - Condicionais: são os comandos capazes de tomar decisões e desviar o fluxo de processamento com base em condições que podem ser expressas em múltiplos resultados;
  - Operadores: os operadores fornecem suporte para as expressões aritméticas, relacionais, lógicas e de *string*<sup>2</sup>; e,
  - Dados: envolvem o armazenamento, recuperação e atualização de valores das variáveis (um único número ou sequência) ou lista (uma coleção de números ou string).
- **Práticas computacionais:** são as práticas que os projetistas desenvolvem enquanto programam, estão concentradas nos processos de pensar e de aprender.
  - Iterativo e incremental: é a elaboração de um plano de *design* e implementação do código, descrevendo ciclos iterativos de imaginação e construção evolutiva, com base em suas experiências e novas ideias;
  - Teste e depuração: são as estratégias e práticas para encontrar os erros e defeitos (ou prever problemas) na implementação;
  - Reúso e remixagem: estão relacionados a capacidade crítica de leitura de código, trabalho cooperativo e colaborativo, envolve a reutilização de códigos desenvolvidos anteriormente ou ajustes e adaptações de códigos de terceiros; e,
  - Abstração e modularidade: são caracterizadas com a capacidade de construir ou resolver um problema, por meio de uma coleção de partes de códigos realizando a integração entre eles. Basicamente, consiste em dividir em partes consistentes para auxiliar em uma solução.

---

<sup>2</sup> **String** são cadeias de caracteres.

- **Perspectivas computacionais:** São as perspectivas que os *designers* formam sobre o mundo ao seu redor e sobre si mesmos durante a programação.
  - Autoexpressão: está ligada com a capacidade de aprender a se expressar, apresentar as suas ideias e projetos;
  - Conectividade: é a capacidade de conectar pessoas por meio de interações face a face, de redes online, ter acesso a novas perspectivas e projetos; e,
  - Questionamento: envolve a capacidade de interrogar um determinado dado como certo, é a propensão de fazer perguntas sobre tecnologia e como a tecnologia pode atuar em diferentes realidades do mundo.

#### 4. Resultados

Obteve-se 34 inscritos, foram pré-selecionados 31 e 27 confirmaram a participação no curso de extensão *JogLog*. Destaca-se que 89% dos alunos são de colégios estaduais urbanos do município e 7% de colégios particulares, por outro lado, apenas 4% são de colégios estaduais da zona rural.

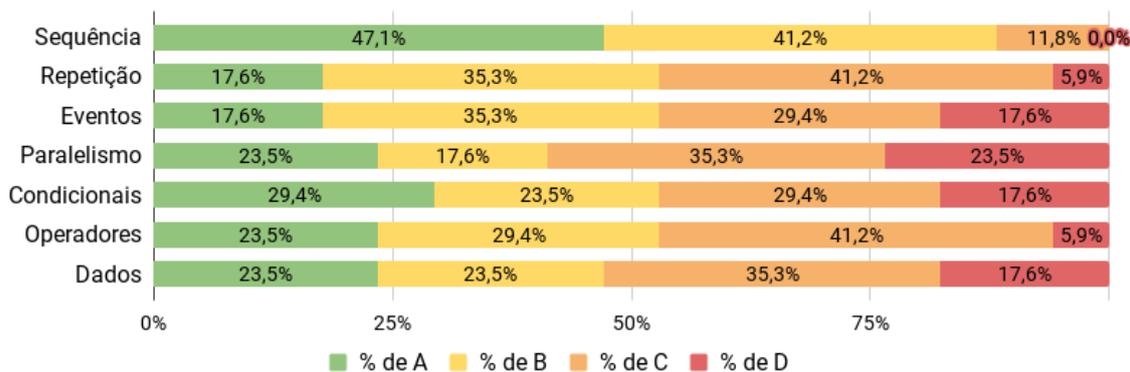
Esse processo de análise verificou que 93% dos alunos são de escolas públicas e 76% estão no 9º ano do ensino fundamental. A caracterização da pesquisa também revelou que 85,7% dos alunos não tiveram uma disciplina exclusivamente de informática no ensino fundamental e que apenas 23,8% tiveram os conteúdos de raciocínio lógico em outras disciplinas.

A avaliação do pensamento computacional dos alunos do curso de extensão *JogLog* foi realizada de acordo com o *framework* definido por Brennan e Resnick (2012) que mostra de maneira clara e objetiva os parâmetros a serem avaliados no desempenho dos alunos. Esse *framework* foi desenvolvido para avaliação das competências referentes ao Scratch, portanto, a avaliação aconteceu antes do início das atividades na Linguagem de Programação em C, avaliando os jogos, as animações e as histórias interativas que os alunos desenvolveram.

A avaliação se deu por meio de conceitos, seguindo as normas de avaliação dos processos de ensino-aprendizagem no âmbito do IFPR, onde:

- *Conceito A* – Quando a aprendizagem do aluno foi PLENA e atingiu os objetivos propostos no processo ensino-aprendizagem;
- *Conceito B* – A aprendizagem do aluno foi PARCIALMENTE PLENA e atingiu níveis desejáveis aos objetivos propostos no processo ensino-aprendizagem;
- *Conceito C* – A aprendizagem do aluno foi SUFICIENTE e atingiu níveis aceitáveis aos objetivos propostos; e,
- *Conceito D* - A aprendizagem do aluno foi INSUFICIENTE e não atingiu os objetivos propostos.

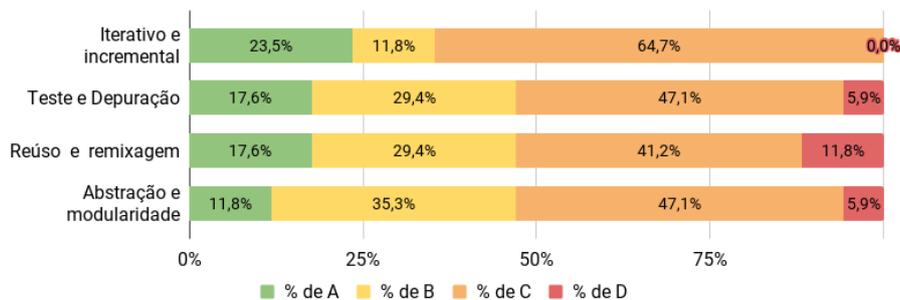
Na Figura 3, apresenta-se os sete conceitos fundamentais para o pensamento computacional e nesta dimensão, observa-se que a maioria dos alunos obtiveram sucesso no processo de aprendizagem. Examinou-se no desenvolvimento dos projetos que os conceitos sequência, repetição, eventos, condicionais, operadores e dados (declaração de variáveis) foram aplicados corretamente na maioria dos casos. Nota-se um aprendizado de 100% dos alunos em comandos sequências. Devido a importância do conceito de repetição para os demais conceitos, dedicou-se mais atenção e isso fica visível pois 94,1% dos alunos atingiram no mínimo conceito C.



**Figura 3. Resultado da avaliação da dimensão: Conceitos computacionais.**

O conceito que os alunos mais tiveram dificuldade foi o paralelismo, onde 23,5% dos alunos obtiveram aprendizado insuficiente, mesmo que aplicado corretamente em alguns casos, ficou notória a dificuldade dos alunos em compreender completamente esse conceito, pois verificou-se que quando eles tinham várias de ações a serem executadas ao mesmo tempo isso causava confusão e falta de domínio.

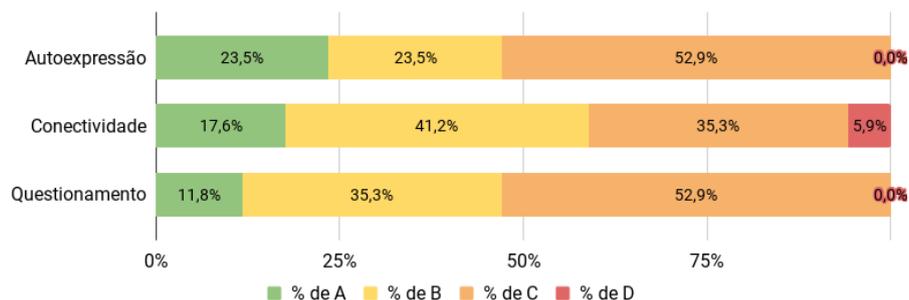
Na Figura 4, apresenta-se as práticas computacionais avaliadas no curso de extensão. Percebe-se que a prática de iteração e incremento obteve 100% de compreensão dos alunos, isso se reflete no nível de qualidade dos jogos, animações e histórias desenvolvidas, com boas apresentações de *game design* e ciclos iterativos desde jogos com fases, personagens e pontuação, destacando a implementação completa das práticas.



**Figura 4. Resultado da avaliação da dimensão: Práticas computacionais.**

Dentre as práticas computacionais dessa dimensão destaca-se a abstração e modularidade que obteve 94,1% dos alunos com conceitos de aprovação, pois nos treinamentos precedentes ao desenvolvimento da atividade de criação livre, dedicou-se uma atenção para a modularidade, isso devido a criação dos seus próprios blocos no scratch, focando na definição de responsabilidade única e nos parâmetros.

Na Figura 5, demonstra-se as perspectivas do pensamento computacional, as quais tratam da autoexpressão que é a capacidade de se expressar e apresentar as suas ideias, isso foi visível nas apresentação das criações dos alunos, portanto 100% dos alunos conseguiram transmitir as suas ideias por mais que não estivessem implementadas completamente.



**Figura 5. Resultado da avaliação da dimensão: Perspectivas computacionais.**

Outra perspectiva do *framework* é referente a conectividade dos alunos, que pode ser face a face ou online, na avaliação dos projetos examinou-se a capacidade dos alunos em disseminar o seu projeto na turma e também de maneira online, e constatou-se que apenas 5.9% dos alunos tiveram dificuldades, isso particularmente na interação online. Por fim, nota-se que nenhum aluno obteve problemas de compreensão referente ao nível de questionamento.

## 5. Considerações Finais

Neste estudo considerou-se a importância do pensamento computacional associada com a abordagem *gamification* no processo de ensino-aprendizagem na área da computação, o estudo de caso foi aplicado no curso de extensão *JogLog* – Jogos de Raciocínio Lógico para Alunos do Ensino Fundamental associando estas duas metodologias e um *framework* de avaliação das competências computacionais.

No decorrer do curso verificou-se que os alunos, sem nenhum contato prévio com a programação de computadores, conseguiram estabelecer um aprendizado de lógica de programação por meio das competências do pensamento computacional, sem vínculo com linguagem de programação, que somente foi introduzida na fase 7 (linguagem de programação C) e o rendimento dos estudantes foi avaliado como bom em definir soluções computacionais.

Portanto, a principal constatação desse estudo de caso é de que o processo de ensino-aprendizagem de programação deve ser aplicado em duas etapas, a primeira é ensinar raciocínio lógico e lógica de programação e somente depois das competências computacionais terem sido compreendidas completamente, deve-se iniciar o ensino de linguagem de programação, pois essa abordagem suaviza a curva de aprendizagem em computação.

Por fim, a investigação apresentada neste estudo de caso comprova a eficácia do uso dos jogos como ferramenta de ensino, e que o uso de jogos de maneira introdutória ao ensino de programação aumenta o engajamento e a motivação no aprendizado.

## Referências

Bell, T., Witten, I., e Fellows, M. (2011). *Computer science unplugged – ensinando ciência da computação sem o uso do computador*. Tradução de Luciano Porto Barreto, 2011. Disponível em: <http://csunplugged.org/>. Acessado em junho de 2019.

- Brennan, K. e Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American Educational Research Association*, pages 1–25, Vancouver, Canada. American Educational Research Association.
- Cavalcante, A. F., Costa, L. S., e Araújo, A. L. S. O. (2016). Um estudo de caso sobre competências do pensamento computacional estimuladas na programação em blocos no code.org. In *Anais dos Workshops do V Congresso Brasileiro de Informática na Educação*, pages 1117–1126, Uberlândia, MG, Brasil.
- Fernandes, K. T., da Silva Aranha, E. H., e Lucena, M. J. N. R. (2018). Estratégias para elaboração de game design de jogos digitais educativos: uma revisão sistemática. In *Anais do XXIX Simpósio Brasileiro de Informática na Educação (SBIE 2018)*, pages 585–594, Fortaleza, CE, Brasil.
- Gomes, D. N., Araújo, L., Machado, T. L. A., Filho, A. J. C. B., e de Souza, S. (2015). Ensino aprendizagem através do desenvolvimento de jogos. In *Proceedings of the XIV SBGames 2015*, pages 767–775, Teresina, PI, Brasil.
- Kapp, K. M. (2012). *The Gamification of Learning and Instruction: Game-Based Methods and Strategies for Training and Education*. Pfeiffer, San Francisco, CA , USA.
- Kologeski, A. L., Silva, C. G., Barbosa, D. N. F., Mattos, R. R., e Miorelli, S. T. (2016). Desenvolvendo o raciocínio lógico e o pensamento computacional: Experiências no contexto do projeto logicando. *RENOTE: Revista Novas Tecnologias da Educação*, 14, pg. 1–10.
- Lockwood, J. e Mooney, A. (2018). Computational thinking in secondary education: Where does it fit? a systematic literary review. *International Journal of Computer Science Education in Schools*, 02, pg. 1–10.
- Ramos, F. O. e Teixeira, L. d. S. (2015). Significação da aprendizagem através do pensamento computacional no ensino médio: Uma experiência com scratch. In *Anais do XXI Workshop de Informática na Escola (WIE)*, pages 217–226, Brasília, DF, Brasil.
- Raposo, E. H. S. e Dantas, V. (2016). O desafio da serpente - usando gamification para motivar alunos em uma disciplina introdutória de programação. In *Anais do XXVII Simpósio Brasileiro de Informática na Educação (SBIE 2016)*, pages 577–586, Uberlândia, MG, Brasil.
- Resnick, M., Maloney, J., Monroy-Hernandez, A., Rusk, N., Eastman, E., Brennan, K., e Kafai, Y. (2009). Scratch: Programming for all. *Communications of the ACM*, 52(5), pg. 31–33.
- Sobreira, E. S. R., Takinami, O. K., e Santos, V. G. (2018). Programando, criando e inovando com o scratch: em busca da formação do cidadão do século xxi. In *Anais da VII Jornada de Atualização em Informática na Educação (JAIE 2018)*, pages 126–152, Fortaleza, CE, Brasil.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), pg. 33–35.