

Recursos e estratégias para desenvolvimento e avaliação do Pensamento Computacional na escola

Carla Lopes Rodriguez (UFABC)¹

Rachel Carlos Duque Reis (UFV)²

Seiji Isotani (ICMC/USP)³

Resumo

Pesquisas na área de Informática na Educação ressaltam a importância de promover na escola o desenvolvimento de competências associadas aos princípios do pensamento computacional. Nesse sentido, este artigo tem como objetivo apresentar os resultados de um projeto de Pré-Iniciação Científica, desenvolvido com alunos do Ensino Fundamental, Médio e Médio-Técnico, para a aquisição, aprimoramento e avaliação do pensamento computacional, por meio de desafios de lógica usando a linguagem de programação *Scratch*. Este estudo insere-se no âmbito de uma pesquisa mista, utilizando uma abordagem interpretativa para a análise dos dados. Os resultados foram analisados e discutidos com base nas dimensões do pensamento computacional: conceitos, práticas e perspectivas, confirmando que as estratégias de avaliação adotadas contribuíram para evidenciar o grau evolutivo de desempenho dos alunos.

Palavras-chave: Avaliação; Desafios de lógica; Pensamento computacional; Dr. Scratch.

Abstract

Research on computers in education emphasizes the importance of helping k-12 students to develop new skills associated with the principles of computational thinking. In this context, this paper aims at presenting the development and evaluation of computational thinking activities, through logic challenges using Scratch (a visual programming language). The activities were carried out with students from Elementary School, High School and Technical Education. We conduct a mixed research study with emphasis on qualitative data using an interpretative approach to analyze collected information. The obtained results are discussed based on three dimensions of computational thinking: concepts, practice and perspectives. Our results provide some evidences that the developed activities and evaluation strategies contribute to improve students' performance.

Keywords: Evaluation; Challenges of logic; Computational thinking; Dr. Scratch.

¹ Contato: carlalrodriguez@gmail.com

² Contato: rachel.reis@ufv.br

³ Contato: sisotani@icmc.usp.br

1. Introdução

A sociedade atual requer dos indivíduos o aprimoramento de habilidades que vão além do conhecimento instrumental para leitura, escrita e realização de cálculos básicos de matemática. Com os avanços tecnológicos cada vez mais inseridos na escola, observa-se a necessidade do desenvolvimento de competências que permitam aos alunos aprender a usar as tecnologias disponíveis de forma criativa e inovadora para melhor conceituar, analisar e resolver problemas (Blikstein, 2008). Assim, surgem habilidades relacionadas ao que se denomina “pensamento computacional”, conceito que tem sido investigado por diversos pesquisadores da comunidade científica nacional e internacional (Papert, 1993; Wing, 2006; Blikstein, 2008; Brennan; Resnick, 2012; Ramos; Espadeiro, 2014; França; Tedesco, 2015; Rodriguez et al., 2015).

O pensamento computacional é caracterizado como uma abordagem para resolução de problemas, que utiliza os conceitos da área da Ciência da Computação (e.g., abstração, lógica, representação dos dados) para processar e analisar os dados e, a partir disso, criar novos artefatos e conhecimentos (Wing 2010, 2014), podendo ser aplicada a qualquer área do conhecimento (e.g., Física, Química, Biologia e História). Do ponto de vista pedagógico, Brennan e Resnick (2012) descrevem os princípios do pensamento computacional em três dimensões: *conceitos*, *práticas* e *perspectivas* computacionais. Os *conceitos* estão relacionados ao que os alunos mobilizam e aprendem enquanto programam utilizando uma linguagem de programação (e.g., sequência, laços, paralelismo). As *práticas* são desenvolvidas pelos alunos durante as atividades de desenvolvimento e implementação (e.g., abstração e modulação, teste e depuração) de um projeto. Finalmente, as *perspectivas* dizem respeito à compreensão dos alunos sobre o mundo que o cerca e sobre eles mesmos (e.g., capacidade de se expressar, conectar-se aos colegas, questionar).

Um dos desafios desse campo de pesquisa consiste na seleção e/ou definição de abordagens (e.g. ferramentas, práticas, estratégias, métodos) que sejam adequadas à avaliação da aplicação dos princípios do pensamento computacional em situações educacionais (Ramos; Espadeiro, 2014). Nesse contexto, Ramos e Espadeiro (2015) apresentam, por meio de uma revisão sistemática da literatura, as principais *estratégias* e *práticas* utilizadas para avaliação do pensamento computacional em escolas do ensino básico e secundário de Portugal. Com relação às *estratégias*, os resultados da revisão sistemática mostraram a *avaliação formativa* como a mais adotada nas escolas. Em menor número, foram identificadas estratégias de *auto avaliação*, *avaliação por pares*, e *trabalho em grupo*. No caso das *práticas* de avaliação, a técnica mais utilizada foi *avaliação dos artefatos e projetos finais*, seguido dos *roteiros de observação*, *questionários*, *roteiros de*

avaliação e *entrevistas*. No que diz respeito ao uso de ferramentas, Moreno-León, Robles e Chusig (2014) apresentam a ferramenta *Dr. Scratch* (<http://drscratch.programamos.es>) para análise e avaliação automática de projetos criados com o *Scratch* (www.scratch.mit.edu/). A ferramenta, além de analisar os blocos de comandos e localizar possíveis erros de código, identifica as competências demonstradas pelo desenvolvedor com base em sete princípios do pensamento computacional: abstração, paralelismo, pensamento lógico, sincronização, controle de fluxo, interatividade do usuário e representação dos dados.

Com base no exposto anteriormente, este trabalho tem como objetivo contribuir para a ampliação das reflexões sobre as abordagens usadas para o desenvolvimento e avaliação da aquisição dos princípios do pensamento computacional em contextos educacionais. Para isso apresentamos o desenvolvimento e os resultados alcançados com um projeto de Pré-Iniciação Científica (Pré-IC), que envolveu um grupo de alunos do Ensino Fundamental, Médio e Médio-Técnico, para investigar como as dimensões (conceitos, práticas e perspectivas) do pensamento computacional, apresentadas por Brennan e Resnick (2012), podem ser trabalhadas por meio de atividades que envolvem a resolução de desafios de lógica, usando a linguagem de programação *Scratch*. Especificamente, pretendeu-se analisar a contribuição das estratégias e instrumentos (questionários, roteiros de observação e ferramenta de avaliação automática *Dr. Scratch*) utilizados na avaliação da aplicação dos princípios do pensamento computacional nos jogos digitais produzidos pelos alunos.

Considerando os objetivos definidos, e o contexto onde está inserido o estudo proposto, foram definidas três questões de pesquisa:

1. Resolver desafios de lógica com o apoio da programação de jogos interativos, no caso usando o *Scratch*, pode apoiar o desenvolvimento/ampliação dos princípios do pensamento computacional?
2. Quais conceitos, práticas e perspectivas computacionais foram desenvolvidos/aprimorados pelos alunos por meio da realização de atividades de programação de jogos digitais interativos usando o *Scratch*?
3. Quais estratégias podem ser efetivas para avaliar a aplicação dos princípios do pensamento computacional pelos alunos quando realizam atividades de programação de jogos digitais interativos usando a linguagem de programação *Scratch*?

2. Pensamento Computacional no contexto educacional

A necessidade de mudança de paradigma no processo de ensino e de aprendizagem das escolas de ensino básico é discutida por Ramos e Espadeiro (2014) que apresentam os anseios da sociedade pela formação de indivíduos capazes de se adaptar às rápidas mudanças tecnológicas, e que saibam entender e utilizar as tecnologias de forma crítica

para a construção de novos conhecimentos nas mais diversas profissões e áreas de conhecimento. Essa preocupação é reforçada pelo professor e pesquisador Paulo Blinkstein (Goulart, 2011), que ressalta a importância de requalificar o ensino básico em todos os países, com base no desenvolvimento de novas competências relacionadas, por exemplo, aos princípios do pensamento computacional (Blinkstein, 2008), de forma a despertar nos indivíduos a paixão pelas ciências e estimular a criação de gerações de inovadores.

Com relação ao cenário educacional brasileiro, embora as discussões sobre a inserção dos conceitos introdutórios de computação no currículo do ensino básico sejam ainda incipientes, observa-se o aumento crescente de projetos e iniciativas com a proposta de *desenvolvimento* e *avaliação* do pensamento computacional nas escolas. Nesse sentido, às subseções a seguir são destinadas ao detalhamento de cada uma dessas propostas.

2.1. Desenvolvimento do Pensamento Computacional

O projeto “Reinventando o Ensino Médio” (Carvalho; Chaimowicz; Moro, 2013) tem como proposta o desenvolvimento e disseminação do pensamento computacional em escolas da rede pública de Minas Gerais. Esse projeto foi criado pela Secretaria de Educação do Estado e implantado com o auxílio de docentes dos cursos de Computação da Universidade Federal de Minas Gerais (UFMG) em onze escolas estaduais de Belo Horizonte (em 2012) e em outras 122 escolas do estado (em 2013), atingindo mais de 65.000 alunos da rede pública. Outro projeto em desenvolvimento é o “EXP-PC - Explorando o Pensamento Computacional para a Qualificação do Ensino Fundamental” (UFPel, 2014), formado por professores e alunos (bolsistas e voluntários) da Universidade Federal de Pelotas, para implementação de ações educacionais em escolas públicas da rede municipal da cidade de Pelotas-RS. Algumas dessas ações foram publicadas no trabalho de Santos et al. (2015) que propõe a realização de atividades para o desenvolvimento do pensamento computacional no ensino público fundamental sem o uso do computador, conforme proposto por Bell, Witten e Fellows (2011).

Outro método de disseminação do pensamento computacional utilizado por iniciativas nacionais é o desenvolvimento de projetos de jogos digitais usando a linguagem de programação *Scratch*. O *Scratch* foi desenvolvido pelo grupo de pesquisa *Lifelong Kindergarten* do *MIT Media Lab* para a construção de projetos interativos utilizando blocos de comandos. De acordo com Resnick et al. (2009), a construção de projetos com a linguagem *Scratch* favorece o desenvolvimento do pensamento computacional ao possibilitar: (i) o aprendizado de conceitos matemáticos e de computação; (ii) o desenvolvimento do pensamento sistemático; (iii) a criatividade para solucionar problemas; e (iv) o trabalho colaborativo. Dentre as iniciativas criadas, o trabalho de França e Amaral

(2013) apresenta uma oficina utilizando a linguagem *Scratch* para ensinar os fundamentos de computação com base nos conceitos e práticas computacionais de Brennan e Resnick (2012). Os resultados mostraram-se satisfatórios quanto ao uso do *Scratch* para o desenvolvimento do pensamento computacional de estudantes na faixa de idade entre 13 e 14 anos. Utilizando o *Scratch* em uma abordagem interdisciplinar com alunos do 1º ano do ensino fundamental, Wangenheim, Nunes e Santos (2014) desenvolveram uma unidade instrucional (conjunto de exercícios e atividades) para ensinar os conceitos de computação de forma integrada ao conteúdo das disciplinas de Língua Portuguesa e Artes. O estudo apresentou indícios de que é possível usar o *Scratch* com alunos nesse nível de escolaridade. Outra experiência positiva envolvendo alunos do Ensino Médio foi desenvolvida por Scaico et al. (2013). Os autores utilizaram a linguagem *Scratch* em conjunto com outros métodos de disseminação do pensamento computacional (atividades sem o uso do computador) para introdução dos conceitos de computação. Os resultados mostraram que apesar das deficiências de escrita, leitura e fundamentos lógico-matemáticos, os estudantes foram capazes de assimilar os conceitos básicos de programação.

2.2. Avaliação do desenvolvimento do Pensamento Computacional

Segundo Ramos e Espadeiro (2014) ainda são poucos os trabalhos na literatura que apresentam propostas de avaliação da aprendizagem com vistas ao desenvolvimento do pensamento computacional. Os autores apontam duas tendências de avaliação da aprendizagem: pedagógica e tecnológica. A primeira tendência combina métodos e técnicas pedagógicas para avaliação do desenvolvimento do pensamento computacional. Nesse contexto, o trabalho de França e Amaral (2013) utiliza como estratégia de avaliação, a avaliação formativa, ou seja, a verificação da aplicação dos conceitos “sequência”, “evento”, “paralelismo”, dentre outros, no processo de construção dos quatro projetos desenvolvidos pelos alunos usando a linguagem *Scratch*. No estudo desenvolvido por Wangenheim, Nunes e Santos (2014) as práticas pedagógicas de avaliação foram realizadas com base na *observação* dos instrutores e professores e no resultado dos *projetos* produzidos pelos alunos usando o *Scratch*. No intuito de verificar se os objetivos foram atingidos foi levado em consideração se os alunos: (i) utilizaram os recursos adequados do ambiente para desenvolvimento dos projetos em *Scratch* para ilustrar pensamentos e ideias (ferramentas de edição, câmeras digitais e ferramentas de desenho); (ii) demonstraram competências para trabalhar de forma colaborativa; (iii) conseguiram criar histórias interativas por meio de sequências de instruções; dentre outros. Utilizando uma abordagem de avaliação diferente, Scaico et al. (2013) desenvolveram uma Olimpíada Interna de Programação com *Scratch*,

simulando um cenário de maratona de programação, com o propósito de avaliar as habilidades desenvolvidas pelos alunos. O desafio foi composto por cinco questões, com níveis de dificuldade diferentes, que foram avaliados levando em consideração fatores como a criatividade nas soluções, o uso adequado dos comandos da linguagem *Scratch*, a identificação e resolução do erro inserido nos desafios, dentre outros.

A segunda tendência de avaliação da aprendizagem, definida por Ramos e Espadeiro (2014) como “tecnológica”, consiste no desenvolvimento de aplicações computacionais para automatizar o processo de avaliação. Nesse contexto, Moreno-León, Robles e Chusig (2014) apresentam a ferramenta web *Dr. Scratch* (gratuita e de código aberto) para análise de projetos desenvolvidos utilizando a linguagem de programação *Scratch*. Essa ferramenta oferece ao desenvolvedor um retorno (*feedback*) sobre a eficiência dos códigos e comandos utilizados nos projetos com base em sete princípios (abstração e decomposição do problema, paralelismo, pensamento lógico, sincronização, controle de fluxo, interatividade do usuário e representação dos dados). O *feedback* oferecido pode ser usado para melhorar as habilidades de programação e o desenvolvimento do pensamento computacional. Além disso, essa ferramenta é capaz de identificar práticas insatisfatórias de programação e/ou erros de código (e.g., código repetido e/ou que nunca será usado, inicialização incorreta de objetos).

Considerando que não foram encontrados estudos científicos nacionais que utilizem soluções computacionais para a avaliação automática da dimensão tecnológica do desenvolvimento do pensamento computacional, nesse trabalho é proposto o uso da ferramenta *Dr. Scratch*, combinada a outras estratégias de avaliação da aprendizagem (e.g., questionários, roteiros de observação, relatórios), para avaliar a aplicação dos princípios do pensamento computacional nos jogos digitais produzidos pelos alunos participantes.

3. Metodologia

O estudo aqui apresentado foi realizado no âmbito do programa de Pré-IC do Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (ICMC-USP), que ocorreu entre 2014 e 2016. Neste contexto, a proposta de trabalho, sob o tema “*Computação e Criatividade: criando jogos digitais usando o Scratch*”, foi desenvolvida no final do ano letivo de 2015, sendo os dados coletados de Outubro à Dezembro. Foram nove encontros presenciais, com duração média de quatro horas cada, para o desenvolvimento de atividades que estimulassem os alunos participantes a trocar ideias para resolver desafios de lógica (individualmente e em dupla) e implementá-los computacionalmente com os recursos da linguagem de programação *Scratch*. A ideia era introduzir conceitos

relacionados aos princípios do pensamento computacional e estimular o raciocínio lógico para a resolução de problemas por meio da programação de jogos digitais. Nesse sentido, os alunos foram apresentados ao Scratch e criaram jogos digitais interativos de forma livre e autônoma, usando os blocos de comandos para compor a lógica que cada desafio proposto exigia.

Os alunos, provenientes de duas escolas públicas do município, se reuniam com os monitores (alunos de pós-graduação) no laboratório de informática da Universidade uma vez por semana. Além das atividades de ambientação, apresentação e exploração dos recursos do *Scratch*, foram desenvolvidas outras três atividades, que envolveram a resolução de desafios de lógica, usando a linguagem de programação Scratch: i) Desafio 1 - Jogo de perguntas e respostas; ii) Desafio 2 - Caixas de frutas e; iii) Desafio 3 - A travessia do rio. A Figura 1 exemplifica a apresentação do Desafio 3 (D3).

Figura 1 - Apresentação do Desafio 3 - A travessia do rio



Implemente usando o Scratch - A travessia do rio

Um homem precisa atravessar um rio em uma canoa. Com ele tem um fardo de grama, uma ovelha e um lobo.

Ele só pode levar ou trazer um dos três de cada vez.

O lobo devora a ovelha se os dois ficarem sozinhos.

A ovelha come a grama se ficar sozinha com ela.

<http://rachacuca.com.br/jogos/o-lobo-e-a-ovelha/>

Fonte: Autores

Nas atividades de ambientação os alunos responderam individualmente a um questionário *online* que coletou dados sobre a idade, a escolaridade e os conhecimentos prévios sobre linguagens de programação, e foram apresentados ao *Scratch* por meio de orientações dirigidas e exploração livre dos recursos disponíveis. As atividades que envolveram os desafios de lógica foram organizadas por grau de complexidade (do menor para o maior grau). A proposta para o Desafio 1 (D1), mais simples de ser resolvido, era explorar os elementos condicionais por meio da elaboração de um jogo de perguntas e respostas. Para resolver o Desafio 2 (D2), com um grau de complexidade um pouco mais elevado, era necessário construir uma solução interativa em que o jogador distribuía as

frutas nas caixas corretas. Finalmente, para concluir o Desafio 3 (D3), mais complexo, os alunos deveriam propor um jogo em que o jogador manipulava vários personagens até completar corretamente a travessia do rio.

A dinâmica usada pelos monitores nas atividades dos desafios de lógica consistiu em apresentar e discutir o problema com o grupo e, em seguida, orientar os alunos na implementação da resolução que cada um imaginou usando os recursos do *Scratch*, auxiliando-os quando solicitados. Apesar de contarem com um computador para cada, os alunos eram incentivados a discutir e compartilhar suas descobertas com os colegas no Estúdio criado para a turma no *Scratch* (<https://scratch.mit.edu/studios/1547222/>).

Por fim, para encerrar as atividades do projeto, os alunos elaboraram um relatório individual onde registraram suas reflexões sobre o percurso de aprendizagem e sua evolução em relação aos conceitos, às práticas e perspectivas computacionais.

3.1. Alunos participantes

As atividades foram realizadas com um grupo de 14 alunos do projeto Pré- IC, seis do sexo feminino e oito do sexo masculino, com idades entre 13-19 anos. Dos 14 alunos, a maioria (11 alunos) frequentava o Ensino Médio-Técnico, sendo nove alunos do curso de Programação de Jogos Digitais e dois alunos do curso de Processamento de Dados; dois alunos estavam no 8º ano do Ensino Fundamental e um aluno no Ensino Médio. Em relação ao nível de conhecimento relacionado à área da Computação, os onze alunos provenientes do Ensino Médio-Técnico possuíam algum conhecimento sobre linguagem de programação (C, C++), enquanto os outros três alunos, do Ensino Fundamental e Médio, não tinham qualquer contato com conteúdos da área da Computação.

Apesar de coletar e analisar os resultados de todos os alunos participantes, para compor a análise detalhada das estratégias de avaliação apresentada neste trabalho, exploramos os dados provenientes de uma amostra de três alunos. Os dados resultantes da participação dos três alunos foram selecionados por serem eles alunos do Ensino Fundamental e Médio, que não possuíam contato anterior com linguagens de programação, e evidenciam os elementos do pensamento computacional que nos propomos a discutir nesse trabalho. Além disso, a análise dos dados possibilitou verificar se os instrumentos escolhidos para avaliar a aquisição/ evolução do pensamento computacional eram satisfatórios. Apesar de incluirmos na amostra alunos de diferentes níveis de ensino (Fundamental e Médio), não houve intenção de comparar o desempenho dos alunos e sim avaliar a evolução individual de cada um.

3.2. Procedimentos e métodos

A pesquisa realizada pode ser considerada de natureza mista, sendo que os procedimentos metodológicos adotados englobaram instrumentos de coleta de dados como questionários, avaliação automática das atividades desenvolvidas no *Scratch*, observação das atividades práticas realizadas no laboratório da Universidade e análise documental dos relatórios finais, entregues por cada participante no final do projeto.

A análise dos dados coletados seguiu uma abordagem interpretativa, pois o objetivo principal do estudo era investigar se o pensamento computacional pode ser trabalhado por meio da realização de desafios de lógica usando o Scratch e verificar se tal estratégia contribui para o desenvolvimento e ampliação do arcabouço de conceitos relacionados aos princípios do pensamento computacional.

Para apoiar a análise foram consideradas as ações descritas para as três dimensões (Conceitos, Práticas e Perspectivas) apresentadas por Brennan e Resnick (2012), que configuram as “conexões do pensamento computacional” (Quadro 1).

Quadro 1 – Conexões do Pensamento Computacional

Dimensão	Conceitos	Descrição das ações
Conceitos Computacionais	Sequência	identificar uma série de etapas de uma tarefa
	Ciclos	executar a mesma sequência várias vezes
	Paralelismo	fazer as ações decorrerem ao mesmo tempo
	Eventos	fazer um acontecimento causar outro acontecimento
	Condições	tomar decisões com base em condições
	Operadores	expressar operações matemáticas e lógicas
	Dados	armazenar, recuperar e atualizar valores
Dimensão	Práticas	Descrição das ações
Práticas Computacionais	ação iterativa e incremental	desenvolver um pouco, depois verificar se funciona e, em seguida, desenvolver um pouco mais
	teste e depuração	certificar-se de que tudo funciona e encontrar e corrigir erros
	reutilização e reformulação	fazer algo utilizando o que outros (ou o próprio) já fizeram
	abstração e modulação	construir algo grande unindo conjuntos de partes menores
Dimensão	Perspectiva	Descrição das ações
Perspectivas Computacionais	Expressar	perceber que a computação é um meio de criação
	Conectar	reconhecer a vantagem de criar com e para outros
	Questionar	sentir que se pode fazer perguntas sobre o mundo

Fonte: Baseado em Brennan e Resnick (2012)

Além dos dados coletados pelo questionário *online*, aplicado no início do projeto para traçar o perfil dos alunos participantes, foram utilizados outros três instrumentos de coleta de dados: i) as telas com o resultado da avaliação automática dos projetos desenvolvidos pelos alunos, geradas com a aplicação da ferramenta *Dr. Scratch*; ii) o documento de registro das observações das práticas e; iii) os relatórios individuais finais, com as reflexões sobre o percurso de aprendizagem apresentados por cada participante.

Assim, considerando as dimensões, conceitos e respectivas ações apresentadas no Quadro 1 foram definidas as categorias e os indicadores para análise (Quadro 2), de acordo com os registros dos instrumentos que continham os dados coletados.

Quadro 2 – Dimensões, categorias e indicadores da análise

Dimensões	Categorias (conceitos)	Indicadores	Instrumentos e registros
Conceitos computacionais	abstração (sequência)	níveis de abstração e decomposição do problema	Códigos dos projetos compartilhados no Estúdio do <i>Scratch</i> ; Resultados da avaliação automática (<i>Dr. Scratch</i>)
	controle de fluxo (ciclos)	noções algorítmicas e controle de fluxo de dados	
	paralelismo (paralelismo)	rotinas de execução em paralelo	
	sincronização (eventos)	grau de sincronização entre um acontecimento e outro	
	lógica (condições e operadores)	uso da lógica condicional para tomada de decisões com o apoio de operações matemáticas	
	representação de dados (dados)	uso de variáveis para armazenar, recuperar e atualizar valores	
	interatividade (não há conceito relacionado)	recursos que promovem a interatividade com o usuário	
Práticas Computacionais	ação iterativa e incremental	frases que exteriorizam o processo de imaginar e construir, testar e buscar implementar as ideias novas em ações incremental	Observação das práticas durante o desenvolvimento dos projetos;

	teste e depuração	movimento de experimentar a solução proposta, identificar os erros e depurar o código	relatórios individuais finais
	reutilização e reformulação	evidências de reutilização de códigos, leitura crítica de outros projetos e trocas de ideias para reformulação dos projetos	
	abstração e modulação	expressões que denotam o exercício de compreender o problema, decompor-lo em partes e juntá-las para compor o todo	
Perspectivas Computacionais	Expressar	evidências de que o computador pode ser usado para criar e expressar as próprias ideias	Observação das práticas durante o desenvolvimento dos projetos; relatórios individuais finais
	Conectar	indícios de valorização da partilha tanto no que diz respeito a fazer com os outros como no fazer para os outros	
	Questionar	comprovações sobre o ato de interrogar as potencialidades e limites das tecnologias	

Fonte: Autores

É importante observar que as categorias definidas para a dimensão “Conceitos Computacionais”, apresentadas no Quadro 2, foram renomeadas em função dos registros coletados pelo instrumento utilizado para a avaliação dos códigos implementados pelos alunos, no caso, a ferramenta *Dr. Scratch*. Os conceitos correspondentes ao Quadro 1 estão descritos entre parênteses.

Posteriormente, as categorias definidas foram confrontadas com os resultados da avaliação automática dos projetos (conceitos computacionais) e com os registros sobre as práticas e as perspectivas computacionais, extraídos das reflexões registradas pelos alunos participantes nos relatórios individuais. Essa comparação permitiu identificar elementos relevantes para o estudo em questão e propiciou uma reflexão crítica sobre os resultados alcançados.

Na próxima seção são apresentados os resultados alcançados com o estudo, bem como a análise dos dados e a discussão dos aspectos mais relevantes para o trabalho.

4. Resultados, análise e discussão

Os jogos digitais interativos criados pelos alunos durante a participação nas atividades que envolveram os três desafios de lógica (D1, D2 e D3), mostram a evolução de

cada aluno no que diz respeito à aprendizagem dos conceitos relacionados aos princípios do pensamento computacional, independente do nível de ensino que cursava e do conhecimento técnico que possuía. É importante ressaltar que não houve intenção de comparar o desempenho entre os alunos e sim a evolução individual de cada um. Os jogos criados podem ser conferidos na galeria <https://scratch.mit.edu/studios/1547222/>.

De forma geral foi possível observar que os alunos evoluíram, tanto no grau de complexidade dos elementos que usaram para compor a interface, quanto na escolha dos blocos de comandos que constituem o código do jogo criado. Por exemplo, o recorte do jogo elaborado para o Desafio 1 (D1) apresentado na Figura 2 mostra que o aluno estava em uma fase inicial da aprendizagem dos conceitos de lógica de programação, incluindo poucos elementos na composição da interface e fazendo uma utilização elementar dos blocos de comando da linguagem para compor o código. Uma evolução gradativa pode ser observada em outro exemplo (Figura 3), no qual identifica-se uma exploração maior dos elementos que compõem a interface mais interativa e um uso mais estratégico dos blocos de comandos. Os jogos completos podem ser acessados na galeria <https://scratch.mit.edu/users/CarollinaM/>.

Figura 2 – Recorte da interface e código criado para o D1: Jogo Perguntas e Respostas



Fonte: Autores

Figura 3 – Recorte da interface e código criado para o D3: Jogo A travessia do rio



Fonte: Autores

A confirmação da evolução da aprendizagem dos alunos foi feita por meio da análise dos dados coletados pelos instrumentos mencionados e pela aplicação da ferramenta de avaliação *Dr. Scratch*. Assim, observando as categorias e indicadores apresentados no Quadro 2, e os registros de cada instrumento utilizado na coleta dos dados, foi possível classificar os resultados obtidos e distribuí-los nas três dimensões (conceitos, práticas e perspectivas) que configuram o pensamento computacional. Então, a partir da avaliação automática dos projetos desenvolvidos no *Scratch* (Tabela 1), realizada com a ferramenta *Dr. Scratch*, foi possível identificar a presença de elementos da dimensão “conceitos computacionais” e a evolução dos alunos no que diz respeito à aplicação desses conceitos em seus projetos.

Tabela 1 – Exemplos dos resultados da avaliação automática *Dr. Scratch*

Alunos		Aluno 1			Aluno 2			Aluno 3		
Desafios		D1	D2	D3	D1	D2	D3	D1	D2	D3
Pontuação <i>Dr. Scratch</i> para cada categoria da dimensão	lógica	3	3	3	3	3	3	3	3	3
	paralelismo	-	-	-	-	-	3	1	3	1
	interatividade	2	2	2	2	2	2	2	2	2

"Conceitos computacionais"	representação de dados	-	3	1	3	3	3	3	2	3
	controle de fluxo	3	2	1	3	1	1	3	2	3
	sincronização	-	-	2	2	-	2	2	2	-
	abstração	-	-	1	1	-	1	1	2	1
Total		8	10	10	14	9	15	15	16	13

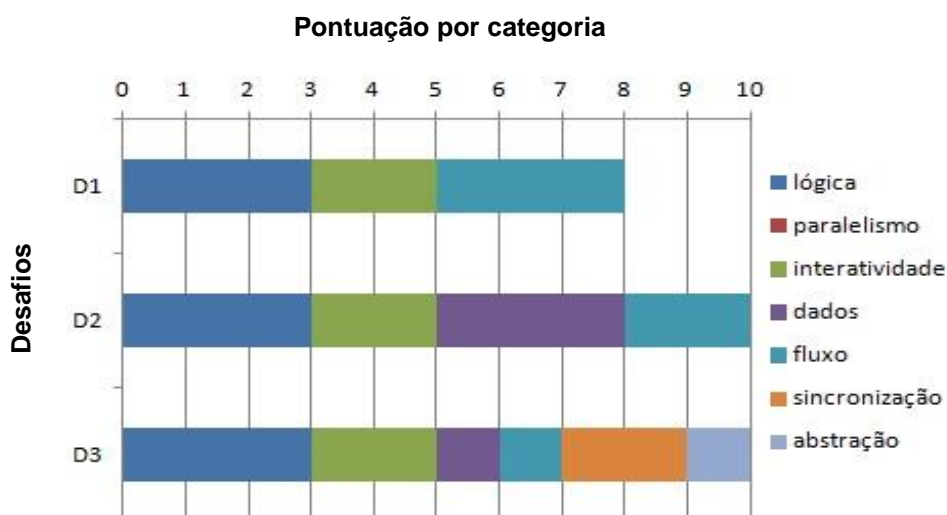
Fonte: Autores

Na Tabela 1 observa-se um exemplo da pontuação obtida por três alunos (A1, A2 e A3) em cada categoria da dimensão "conceitos computacionais", bem como o total de pontos alcançados para cada desafio de lógica proposto (D1, D2 e D3). A ferramenta *Dr. Scratch* foi aplicada como avaliação final, após a conclusão de todos os desafios propostos. A pontuação para cada categoria varia de zero (para códigos que não contemplam os elementos desejados em sua estrutura) a três pontos (quando todos os elementos desejados são contemplados no código) e o total geral pode atingir 21 pontos.

Verifica-se na Tabela 1 que todos os alunos conseguiram com expressividade representar os conceitos de "lógica", obtendo a pontuação máxima (3) para a categoria, e "interatividade com o usuário", conquistando a pontuação mediana (2), nos três desafios propostos. Este resultado era esperado, pois, em aplicações do tipo jogos digitais, os elementos relacionados à tomada de decisão (estruturas condicionais e operadores) e à interatividade (funções de interação com os personagens) são essenciais para gerar diferentes ações de acordo com a situação definida. Da mesma forma, o conceito "controle de fluxo", mesmo com menor expressividade, foi utilizado por todos os alunos, sendo que a pontuação variou entre a mínima (1) e a máxima (3). A "representação de dados" também foi um conceito explorado pelos alunos, sendo exceção apenas o Aluno 1, que no primeiro desafio (D1) não aplicou o conceito. No caso do conceito "controle de fluxo", era desejado que os alunos explorassem os blocos de comandos, que contém as estruturas de repetição (*repita até que... repita sempre*), em seus projetos. Para representar os dados, os alunos deveriam utilizar variáveis e, em um nível mais avançado, lançar mão das listas. Ambos os conceitos (controle de fluxo e representação de dados) demandam um alto nível de reflexão. Por outro lado, os conceitos "paralelismo", "sincronização" e "abstração" foram os conceitos menos explorados pelos alunos e, portanto, os que obtiveram a menor pontuação. O uso reduzido do conceito "paralelismo", por exemplo, se justifica porque para fazer com que a aplicação realize determinadas ações independentes ou conectadas ao mesmo tempo é

necessário compreender que é possível adicionar os comandos em blocos separados. Da mesma forma, o conceito “sincronização” envolve a gestão de múltiplos processos concorrentes que acessam um mesmo recurso e, a “abstração”, a habilidade de analisar um problema e decompô-lo em partes, para se concentrar nos detalhes relevantes para sua resolução. Ambos os conceitos não são triviais para iniciantes em programação e demandam mais tempo para serem aprendidos (França; Amaral, 2013). No que diz respeito à evolução particular dos alunos em cada categoria, observamos na Tabela 1, que o Aluno 1 não aplicou o conceito “paralelismo” em nenhum dos desafios propostos. Também não obteve pontuação para os conceitos “sincronização” e “abstração” nos primeiros dois desafios, utilizando tais conceitos apenas no Desafio 3 (D3), quando obteve 2 pontos para “sincronização” e 1 ponto para “abstração”. Houve ainda um decréscimo no uso do conceito “controle de fluxo” e “representação de dados”. Dessa forma, o total de pontos obtidos pelo Aluno 1 não variou muito de um desafio para o outro, mas houve uma evolução de dois pontos do Desafio 1 (8) para o Desafio 2 (10), mantendo-se estável no Desafio 3 (10). Considerando que o Aluno 1 cursava o 8º ano do Ensino Fundamental, e não havia tido contato prévio com os conceitos da área da computação, constata-se que houve uma significativa evolução no seu desempenho pois, com exceção do conceito “paralelismo”, todos os demais foram gradativamente contemplados (Gráfico 1).

Gráfico 1 - Desempenho do Aluno 1 na dimensão “Conceitos Computacionais”



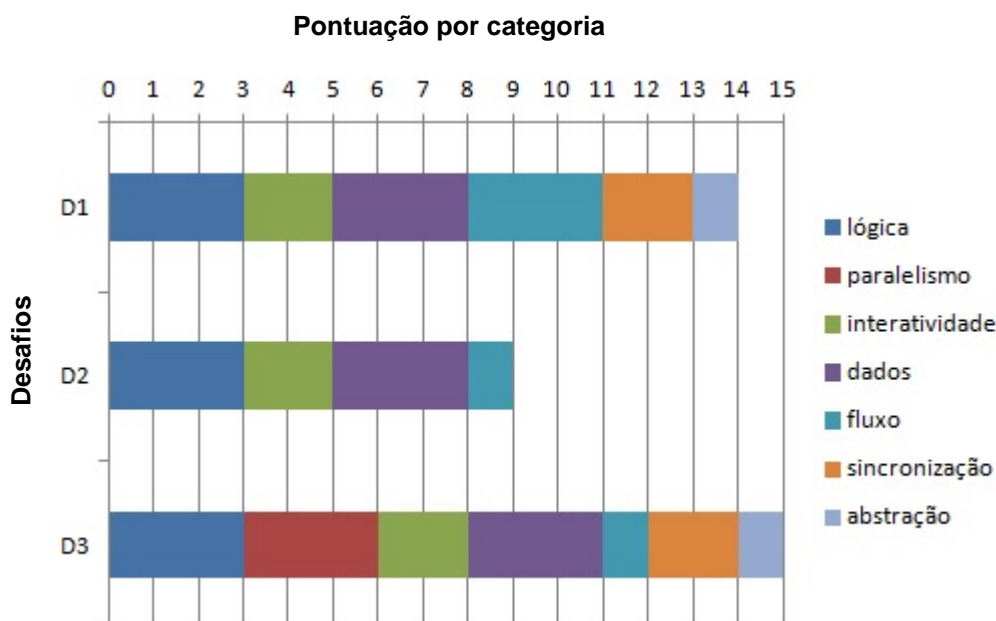
Fonte: Autores

Conforme apresentado no Gráfico 1, ao resolver o Desafio 1 (D1), o Aluno 1 aplicou somente três conceitos (lógica, interatividade e controle de fluxo). No Desafio 2 (D2) houve um pequeno avanço com a incorporação do conceito “representação de dados”. E, no

Desafio 3 (D3), o Aluno 1 utilizou quase todos os conceitos computacionais desejados, comprovando, portanto, uma significativa melhora na sua performance em relação à aplicação dos conceitos computacionais.

O Aluno 2 usou o conceito “paralelismo” nos Desafios 2 e 3, conquistando 3 pontos no Desafio 3, enquanto os conceitos “sincronização” e “abstração” foram usados nos três desafios, alcançando 2, 1 e 2 ponto(s) respectivamente. Entretanto, houve um decréscimo na aplicação dos conceitos “sincronização” e “abstração” no Desafio 2 (D2=1), sendo estes novamente aplicados com pontuação maior (D3=2) no Desafio 3. Assim, o total de pontos obtidos pelo Aluno 2 decaiu do Desafio 1 (14) para o Desafio 2 (12), evoluindo significativamente no Desafio 3 (16). Considerando que o Aluno 2 cursava o 1º ano do Ensino Médio, e não havia tido contato prévio com os conceitos da área, era de se esperar que tivesse dificuldade em aplicar alguns conceitos básicos, relacionados à programação de computadores, em todos os desafios propostos. Porém, a evolução de seu desempenho foi confirmada no Desafio 3, quando todos os conceitos computacionais foram aplicados (Gráfico 2).

Gráfico 2 - Desempenho do Aluno 2 na dimensão “Conceitos Computacionais”



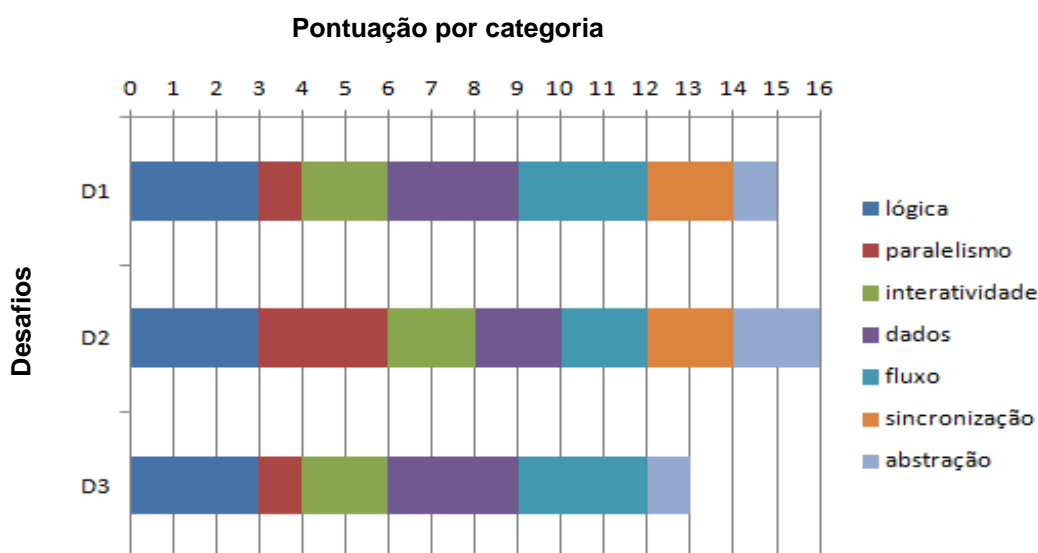
Fonte: Autores

Conforme apresentado no Gráfico 2, ao resolver o D1, o Aluno 2 aplicou seis conceitos (lógica, interatividade, representação de dados, controle de fluxo, sincronização e abstração) relacionados ao pensamento computacional. No D2 houve um considerável

decréscimo, com a supressão dos conceitos “sincronização” e “abstração”, ausência que reforça a dificuldade dos iniciantes em programação no gerenciamento de múltiplos processos e na decomposição de um problema em partes (França; Amaral, 2013). Entretanto, no D3, o Aluno 2 utilizou todos os conceitos computacionais desejados, comprovando, portanto, um aumento da sua performance.

Por sua vez, o Aluno 3 representou o conceito “paralelismo” em dois desafios propostos, obtendo pontuação mediana (2) no D2 e pontuação mínima (1) no D3. Em relação à aplicação do conceito “sincronização”, alcançou 2 pontos nos dois primeiros desafios e não pontuou no D3. Quanto ao conceito “abstração”, conseguiu aplicá-lo nos três desafios, ainda que com pontuação mínima (1) nos Desafios 1 e 3 e pontuação mediana (2) no Desafio 2. O total de pontos obtidos pelo Aluno 3 manteve-se relativamente equilibrado ao longo dos três desafios (D1=13; D2=15 e D3=11), havendo um decréscimo no Desafio 3. Considerando que o Aluno 3 cursava o 8º ano do Ensino Fundamental, e não havia tido contato prévio com os conceitos da área da computação, era esperado que tivesse dificuldade em aplicar alguns conceitos básicos, relacionados à programação de computadores, em todos os desafios propostos. Entretanto, curiosamente, o aluno demonstrou um equilíbrio no domínio dos conceitos nos dois primeiros desafios (D1=13), sendo que o ponto alto da performance foi confirmado no Desafio 2 (D2=15), quando todos os conceitos computacionais foram aplicados (Gráfico 3).

Gráfico 3 - Desempenho do Aluno 3 na dimensão “Conceitos Computacionais”



Fonte: Autores

De acordo com o Gráfico 3, ao resolver o Desafio 1 (D1), o Aluno 3 aplicou todos os sete conceitos computacionais (lógica, paralelismo, interatividade, representação de dados, controle de fluxo, sincronização e abstração) investigados. Também no Desafio 2 (D2), todos os conceitos foram aplicados e, apesar do decréscimo na pontuação dos conceitos “representação de dados” e “controle de fluxo”, houve um acréscimo na pontuação dos conceitos “paralelismo” e “abstração”. Contrariando a expectativa, o desempenho do Aluno 3 decaiu no Desafio 3 (D3), quando além de não aplicar o conceito “sincronização” houve um decréscimo na pontuação dos conceitos “paralelismo” e “abstração”.

Com base nos resultados apresentados é possível afirmar que, no contexto dessa pesquisa, houve uma maior dificuldade por parte dos alunos em aplicar os conceitos computacionais “paralelismo”, “sincronização” e “abstração”. Dificuldade encontrada, por exemplo, pelo Aluno 1, do 8º ano do Ensino Fundamental, que não havia tido contato com a área da Computação. Porém, é possível observar que o Aluno 1 evoluiu, e ampliou sua performance em relação à apropriação dos conceitos computacionais. Os Alunos 2 e 3, provenientes do Ensino Médio e 8º ano do Ensino Fundamental respectivamente, apesar do pouco ganho em performance, conseguiram em um ou outro desafio proposto contemplar todos os conceitos computacionais sugeridos.

Com relação à dimensão “Práticas Computacionais” (Quadro 1), e com base nos indicadores de análise apresentados no Quadro 2, foi possível identificar nos relatórios individuais a presença, em menor e maior grau, das quatro categorias: i) ação iterativa e incremental; ii) teste e depuração; iii) reutilização e reformulação e iv) abstração e modulação. A “ação iterativa e incremental”, a “reutilização e reformulação” e o “teste e depuração” foram as práticas computacionais mais observadas e evidenciadas pelos alunos, e podem ser confirmadas por algumas frases extraídas dos relatórios individuais. Por exemplo, trechos destacados em negrito de frases que exteriorizam o processo de imaginar e construir, testar e implementar ideias novas em ações incrementais (ação iterativa e incremental) e o movimento de experimentar a solução proposta, identificar os erros e depurar o código (teste e depuração):

- *“as atividades de certa forma nos obrigavam a **entender o problema, e procurar uma solução**, e depois **reproduzindo isso com código**, o que na minha opinião foi um sucesso, pois realmente deu certo”.*

- *“(programar) nos ajuda no nosso desenvolvimento de raciocínio lógico e exercita a paciência para **corrigir os erros**... aprendi **que se aprende errando e acertando** e que a perfeição só se chega com a prática”.*

Com menor expressividade, mas presentes em alguns relatos dos alunos, as práticas de “abstração e modulação” também foram adotadas e podem ser confirmadas em expressões, como as destacadas em negrito, que denotam o exercício de compreender o problema, decompô-lo em partes e juntá-las para compor o todo, por exemplo:

- “foi muito divertido **construir os jogos e quebrar a cabeça** para determinar o que os personagens deveriam dizer e fazer”.

- “esses exercícios de lógica me ajudaram a **melhorar a minha capacidade de pensar**, por exemplo, em matemática, física, sociologia, filosofia e eletrônica”.

Na dimensão “Perspectivas Computacionais”, a observação realizada durante a resolução dos desafios e, principalmente, as reflexões dos alunos expressas nos relatórios individuais, permitiram, por meio dos indicadores definidos no Quadro 2, identificar as três categorias: i) expressar; ii) conectar e; iii) questionar, inerentes às perspectivas do pensamento computacional. Por exemplo, em trechos destacados em negrito de frases que evidenciam que o computador pode ser usado para criar e expressar as próprias ideias (expressar):

- “acredito que **revi alguns conceitos e aprendi a programar algoritmos** de uma forma mais lógica e objetiva”.

- “o primeiro desafio foi que **me fez mais bater a cabeça**, no segundo **tive dificuldade na lógica e facilidade na prática** no Scratch, no terceiro foi ao contrário do segundo, ou seja, **tive facilidade na lógica e dificuldade na prática**”.

Indícios de valorização da partilha, tanto no que diz respeito a fazer *com* os outros como no fazer *para* os outros (conectar), também surgiram nos trechos destacados em negrito de frases extraídas dos relatos dos alunos:

- “outra coisa que percebi foi **minha capacidade de trabalhar em grupo**, algo que ainda estou desenvolvendo, algumas vezes é fácil, mas outras é bem difícil, **é preciso paciência para lidar com pessoas, e saber ouvir o que os outros tem a dizer**”.

- “mas quando fiz uma das etapas dessa atividade **que era em dupla**, lembrei muito mais, **aprendi descobrindo** sobre as qualidades e defeitos dessa atividade também, como **ser aberta para novas experiências**”.

Por fim, outros trechos em negrito, destacados dos relatos dos alunos, acabam por trazer comprovações sobre a perspectiva de interrogar as potencialidades e limites das tecnologias (questionar):

- “o Scratch **desperta um pouco mais a criatividade** de programar jogos e **não fica, às vezes, sem graça** como a linguagem de programação normal, **crua**, por causa das cores e

até por ser em balões os comandos **o que ajuda até para uma criança aprender sobre programação de jogos e desperta mais o interesse**".

- "gostei muito de utilizar o Scratch para desenvolver os exercícios, pois pra mim, que entendo pouco de programação, **foi uma ótima ideia trabalhar com ele, focando apenas na lógica do exercício, sem precisar se preocupar com códigos, etc**".

5. Considerações finais

No intuito de contribuir para a ampliação das reflexões sobre as abordagens usadas na avaliação da aquisição dos princípios do pensamento computacional no contexto educacional brasileiro, apresentamos nesse trabalho os resultados alcançados com um projeto de pré-iniciação científica desenvolvido com um grupo de alunos do Ensino Fundamental, Médio e Médio-Técnico. O projeto em questão buscou investigar como as dimensões (conceitos, práticas e perspectivas) do pensamento computacional, apresentadas por Brennan e Resnick (2012), podem ser trabalhadas por meio de atividades que envolvem a resolução de desafios de lógica, usando a linguagem de programação *Scratch*, e apresentou a contribuição de instrumentos (questionários, roteiros de observação e ferramenta de avaliação automática *Dr. Scratch*) utilizados na avaliação da aplicação dos princípios do pensamento computacional nos jogos digitais produzidos pelos alunos. Os resultados apresentados nesse trabalho foram extraídos de uma amostra dos dados coletados junto a três alunos (dois alunos do 8º ano do Ensino Fundamental e um aluno do Ensino Médio) participantes do projeto.

Para orientar o estudo foram elaboradas três questões de pesquisa. A primeira questão investigou se os princípios do pensamento computacional poderiam ser desenvolvidos junto dos alunos por meio de desafios de lógica, usando o aplicativo *Scratch*. Os resultados apontam que, se bem orientadas, é possível trabalhar os princípios do pensamento computacional por meio de atividades que envolvem a resolução de desafios de lógica, aplicando os recursos do *Scratch*, e contribuir para a ampliação do arcabouço conceitual e para a prática de resolução de problemas. Nesse sentido, é importante observar que a escolha dos níveis de dificuldade dos desafios propostos influenciou o desempenho dos alunos e, conseqüentemente, a motivação para implementarem os projetos no *Scratch*.

A segunda questão de pesquisa, de cunho mais específico, buscou identificar quais os conceitos, práticas e perspectivas computacionais foram desenvolvidos/aprimorados pelos alunos por meio da realização de atividades de programação de jogos digitais interativos usando o *Scratch*. Por meio da análise dos dados coletados foi possível

identificar e classificar, em maior e menor grau de dificuldade, o desempenho individual dos alunos na aplicação dos conceitos de “lógica”, “interatividade com o usuário”, “controle de fluxo” e “representação de dados” (maior grau de expressividade), e os conceitos “paralelismo”, “sincronização”, “abstração” (menor grau de expressividade). Em relação às práticas computacionais, a “ação iterativa e incremental”, a “reutilização e reformulação” e o “teste e depuração” foram as mais observadas e, com menor expressividade, surgiram práticas de “abstração e modulação”. No que diz respeito às perspectivas computacionais, as três categorias “expressar”, “conectar” e “questionar” foram identificadas nas atitudes dos alunos ao programarem os seus projetos no *Scratch*.

Por fim, a terceira questão de pesquisa estava relacionada às estratégias que poderiam ser efetivas para avaliar a aquisição e o aprimoramento dos princípios do pensamento computacional pelos alunos quando realizam atividades de programação de jogos digitais interativos usando a linguagem de programação *Scratch*. Considerando que foram utilizados três tipos de instrumentos de avaliação (análise automática dos projetos - *Dr. Scratch*, observação das práticas e relatórios individuais dos alunos), que aportaram resultados distintos e complementares, é possível afirmar que essa combinação de instrumentos foi positiva para o contexto estudado, pois contribuiu para evidenciar o grau evolutivo de desempenho dos alunos. Entretanto, vale ressaltar que, por serem instrumentos que apoiam estratégias avaliativas de cunho mais diagnóstico, não possibilitam que seus resultados gerem uma intervenção mais formativa, durante o desenvolvimento dos projetos. Nesse sentido, Brennan, Balch e Chung (2014) propõem uma estratégia de avaliação “no processo”, criando oportunidades para os alunos falarem sobre a sua própria criação e sobre a criação dos colegas e também sobre suas práticas. Esse tipo de estratégia valoriza o engajamento do aluno em seu percurso de aprendizagem e dá importância ao erro como possibilidade de reavaliação do processo. Assim, sugere-se como trabalho futuro a aplicação dos instrumentos de avaliação apresentados durante todo o processo de desenvolvimento dos projetos. Por exemplo, a ferramenta de avaliação automática *Dr. Scratch*, quando aplicada em vários momentos do desenvolvimento dos projetos, pode ser um bom recurso para o aluno se auto-avaliar, aprender e refazer trechos do seu código.

Esperamos que os resultados do estudo aqui apresentado possam contribuir para as reflexões sobre propostas e instrumentos de avaliação que acompanham as iniciativas de promoção do pensamento computacional na escola e apoiem a realização de outros trabalhos na área.

6. Referências

BELL, T.; WITTEN, I. H.; FELLOWS, M. **Computer Science Unplugged**: Ensinando Ciência da Computação sem uso do computador. 2011. Tradução por Luciano Porto Barreto. Disponível em: <<http://csunplugged.org>>. Acesso em: 12 ago. 2016.

BLIKSTEIN, P. **O Pensamento Computacional e a Reinvenção do Computador na Educação**, 2008. Disponível em: <<http://bit.ly/1IXIbNn>>. Acesso em: 12 de ago. 2016.

BRENNAN, K.; BALCH, C.; CHUNG, M. **Creative computing**: An Introductory Computing Curriculum Using Scratch. MIT Harvard Graduate School of Education, 2014.

BRENNAN, K.; RESNICK, M. New frameworks for studying and assessing the development of computational thinking. **American Educational Research Association meeting**, 2012. Disponível em: <<https://goo.gl/Fr8gtF>>. Acesso em: 12 ago. 2016.

CARVALHO, M. L. B. de; CHAIMOWICZ, L.; MORO, M. M. Pensamento Computacional no Ensino Médio Mineiro. **XXI Workshop sobre Educação em Computação**, p. 641-650, 2013.

FRANÇA, R. S. de; AMARAL, H. J. C. do. Proposta Metodológica de Ensino e Avaliação para o Desenvolvimento do Pensamento Computacional com o Uso do Scratch. **XIX Workshop de Informática na Escola**, p. 179-188, 2013.

FRANÇA, R. S. de; TEDESCO, P. C. de A. R. Desafios e oportunidades ao ensino do pensamento computacional na educação básica do Brasil. **XXI Workshop de Informática na Escola**, p. 1464-1473, 2015.

GOULART, N. Precisamos despertar paixões pelas ciências. **Revista Veja**, Acervo Digital, 2011. Disponível em: <<http://abr.ai/1kUXq7h>>. Acesso em: 12 de ago. 2016.

MORENO-LEÓN, J.; ROBLES, G.; CHUSIG, C. **Dr scratch**, Automatic analysis of Scratch projects to assess the development of CT, 2014. Disponível em: <<https://goo.gl/LIEcgl>> Acesso em: 12 ago. 2016.

PAPERT, S. **Mindstorms**: Children, Computers, and Powerful Ideas. 2nd ed. NY: Basic Books, 1993.

RAMOS, J. L.; ESPADEIRO, R. G. Os futuros professores e os professores do futuro. Os desafios da introdução ao pensamento computacional na escola, no currículo e na aprendizagem. **Educação, Formação & Tecnologias**, 7(2): 4-25, 2014.

RAMOS, J. L.; ESPADEIRO, R. G. Pensamento computacional na escola e práticas de avaliação das aprendizagens. Uma revisão sistemática da literatura. **IX Conferência Internacional de TIC na Educação**, Challenges, 2015. Disponível em: <<https://goo.gl/taWGEY>> Acesso em: 12 ago. 2016.

RESNICK, M. et al. Scratch: Programming for all. **Comm. ACM** 52, 11, 60–67, 2009.

RODRIGUEZ, C. L.; ZEM-LOPES, A. M.; MARQUES, L.; ISOTANI, S. Pensamento Computacional: transformando ideias em jogos digitais usando o Scratch. **XXI Workshop de Informática na Escola**, p. 62-71, 2015.

SANTOS, G.; SILVA, W.; CAVALHEIRO, S.; FOSS, L.; AGUIAR, M.; PERNAS, A. M.; BOIS, A. D.; REISER, R. Proposta de atividade para o quinto ano do ensino fundamental: Algoritmos Desplugados. **XXI Workshop de Informática na Escola**, p. 246-255, 2015.

SCAICO, P. D.; LIMA, A. A. de; SILVA, J. B. B. da; AZEVEDO, S.; PAIVA, L. F.; RAPOSO, E. H.; ALENCAR, Y.; MENDES, J. P.; SCAICO, A. Ensino de Programação no Ensino Médio: Uma Abordagem Orientada ao Design com a linguagem Scratch. **Revista Brasileira de Informática na Educação**, 21(2): 92-103, 2013.

UFPel **EXP-PC** - Explorando o Pensamento Computacional para a Qualificação do Ensino Fundamental. 2014. Disponível em: <<https://goo.gl/UKSfKv>>. Acesso em: 12 ago. 2016.

WANGENHEIM, C. G. von; NUNES, V. R.; SANTOS, G. D. dos. Ensino de Computação com SCRATCH no Ensino Fundamental – Um Estudo de Caso. **Revista Brasileira de Informática na Educação**, 22(3): 115-125, 2014.

WING, J. Computational Thinking. **Communication of the ACM**, 49(3): 33-36, 2006.

WING, J. M. Computational Thinking - What and Why? **Carnegie Mellon School of Computer Science**, 2010. Disponível em: <<https://goo.gl/V8PdPi>>. Acesso em: 12 ago. 2016.

WING, J. Computational Thinking Benefits Society. 2014. **Social Issues in Computing**. Disponível em: <<http://socialissues.cs.toronto.edu>>. Acesso em: 12 ago. 2016.