Desenvolvendo o Pensamento Computacional no Ensino Fundamental com o uso do *Scratch*

Sivaldo Joaquim de Santana[1], Wilk Oliveira[2]

[1]Núcleo de Excelência em Tecnologias Sociais (NEES) – Instituto de Computação (IC)
Universidade Federal de Alagoas (UFAL) – CEP: 57.072-900 – Maceió, AL – Brasil.

[2]Laboratório de Computação Aplicada à Educação e Tecnologia Social Avançada (CAEd) – Instituto de Ciências Matemáticas e de Computação (ICMC)
Universidade de São Paulo (USP) – CEP: 13.566-590 – São Carlos, SP – Brasil.

sivaldojoaquim@ic.ufal.br, wilk.oliveira@usp.br

Resumo. *O desenvolvimento do Pensamento Computacional possibilita a aquisição de habilidades essenciais para resolver problemas do mundo real. No entanto, abordagens para o desenvolvimento dessa habilidade ainda são raros no cenário educacional e um grande desafio a ser superado nas escolas brasileiras. Este artigo tem por objetivo relatar a experiência de uso da ferramenta Scratch com 137 estudantes do Ensino Fundamental Brasileiro. Para isso, foi utilizada uma abordagem de pesquisa mista (quantitativa e qualitativa). Como resultados, observou-se que houve desempenho positivo em conceitos fundamentais do Pensamento Computacional, além da ampliação da interação entre os estudantes e da promoção do desenvolvimento de habilidades importantes para o crescimento profissional e social dos estudantes.*

Abstract. *The development of Computational Thinking enables the acquisition of skills essential to solving real-world problems. However, approaches to developing this ability are still rare in the educational setting and an important challenge to be overcome in Brazilian schools. This paper aims to report the experience of using the Scratch tool with 137 Brazilian elementary students. For this, a mixed research approach (quantitative and qualitative) was used. As results, it was observed that there was a positive performance in fundamental concepts of Computational Thinking, in addition to the increase of the interaction between the students and the promotion of the development of important skills for the professional and social growth of the students.*

1. Introdução

Atualmente, os avanços tecnológicos têm despertado atenção da comunidade acadêmica e científica para a utilização das Tecnologias Digitais da Informação e Comunicação (TDIC) de forma adequada, crítica e reflexiva em diferentes níveis e contextos educacionais [Santana e Oliveira 2019; Valente 2019; Almeida e Valente 2019]. Neste sentido, as práticas pedagógicas com o uso das TDIC para resolver problemas complexos do mundo real e para desenvolver o pensamento crítico dos estudantes podem contribuir positivamente para a aquisição de habilidades essenciais do século XXI e sua qualificação para o mercado de trabalho [Hinterholz e Santos 2017].

DOI: 10.5753/cbie.wie.2019.158

Neste contexto, programar um computador requer habilidades como analisar problemas, além de desenvolver e testar soluções computacionais, tais como: aplicativos e jogos digitais. Desse modo, aplicar as técnicas de programação em contextos interdisciplinares como prática pedagógica na Educação Básica visa promover o desenvolvimento do Pensamento Computacional (PC) dos estudantes. Para [Wing 2006] o PC é uma estratégia para formulação de problemas e aplicação do raciocínio heurístico na descoberta de uma solução. Entretanto, vale ressaltar, que o PC não está diretamente relacionado à Computação, sendo, de acordo com [Brackmann 2017, p. 31] "uma habilidade que qualquer pessoa deveria saber, independentemente da área de conhecimento ou atividade profissional, assim como saber ler, escrever e calcular".

Para [Wing 2006], o PC é uma habilidade fundamental para todos, e não apenas para cientistas da computação. Segundo a pesquisadora, o PC envolve a resolução de problemas, projeção de sistemas, a decomposição de questões difíceis em problemas menores, de modo a facilitar o processo de resolução, desenvolver a capacidade de abstração, automação e análise embasada nos conceitos da Ciência da Computação. Neste contexto, pesquisas recentes [*e.g.,* Bordini *et al.* 2017; Torres *et al.* 2018] sobre a inserção do PC em diferentes níveis e contextos educacionais, têm buscado a utilização de ferramentas para auxiliar na construção do PC como os ambientes de programação *Scratch, Alice, AgentCubes* e *AgentSheets*, propostas de atividades, estratégias de aplicação, práticas pedagógicas, desenvolvimento do raciocínio lógico e avaliação de habilidades dos estudantes da Educação Básica.

Diante disso, a necessidade de inclusão dos estudantes da Educação Básica na cultura digital é reforçada pelo próprio Ministério da Educação (MEC) como uma das dez competências gerais na Base Nacional Comum Curricular (BNCC) visando "compreender, utilizar e criar TDIC de forma crítica, significativa, reflexiva e ética nas diversas práticas sociais (incluindo as escolares) (...)" [Brasil 2018, p. 9]. A Sociedade Brasileira de Computação (SBC) tem discutido a inserção do Ensino de Computação na Educação Básica com foco nos eixos: PC, mundo digital e cultura digital, visando desenvolver as competências fundamentais dos estudantes do século XXI [SBC 2019]. Com base nisso, organizações sem fins lucrativos, independentes, também têm buscado contribuir para este fim, como o Centro de Inovação para a Educação Brasileira (CIEB) que criou um Currículo de Referência em Tecnologia e Computação com ênfase na Educação Infantil e Ensino Fundamental, organizado em três eixos principais: cultura digital, PC e tecnologia digital [CIEB 2019].

Neste sentido, o presente artigo relata a experiência de uso do ambiente de programação visual *Scratch 3.0* [Scratch 2019], com objetivo de desenvolver o PC dos estudantes do Ensino Fundamental II em uma escola de Educação Básica, localizada ao Sul do Estado de Alagoas. Para isso, utilizou-se uma abordagem de pesquisa mista (qualitativo e quantitativo). Os principais resultados obtidos mostram que a utilização da ferramenta de programação contribuiu positivamente para o desenvolvimento do trabalho colaborativo, raciocínio lógico, interatividade com o usuário, representação de dados e sincronização, bem como, enfatiza a necessidade de melhorar outras características essências relacionadas ao PC (paralelismo, controle de fluxo e abstração), habilidades fundamentais do XXI.

2. Trabalhos Relacionados

De acordo com recentes estudos secundários [*e.g.*, Santos *et al*. 2018] Diferentes estudos têm sido conduzidos visando à aplicação de diversas ferramentas para o desenvolvimento do PC na Educação Básica. Neste contexto, no trabalho de [Rodriguez *et al.* 2015] é relatada a experiência vivenciada em um projeto desenvolvido no período de maio de 2014 a abril de 2015 envolvendo 7 estudantes, com idade entre 14 e 16 anos, sendo seis participantes do gênero feminino e apenas um do gênero masculino, do primeiro ano do Ensino Médio de uma escola da rede pública de ensino, com objetivo de desenvolver noções básicas do PC com o uso do *Scratch*. Como resultados, os pesquisadores destacam a descoberta e a materialização de conceitos relacionados ao PC, bem como o desenvolvimento de competências interpessoais e de pesquisa. Além disso, as atividades conduzidas com o ambiente de programação *Scratch* possibilitou estimular o raciocínio lógico, a resolução de problemas de forma lúdica e dinâmica e o desenvolvimento de noções básicas de programação.

Mais adiante, [Torres *et al.* 2018] realizaram no período de 2015 a 2016 cursos sobre o PC que resultou na formação de 89 estudantes do Ensino Médio de três municípios situados no Alto Sertão da Paraíba, visando relatar o desenvolvimento do PC em escolas públicas e caracterizar a eficácia dos cursos de introdução à programação. Para isso, os pesquisadores utilizaram duas ferramentas: *LibreLogo*[1] em 4 aulas e *Scratch*[2] utilizado em 6 aulas. Os principais resultados revelam que o desempenho escolar dos estudantes melhorou. Além disso, na visão dos estudantes o curso contribuiu para melhoria da vida profissional e pretendem aplicar no mercado de trabalho, motivando a condução de novos estudos similares que possam beneficiar outros estudantes.

Recentemente, [Carlos *et al*. 2018] apresentaram um relato de experiência sobre o ensino de programação envolvendo 58 crianças, sendo 42 meninos e 16 meninas, na faixa etária entre 5 e 12 anos de idade com objetivo de oferecer um ambiente divertindo e interativo de ensino de programação nas férias escolares com o uso das ferramentas: *Code Studio*[3]*, Scratch, Monster Coding*[4]*, Pencil Code*[5] e *CodeMonkey*[6] para condução dos conteúdos e dinâmicas das aulas planejadas. Os principais resultados mostram que, das 58 crianças participantes, 40 concluíram o curso e 100% desses estudantes afirmaram que gostaram muito ou gostaram de terem participado. Na avaliação dos estudantes o *Scratch* foi escolhido como a ferramenta favorita e *Monster Coding* como a que menos gostaram. Apesar disso, 80% dos estudantes avaliaram as ferramentas como positivas e 85% dos estudantes afirmaram ter interesse em aprender mais sobre Ciência da Computação. Além disso, a maioria dos estudantes responderam corretamente as perguntas relacionadas aos conceitos de programação.

Diferentemente dos trabalhos de [Torres *et al.* 2018 e Carlos *et al*. 2018] que fizeram uso de várias ferramentas para ensinar conceitos relacionados a programação de computadores e [Rodriguez *et al.* 2015] que utilizaram o *Scratch* visando o

[1] Disponível em: http://librelogo.org/en/
[2] Disponível em: https://scratch.mit.edu
[3] Disponível em: https://code.org/
[4] Disponível em: http://monstercoding.com/
[5] Disponível em: https://pencilcode.net/
[6] Disponível em: https://www.playcodemonkey.com/

desenvolvimento do PC com um grupo reduzido de estudantes (apenas sete estudantes e uma professora) do primeiro ano do Ensino Médio. O presente estudo foi conduzido com aulas teóricas e práticas através do ambiente de programação *Scratch 3.0* com 137 estudantes do Ensino Fundamental II, sendo 66 estudantes do sexo masculino e 71 do sexo feminino, durante o período de fevereiro a junho de 2019 no laboratório de informática da própria escola. Além disso, foi aplicada uma abordagem de pesquisa mista (qualitativa e quantitativa) na avalição dos projetos desenvolvidos pelos estudantes.

3. Materiais e Métodos

O presente estudo foi realizado com quatro turmas do Ensino Fundamental II conforme mostrado na Tabela 1, com um total de N=137 estudantes, sendo 66 do sexo masculino e 71 do sexo feminino com o uso da ferramenta *Scratch*, em uma escola de Educação Básica, situada na região do Baixo do São Francisco, localizado ao Sul do Estado de Alagoas.

Tabela 1. Relação de anos e turmas de estudantes do Ensino Fundamental II

Nível	Ano/Turma	Gêneros		Número de Estudantes por Turma
		Masculino	Feminino	
Ensino Fundamental II	7º Ano "A"	28	16	44
	7º Ano "B"	8	6	14
	8º Ano "A"	15	27	42
	9º Ano "A"	15	22	37
	Total de estudantes			**137**

De acordo com [Gil 2012, p. 8] para que o "conhecimento possa ser considerado científico, torna-se necessário identificar operações mentais e técnicas que possibilitam a sua verificação". Diante disto, foi utilizada pesquisa de métodos mistos (quantitativo e qualitativo). Segundo [Creswell 2010, p. 27] o uso de métodos mistos é "uma abordagem da investigação que combina ou associa as formas qualitativa e quantitativa". Para isso, foram executadas as seguintes etapas: *(i)* seleção do ambiente de programação; *(ii)* escolha de materiais para auxiliar nos processos de ensino e aprendizagem; *(iii)* execução das aulas/atividades; *(iv)* observação e coleta dos dados; e *(v)* análise dos resultados e discussão.

Na primeira etapa, buscou-se selecionar com base em estudos científicos [Bordini *et al.* 2017; SB Ortiz e Pereira 2018] um ambiente de desenvolvimento gratuito que proporcionasse uma maior interatividade e usabilidade no processo de ensino e aprendizagem. Como resultado, foi escolhido o *Scratch 3.0*, um *software* gratuito e disponível em duas versões: uma versão denominada *Scratch 3.0 Desktop*[7] disponível para *download* e instalação no computador e outra versão *Scracth Online*[8] para utilizar através de um *browser*. O *Scratch* é um ambiente de programação visual, podendo ser utilizado para criação de jogos, animações e histórias interativas, desenvolvido por pesquisadores em novas tecnologias do *Lifelong Kindergarten Group* no *MIT Media Lab*, um grupo de pesquisa do *Massachusetts Institute of Technology – MIT* [Scratch 2019].

[7] Disponível em: https://scratch.mit.edu/download/
[8] Disponível em: https://scratch.mit.edu/projects/editor/

Na segunda etapa, realizou-se uma pesquisa através da *web* visando selecionar materiais de referências como suporte aos processos de ensino e de aprendizagem. Nesta fase, foi selecionado e adotado como material de apoio o *e-book*: "Programação de Computadores na Educação: um passo a passo utilizando o Scratch[9]" por apresentar uma abordagem didática, elaborado por pesquisadores do Grupo de Estudo e Pesquisa em Inclusão Digital (GEPID) da Universidade de Passo Fundo (UPF) [Oro, Pazinato e Teixeira 2016].

Na terceira etapa, foram conduzidas as aulas no laboratório de informática no período de seis meses, uma vez por semana com duração de 50 minutos/aula. Neste período, os estudantes conheceram o ambiente de programação, exploraram alguns projetos compartilhados por outros usuários e tutoriais disponíveis no *site* do *Scratch*. Foram ministradas aulas com abordagem sobre os conceitos de algoritmos, variáveis, operadores aritméticos, relacionais e lógicos, estruturas condicionais simples, composta e estruturas de repetição. Os estudantes desenvolveram várias atividades práticas e ao término de cada assunto, foram estimulados a resolver uma atividade desafio. Além disso, no final do semestre foi proposta a elaboração de um projeto interdisciplinar em equipes com no mínimo três e no máximo seis estudantes. Para isso, foi utilizado um modelo de planilha[10] na fase de planejamento para concepção dos jogos [Cruz Alves *et al.* 2016; Iniciativa Computação na Escola 2019].

Durante esse período (etapa *iv*), foram feitas anotações em um diário de bordo do pesquisador com as observações *in loco*, fotos, informações descritivas, as falas e comportamentos dos estudantes com a utilização do *Scratch* no laboratório de informática. Segundo [Lüdke e André 2013, p. 38] "essas medidas têm um caráter meramente prático, no sentido de ajudar a organização e a análise dos dados" qualitativos. Por último (etapa *v*), foi utilizada a ferramenta *Dr. Scratch*[11] para análise quantitativa dos projetos desenvolvidos no ambiente de programação *Scratch* pelos estudantes [Avila *et al.* 2017; Ribeiro *et al*. 2018].

O *Dr. Scratch* é uma ferramenta *online* e gratuita, desenvolvida por pesquisadores da *Universidad Rey Juan Carlos*, em Madri, Espanha [Webber *et al.* 2016] que além de analisar os blocos de comandos e mostrar possíveis erros de código nos projetos, identifica as competências com base em sete categorias/conceitos do PC: *paralelismo, lógica, controle de fluxo, interatividade com o usuário, representação dos dados, abstração e sincronização.* Na avaliação automática dos projetos, a ferramenta atribui para cada categoria uma pontuação de 0 a 3 pontos (podendo totalizar 21 pontos em cada projeto) [Moreno-León e Robes 2015; Rodriguez, Reis e Isotani 2017]. Além disso, os dados foram tabulados e processados com o uso do *software R*[12] para análise da estatística descritiva.

4. Resultados e Discussão

Ao longo do estudo, foram criados 30 jogos digitais em formato de *quiz* com abordagens interdisciplinares com o uso da ferramenta *Scratch 3.0* pelos estudantes do 7º ano, 8º ano e 9º ano do Ensino Fundamental II, como parte de um projeto da

[9] Publicado pela editora UPF. Disponível em: https://bit.ly/2xf4qdy
[10] Disponível em: https://bit.ly/2XyZxKY
[11] Disponível em: http://www.drscratch.org/
[12] Disponível em: https://www.r-project.org/

disciplina de Informática. Todos os jogos estão disponíveis em um estúdio criado pelo professor responsável pela disciplina no *link:* https://bit.ly/2X3COao. Na Tabela 2, observa-se a estatística descritiva com as medidas de tendência central dos sete conceitos do PC obtidos a partir dos dados gerados pela ferramenta *Dr. Scratch* [Moreno-León e Robes 2015], tabulados e processados pelo *software R*.

Tabela 2. Sumarização dos dados relativos às sete características do PC.

	Lógica	**Paralelismo**	**Interatividade com o usuário**	**Representação de dados**	**Controle de fluxo**	**Sincronização**	**Abstração**
Mínimo	1.000	0.000	2.000	0.000	1.000	0.000	0.000
1º Quartil	2.000	0.000	2.000	1.000	1.000	2.000	1.000
Mediana (2º Quartil)	2.000	0.500	2.000	2.000	1.000	2.000	1.000
Média	2.100	1.100	2.000	1.530	1.330	1.730	0.800
3º Quartil	2.000	3.000	2.000	2.000	1.750	2.000	1.000
Máximo	3.000	3.000	2.000	3.000	3.000	3.000	2.000

Conforme mostrado na Tabela 2, a média do desenvolvimento do pensamento **lógico** dos estudantes foi de 2.100 e mediana 2.000. Isso significa dizer que, no geral os projetos desenvolvidos estão acima da média por possuir em suas estruturas, operadores lógicos (***E, OU*** e ***NÃO***) que combinam as condições, de modo a apresentar resultados diferentes dependendo das situações programadas. Resultado semelhante foi obtido no conceito de **interatividade com o usuário** (média = 2.000) e (mediana = 2.000). Esse resultado indica que foram aplicados diferentes formas para o usuário interagir com o jogo, ou seja, podendo utilizar o ***mouse***, **teclado, *webcam*** ou **microfone**, exemplo: no jogo "*II War Quiz*"[13] a equipe programou para o usuário selecionar a alternativa correta com o uso do *mouse*, já no jogo "*Historiando*"[14] o usuário responde as perguntas através do teclado. No geral, foram utilizadas apenas essas duas formas de interação.

Desempenho relevante foi obtido também no conceito de **representação de dados** (média = 1.530 e mediana = 2.000) isso indica que os estudantes utilizaram em seus projetos um conjunto de informações a respeito dos personagens para funcionar corretamente, exemplo: a posição de um personagem nos eixos X, Y, direção, tamanho, variáveis, listas, etc. Nos projetos, observou-se a definição da posição dos personagens nos eixos X, Y e a criação de variáveis para mostrar a pontuação ou o número de vidas nos jogos, como exemplo, no jogo "*QuizDNA*"[15] a equipe implementou variáveis para mostrar os números de questões respondidas certas e erradas.

Além disso, nota-se um desempenho relevante no princípio de **sincronização** (média = 1.730 e mediana = 2.000) isso indica a ordem com que as ações dos personagens do jogo ocorrem, exemplo, no jogo "*Seja um jovem cientista*"[16] ao clicar com o *mouse* na bandeira "verde" para iniciar, simultaneamente o personagem "troféu" é escondido e o "professor" é apresentado com a mensagem: *"Olá, seja bem vindo ao*

[13] Disponível em: https://scratch.mit.edu/projects/311589767/
[14] Disponível em: https://scratch.mit.edu/projects/309795521/
[15] Disponível em: https://scratch.mit.edu/projects/309506272/
[16] Disponível em: https://scratch.mit.edu/projects/308193470/

nosso quiz de ciências!" por 3 segundos, em seguida, a equipe programou para esperar mais 1 segundo e então apresenta as regras do jogo: *"Regras: ESCREVA A RESPOSTA CORRETA NO QUADRO ABAIXO E SE ERRAR TENTE NOVAMENTE",* na sequência é exibida a questão 1 e espera o usuário digitar a resposta, se estiver correta é mostrada a mensagem *"resposta correta"* por 2 segundo, em seguida, avança para próxima questão, caso contrário, surge a mensagem: *"resposta errada, tente novamente!"* por 2 segundos e repete a mesma pergunta para o usuário tentar responder novamente. No total, o jogo possui 12 perguntas e apenas no final é mostrado o "troféu" para o usuário.

Entretanto, observa-se um desempenho baixo no conceito de **paralelismo** (média = 1.100 e mediana = 0.500) o que significa afirmar que apenas alguns jogos foram programados para executar várias ações ao mesmo tempo, ou seja, simultaneamente. Além disso, observou-se poucas implementações de blocos de códigos para o **controle de fluxo** (média = 1.330 e mediana = 1.000) de modo a controlar o comportamento dos personagens nos jogos desenvolvidos pelos estudantes, como por exemplo: o uso de estruturas de repetição (*sempre, repita "x" vezes* e *repita até que "x"*) e estruturas condicionais: simples (*se então*) e composta (*se então, senão*), o mesmo foi aconteceu no item relacionado à **abstração** (média = 0.800 e mediana = 1.000) habilidade de abstrair e decompor problemas em partes menores, de modo que torne mais fácil a compreensão do algoritmo e facilite as futuras correções e/ou alterações no código.

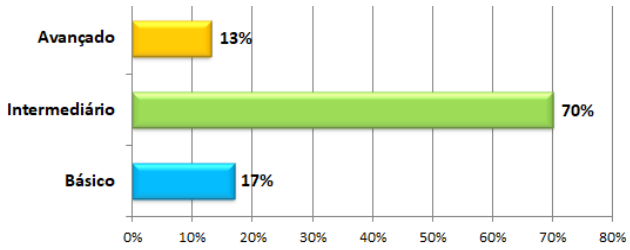

Figura 1. Níveis dos projetos.

Observa-se no gráfico da Figura 1, os níveis dos projetos desenvolvidos pelos estudantes com o uso do ambiente de programação *Scratch*. Do total geral de 30 jogos criados pelos estudantes, 5 (17%) foram avaliados como básicos, 21 (70%) foram classificados como intermediários e 4 (13%) dos jogos foram avaliados como avançados. Vale ressaltar, que os participantes deste estudo não conheciam o ambiente de programação *Scratch* e pela primeira vez foi introduzido programação de computadores na disciplina de Informática para os estudantes do Ensino Fundamental II. Apesar disso, nota-se que, um percentual importante dos projetos desenvolvidos, foram classificados como intermediário (70%) e 13% dos projetos no nível avançado, isso mostra que os conteúdos trabalhados durante o semestre, devem ter sido compreendidos e aplicado no desenvolvimento dos jogos.

Com relação aos resultados observados no estudo qualitativo, notou-se interação e colaboração dos estudantes durante o desenvolvimento dos jogos com o uso da ferramenta *Scratch 3.0*. Em determinados momentos, algumas equipes se reuniram para analisar as estratégias do jogo, como por exemplo: o modo como usuário interage com o jogo (*mouse* ou teclado), as regras do jogo (*e.g.,* se o usuário errar uma quantidade "x" de questões, ele perde o jogo e volta à questão inicial), atribuição de pontos para cada questão respondida corretamente. Neste contexto, um dos membros da equipe comentou: *"Professor, veja só, vamos fazer assim: quando o jogador iniciar o jogo vai*

mostrar 5 vidas. Cada questão respondida errada, perde uma vida. Se perde todas as 5 vidas, o jogo termina. Game over!". Outro estudante da mesma equipe acrescentou: *"isso vai ficar muito legal!"*. Resultados semelhantes de interação e colaboração foram encontrados em um estudo de caso conduzido por [Cruz Alves *et al.* 2016].

Após as equipes finalizarem os projetos, todos os estudantes apresentaram os jogos desenvolvidos no laboratório de informática com o uso de computador e *datashow*. Nesse momento, observou-se a participação dos estudantes na tentativa de responder corretamente as questões propostas nos jogos. Além disso, puderam acessar e jogar os jogos criados pelos colegas na galeria do projeto (https://bit.ly/2X3COao) criando assim, momentos de diversão e aprendizado sobre temas específicos como ciência, história e conhecimentos gerais. Achados semelhantes são descritos no trabalho de [Aono *et al.* 2017]. Ao interagir com os jogos, alguns estudantes discutiram com os colegas sugestões de melhoria dos mesmos. Em um determinado momento, o estudante "A" dialogou com os membros de outra equipe e sugeriu: *"olha, esse tipo de jogo é melhor se você criar alternativas, tipo: A, B, C, D. Assim, ao invés do jogador digitar a resposta completa, ele apenas analisa a questão e digita ou clica na alternativa que achar correta"*.

Diante de situações como essa, notou-se o desenvolvimento do trabalho colaborativo entre os estudantes na construção dos jogos. Outro aspecto observado foi à motivação dos estudantes durante a execução dos projetos com o uso da ferramenta *Scratch*. Contudo, diferentemente do resultado encontrado por [Aono *et al.* 2017] houve em alguns momentos problemas com indisciplina e falta de atenção de alguns estudantes durante as aulas no laboratório de informática. Fatos isolados dessa natureza, possivelmente deram-se devido à grande quantidade de estudantes por turma (7º Ano "A" = 44 estudantes; 8º Ano "A" = 42 estudantes e 9º Ano "A" = 37 estudantes), dado que houve uma maior participação, atenção e interesse na turma com menos estudantes (7º Ano "B" = 14 estudantes).

Outro fator observado foi a motivação dos estudantes em querer continuar desenvolvendo novos jogos mesmo após a entrega dos projetos. Uma semana após a entrega dos trabalhos, muitos estudantes procuraram o professor no laboratório de informática para mostrar os novos jogos desenvolvidos individualmente com o ambiente de programação usando na disciplina. Esse interesse de continuar desenvolvendo jogos fora do ambiente escolar vai ao encontro dos resultados apresentados por [Wangenheim, Numes e Santos 2014], pois, de modo geral, notou-se que as atividades desenvolvidas com a ferramenta *Scratch* possibilitaram uma maior interação entre os estudantes, promoveu um ambiente aprendizagem lúdico e aquisição de habilidades relacionadas ao PC dos estudantes do Ensino Fundamental.

5. Considerações Finais

O presente trabalho relatou a utilização do ambiente de programação visual *Scratch 3.0* com 137 estudantes de 4 turmas do Ensino Fundamental II. Os principais resultados obtidos mostraram que, houve desempenho significativo em aspectos fundamentais para o desenvolvimento do PC, como: lógica, interatividade com o usuário, representação de dados e sincronização, bem como, enfatizam a necessidade de melhorar outras características essenciais do PC (paralelismo, controle de fluxo e abstração). Como trabalhos futuros, pretende-se desenvolver unidades instrucionais para o ensino de

conceitos de computação visando o desenvolvimento do PC, em domínios específicos do conhecimento para potencializar o raciocínio lógico, trabalho em equipe e a construção de soluções para resolver de problemas complexos do mundo real com estudantes de diferentes níveis educacionais.

Referências

Almeida, M. E. B., & Valente, J. A. (2019). Pensamento Computacional nas Políticas e nas Práticas em alguns Países. *Revista Observatório, 5*(1), 202-242.

Aono, A. H., Rody, H. V. S., Musa, D. L., Pereira, V. A., & Almeida, J. (2017). A utilização do scratch como ferramenta no ensino de pensamento computacional para crianças. In *25º Workshop sobre Educação em Computação (WEI 2017)* (Vol. 25, No. 1/2017). SBC.

Avila, C., Cavalheiro, S., Bordini, A., Marques, M., Cardoso, M., & Feijo, G. (2017, October). Metodologias de Avaliação do Pensamento Computacional: uma revisão sistemática. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)* (Vol. 28, No. 1, p. 113).

Bordini, A., Avila, C., Marques, M., Foss, L., & Cavalheiro, S. (2017). Pensamento computacional nos ensinos fundamental e médio: uma revisão sistemática. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)* (Vol. 28, No. 1, p. 123).

Brackmann, C. P. (2017). Desenvolvimento do pensamento computacional através de atividades desplugadas na educação básica.

Brasil, Ministério da Educação. (2018). Base Nacional Comum Curricular. Disponível em: http://basenacionalcomum.mec.gov.br/

Carlos, L., Godinho, J., & Gomide, J. (2018). Um Relato de Experiência da Escola de Verão de Programação para Crianças. In *Anais do Workshop de Informática na Escola* (Vol. 24, No. 1, p. 41).

CIEB, Centro de Inovação para a Educação Brasileira. (2019). Currículo de Referência em Tecnologia e Computação. Disponível em: http://curriculo.cieb.net.br/

Creswell, John W. (2010). Projeto de Pesquisa: métodos qualitativo, quantitativo e misto. 3ª ed. – Porto Alegre: Artmed.

Cruz Alves, N., Rodrigues, P. E., Borgatto, A. F., Gresse von Wangenheim, C., & Hauck, J. C. R. (2016). Ensino de Computação de Forma Multidisciplinar em Disciplinas de História no Ensino Fundamental--Um Estudo de Caso. *Revista Brasileira de Informática na Educação, 24*(3).

Gil, A. C. (2012). Métodos e técnicas de pesquisa social. 6ª ed. – São Paulo: Atlas.

Hinterholz, L., & Santos, W. O. (2017). Aprendizagem baseada em projetos: Relato de introduçao da lógica no ensino fundamental. In *Anais do Workshop de Informática na Escola* (Vol. 23, No. 1, p. 1154).

Iniciativa Computação na Escola. (2019). Unidade instrucional para ensinar programação de jogos de forma interdisciplinar. Disponível em: http://www.computacaonaescola.ufsc.br/?page_id=1476

Lüdke, M.; André, M. E. D. A. (2013). Pesquisa em educação: abordagens qualitativas. 2ª ed. – Rio de Janeiro: E.P.U.

Moreno-León, J., Robles, G. (2015). Dr. scratch: A web tool to automatically evaluate scratch projects. In *Proceedings of the Workshop in Primary and Secondary Computing Education,* pages 132–133. ACM.

Oro, N. T., Pazinato, A. M., Teixeira, A. C. (2016). Programação de computadores na educação: um passo a passo utilizando o Scratch. Passo Fundo: Ed. Universidade de Passo Fundo.

Ribeiro, R., Barcelos, T., Souza, A., & Silva, L. A. (2018). Mensurando o desenvolvimento do Pensamento Computacional por meio de Mapas Auto-Organizáveis: Comparação de métricas de complexidade de Software com Dr. Scratch e CT-Test. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação* (Vol. 7, No. 1, p. 609).

Rodriguez, C., Reis, R. C. D., Isotani, S. (2017). Recursos e estratégias para desenvolvimento e avaliação do Pensamento Computacional na escola. *Tecnologias, Sociedade e Conhecimento*, Campinas, v. 4, n. 1.

Rodriguez, C., Zem-Lopes, A. M., Marques, L., & Isotani, S. (2015). Pensamento Computacional: transformando ideias em jogos digitais usando o Scratch. In *Anais do Workshop de Informática na Escola* (Vol. 21, No. 1, p. 62).

Santana, S. J., & Oliveira, W. (2019). Jogos Educacionais como Ferramenta de Auxílio ao Processo de Alfabetização. In *Anais do Workshop de Informática na Escola* (Vol. 25, No. 1).

Santos, W. O., Silva, F. C., Hinterholz, L. T., Isotani, S., & Bittencourt, I. I. (2018) Computação Desplugada: Um Mapeamento Sistemático da Literatura Nacional. RENOTE, 16(2).

SB Ortiz, J., & Pereira, R. (2018). Um Mapeamento Sistemático Sobre as Iniciativas para Promover o Pensamento Computacional. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)* (Vol. 29, No. 1, p. 1093).

SBC, Sociedade Brasileira de Computação. (2019). Diretoria de Educação Básica. Disponível em: http://www.sbc.org.br/educacao/diretoria-de-educacao-basica

Scratch. (2019). Disponível em: https://scratch.mit.edu/

Torres, F., Junior, M., Dantas, R., Mascena, L., & Damasceno, A. (2018). Ensinando o pensamento computacional usando linguagens de programação no Alto Sertão da Paraíba. In *Anais do Workshop de Informática na Escola* (Vol. 24, No. 1, p. 660).

Valente, J. A. (2019). Pensamento Computacional, Letramento Computacional ou Competência Digital? Novos desafios da educação. *Revista Educação e Cultura Contemporânea, 16*(43), 147-168.

Wangenheim, C. G., Nunes, V. R., & dos Santos, G. D. (2014). Ensino de computaçao com scratch no ensino fundamental–um estudo de caso. *Revista Brasileira de Informática na Educação, 22*(3), 115-125.

Webber, C. G., Spindola, M. M., Otobelli, E. S., Giron, G. R., Dall, G., Poloni, L., ... & do Prado, M. D. F. W. (2016). Reflexões sobre O Software Scratch no Ensino de Ciências e Matemática. *RENOTE, 14*(2).

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33-35.