



Metodologia Para Ensino do Pensamento Computacional para Crianças Baseada na Alternância de Atividades Plugadas e Desplugadas

Methodology For Teaching Computational Thinking For Children Based On The Alternation Of Plugged And Unplugged Activities

Letícia Mara Berto
UFSCar - campus Sorocaba
leticia.mberto@gmail.com

Luciana Aparecida Martinez Zaina
UFSCar - campus Sorocaba
lzaina@ufscar.br

Tiemi Christine Sakata
UFSCar - campus Sorocaba
tiemi@ufscar.br

Resumo

Ensinar o pensamento computacional às crianças tem sido cada vez mais reconhecido nos últimos anos como uma atividade importante, principalmente nos países da Europa e nos Estados Unidos. No Brasil, a Sociedade Brasileira de Computação (SBC) recomenda que as habilidades de raciocínio computacional sejam trabalhadas desde estágios iniciais da educação. O contato com conceitos de pensamento computacional fornece experiências que podem encorajar as crianças a desenvolver habilidades diferentes, como o raciocínio lógico, a solução de problemas, reconhecimento de padrões, decomposição e generalização, e assim por diante. Este artigo apresenta uma metodologia, chamada MEPeCoC (Metodologia para Ensino de Pensamento Computacional para Crianças), que visa apoiar o ensino do pensamento computacional para as crianças. Ao executar atividades desplugadas e plugadas alternadamente e complementá-las com o desenvolvimento de um projeto, a MEPeCoC visa introduzir os fundamentos de forma sinérgica e incremental. A proposta foi avaliada em um estudo de caso com participação de alunos do ensino fundamental pertencentes a uma escola pública de Sorocaba - SP. Os resultados demonstraram que os elementos que compõem a metodologia trazem um grande potencial para auxiliar o ensino do pensamento computacional.

Palavras-Chave: *pensamento computacional; ensino de programação; computação desplugada*

Abstract

The teaching of computational thinking to children has become an increasingly important activity in recent years, especially in Europe and in the United States. In Brazil, the Brazilian Society of Computing (SBC) recommends that computational reasoning skills should be developed in the early stages of education. The contact with computational thinking concepts provides experiences that can encourage children to develop different abilities, such as logical reasoning, problem-solving, pattern recognition, decomposition and generalization. This article presents a methodology, called MEPeCoC, that aims to support the teaching of computational thinking to children. Alternating between unplugged and plugged activities and complementing them with the development of a project, MEPeCoC aims to introduce the fundamentals of CT skills in a synergic and incremental way. The proposal was evaluated in a case study with the participation of elementary school students from a public school in Sorocaba - SP. The results demonstrated that the elements that composed the methodology bring a great potential to aid the teaching of computational thinking.

Keywords: *computational thinking; programming teaching; unplugged computing*



1 Introdução

A Internet viabilizou a conexão e a comunicação das pessoas. *Smartphones, tablets* e computadores tornaram-se itens básicos de grande parte dos indivíduos, inclusive para as crianças. Porém, poucas pessoas (mesmo entre os adultos) conhecem como os computadores funcionam e as diversas habilidades que podem desenvolver utilizando-os.

Com a facilidade da internet, o avanço da tecnologia e o desenvolvimento de equipamentos cada vez mais sofisticados, temos como resultado uma sociedade cada vez mais moderna, o que requer o desenvolvimento de novas habilidades, entre elas o conhecimento e domínio de tecnologias. A Lei de Diretrizes e Bases (LDB), em seu Artigo 32, afirma que, além dos conhecimentos básicos como o pleno domínio da leitura, escrita e cálculo, é objetivo da formação básica do cidadão a compreensão do ambiente natural e social, do sistema político, da tecnologia, das artes e dos valores em que se fundamenta a sociedade. Com isso, torna-se necessário realizar mudanças no século XXI para o exercício pleno da cidadania (França, Santos, Almeida, & Amaral, 2014).

O pensamento computacional (em inglês, *Computational Thinking* – PC) foi apresentado por (Wing, 2006) e está associado ao processo cognitivo para resolução de problemas, design de sistemas e compreensão do comportamento humano, norteados por conceitos fundamentais da Ciência da Computação. As características do PC, descritas por (Wing, 2006) são:

- *Conceituar, não apenas programar*: não se refere somente a programar computadores, mas pensar de forma abstrata, em múltiplos níveis;
- *É uma habilidade fundamental*: não é uma habilidade mecânica ou utilitária, mas que permite a resolução de problemas através do computador;
- *É uma maneira de pensar*: forma que as pessoas utilizam para resolver problemas, não um modo de fazê-las agir como computadores;
- *Complementa e combina matemática e engenharia*: a ciência da computação é derivada do pensamento matemático, em seus fundamentos formais e do pensamento engenheiro na construção de sistemas do mundo real;
- *Gera ideias e não artefatos*: não necessariamente produz *hardware* e *software*, mas gera a oportunidade de abordar os conceitos de computação para resolver problemas do cotidiano e interagir com outras pessoas;
- *Para todos, em qualquer lugar*: pode ser útil para todas as pessoas.

Embora PC descreva o tipo de pensamento que os cientistas da computação utilizam, as habilidades desenvolvidas ao praticar seus conceitos podem ser usadas no cotidiano e são essenciais para os profissionais de qualquer área. No mundo atual, é imprescindível que todos tenham competências para resolver problemas de forma correta e eficiente (Wing, 2006).

O ensino do pensamento computacional na educação básica é um tema defendido tanto em trabalhos acadêmicos quanto em sociedades científicas (Oliveira, Souza, Barbosa, & Barreiros, 2014; Raabe et al., 2017). Fora do Brasil, o movimento é ainda mais forte, principalmente nos EUA e nos países da Europa, que já implementaram o ensino de computação no currículo (CSTA, 2005).

Este artigo apresenta a MEPeCoC (**M**etodologia para **E**nsino de **P**ensamento **C**omputacional para **C**rianças) que visa apoiar o ensino de pensamento computacional para crianças utilizando como base recursos de Ciência da Computação. A MEPeCoC intercala 3 tipos



de atividades: (i) atividades plugadas (que utilizam recursos digitais) para ensinar conceitos de pensamento computacional; (ii) atividades desplugadas (sem o uso de recursos digitais) para avaliar se o conceito foi aprendido e; (iii) desenvolvimento de um projeto (realizado em etapas) para relacionar e consolidar o aprendizado dos conceitos. No entendimento dos autores, não há uma metodologia que une essas três abordagens no formato proposto neste estudo. Como estudo de caso, a metodologia foi aplicada na forma de um curso de 10 horas-aula, para 32 alunos do 5º ano do Ensino Fundamental I. Em uma visão geral, notou-se que a proposta da MEPeCoC auxiliou a manter as crianças interessadas em resolver problemas de uma forma divertida e descontraída. Também foi possível observar que as atividades estimularam o desenvolvimento de habilidades como persistência, raciocínio lógico, abstração e colaboração.

As próximas seções estão organizadas da seguinte maneira: a Seção 2 apresenta os trabalhos relacionados, enquanto a Seção 3 descreve a metodologia proposta. A descrição da aplicação do curso e os resultados são mostrados nas Seções 4 e 5, respectivamente. Por fim, na Seção 6, as considerações finais são apresentadas.

2 Trabalhos Relacionados

O pensamento computacional (PC) é definido como um conjunto de habilidades focadas em fundamentos da Ciência da Computação que auxiliam na interpretação e solução de problemas de todas as áreas do conhecimento. Essa competência de "pensar computacionalmente" é considerada tão importante quanto as demais áreas como Matemática, Linguagens e Ciências (Raabe et al., 2017). Países como Inglaterra, Irlanda, Portugal e Israel incluíram em seus currículos o ensino de PC (Balanskat & Engelhardt, 2015). Nos EUA, as escolas seguem normas criadas pelas organizações Computer Science Teachers Association (CSTA) e International Society for Technology in Education (ISTE) para a inclusão de PC para todos os níveis de escolaridade. Além disso, existem iniciativas que produzem e distribuem materiais didáticos e ferramentas gratuitas para estimular a realização de atividades que promovem o raciocínio computacional nas crianças. Algumas das iniciativas desses países são: o projeto britânico Computing At School ou CAS (<https://www.computingschool.org.uk>), o CS Unplugged (<http://csunplugged.org>) da Nova Zelândia, a plataforma Code.org (<https://studio.code.org>), e a formação online Exploring Computational Thinking (<https://edu.google.com>), para docentes, da Google for Education.

No Brasil, existe um esforço da Sociedade Brasileira de Computação (SBC) e iniciativas de universidades para incluir PC no Ensino Básico. Os primeiros trabalhos existentes na literatura sobre ensino de PC no Brasil são, em sua maioria, focados em iniciar princípios da Computação no Ensino Médio por meio do ensino de linguagens de programação (Python, C, JavaScript, Logo, entre outros) e/ou uso de robótica (Lego e Arduíno). Recentemente, houve um crescimento de trabalhos que promovem a inclusão de PC no Ensino Fundamental e no Ensino Infantil (Santos, Araujo, & Bittencourt, 2018).

O ensino de programação, seja por linguagens de programação ou linguagens por blocos, com uso de jogos digitais, IDE ou Arduíno, para crianças ou jovens, é uma estratégia comum para estimular as habilidades do pensamento computacional (Aono, Rody, Musa, Pereira, & Almeida, 2017; Batista et al., 2015; Gomes, Melo, & Tedesco, 2016; Alves, Rodrigues, Borgatto, Wangenheim, & Hauck, 2016; Torezani, Chagas, & Tavares, 2013). Isso se deve ao fato de que a elaboração de uma solução computacional desenvolve intrinsecamente o raciocínio computacional, pois requer a capacidade de compreender e simplificar problemas, desenvolver o raciocínio lógico, analisar fatos, reduzir problemas em partes menores e gerenciáveis, entre outros procedimentos mentais. O estudo realizado por (Sousa, Silva, Raiol, Sarges, & Bezerra, 2015)



indica que os alunos que realizaram o curso de programação, ofertado pelos autores, melhoraram o desempenho na disciplina de matemática. Além disso, enquanto aprendem a programar desenvolvem habilidades cognitivas como abstração, comparação e análise crítica de soluções, como é relatado no estudo realizado por (Toresani et al., 2013), no qual os autores apresentam um ambiente criado especificamente para apoiar o processo de aprendizagem de programação em crianças do ensino fundamental.

O ensino de programação na Educação Fundamental também tem sido realizado com o uso da robótica educacional como mediadora. A robótica permite trabalhar em ambiente lúdico, desenvolve habilidades de coordenação motora e concentração, trabalho em equipe, além de motivar o aprendizado de programação (Martins, Brelaz, Nascimento, Alfaia, & Martins, 2016). Algumas competências do pensamento computacional que podem ser estimuladas pela robótica educacional são algoritmo, abstração, decomposição, depuração, simulação e paralelismo (Oliveira & Araújo, 2016). Apesar da robótica educacional ser um ótimo mecanismo para potencializar o desenvolvimento do pensamento computacional, seu uso é restrito às escolas que possuem o material (kits) apropriado.

Estudos apontam que o uso de jogos para o ensino de programação com crianças até 12 anos favorece a apresentação dos conteúdos e é muito eficaz em manter os estudantes engajados durante as aulas (Gomes et al., 2016). Os jogos podem ser utilizados como meios de condução do trabalho (procedimentos de como as atividades são conduzidas), como é apresentado por (Batista et al., 2015) que utilizou em sua oficina de programação uma abordagem que mesclou desafios e ambiente lúdico. Outra maneira de motivar os alunos é com o desenvolvimento de jogos usando uma linguagem de programação no formato de um projeto, ou seja, elaborando etapas com atividades relacionadas ao jogo. (Aono et al., 2017) realizaram um curso de 8 semanas com o objetivo de ensinar pensamento computacional e programação e ao fim de cada aula, os estudantes tinham a oportunidade de desenvolver subpartes de um jogo aplicando de forma prática o que foi aprendido em aula. O desenvolvimento de jogos também possibilita o aprendizado interdisciplinar, como realizado pelos alunos do 5º e 7º ano que criaram jogos sobre conteúdo de História (Alves et al., 2016).

Outra prática bastante utilizada para promover o pensamento computacional no Ensino Fundamental é o uso de atividades desplugadas (sem uso de recurso digital). Essas atividades não requerem laboratório de computadores e proporcionam ambiente lúdico que não dependem de materiais de alto custo. Os trabalhos identificados na literatura utilizam as atividades desplugadas como complementar às atividades plugadas. Por exemplo, (Machado et al., 2010) utilizaram atividades plugadas e desplugadas para preparar alunos para a Olimpíada Brasileira de Informática. (Sousa et al., 2015) utilizaram atividades desplugadas para introduzir conceitos iniciais de computação, antes das aulas plugadas de ensino de programação com Logo. Já o minicurso ofertado por (Araújo, Rodrigues, Silva, & Soares, 2015) utilizou a plataforma Scratch para ensino de programação e as atividades desplugadas foram usadas como meio de avaliação dos conceitos aprendidos.

Todos os trabalhos relacionados acima são focados em público do Ensino Fundamental, utilizaram atividades plugadas (programação, robótica ou ferramentas computacionais), desplugadas (quando a atividade não envolve uso de dispositivos digitais) e/ou desenvolvimento de projetos (os trabalhos encontrados relatam o desenvolvimento de jogos em etapas). A Tabela 1 resume as abordagens adotadas pelos artigos mencionados nesta seção.

Como pode-se observar na Tabela 1, todos os trabalhos investigados nesta seção têm foco no uso de atividades plugadas, sendo a maior parte focados ao ensino de programação. Alguns artigos relacionados utilizam a robótica como abordagem, o que pode impedir replicações das



experiências tendo em vista que nem todas as escolas têm condições de adquirir tais equipamentos. Não foram encontrados trabalhos que usam atividades desplugadas e desenvolvimento de projetos como abordagem de ensino. A proposta deste artigo se diferencia dos demais por propor uma metodologia que combina o uso de atividades desplugadas, plugadas e desenvolvimento de projeto como forma de estimular a participação dos alunos em aperfeiçoar suas habilidades do pensamento computacional. Além disso, essa metodologia se diferencia dos demais pela forma em que as atividades são aplicadas: ensino dos conceitos (atividades plugadas), avaliação formativa (atividades desplugadas), integração de todo conteúdo aprendido (projeto).

Tabela 1: Resumo das abordagens utilizadas pelos trabalhos relacionados.

Referência	Plugada	Desplugada	Projeto
Machado et al., 2010	✓	✓	
Sousa et al., 2015	✓	✓	
Batista et al., 2015	✓		
Martins et al., 2016	✓		✓
Gomes et al., 2016	✓		
E. J. S. d. Oliveira & Araújo, 2016	✓		✓
Alves et al., 2016	✓		
Araújo et al., 2015	✓	✓	
Aono et al., 2017	✓		✓
Torezani et al., 2013	✓		
MEPeCoC	✓	✓	✓

3 MEPeCoC: Metodologia para Ensino de Pensamento Computacional para Crianças

Este artigo propõe uma metodologia chamada MEPeCoC – Metodologia para Ensino de Pensamento Computacional para Crianças. A MEPeCoC intercala atividades desplugadas, plugadas e desenvolvimento de projeto, sendo que cada uma dessas abordagens tem um propósito diferente na metodologia.

As atividades plugadas utilizadas nesta metodologia têm o objetivo de introduzir o pensamento computacional por meio do ensino de programação. O recurso de programação foi escolhido pois permite a representação de uma solução para uma determinada situação na linguagem algorítmica, que é uma das competências fundamentais do PC (CSTA, 2012).

As atividades desplugadas são atividades que favorecem o ensino dos conceitos de computação sem o uso dos computadores (Bell, Witten, & Fellows, 2015). Essas atividades usam de recursos como papel, lápis, lousa, *post-its* para apresentar o raciocínio desenvolvido na solução de problemas tipicamente estudados na computação. Nesta proposta, as atividades desplugadas foram utilizadas como meio de reforçar conceitos aplicados nas atividades plugadas e como um mecanismo para avaliação formativa do aprendizado. A avaliação formativa tem um caráter de acompanhamento da evolução sem caráter classificatório, focando na eficácia do aprendizado (Zaina, Ruggiero, & Bressan, 2012). As atividades desplugadas permitem sanar as dúvidas individuais dos alunos e ajustar o ensino de acordo com a aprendizagem de cada aluno antes dele progredir para uma próxima atividade.



Como forma de consolidar os conhecimentos adquiridos a partir das atividades plugadas e desplugadas, esta metodologia inclui o desenvolvimento de um projeto (trabalho dividido em partes, realizado em grupo, divisão de tarefas, mais complexo e que demanda maior prazo comparado com as atividades desenvolvidas em aula). Diferente de uma metodologia baseada em projetos, onde ensino acontece a partir da proposta de um projeto (English & Kitsantas, 2013), na MEPeCoC o desenvolvimento do projeto funciona como um mecanismo para que os participantes desenvolvam a habilidade de relacionar os diferentes conceitos sobre pensamento computacional apreendidos por meio das atividades plugadas e desplugadas. Além disto, o desenvolvimento do projeto estimula a curiosidade e a criatividade, além de motivá-los a superar desafios e a trabalhar colaborativamente.

Todas as atividades que compõem a metodologia têm como base os conceitos e abordagens envolvidos para trabalhar o PC segundo a Barefoot (CAS, 2014). Neste trabalho, os conceitos são compreendidos como habilidades¹ a serem estimuladas no indivíduo que caracterizam o pensamento computacional. Os conceitos são:

- *Lógica*: capacidade de estabelecer e verificar fatos, fazer previsões;
- *Algoritmo*: capacidade de desenvolver uma sequência precisa de instruções para executar uma tarefa;
- *Decomposição*: capacidade de quebrar um problema ou sistema em suas partes;
- *Padrões*: capacidade em reconhecer padrões para fazer previsões, criar regras e resolver problemas;
- *Abstração*: capacidade de sintetizar fatos com o intuito de decidir o que é importante e ignorar detalhes desnecessários;
- *Avaliação*: capacidade em analisar diferentes fatores e critérios ao desenvolver sua solução.

As abordagens indicam as características chaves presentes em práticas de ensino que podem propiciar o pensamento computacional, sendo elas:

- *Mexer-se*: utilizar diferentes estratégias para promover diferentes experiências aos alunos;
- *Criar*: propiciar um ambiente prático para que seja possível o aluno planejar, executar e avaliar sobre um problema;
- *Depuração*: instigar a análise constante durante a construção de soluções sobre um problema e corrigir seus erros;
- *Perseverança*: incentivar a resiliência frente a um problema difícil para nunca desistir, ser determinado;
- *Colaboração*: fomentar a solução de problemas por meio de grupos e trabalho colaborativo para garantir o melhor resultado.

A Figura 1 ilustra a organização da MEPeCoC. Esta metodologia propõe que o ensino do pensamento computacional seja realizado em 10 intervenções (indicadas pelos números). Uma intervenção pode ser considerada um dia, uma semana ou ocorrência em dias alternados. As setas e a numeração da intervenção indicam dependências entre as atividades. As setas que ligam as

¹ As habilidades são aprendizagens relacionadas a diferentes conteúdos, ações, conceitos e processos (BNCC, 2017)



atividades desplugadas ao projeto não estão relacionadas com uma semana específica do projeto, indicando que estas influenciam o desenvolvimento do projeto como um todo.

A metodologia é flexível para se adaptar a diferentes formas temporais de aplicação. Na figura é possível observar os conceitos de Ciência da Computação e sua relação com os conceitos de pensamento computacional que são trabalhados a cada intervenção. Os símbolos em cada bloco indicam os conceitos e abordagens do PC utilizados em cada atividade. Para facilitar a explanação da proposta, será suposto a partir daqui que cada intervenção corresponde a uma semana.

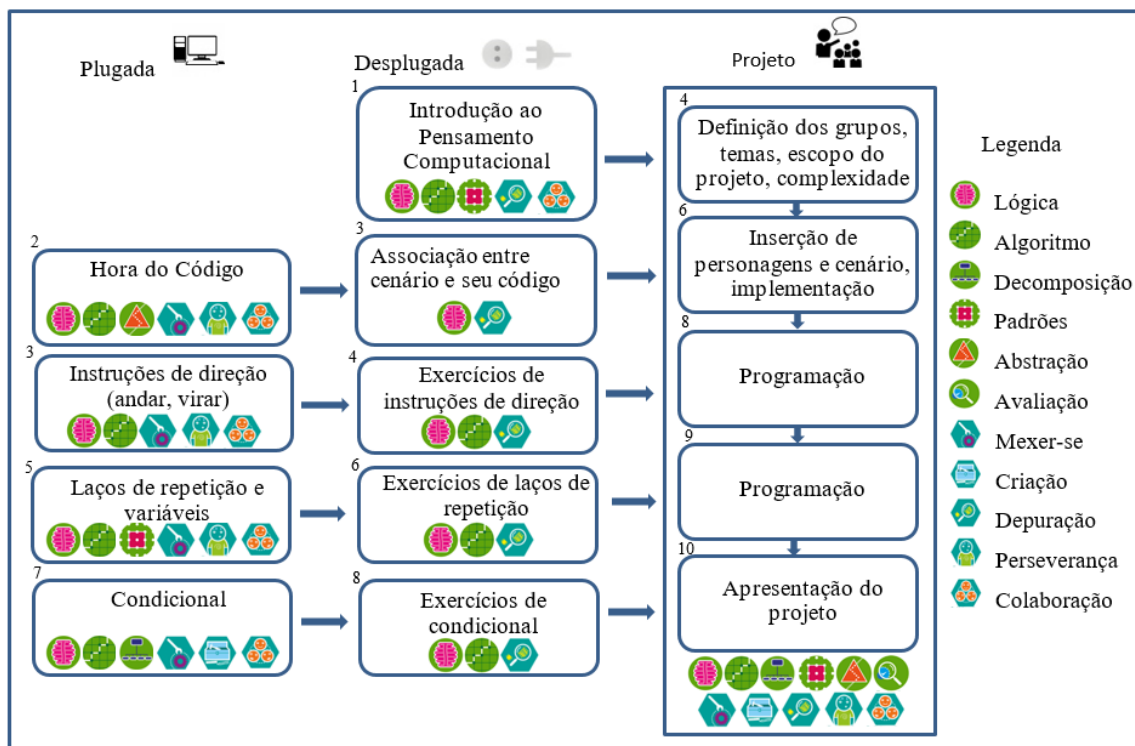


Figura 1: Esquema da metodologia MEPeCoC.

As atividades iniciam-se com atividades desplugadas (Semana 1). Essas atividades devem ser selecionadas para mostrar aos estudantes como o computador funciona e o que ele pode executar. É fundamental também introduzir conceito de algoritmos já nesse primeiro encontro. A atividade de programação é iniciada com a Hora do Código (HoC) na Semana 2. A metodologia é independente de plataforma, porém se sugere o uso de uma das atividades disponibilizadas pela **Code.org**, onde se encontram diversas práticas simples, com interface em português e que não requerem conhecimentos avançados para serem realizadas. Os principais conceitos de PC abordados são lógica, algoritmo, abstração; e as abordagens são mexer-se, perseverança e colaboração.

A partir da Semana 3, os participantes passam então a mergulhar na aprendizagem sobre os conceitos de PC. Os conceitos de programação introduzidos pela MEPeCoC são: (i) instruções de direção; (ii) laços de repetição; (iii) variáveis; (iv) condicional. Para trabalhar com tais conceitos, sugere-se o uso de Scratch, pois esse ambiente possui interface em português e dispõe de diversos materiais de apoio para o aprendizado. Isto se deve ao fato da linguagem já ser amplamente utilizada, principalmente fora do Brasil, pelas escolas que já incluem programação em seus currículos.

Todos os conceitos de programação são introduzidos por atividades plugadas (Semanas 2, 3, 5 e 7), seguidas por atividades desplugadas (Semanas 3, 4, 6 e 8) realizadas na semana seguinte.



O objetivo da atividade desplugada é identificar se o conteúdo que foi ensinado na semana anterior está sendo compreendido pelos participantes e identificar quais pontos são necessários revisar. As atividades plugadas são desenvolvidas em dupla, e a desplugada individualmente (com exceção da primeira, pois tem um foco diferente de introdução aos conceitos e não como forma avaliativa).

Em todas as atividades desplugadas, os principais conceitos e abordagens de PC vistos são lógica, algoritmo, depuração. Já nas atividades plugadas os conceitos de lógica e algoritmo são ensinados em todas as atividades. Adicionalmente, no ensino de laços de repetição trabalha-se o conceito de padrões, e em condicional, o conceito de decomposição. As abordagens mais comuns adotadas para execução das atividades plugadas são mexer-se e perseverança, sendo que no aprendizado de condicional também é necessária a abordagem criar.

Um dos diferenciais da MEPeCoC é a inclusão do desenvolvimento de um projeto de forma incremental e iterativa. Este não deve ser aplicado isoladamente ao final de todas as atividades, mas sim realizado de forma intercalada com o ensino dos conceitos de programação. Assim, é possível dar a oportunidade aos alunos de aplicar os conceitos aprendidos nas semanas anteriores, aprender novos conceitos e descobrir recursos oferecidos pela plataforma (sons, cenário, personagens, etc.). No projeto, os alunos têm a oportunidade de trabalhar todos os conceitos do PC, bem como todas as abordagens. Sugere-se que o projeto a ser desenvolvido tenha temática livre, desde que não viole a integridade e respeito dos alunos e das instituições envolvidas.

4 Estudo de caso: aplicando a MEPeCoC em um curso

Para avaliar a metodologia proposta foi realizado um estudo de caso. Para realização do estudo de caso foi planejado, criado e aplicado um curso baseando-se na MEPeCoC. O curso foi composto por 10 horas-aula, 1 hora por semana, e foi aplicado para o 5º ano da escola Escola Estadual Arquimínio Marques da Silva, na cidade de Sorocaba - SP. A aplicação do curso contou com a ajuda voluntária de 6 alunos do curso de Ciência da Computação da Universidade Federal de São Carlos - campus Sorocaba em parceria com o projeto de extensão *Computing to You!(C2Y!)*². A equipe foi coordenada por um professor doutor pesquisador na área de PC.

Primeiramente foi divulgado o curso na escola para todas as turmas de 5º ano, como atividade extracurricular no horário contraturno dos alunos. Essa abordagem fez com que somente aqueles que tivessem interesse e curiosidade manifestassem interesse em participar, pois eles ficariam horas a mais na escola. As inscrições e o termo de consentimento dos pais foram gerenciados e coletados pela escola.

Das 3 turmas de 5º ano que continham aproximadamente 30 alunos, um total de 32 alunos realizaram inscrição no curso, sendo que 40,62% dos alunos eram meninas e 59,37% eram meninos.

Em uma pesquisa informal foi verificado que todos os alunos possuíam algum conhecimento em tecnologia (uso de computadores, celular, *tablets* e videogames), porém nem todos afirmaram ter esses aparelhos em casa. Além disso, apenas um aluno possuía experiência prévia com programação, enquanto os demais não tinham conhecimento sobre o assunto.

² “Computing to You!(C2Y!)”. <http://uxleris.sor.ufscar.br/c2y/index.html>

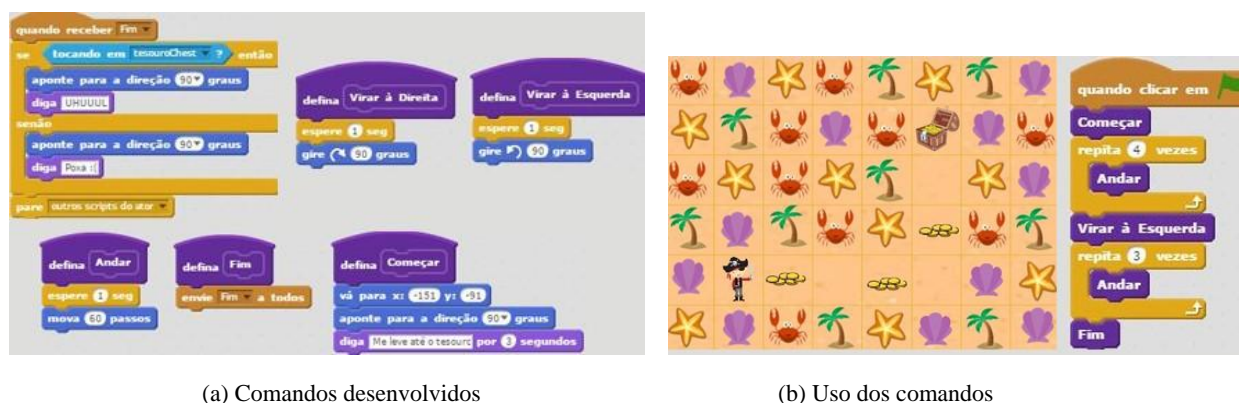


Para o desenvolvimento das atividades desplugadas, utilizou-se como base o material da Computer Science without a computer³. Para as atividades plugadas foram usadas as plataformas Code.org⁴ e Scratch⁵. Para que fosse possível analisar os dados e gerar os resultados, todas as atividades plugadas foram gravadas por meio de captura de telas, usando o software *FastStone Capture*⁶.

Na primeira aula, utilizaram-se duas atividades desplugadas. Uma das atividades teve foco em desenvolver algoritmos por meio do uso de imagens⁷ e a outra na detecção de erros usando paridade⁸. Assim, os principais conceitos e abordagens do PC apresentados foram lógica, algoritmo, padrões, depuração e colaboração.

Para a hora do código foi utilizado o Code.org, pois as atividades disponibilizadas por essa plataforma têm o propósito de iniciar o contato das pessoas com programação a partir de atividades que podem ser realizadas em uma hora. A atividade “Aventureiro de Minecraft”⁹ foi escolhida devido à ludicidade e grande aceitação entre o público infantil desta plataforma.

As aulas de programação foram realizadas com o uso da linguagem Scratch. Para reduzir a complexidade de opções de ação e construção de cenários do Scratch e para dar foco ao ensino de cada conceito como proposto na MEPeCoC, foram criados pela equipe que conduziu o curso, cenários e comandos (blocos) específicos. Os exercícios propostos em Scratch para o ensino desses conceitos estão disponíveis para acesso¹⁰. A Figura 2(a) mostra alguns comandos criados e a Figura 2(b) ilustra um cenário criado e os comandos necessários para concluir o exercício proposto.



(a) Comandos desenvolvidos

(b) Uso dos comandos

Figura 2: Exemplo de Cenário e comandos desenvolvidos.

³ "CS Unplugged." <https://csunplugged.org/>. Accessed 25 Sep. 2018.

⁴ "Code.org." <https://code.org/>. Accessed 25 Sep. 2018.

⁵ "Scratch - Imagine, Program, Share." <https://scratch.mit.edu/>. Accessed 25 Sep. 2018.

⁶ "FastStone Screen Capture - The Best Screen Capture Software." <http://www.faststone.org/FSCaptureDetail.htm>. Accessed 25 Sep. 2018.

⁷ "Programming Languages - Computer Science Unplugged." <https://classic.csunplugged.org/programming-languages/>. Accessed 25 Sep. 2018.

⁸ "Error Detection - Computer Science Unplugged - CS Unplugged." <https://classic.csunplugged.org/error-detection/>. Accessed 25 Sep. 2018.

⁹ "Minecraft Hour of Code - Studio Code.org." <https://studio.code.org/s/mc/reset>. Accessed 25 Sep. 2018.

¹⁰ <https://scratch.mit.edu/users/cursoufscar/>



As atividades desplugadas, aplicadas após cada aula plugada de introdução de novos conceitos, eram realizadas logo no início da aula, de forma individual e com tempo estabelecido pelos membros da equipe que conduziam a atividade. Todas as atividades eram impressas e possuíam cenários e personagens de acordo com os gostos das crianças. Essas atividades estão disponíveis para acesso em <https://goo.gl/v9UyuR>. É importante ressaltar que o curso foi planejado de forma simples e organizada o suficiente para que os professores da escola possam reproduzir o curso e/ou dar continuidade na atividade posteriormente.

A equipe foi responsável por abrir a sala, ligar e fazer *login* nos computadores, arrumar a sala, ligar o projetor, instalar os programas de gravação de telas nos computadores. A condução das aulas plugadas foi realizada por 2 membros da equipe, sendo que um realizava a explicação do conteúdo e outro demonstrava a execução por meio de um projetor multimídia para as crianças. Enquanto isso, os demais membros da equipe executora percorriam a sala para auxiliar os participantes e esclarecer dúvidas. Contudo, os membros da equipe somente agiam quando requisitados pelas crianças, deixando assim que elas promovessem a descoberta sozinhas primeiramente.

Nas aulas de programação, a falta de um computador por aluno (16 computadores para 32 alunos) propiciou a aplicação da prática de *pair programming*, que consiste em programar em uma equipe de duas pessoas em um único computador, sendo que um membro da dupla geralmente é o motorista, que controla o teclado e o *mouse*, e o outro é o navegador, que analisa possíveis problemas e tem em mente como tudo se encaixa (Prottsman, 2014). O desenvolvimento das atividades em dupla também permitiu a possibilidade de discussão sobre uma determinada solução e os próprios alunos faziam questão de revezar entre motorista e navegador, de forma que ambos trabalhassem nos exercícios. Após as explicações, os alunos realizavam os exercícios referentes à aula com o acompanhamento de um monitor, membro da equipe responsável. A Figura 3 ilustra alguns desses momentos.



Figura 3: Realização das atividades plugadas e desplugadas.

Cada monitor supervisionava, em média, duas duplas de alunos. Além de auxiliar os alunos nas atividades, os monitores observavam as dificuldades e dúvidas dos alunos e realizavam anotações em papel sobre o comportamento deles. Ademais, a equipe gravou as telas produzidas pelos alunos que continham as soluções dos exercícios plugados para futuras avaliações.

Durante a execução do curso, foram coletados dados quantitativos e qualitativos em que os membros da equipe realizaram a observação da interação do aluno e de suas reações (Lazar, Feng, & Hochheiser, 2010). Para apoiar a observação os membros da equipe realizavam anotações sobre dificuldades encontradas, problemas que apareciam durante o desenvolvimento da atividade, reações de euforia ou tristeza observadas durante as atividades. Essas anotações eram



feitas individualmente pelos membros da equipe durante as sessões das atividades e eram consolidadas em um relatório semanal sobre o curso.

5 Resultados

Os resultados analisados nesta seção são baseados em dados coletados durante o curso como anotações escritas, gravações em vídeo da interação dos participantes, desempenho nas atividades plugadas, desplugadas e no desenvolvimento do projeto.

Os relatórios de observações semanais indicam que os alunos eram receosos no início do curso e não faziam perguntas quando tinham dúvidas, pois ficavam com vergonha. Esse fato pode ser concluído devido às expressões faciais apresentadas pelos alunos e pelo fato de perguntarem novamente os conceitos recém explicados de forma particular aos monitores. Percebendo tal fator, os professores faziam perguntas sobre o conceito recém-explicado e pelas respostas (ou falta delas) re-explicavam caso necessário. No decorrer do curso, os alunos se sentiram mais à vontade e participaram ativamente, tirando dúvidas e respondendo as perguntas. Outro ponto interessante foi em relação ao fato dos alunos se ajudarem mutuamente quando não compreendiam o assunto, de forma que o conceito fosse explicado por um aluno com linguagem e exemplos mais próximos deles.

As expressões mais comuns eram de dúvida ao iniciar um exercício, desafio e por vezes tristeza ao errar a solução do exercício e felicidade ao acertar. Quando acertavam o exercício costumavam bater palmas, pular, comemorar levantando os braços, dançar e dizer em alto tom palavras de comemoração (“uhuul”, “ae”, “isso”). O progresso dos alunos em relação à compreensão do conteúdo e acompanhamento das atividades propostas foi melhor do que o esperado, o que nos permitiu ter aulas mais dinâmicas e participativas.

5.1 Atividades plugadas

A atividade Hora do Código Minecraft, disponível pelo Code.org, foi analisada separadamente, pois esta foi uma atividade de aquecimento e teve como objetivo fazer o primeiro contato dos alunos com a programação e não o ensino de um conceito específico (Lazar et al., 2010). Das 14 fases propostas por essa atividade, todos os alunos completaram ao menos 8 delas, sendo que 4 alunos completaram 12 fases e 2 alunos completaram 13 fases. Tendo em vista que as crianças tiveram cerca de 40 minutos e não uma hora como é a proposta da atividade (os outros 20 minutos foram usados para a organização da sala e apresentação da plataforma), esses resultados demonstram que as crianças tiveram facilidade com o uso da plataforma e entenderam o funcionamento de encaixe de blocos e a execução dos comandos de cada bloco.

Nas aulas das atividades plugadas seguintes houve uma variação no número de alunos presentes, pois por ser no contraturno alguns alunos precisavam comparecer a outros compromissos previamente agendados (consultas médicas, por exemplo). Assim, para a análise dos resultados destas atividades, decidiu-se selecionar oito amostras aleatórias de cada exercício. Cada amostra equivale ao resultado alcançado por uma dupla de alunos. É importante ressaltar que nem todas as amostras iniciaram ou terminaram todos os exercícios (por falta de tempo ou desistência). As aulas das atividades plugadas correspondem às aulas das Semanas 3, 5 e 7 e o número de exercícios propostos para essas aulas foi de 3, 7 e 9 exercícios, respectivamente.

Foi analisada a aplicação dos seguintes conceitos pelos alunos: direção (InstD), laços de repetição (LRep), variáveis (Var) e condicional (Cond). A Tabela 2 apresenta o resultado para



cada um dos conceitos e descreve, para cada exercício proposto¹¹, a quantidade de amostras que finalizaram o exercício (NumFin), quantidade de amostras que não tentaram (NumNTentou, devido a falta de tempo) e o desempenho das amostras por conceito. Todos os valores são considerados utilizando o universo das oito amostras analisadas.

O desempenho (D), relacionado ao grau de dificuldade dos estudantes, foi calculado pela Equação 1, onde NumDificuldade é a quantidade de alunos que mostraram alguma dificuldade para resolver o exercício, isto é, cometeram algum erro no uso do conceito, mesmo tendo chegado, em algum momento, na solução correta. NumNTentou é o número de amostras que não chegaram a tentar nenhuma vez resolver o exercício. Assim, esta equação não considera a dificuldade de execução de amostras que não executaram determinado exercício.

Tabela 2: Resultados das atividades plugadas por conceito.

Exercício	NumFin	NumNTentou	InstD	LRep	Var	Cond
3.1	7	1	☹️	-	-	-
3.2	5	1	☹️	-	-	-
3.3	3	4	☹️	-	-	-
5.1	8	0	😊	😊	-	-
5.2	8	0	☹️	😊	-	-
5.3	7	0	☹️	-	-	-
5.4	4	4	-	😊	😊	-
5.5	8	0	☹️	😊	😊	-
5.6	8	0	-	😊	😊	-
5.7	6	1	☹️	☹️	😊	-
7.1	5	2	😊	☹️	😊	-
7.2	5	2	☹️	😊	😊	-
7.4	7	0	☹️	☹️	-	😊
7.5	6	1	☹️	☹️	-	😊
7.6	4	4	☹️	😊	-	☹️
7.7	4	4	-	😊	😊	😊
7.8	4	4	-	☹️	-	😊
7.9	3	5	☹️	😊	😊	😊

$$D = \left(1 - \frac{\text{NumDificuldade}}{8 - \text{NumNTentou}}\right) * 100 \quad (1)$$

Com o valor D obtido, foi feita a classificação seguindo as faixas determinadas pelos autores como parâmetros para avaliação de desempenho em cada atividade:

- 😊 = 65% ou mais das amostras não tiveram dificuldade;
- ☹️ = dificuldade média (36% a 64% das amostras tiveram alguma dificuldade);

¹¹ A numeração dos exercícios descrita neste artigo está relacionado com a semana em que cada exercício foi realizado. Já no estúdio (repositório público) do Scratch, os exercícios iniciam no número 3, que é referente à semana da primeira aula plugada, e a partir disso segue sequencialmente.



- 😞 = 35% ou menos das amostras não tiveram alguma dificuldade.

A Tabela 2 mostra que a maioria dos alunos teve dificuldade com o conceito de lateralidade. Essa dificuldade também foi observada no trabalho de (Raabe et al., 2015). Esse conceito foi introduzido na Semana 3 e foi explorado em praticamente todos os exercícios propostos. Acredita-se que as imagens estáticas utilizadas nos cenários do Scratch não auxiliaram na compreensão de direção e por isso, a maioria não conseguia acertar o exercício sem errar. Apesar de tudo, pode-se notar que a maioria das amostras (média de 73,21%) conseguiu finalizar corretamente as atividades propostas.

O conceito de repetição, introduzido na Semana 5, foi explorado em todos os exercícios propostos, com exceção do 5.3, a partir dessa aula. O número de finalizações corretas foi elevado (diferentemente do resultado obtido por (Gomes, Falcão, & Tedesco, 2018), no qual os alunos preferiam usar os mesmos comandos diversas vezes), porém em praticamente todos os exercícios houve dúvidas. A taxa de amostras que não tentaram foi baixa na aula introdutória do conceito (exercícios 5.1 a 5.7), mas aumentou na aula seguinte, possivelmente por misturar este conceito de repetição com outros conceitos.

O conceito de variável, introduzido na Semana 5, foi explorado em pelo menos metade dos exercícios após ensiná-lo. Com exceção de um exercício, ao menos metade das amostras conseguiu finalizar as atividades. Além disso, em praticamente todos os exercícios, os alunos apresentaram alguma dificuldade, embora esse número tenha sido baixo.

O conceito de condicional foi introduzido na Semana 7. O número de amostras que finalizou os exercícios foi menor que em outras aulas. Muitos não conseguiram nem tentar resolver os últimos exercícios (7.6 a 7.9). Porém, as amostras que tentaram apresentaram poucas dificuldades nessa aula, e várias destas conseguiram solucionar os exercícios na primeira tentativa.

É importante ressaltar que em cada nova aula, um novo conceito foi introduzido, ou seja, o curso introduziu os conceitos de maneira a tornar a aprendizagem incremental. De uma semana para outra os participantes trabalhavam com um conjunto de conceitos para resolução de problemas. Porém, o número de exercícios que puderam ser trabalhados aumentou a cada aula, o que indica que os alunos, no decorrer do curso, foram desenvolvendo habilidades para usar a ferramenta e para o desenvolvimento do raciocínio algorítmico.

Além desses dados, foram analisados o número médio de blocos (comandos) usados (NumBlUs) Vs. número de blocos esperados (NumBlEsp) das amostras de cada exercício. É importante ressaltar que para esta análise, foram consideradas apenas as amostras que finalizaram o exercício correspondente. Como pode-se observar na Tabela 3, os exercícios 3.1, 7.4, 7.5, 7.6, 7.7 apresentam número médio de blocos usados inferior à quantidade mínima de blocos necessários para realizar o exercício. Ao observar os vídeos foi possível notar que alguns alunos encontraram outra forma de fazer o exercício (alguns tiraram personagens do cenário, outros encontraram modos de alterar o código base para funcionar conforme eles desejavam). Essas soluções não foram avaliadas como incorretas, tendo em vista que os alunos usaram a criatividade para resolver o problema, mesmo que não tenha sido da forma esperada. No exercício 7.8 é possível notar uma diferença muito grande entre o número de blocos usados (22,5) e esperados (9), que pode ser explicada devido ao fato do NumBlEsp ter sido contabilizado utilizando laços de repetição, e os alunos não utilizaram o conceito para fazer o exercício.

A análise das atividades plugadas permitiu concluir que uma das principais dificuldades foi em relação ao conceito de direção (lateralidade). Os demais conceitos foram executados sem



muita dificuldade, o que comprova que os alunos evoluíram e amadureceram durante o curso, buscando alcançar melhor desempenho mesmo em problemas mais complexos.

5.2 Atividades desplugadas

As atividades desplugadas foram utilizadas para analisar a compreensão de cada aula plugada proposta e eram aplicadas na semana seguinte à realização de uma atividade plugada. Todas as atividades usaram diferentes cenários (níveis de complexidade, temática, enredo). A Tabela 4 mostra o resultado da correção dos exercícios de cada atividade desplugada aplicados após cada atividade plugada. Essa tabela descreve o número total de alunos que estavam presentes no dia do exercício, o número de alunos que resolveram o exercício corretamente e o número de alunos que usaram o conceito corretamente (mesmo sem conseguir chegar na solução correta do exercício), e o tempo total que foi permitido usar para todos os exercícios do conceito.

A atividade desplugada referente à Hora do Código (HoC) foi composta por 2 exercícios de associação, um deles para depurar um código e as opções de diferentes cenários, e para o outro vários códigos com seus respectivos cenários. Praticamente todos os alunos conseguiram resolver os exercícios.

Tabela 3: Relação do número médio de blocos utilizados e esperado.

Exercício	NumBIEsp	NumBIUs
3.1	13	12,71
3.2	19	19,00
3.3	12	12,00
5.1	7	7,62
5.2	20	20,12
5.3	9	9,00
5.4	6	6,00
5.5	9	9,12
5.6	5	5,00
5.7	9	15,00
7.1	14	14,60
7.2	15	15,40
7.4	18	17,57
7.5	23	22,23
7.6	22	21,50
7.7	12	11,75
7.8	9	22,50
7.9	18	19,33

Tabela 4: Resultado da correção dos exercícios desplugados.

	Número alunos	Exercício 1		Exercício 2		Exercício 3		Tempo (minutos)
		Correto	Conceito	Correto	Conceito	Correto	Conceito	
HoC	26	26	26	25	25	-	-	10
InstD	25	1	19	5	16	6	17	15
LRep	25	2	19	5	5	7	15	15
Cond	28	25	25	-	-	-	-	10



A atividade desplugada de direções (InstD) foi composta por 3 cenários em níveis gradativos de dificuldade, para os quais era necessário escrever os algoritmos de solução. Nesse caso, os alunos tiveram como principal dificuldade escrever as instruções, pois o sentido de direção estava relacionado à direção da visão do personagem e não dos alunos. Assim, muitos erraram a instrução de direita e esquerda em algum momento, não conseguindo finalizar os exercícios. Porém, a maioria conseguiu usar o número de avanços corretamente, conseguindo usar um dos conceitos ensinados. Apesar do exercício 3 ser considerado o mais difícil pelo número de instruções relacionadas, nota-se que os alunos foram entendendo o raciocínio e obtiveram melhor desempenho (Tabela 4).

Nos exercícios de laço de repetição (LRep), o problema continuou sendo a falta de compreensão de direção. Os alunos demonstraram entender o funcionamento da instrução de repetição, mas frequentemente erravam na direção do personagem, não conseguindo alcançar o objetivo. Apenas 2 alunos resolveram corretamente os 3 exercícios e a maioria conseguiu usar de forma correta a instrução de repetição. O erro mais comum, próprio deste conceito, foi no número de repetição. O melhor desempenho ocorreu no exercício 3, sendo que 28% da turma resolveu corretamente o exercício e 60% dos alunos usaram corretamente a instrução de LRep.

A atividade desplugada sobre condicional ocorreu na 8ª Semana e os alunos se mostraram treinados e não tiveram dificuldade com as instruções de direção. Neste caso, 89,2% dos alunos resolveram corretamente o exercício na primeira tentativa (conclusão por observação). Os 3 alunos restantes não utilizaram o conceito de condicional, mas demonstraram ter entendido o conceito de direção. Com isso, é possível concluir que o uso de atividades desplugadas como forma de avaliação formativa permite acompanhar o desenvolvimento das habilidades de forma específica e personalizada com cada aluno, possibilitando compreender os pontos que necessitam de reforço, além de trabalhar o conteúdo já visto por uma abordagem diferente e que motiva os alunos com novos desafios.

5.3 Desenvolvimento do projeto

O desenvolvimento do projeto iniciou na Semana 4 (de acordo com a MEPeCoC, Seção 3, Figura 1) e foi intercalado com as demais atividades. O desafio foi de construir um jogo de tema livre. O tema jogo foi um motivador importante para as crianças. O projeto foi realizado por grupos de até 6 alunos e, para cada grupo, foi designado um líder, membro da equipe pesquisadora, para acompanhar as atividades. Os grupos discutiram o tema, o enredo e escopo do jogo. Para a programação do jogo, eles se dividiram em subgrupos, os quais trabalhavam focados em determinado assunto (pesquisar personagens e cenários, programar trocas de cenário, programar funcionalidade de cada personagem, pontuação do jogo, criação do enredo, desenvolver diferentes fases do jogo) e depois explicavam o que haviam feito para os demais, de forma que todos aprendessem. No projeto, os alunos tiveram a oportunidade de unir os conceitos aprendidos em aula e novos conceitos que eram ensinados pelo seu líder.

Os jogos foram analisados a partir da leitura dos códigos e execução dos mesmos. Para cada jogo foi analisado se os conceitos ensinados em aula foram utilizados, tomando como referência os blocos existentes em cada seção no Scratch. Para cada conceito, foi analisado se havia a presença dos seguintes blocos:

- *Instruções de direção* (InstD): Blocos da seção “Movimento”, que correspondem às posições do personagem (girar, andar, deslizar, etc);



- *Laços de repetição* (LRep): Blocos da seção “Controle”, que contêm blocos com laços de repetição (sempre, repita x vezes, etc).
- *Variáveis* (Var): Blocos da seção “Variáveis”, que contêm blocos que permitem criar e manipular variáveis.
- *Condicional* (Cond): Blocos da seção “Controle”, que contêm blocos com condições (Se x então, Se x então - senão).

Os resultados obtidos são apresentados na Tabela 5. Como pode-se observar, todos os conceitos ensinados em aulas de atividades plugadas foram utilizados pelos grupos, com exceção do grupo com o projeto “Pokemon”, que não utilizou variáveis. Como consta na Figura 1 ao trabalhar com instruções de direção, os alunos desenvolveram lógica e algoritmos; em laços de repetição e variáveis puderam exercitar lógica, algoritmos e padrões; e por fim, em condicional puderam aperfeiçoar lógica e algoritmos além de desenvolver a decomposição.

Tabela 5: Resultados dos projetos por conceitos ensinados em aulas plugadas.

Ensinado no curso	Caça ao tesouro	Global offensive	Pokemon	Perdido na Ilha	Suicide Squad	PacMan Vs. Goblin
InstD	✓	✓	✓	✓	✓	✓
LRep	✓	✓	✓	✓	✓	✓
Var	✓	✓		✓	✓	✓
Cond	✓	✓	✓	✓	✓	✓

A proposta de desenvolvimento do projeto trouxe outras vantagens ao permitir que os alunos pudessem explorar a ferramenta e criar situações que não haviam sido previamente abordadas ou ensinadas. Os alunos tiveram a capacidade de aprender sozinhos (alguns com pouca ajuda dos líderes) novos recursos de programação e com isso desenvolver ainda mais o pensamento computacional. A Tabela 6 contém esses conceitos, que foram verificados a partir da existência dos seguintes blocos:

- *Paralelismo*: Blocos da seção “Eventos” e “Sensores”, que correspondem a eventos possíveis de ocorrerem (ex: quando receber mensagem, executar o mesmo trecho de código para vários personagens, etc);
- *Interatividade com o usuário*: Blocos da seção “Sensores” e “Aparência”, que contêm blocos com entrada (teclado, mouse, webcam, microfone) e saída (tela) (ex: quando a tecla y for pressionada, mostrar uma frase na tela);
- *Sincronização*: Blocos da seção “Eventos” e “Controle”, que contêm blocos que permitem enviar mensagens e controlar a execução de *scripts* respeitando as dependências;
- *Função*: Blocos da seção “Controle” e “Mais blocos” que possibilitam a criação de novos comandos (blocos) e clones (criar um novo personagem em tempo de execução igual ao já existente).

Tabela 6: Resultados dos projetos por conceitos não ensinados em aulas plugadas.

Não ensinado no curso	Caça ao tesouro	Global offensive	Pokemon	Perdido na Ilha	Suicide Squad	PacMan Vs. Goblin
Paralelismo	✓	✓	✓	✓	✓	✓



Interatividade com o usuário	✓	✓	✓	✓	✓	✓
Sincronização	✓			✓	✓	✓
Função		✓				

Como pode-se observar, todos os grupos utilizaram Paralelismo e Interatividade com o usuário; com exceção de dois grupos, todos utilizaram Sincronização e apenas um fez uso de Funções. Durante a análise dos projetos foi identificada também a junção de blocos previamente ensinados com usos não ensinados, como: condicionais e laços de repetição aninhados e com operações lógicas, trabalhar com coordenadas x e y, envio de mensagens, entrada de teclado e mouse. Esses conceitos avaliados, apesar de não terem sido ensinados como conteúdo das aulas e sim por cada líder, apresentaram bons resultados em sua utilização, trazendo indícios de que os alunos conseguem transpor os limites do que já haviam aprendido. Esses resultados demonstram que os alunos de fato aprenderam os conceitos e que possuem capacidade de aplicá-los em projetos e não somente na aula em que é ensinado o assunto. Como o aprendizado é incremental, é possível concluir que no final do curso, os alunos estavam amadurecidos em conhecimento de algoritmos e programação.

Além da correção a partir do código, também foram utilizadas na análise, as anotações feitas pelos líderes dos grupos, que são sumarizados na Tabela 7.

Tabela 7: Observações sobre os projetos desenvolvidos.

Nome	Tema	Visão do líder
Caça ao tesouro	Guerreiro tem a missão de combater os zumbis.	Grupo com facilidade em programação. Trabalharam de forma incremental no projeto. Todos os alunos faziam questão de trabalhar na implementação e se ajudavam nas dúvidas.
Global Offensive	Guerreiro tem a missão de combater os zumbis.	Grupo tinha certeza do tema desde o início do projeto, muito dedicados e focados. Em todas as aulas de projetos já sabiam o que deveria ser implementado e ficavam bastante envolvidos para finalizar.
Pokemon	Pokebolas estão soltas pelo centro de treinamento e o Charizard precisa desviar delas.	Alunos bem dedicados, de fácil consenso entre eles. A única dificuldade foi em relação à comparação constante do seu projeto e com os dos demais grupos.
Perdido na Ilha	Um avião caiu em uma ilha e os sobreviventes se separaram para encontrar recursos. Precisam colher frutas, ferramentas e combater o vilão que derrubou o avião.	O grupo queria implementar um jogo como os existentes com grande poder gráfico e computacional, não entendendo que com os recursos e tempo disponível não seria possível. Por ser um grupo bastante heterogêneo, foi difícil fazer com que os alunos entrassem em consenso.
Suicide Squad	Arlequina precisa chegar até o Coringa. Para isso ela precisa desviar dos inimigos e suas armadilhas.	Grupo focado, com facilidade em tomada de decisões e de fácil consenso. Não apresentou dificuldades para desenvolver o projeto e conseguiram implementar tudo o que foi planejado.
PacMan Vs. Goblin	O PacMan caiu em uma caverna de Goblins e precisa combater-los, para isso conta com seus poderes.	Grupo com várias ideias diferentes de mesma temática no início. Entraram em consenso de forma fácil e fizeram incrementos de acordo com o tempo disponível. Teve dificuldade para evitar conversa, atrapalhando o desenvolvimento do projeto e a comunicação entre eles.



Na última aula ocorreram as apresentações dos projetos (Figura 4). Estavam presentes os alunos do curso, a equipe envolvida, incluindo a coordenadora da atividade, a coordenadora e a diretora da escola, além da supervisora da diretoria de educação da cidade. Os alunos apresentaram seus projetos junto com o líder responsável, responderam as perguntas feitas pela coordenação do colégio e relataram como foi o curso. A direção da escola demonstrou satisfação com os resultados e solicitou uma nova edição do curso para o próximo ano, incluindo também o ensino para os professores. Os alunos também pediram para dar continuidade ao curso, devido ao fato de ter promovido um conhecimento diferenciado e muito divertido para eles.



Figura 4: Apresentações dos projetos finais.

6 Discussão e Conclusões

Após a realização do estudo de caso pode-se concluir que a MEPeCoC auxiliou na introdução gradativa dos conceitos de computação, possibilitando maior compreensão dos alunos, mesmo que estes não tenham conhecimento prévio do assunto. O encadeamento do conteúdo permite que os assuntos façam sentido e sejam melhores relacionados pelos alunos, promovendo com isso o aprendizado evolutivo. O uso de atividades de diferentes formatos possibilita ao aluno diversas maneiras de explorar o conhecimento e colocá-lo em prática.

O curso proposto para avaliar a metodologia foi aplicado em uma escola com envolvimento do público-alvo. Durante as aulas foi possível trabalhar conceitos de PC por meio de atividades desplugadas, plugadas e desenvolvimento de um projeto, promovendo a interdisciplinaridade e trabalhando habilidades relacionadas à solução de problemas, trabalho em grupo, comunicação e perseverança. Cada tipo de atividade utilizada teve uma contribuição diferente para o curso. Os conceitos foram introduzidos com as atividades plugadas, nas quais os alunos puderam colocar em prática as instruções ensinadas e analisar a correção das soluções. As atividades desplugadas contribuíram para o aprendizado construtivo, raciocínio lógico e abstração, fazendo com que os alunos pensassem e depurassem a solução do exercício, sem o apoio de uma ferramenta que executasse o seu código. O projeto permitiu trabalhar todos os conceitos vistos em aula de forma mais profunda, além de possibilitar o aprendizado de novos conceitos. Nele, os alunos puderam explorar a criatividade, a detecção e correção de erros, trabalho em grupo, tomada de decisão, definição de escopo, prazos e narrativa.

O curso aplicado com os estudantes apresentou resultados positivos e motivadores para a sua replicação. Notou-se que assunto que mais tiveram dificuldade (lateralidade) não estava relacionado diretamente a computação, mas sim da dificuldade do conceito no cotidiano. Os assuntos diretamente ligados a computação, como variáveis, laço de repetição e condicional, que envolvem maior poder de abstração, não foram problemas para a maioria dos estudantes, que compreenderam os conceitos rapidamente.



A escola participante do projeto demonstrou grande satisfação com os resultados obtidos e a metodologia utilizada, e segundo a coordenação, apesar do curso ser de curto prazo, o conhecimento adquirido foi extenso. A diretora da escola ainda afirma que o projeto deveria apresentar continuidade, de forma a possibilitar que todas as escolas tenham a oportunidade da experiência. Os relatos de alguns alunos que participaram do curso mostram que observaram melhorias principalmente em 2 aspectos. Primeiro, notaram que é mais fácil decompor um problema de matemática para resolvê-lo. Segundo, enxergaram que organizar ideias e sequenciá-las facilita na escrita de redação.

Trabalhar com as crianças foi uma nova experiência para o grupo, que precisou aprender a trabalhar com todos os níveis de aprendizado dos alunos, comportamento, prazeres, motivação, grau de liberdade e a melhor maneira de corrigi-los. No decorrer do curso algumas dificuldades foram encontradas, pois nenhum dos voluntários havia trabalhado com ensino para crianças. No início não era fácil ter a atenção e respeito dos alunos, que conversavam durante as explicações e não tinham foco para realizar as atividades. Também houve muita dificuldade em relação ao comportamento das crianças, que são bastante curiosas e faziam várias perguntas não relacionadas ao tema da aula e que às vezes deixavam os monitores desconfortáveis. O grupo recebeu bastante auxílio, inclusive a presença da coordenação da escola e da intervenção da professora orientadora durante algumas aulas. Apesar das dificuldades, foi uma experiência de muito aprendizado pessoal e profissional, o que levou aos voluntários o desejo de aplicar uma nova edição do curso.

Ao final do curso, foram entregues certificados para os 29 alunos que tiveram ao menos 75% de presença. Isso mostra que houve uma taxa muito baixa de evasão. Além disso, consta que 100% das meninas finalizaram o curso, enquanto 84,21% dos meninos finalizaram, o que indica o interesse do público feminino pela área.

A evasão do curso foi de 9,37%, o que demonstra que, mesmo sendo no período contraturno, o curso foi bem aceito pelos alunos. Esse número traz evidências de que a metodologia aqui apresentada pode auxiliar a incentivar as crianças e promover o conhecimento sobre PC de forma incremental. Contudo, é importante salientar que a MEPeCoC foi avaliada dentro do contexto nacional, no qual o ensino de Ciência da Computação e Pensamento Computacional não faz parte do currículo escolar.

A principal contribuição deste trabalho é a proposta da MEPeCoC que une de forma sinérgica e flexível a realização de atividades plugadas e desplugadas com o desenvolvimento de projetos como mecanismo auxiliar para o ensino de PC e motivar os alunos. Contudo três outras contribuições particulares são importantes de serem destacadas. A primeira refere-se a incorporação de atividades desplugadas como forma de avaliação formativa. Esta prática potencializa uma maior qualidade no processo de ensino, já que permite identificar falhas rapidamente ao longo do processo e com isso sanar as dúvidas específicas dos alunos. Como consequência, eliminam-se barreiras que podem atrapalhar o processo de ensino. Uma segunda contribuição é a descrição do curso que foi aplicado para avaliação da MEPeCoC. Este curso pode ser replicado em outras instâncias. Por último, é necessário ressaltar o detalhamento realizado na análise dos resultados permitindo que outros trabalhos que tenham o mesmo propósito utilizem a mesma forma de avaliação de dados.



Agradecimentos

Agradecemos o apoio financeiro concedido pela Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) ao projeto através do processo nº 2016/11866-7, aos alunos voluntários de graduação da UFSCar campus Sorocaba para a aplicação do curso e à direção da Escola Estadual Arquimínio Marques da Silva pela oportunidade.

Referências

- Alves, N. D., Rodrigues, P. E., Borgatto, A. F., Wangenheim, C. G. V., & 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), 31–46. doi: [10.5753/RBIE.2016.24.03.31](https://doi.org/10.5753/RBIE.2016.24.03.31) [GS Search]
- Andrade, S. I., Mendes, P., Correa, D. A., & Zaine, M. F. (2011). Conflict of generations in the workplace: a study conducted in the public sector. In *Proceedings of 9^o International Conference on Information Systems and Technology Management (9^o CONTECSI)*. [GS Search]
- 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 *Proceedings of XXV Workshop sobre Educação em Computação (WEI'17)*. [GS Search]
- Araújo, D. D. C., Rodrigues, A. N., Silva, C. V. D. A., & Soares, L. S. (2015). O Ensino de Computação na Educação Básica apoiado por Problemas: Práticas de Licenciandos em Computação. In *Proceedings of XXIII Workshop sobre Educação em Computação (WEI'15)*. [GS Search]
- Balanskat, A., & Engelhardt, K. (2015). *Computing our future: Computer programming and coding. priorities, school curricula and initiatives across europe*. European Schoolnet. Retrieved from <http://www.eun.org/resources/detail?publicationID=661>
- Batista, W. P., Chalegre, H. S., Sena, J. P. S., Santos, A. H. M., Santos, D. M. B. d., Rodrigues, C. A., & Bittencourt, R. A. (2015). Oficinas de Aprendizagem de Programação em uma Escola Pública através do Ambiente Scratch. In *Proceedings of XXIII Workshop sobre Educação em Computação (WEI'15)*. Retrieved from <http://www.lbd.dcc.ufmg.br/colecoes/wei/2015/032.pdf>
- Bell, T., Witten, I. H., & Fellows, M. (2015). *Cs unplugged - ensinando ciência da computação sem o uso do computador*. Retrieved from <http://csunplugged.org/>.
- BNCC. Base Nacional Comum Curricular: Educação é a Base. Brasília: Ministério da Educação, 2017. Retrieved from http://basenacionalcomum.mec.gov.br/wp-content/uploads/2018/06/BNCC_EI_EF_110518_-versaofinal_site.pdf (pp. 28-29).
- CAS, B. (2014). *Computational thinking concepts*. Retrieved from <http://barefootcas.org.uk/barefoot-primary-computing-resources/concepts/>.
- CSTA. (2005). The new educational imperative: Improving high school computer science education. Final report of the CSTA. Curriculum improvement task force. In *Acm*. Retrieved from <https://cse.sc.edu/~buell/References/StudentRecruiting/CSTA-WhitePaperNC.pdf>



- CSTA. (2012). The csta standards task force. csta k-12 computer science standards, 2011. In *New york: Acm computer science teachers association*. Retrieved from <http://csta.acm.org/Curriculum/sub/K12Standards.html>
- English, M. C., & Kitsantas, A. (2013). Supporting student self-regulated learning in problem and project-based learning. *Interdisciplinary Journal of Problem-Based Learning*, 7(2). [[GS Search](#)]
- França, R. S. d., Santos, V. A., Almeida, L. C. F., & Amaral, H. J. C. (2014). A disseminação do pensamento computacional na educação básica: lições aprendidas com experiências de licenciandos em computação. In *Proceedings of XXXIV Congresso da Sociedade Brasileira de Computação (CSBC'14)* (pp. 1505–1514). [[GS Search](#)]
- Gomes, S., Carina, T., Falcão, T. P., & Tedesco, P. C. d. A. R. (2018, June). Exploring an approach based on digital games for teaching programming concepts to young children. *International Journal of Child-Computer Interaction*, 16, 77–84., from <https://linkinghub.elsevier.com/retrieve/pii/S2212868917300338> doi: [10.1016/j.ijcci.2017.12.005](https://doi.org/10.1016/j.ijcci.2017.12.005)
- Gomes, T. C. S., Melo, J. C. B. d., & Tedesco, P. C. d. A. R. (2016). Jogos digitais no ensino de conceitos de programação para crianças. In *Proceedings of XXVII Simpósio Brasileiro de Informática na Educação (SBIE'16)* (pp. 470–479). [[GS Search](#)]
- Lazar, J., Feng, J. H., & Hochheiser, H. (2010). *Research methods in human-computer interaction* (1st ed.). John Wiley & Sons Ltd. [[GS Search](#)]
- Machado, E. Z. d. A., Vasconcelos, I. R., Amorim, K. M., Andrade, A. M. S., Barreto, L. P., & Santosand, D. A. (2010). Uma Experiência em Escolas de Ensino Médio e Fundamental para a Descoberta de Jovens Talentos em Computação. In *Proceedings of XVIII Workshop sobre Educação em Computação (WEI'10)* (pp. 798–807). Retrieved from <http://www.lbd.dcc.ufmg.br/colecoes/wei/2010/004.pdf>
- Martins, L. A. d. S., Brelaz, A. d. S., Nascimento, G. R., Alfaia, R. M., & Martins, T. D. S. (2016). Ensinando lógica de programação aplicada à robótica para alunos do ensino fundamental. In *Proceedings of XXVII Simpósio Brasileiro de Informática na Educação (SBIE'2016)* (pp. 31–41). [[GS Search](#)]
- Oliveira, E. J. S. d., & Araújo, A. L. S. O. d. (2016). Pensamento computacional e robótica: Um estudo sobre habilidades desenvolvidas em oficina de robótica educacional. In *Proceedings of XXVII Simpósio Brasileiro de Informática na Educação (SBIE'16)* (pp. 530–539). [[GS Search](#)]
- Oliveira, M. L. S., Souza, A. A., Barbosa, A. F., & Barreiros, E. F. S. (2014). Ensino de lógica de programação no ensino fundamental utilizando o scratch: um relato de experiência. In *Proceedings of XXXIV Congresso da Sociedade Brasileira de Computação (CSBC'14)* (pp. 1525–1534). [[GS Search](#)]
- Prottzman, K. (2014, November). *3 best practices for pair programming*. Retrieved from <https://www.iste.org/explore/articleDetail?articleid=221>.
- Raabe, A., Rodrigues, A. J., Santana, A., Vieira, M., Rosário, T., & Carneiro, A. C. (2015). Brinquedos de programar na educação infantil: Um estudo de caso. In *Xxi workshop de informática na escola (wie'15)*. [[GS Search](#)]



- Raabe, A. L., Frango, I., Ribeiro, L., Granville, L. Z., Salgado, L., Cruz, M. J. K. d., . . . Fortes, S. (2017, July). Referenciais de Formação em Computação: Educação Básica. *Sociedade Brasileira de Computação - SBC*. Retrieved from <http://www.sbc.org.br/files/ComputacaoEducacaoBasica-versaofinal-julho2017.pdf>
- Santos, P., Araujo, L., & Bittencourt, R. (2018, 07). A mapping study of computational thinking and programming in brazilian k-12 education. In *48th annual frontiers in education conference (fie' 18)*. [[GS Search](#)]
- Sousa, A., Silva, S., Raiol, A. A. C., Sarges, J., & Bezerra, F. (2015). O Universo Lúdico da Programação de Computadores com Logo no Ensino Fundamental. In *Proceedings of XXIII Workshop sobre Educação em Computação (WEI'15)*. [[GS Search](#)]
- Torezani, C., Chagas, L. B. C., & Tavares, O. d. L. (2013, November). NewProg - um ambiente online para crianças aprenderem programação de computadores. Retrieved from <http://www.br-ie.org/pub/index.php/wie/article/view/2637>. doi:[10.5753/CBIE.WIE.2013.140](https://doi.org/10.5753/CBIE.WIE.2013.140)
- Wing, J. M. (2006, March). Computational Thinking. *Communications of the ACM*, 49(13), 33–35. [[GS Search](#)]
- Zaina, L. A. M., Ruggiero, W., & Bressan, G. (2012). Metodologia para acompanhamento da aprendizagem através da web. *Revista Brasileira de Informática na Educação*, 12(1), 20–28. [[GS Search](#)]