

## Incentivos ao desenvolvimento do Pensamento Computacional no Ensino Médio: aprendendo a programar

Fernanda Pires<sup>1,2</sup>, Rafaela Melo<sup>1</sup>, Fábio Michel<sup>1</sup>, João Bernardo<sup>1</sup>, Marcela Pessoa<sup>1,2</sup>, Ricardo S Barboza, Elaine H. T. Oliveira<sup>2</sup>

<sup>1</sup> Universidade do Estado do Amazonas (UEA)  
Av. Darcy Vargas, 1.200 – 69050-020 – Manaus – AM – Brasil

<sup>2</sup>Instituto de Computação (IComp) – Universidade Federal do Amazonas (UFAM)  
Av. General Rodrigo O. Jordão Ramos, 1200 – 69067-005 – Manaus – AM – Brasil

{fpires,rmf.lic16,fmmdl.lic16,jrsbr.lic16,msppessoa,rsbarboza}  
}@uea.edu.br, elaine@icomp.ufam.edu.br

**Abstract.** *This paper presents a didactic sequence for the development of Computational Thinking through programming learning. The learning theory Constructionism was used with a methodological approach of creative learning. The validation of the sequence was done by a case study with 30 high school students from a public school for 6 months. Concepts of programming logic were explored using approaches such as Unplugged Computing, Scratch and Python programming, and Robotics through gamification. The results showed that the activities with the highest level of playfulness were better received by the students.*

**Resumo.** *Este artigo apresenta uma sequência didática para o desenvolvimento do Pensamento Computacional através da aprendizagem de programação. Como teoria de aprendizagem foi utilizado o Construcionismo com uma abordagem metodológica de fundamentação criativa. Para validar a sequência, foi realizado um estudo de caso em que participaram 30 alunos do ensino médio de uma escola pública, durante 6 meses. Foram explorados conceitos de lógica de programação utilizando abordagens como Computação desplugada, programação com Scratch e Python e Robótica através de gamificação. Os resultados evidenciaram que as atividades com maior nível de ludicidade foram mais bem recebidas pelos alunos.*

### 1. Introdução

Neste século XXI, a Computação e o cotidiano estão cada vez mais entrelaçados. Sistemas computacionais influenciam do entretenimento aos cuidados básicos de saúde e segurança, reforçando a importância de introduzir conceitos computacionais ainda na fase escolar (Brackmann, 2017). No Brasil, a introdução desses conceitos se confunde com cursos de informática básica, em que os estudantes aprendem a manipular ferramentas de edição de textos, imagens ou apenas utilizam os computadores para realizar pesquisas escolares (Vicari *et al.*, 2018). Inserir a Computação na educação básica vai muito além disso, envolve Pensamento Computacional (PC), organização algorítmica e sobretudo a visão de que se pode resolver problemas, sendo um protagonista direto da ação (Resnick, 2017).

Papert (1980) foi pioneiro na visão de que a inserção de computadores na escola poderia trazer bons resultados. Discussões sobre o processo de resolução de problemas e conceitos computacionais aconteceram ao longo do tempo (Brackmann, 2017) e ganharam fôlego com o discurso de Wing sobre o Pensamento Computacional (Wing, 2006). Algumas alternativas de introdução a aprendizagem de programação tem se apresentado de forma lúdica, seja com interfaces que buscam auxiliar na redução da complexidade de interpretação, como o Scratch, Game Maker, Alice e Kodu; desenvolvimento de jogos; aplicações de robótica; ou em linguagens de alto nível, como Python e Java (Grover e Pea, 2013). Neste sentido, este trabalho relata a implementação de uma sequência didática para o desenvolvimento do Pensamento Computacional através da aprendizagem de programação, aplicada a estudantes do Ensino Médio. Foram adotadas como estratégias: Computação desplugada, programação com Scratch e Python e robótica gamificada. A proposta foi aplicada a estudantes do 1º ao 3º ano do ensino médio, em uma escola da rede pública da cidade de Manaus.

O artigo está organizado como segue: a Seção 2 apresenta os trabalhos relacionados; a Seção 3 a fundamentação teórica; a Seção 4 descreve a proposta e o design experimental aplicado; na Seção 5 são analisados e discutidos os resultados da pesquisa; e na Seção 6 estão as considerações finais.

## 2. Trabalhos Relacionados

Apesar de existirem diversos trabalhos que buscam inserir o Pensamento Computacional na educação básica, ainda não se conhece um padrão ou forma considerada correta para isso. Weintrop *et al.* (2018) investigaram, durante três anos, abordagens para apoiar a aprendizagem de Ciência da Computação no Ensino Fundamental. Através de um currículo projetado e um ambiente de programação, os autores introduzem conceitos da Computação, fazendo uso de um software semelhante ao Scratch para apresentar a programação em blocos. Ao fim do currículo, os estudantes envolvidos desenvolveram um projeto de cunho livre com *design* próprio. Os autores avaliaram as estruturas utilizadas para a construção do código e a criatividade. Já Duncan e Bell (2015) apontam que a aprendizagem de Ciência da Computação e programação são novas para o ensino pré-superior, por isso há tanta investigação sobre a melhor maneira de incentivar o desenvolvimento do Pensamento Computacional para essa fase. Os autores concentraram suas pesquisas em escolas primárias, baseadas nos currículos inglês e australiano. Eles propõem tópicos a serem abordados de acordo com o público escolhido, e assim montam seu próprio currículo.

Bezerra *et al.* (2017) propõem incentivar o Pensamento Computacional, em alunos do ensino médio, através da técnica de computação desplugada. Abordaram o tema de conversão de números binários. A atividade durou 2 horas, com a participação de 19 alunos. Como resultados, os autores destacam que os estudantes tiveram boa aceitação da atividade e do método utilizado. Já Aono *et al.* (2017) propõem desenvolver o Pensamento Computacional em alunos do ensino fundamental usando o Scratch. Participaram do projeto 20 alunos com faixa etária entre 10 e 11 anos, com duração de 8 aulas. Semelhante ao que aconteceu neste trabalho, os autores começaram com explicações teóricas sobre conceitos computacionais (algoritmos, procedimentos, variáveis, estruturas de decisão, de repetição e processamento de Strings), seguidas de atividades práticas de programação para construção de um jogo. Os autores concluíram

que os estudantes obtiveram êxito, pois foram capazes de construir o jogo, evidenciando que o Scratch pode ser bem utilizado para introdução a programação.

Pereira e Franco (2018) propõem desenvolver o Pensamento Computacional em jovens através de curso de introdução a lógica de programação usando Scratch e Arduino. Participaram 20 estudantes dos últimos anos do ensino fundamental. O curso baseou-se na apresentação de conceitos básicos de software e hardware, seguido de criação de algoritmos com Scratch e construção de projetos com Arduino. Ao fim do curso, os estudantes apresentaram um projeto em uma feira de ciências, de forma a consolidar os conhecimentos adquiridos durante o processo.

Este trabalho utiliza abordagem semelhante às descritas nesta seção, com o diferencial de propor um conjunto diverso de atividades aplicado em um período maior.

### **3. Pensamento Computacional: construindo soluções de forma criativa**

Desenvolver o Pensamento Computacional exige criatividade, pois trata de desenhar soluções usando diferentes estruturas. O pensamento criativo é uma das habilidades essenciais aos indivíduos do século XXI e uma abordagem adequada para que esta habilidade seja desenvolvida é a *kindergarten approach to learning* (abordagem do jardim de infância à aprendizagem) (Resnick e Robinson, 2017).

A proposta de sequência didática aqui apresentada foi fundamentada na Teoria Construcionista de (Papert, 1980) e em aspectos da aprendizagem criativa defendida por Resnick (2017). O Construcionismo tem suas raízes no construtivismo de Piaget (1976) e defende que o indivíduo pode criar suas estruturas cognitivas a partir da interação com o meio, sem a interferência de agentes externos. Assim como Piaget, Papert acredita que os seres humanos devem ser ativos no processo de aprendizagem, construindo suas estruturas intelectuais através da construção de suas ferramentas e isso se dá pelo uso da criatividade.

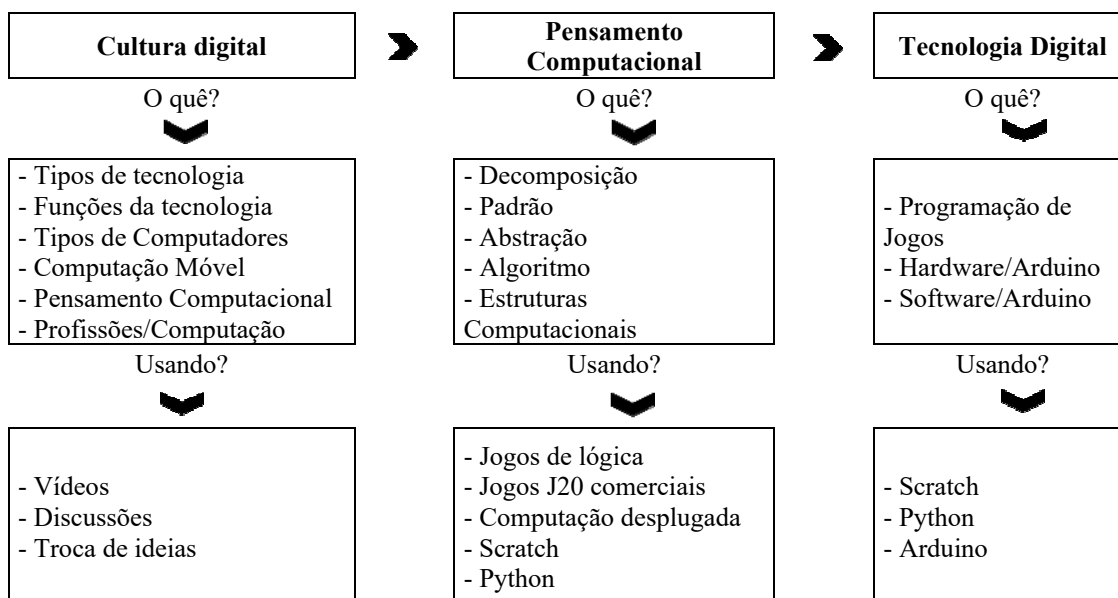
Embora não exista um conceito consolidado, pesquisadores concordam que Pensamento Computacional diz respeito a um conjunto de habilidades que permite resolver problemas usando estratégias utilizadas em Ciência da Computação (Wing, 2008). O Pensamento Computacional se sustenta em 4 pilares: i) decomposição: identificação e redução de um problema em partes menores para facilitar a compreensão; ii) reconhecimento de padrões: verificar se uma solução que atende a uma parte pode ser repetida em outras situações dentro de um problema macro através de pontos que se repetem; iii) abstração: manter o foco no que é importante e excluir o que pode ser descartado e; iv) algoritmo: conjunto de passos que compõe uma solução replicável (Liukas, 2015; Raabe *et al.*, 2018).

### **4. Propostas de estratégias para promover a aprendizagem de programação**

Com o objetivo de incentivar o desenvolvimento do Pensamento Computacional em estudantes do Ensino Médio, foram elaboradas diferentes estratégias para a promoção da aprendizagem de Programação de Computadores e desenvolvimento do Pensamento Computacional. A proposta de sequência didática utiliza uma abordagem imersiva, em estilo *top-down*, baseada no protagonismo dos estudantes e alicerçada no Construcionismo através de uma abordagem criativa. A sequência segue a linha Cultura Digital, Pensamento Computacional e Tecnologia Digital, perfazendo os três eixos de

conhecimento propostos pela Sociedade Brasileira de Computação (SBC) em suas Diretrizes para ensino de Computação na Educação Básica<sup>1</sup>, conforme Figura 1.

A proposta foi dividida em três partes, como segue:



**Figura 1. Divisão da sequência conforme os eixos de conhecimento propostos pela SBC.**

**Etapa I:** utilizou-se abordagem imersiva, aproveitando o conhecimento dos alunos. Foram realizadas rodas de conversas de forma a ambientar os participantes no universo digital e na discussão sobre a criação de produtos utilizados no cotidiano.

**Etapa II:** busca desenvolver os pilares do Pensamento Computacional através da resolução de problemas de lógica em que os indivíduos devem criar formas de explicitar suas estratégias, utilizando diferentes recursos, como: sequência de charadas, jogos, Computação desplugada, resolução de problemas com Scratch e Python.

**Etapa III:** espaço com maior abertura criativa, em que são desenvolvidos produtos baseados nos conhecimentos adquiridos nas etapas anteriores. Os estudantes devem projetar um jogo a ser desenvolvido em Scratch, partindo do projeto de *game design*. As decisões devem ser discutidas com uso de raciocínio lógico e crítico tanto pelo tema do jogo, quanto pelas estruturas computacionais selecionadas. Em seguida, os estudantes veem, em atividade prática, como se dá a comunicação entre hardware e software através da montagem, configuração e programação de kits de Arduino, utilizando uma abordagem gamificada. Os processos são guiados tendo como base a Teoria de Aprendizagem Construcionista e a aplicação do ciclo de aprendizagem criativa imaginar-criar-brincar-compartilhar-refletir, proposto por Resnick (2017).

<sup>1</sup> Disponível em <https://www.sbc.org.br/educacao/diretrizes-para-ensino-de-computacao-na-educacao-basica>

## 5. Aplicação da sequência didática

A sequência didática foi aplicada a alunos da Escola Estadual Aline Salerno. Participaram das atividades, 30 estudantes, sendo 17 meninas e 13 meninos, do 1º ao 3º ano do Ensino Médio com idade entre 15 e 19 anos, durante seis meses. Os estudantes tiveram aulas 3 vezes por semana, por duas horas, como parte de um projeto de extensão da Universidade do Estado do Amazonas. As atividades foram realizadas por alunos e professores do curso de Licenciatura em Computação que alternaram o local entre a universidade e a escola, conforme ilustra a Figura 2.

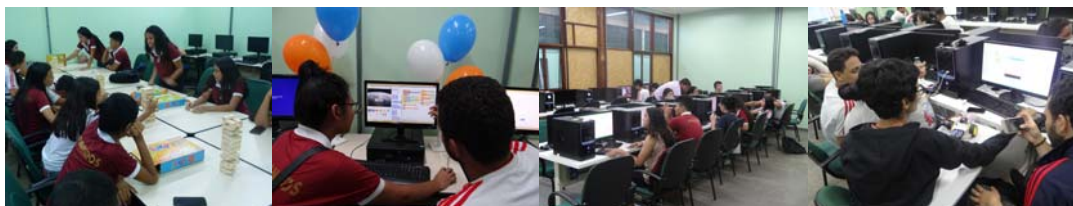


Figura 2 - Alunos durante as etapas do projeto

### 5.1 Introdução à Programação e a Lógica

Foram discutidos temas relacionados a tecnologias digitais, partindo do papel do computador na sociedade. Os estudantes expressaram suas opiniões e houve construção coletiva sobre a função dos computadores e as diferentes formas que eles poderiam assumir. Discutiu-se sobre profissões ligadas à Computação e como as tecnologias digitais ajudam as pessoas a resolverem problemas, chegando à definição de habilidades e competências necessárias a esses profissionais. Em seguida, foram ministradas aulas de Introdução à Lógica de Programação, abordando conceitos como: visualização de problemas, estratégias de divisão e conquista, algoritmos, conceito de programação e linguagens. As atividades duraram aproximadamente 1 mês e estão descritas na Tabela 1.

Tabela 1. Atividades aplicadas na Etapa I

Atividades	Descrição da Atividade
Charada	<b>Fazendeiro</b> - desafio: o fazendeiro atravessar um lobo, uma ovelha e uma alface de um lado para outro do rio. O lobo não pode ficar sozinho com a ovelha e a ovelha não pode ficar com a alface e o fazendeiro só pode transportar um de cada vez no seu barco.
Escrita de Algoritmos	<b>Algoritmos</b> - os alunos receberam uma folha com 6 diferentes desenhos que representavam uma ação. Desafio: descrever uma sequência de passos que concretiza a ação, por exemplo, "baixar um aplicativo" ou "escovar os dentes".
Computação desplugada	<b>Pixels</b> - dado um conjunto de imagens representadas por pixels em matrizes de diferentes tamanhos, escrever um algoritmo com símbolos (setas e linhas) que preencha a matriz e represente a imagem originalmente. (Disponível em: <a href="https://code.org/">https://code.org/</a> )
Computação desplugada	<b>Cupcakes</b> - desafio: analisar uma sequência de cupcakes, completos ou incompletos. Descrever um algoritmo para concluir os <i>cupcakes</i> . Se o <i>cupcake</i> estiver completo, representar com "-", se faltar massa e cobertura, escrever "massa+cobertura" e se faltar só cobertura, "cobertura". (Disponível em: <a href="http://computacional.com.br/">http://computacional.com.br/</a> )

Os estudantes ficaram livres para construir soluções, descrever o processo de resolução do problema e tiveram à sua disposição matérias de desenhos e massa de modelar para realizar a visualização das soluções, o que auxiliou na localização de falhas e otimização das soluções. Foram trabalhados os 4 pilares do PC em que os estudantes foram instigados a dividir os problemas, localizar semelhanças entre problemas, excluir as informações desnecessárias, até conseguirem descrever a sequência de passos que retorna bons resultados.

## 5.2 Programação em Scratch e Python

Essa etapa foi dividida em duas subetapas. A primeira abordou a aprendizagem de estruturas computacionais usando programação em blocos (Scratch). Durante as aulas, os estudantes discutiram coletivamente seu aprendizado e tinham tarefas a fazer em casa. A cada encontro postavam suas soluções em uma sala de aula virtual e levavam para sala de aula para compartilhar. A avaliação da evolução deu-se por meio do desenvolvimento de um projeto de criação de um jogo. Algumas aulas foram destinadas ao desenvolvimento dos projetos. Essa fase do processo está descrita na Tabela 2.

**Tabela 2. Sequência didática da Etapa II – Scratch**

Atividades	Descrição
Apresentação da Plataforma Scratch	Visão geral, explicando o papel dos atores, palco, blocos de programação, fantasias, e suas respectivas funcionalidades.
Eventos e Movimento	
Caneta	Em cada uma dessas aulas foram abordados os diferentes grupos de blocos de programação, através de exemplos práticos e exercícios, além disso, os alunos tinham aulas apenas para tirar dúvidas sobre o conteúdo.
Controle e Sensores	
Variáveis	
Strings e Listas	
Game Design	Foi ministrada uma aula sobre <i>Game Design</i> , para mostrar aos alunos o processo de desenvolvimento de um jogo para que posteriormente eles pudessem desenvolver seu próprio jogo em Scratch.
Projeto Final	Para a finalização dessa etapa, foi proposto um jogo como projeto final em Scratch para ser realizado em duplas. Os jogos obtidos foram apresentados na Semana Nacional de Ciência e Tecnologia.

Na segunda subetapa foram ministradas aulas de programação com a linguagem de programação Python, com os conteúdos: leitura e escrita de variáveis, estruturas condicionais e de repetição. Os estudantes resolveram exercícios na forma de desafios sobre o conteúdo das aulas. Além disso, como prática, eles tiveram que resolver os desafios disponíveis no Code Combat (<https://codecombat.com/>). Esta atividade durou aproximadamente 3 meses, sendo 2 meses para Scratch e 1 mês para Python.

## 5.3 Introdução à Robótica

Os estudantes tiveram contato com os conceitos de robótica, objetivando visualizar como é o processo de comunicação entre software e hardware, como circuitos elétricos e estruturas lógicas de comunicação entre componentes. Foi utilizada uma abordagem constituída a partir de aprendizagem criativa com gamificação, nos moldes de Role Playing Game (RPG).

Os estudantes ficaram imersos em uma narrativa pós-apocalíptica e os processos de aprendizagem foram incorporados à mecânica do jogo. Os elementos de gamificação implementados foram: história, cenário, personificação, combate, restrições, sorte, sistema de pontuação por nível e ranking. O cenário gamificado inicia quando o instrutor narra momentos após um apocalipse no planeta, os sobreviventes se organizam uma sociedade denominada Dakanwa e devem aprender a trabalhar colaborativamente, desenvolvendo habilidades para a resolução de problemas. Para obter treinamento de sobrevivência, os habitantes devem concentrar-se na escola de Dakanwa. Além da necessidade de desenvolver habilidades para a sobrevivência, é fundamental saber lidar com o grupo que discorda das regras de compartilhamento de recursos em Dakanwa e, liderados por Germonkil, criam divisões que podem dificultar a relação entre os sobreviventes. As atividades iniciaram sob a narrativa de que os sobreviventes estão prontos para receber o treinamento na escola tecnológica de Dakanwa. A turma é

dividida em equipes e aquela que tem o melhor desempenho durante o treinamento, tem seus integrantes nomeados como líderes das tropas. As equipes pontuam em cada missão e aquela com melhor desempenho ou que finalize primeiro os desafios, recebe a maior pontuação. Esta atividade durou aproximadamente 1 mês e o processo seguido está descrito na Tabela 3.

**Tabela 3. Sequência de atividades da Etapa IV**

Atividades	Descrição
Projetar uma invenção	O desafio foi projetar um carro que se locomova e tenha itens essenciais para sobrevivência. Foram utilizados materiais como: papelão, massa de modelar, lápis, papéis e pincéis. O objetivo era trabalhar colaborativamente. As equipes foram pontuadas de acordo com a ordem de finalização da atividade. Quem terminou primeiro recebeu a maior pontuação.
Coding Dojo (Blockly)	O desafio DOJO foi realizado com os problemas disponíveis no <a href="https://blockly-games.appspot.com/">https://blockly-games.appspot.com/</a> . A cada 2 minutos trocava-se o membro da equipe que estava programando. A equipe que recebeu a maior pontuação foi a que resolveu mais problemas.
Coding Dojo (mBlock)	O desafio era fazer desenhos utilizando a ferramenta <i>mBlock</i> e foi utilizado o mesmo sistema de pontuação da atividade anterior.
Acendendo o primeiro LED (Semáforo e Pisca-Pisca)	Após acender o primeiro LED utilizando a placa Arduino, as equipes deveriam representar um semáforo e um pisca-pisca utilizando os LEDs. A pontuação foi atribuída com base na ordem de finalização da atividade.
Projetando um carro	Essa atividade foi estruturada da seguinte maneira: montagem do carrinho (utilizando o Kit Chassi Robótico 2WD); construção dos circuitos do carro (utilizando a ponte H, protoboard, Arduino e um <i>bluetooth</i> ); e, programação das funções do carrinho (utilizando o <i>mBlock</i> ).
Mapeando rotas	O desafio foi a resolução de rotas em grafos, primeiramente as equipes deveriam traçar uma sequência de passos no papel, para depois aplicar a resolução traçando a rota manipulando o carro anteriormente programado.

## 6. Resultados e Discussões

O projeto teve duração de 6 meses, com aulas semanais em laboratórios da Universidade e atividades na escola. Durante as aulas observou-se um maior engajamento dos estudantes nas atividades consideradas mais lúdicas, como Computação desplugada, robótica e o Scratch, nessa ordem. As observações foram realizadas por no mínimo 2 pesquisadores, pois em alguns momentos houve 3 ou 4. A técnica adotada foi observação participante, que pode ser adotada no modelo de investigação-ação. Os resultados por fase estão descritos nas seções a seguir.

### 6.1 Avaliação empírica com base na observação

Durante a primeira etapa, cujo foco foi tratar dos conceitos de Computação, do nível de letramento digital dos estudantes, da tecnologia e como ser protagonista nesse processo de inclusão digital, os estudantes começaram com um certo nível de timidez, porém, no decorrer das ações, ao se identificarem com os exemplos utilizados, sentiram-se mais à vontade para compartilhar os seus saberes e perspectivas.

Ao trabalhar o desenvolvimento do Pensamento Computacional através da criação de estratégias para a resolução de problemas utilizando como ferramenta a Computação desplugada, os estudantes se mostraram motivados. Sobretudo nas tarefas que tinham caráter gamificado, mesmo após o término das atividades, os estudantes permaneceram no laboratório que ganhou um aspecto de sala de brincadeiras. Durante essas ações se observou a diminuição do tempo de resolução dos desafios à medida em

que o número de tarefas foi aumentando. Durante as atividades os estudantes tiveram a oportunidade de fazer uma apresentação de seus projetos para a comunidade, durante a Semana Nacional de Ciência e Tecnologia. Os objetos da exposição foram jogos, produzidos no módulo de Scratch como linguagem de programação. Notou-se que o processo de confecção do projeto gerou engajamento e os desenvolvedores indicaram preocupação com o produto que as pessoas jogariam. Os projetos apresentados cumpriram os requisitos de programação, design de jogos e design visual.

Outra atividade que teve avaliação empírica foi a Robótica em que foram apresentados problemas a serem resolvidos com algoritmos e lógica de programação. Os problemas consistiam em mapear rotas entre um ponto de partida e um objetivo, conforme ilustra a Figura 3. A maioria dos estudantes relatou que não teve dificuldade em entender os desafios propostos, no entanto, alguns tiveram dificuldade com noções de graus e lateralidade e por isso, a implementação dos algoritmos ficou comprometida.



Figura 3. Atividade “Mapeando Rotas” durante o módulo de robótica gamificada.

## 6.2 Análise automática dos projetos

Para a etapa de programação com Scratch, foi utilizado o Dr. Scratch, uma ferramenta de código aberto, que realiza análises automáticas de projetos desenvolvidos na plataforma Scratch, atribuindo feedbacks ao usuário sobre o grau de desenvolvimento do Pensamento Computacional do projeto (Moreno-León *et al.*, 2015).

A Tabela 4 apresenta a avaliação dos projetos desenvolvidos pelos estudantes para a mostra da Semana Nacional de Ciência e Tecnologia.

**Tabela 4. Análise dos projetos criados por pares na ferramenta Dr. Scratch. Nas linhas estão os conceitos avaliados e nas colunas estão os projetos das duplas. Todos os conceitos variam de 0 a 3.**

Conceito	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	Média
Abstração	1/3	1/3	1/3	3/3	3/3	3/3	3/3	3/3	1/3	3/3	1/3	1/3	2,0
Paralelismo	3/3	3/3	0/3	3/3	3/3	3/3	3/3	3/3	3/3	3/3	3/3	3/3	2,75
Lógica	1/3	2/3	0/3	3/3	3/3	3/3	3/3	2/3	2/3	3/3	3/3	3/3	2,33
Sincronização	3/3	3/3	0/3	3/3	3/3	3/3	3/3	3/3	3/3	3/3	3/3	3/3	2,75
Controle de fluxo	2/3	3/3	1/3	3/3	3/3	3/3	3/3	3/3	1/3	2/3	3/3	3/3	2,5
Interatividade	2/3	2/3	2/3	2/3	2/3	2/3	2/3	2/3	2/3	2/3	2/3	2/3	2
Representação de Dados	2/3	2/3	1/3	2/3	3/3	3/3	2/3	2/3	2/3	2/3	2/3	2/3	2,08
Total	14/21	16/21	5/21	19/21	20/21	20/21	19/21	18/21	14/21	18/21	17/21	17/21	16,42

Foram avaliados sete conceitos de PC analisados pelo Dr. Scratch: i) Lógica - analisa o uso de condições para a tomada de decisão com o apoio de operações matemáticas; ii) Paralelismo - verifica as rotinas de execução das funções em paralelo; iii) Interatividade - existência de itens que promovem a interatividade com o usuário; iv)



Representação de Dados - uso de variáveis para armazenar, recuperar e atualizar determinados valores; v) Controle de Fluxo - avaliação de noções algorítmicas e visão estruturada das funções utilizadas; vi) Sincronização - o grau de execução múltipla entre um acontecimento e outro; vii) Abstração - os níveis de abstração, atrelados à decomposição do problema [Rodríguez, Zem-Lopes, Marques e Isotani, 2015].

Como forma de avaliação na etapa de programação com Python, utilizou-se a plataforma online de programação Code Combat, que disponibiliza desafios para serem resolvidos em linguagens de programação como Python e JavaScript. Uma questão que chamou a atenção durante o processo é que os estudantes apresentaram melhor desempenho com a linguagem visual, no caso o Scratch, do que com o Python. No início do Python houve uma diminuição no interesse, ocorrendo um decréscimo no número de tarefas enviadas para a plataforma digital de acompanhamento. Inferiu-se pelo padrão demonstrado pelo grupo de estudantes, que isso se deu pelo nível de abstração necessário tanto à programação em si quanto pela linguagem. Os estudantes conseguiam descrever a solução verbalmente, mas não implementar.

## 7. Considerações Finais

O artigo apresentou uma proposta de sequência didática com base nos pilares Cultura Digital, Pensamento Computacional e Tecnologia Digital, perfazendo os três eixos de conhecimento propostos pela SBC em suas Diretrizes para ensino de Computação na Educação Básica. Foram consideradas as arquiteturas humanas de aprendizagem e os requisitos de aprendizagem em Computação, com vistas ao desenvolvimento do Pensamento Computacional através da aprendizagem de programação. Embora não exista um modelo comprovado de como promover o desenvolvimento do Pensamento Computacional no Ensino Médio (ou Fundamental) de maneira eficaz, a inserção de conceitos computacionais nesse contexto é relevante, já que a área da Computação está constantemente presente na vida das pessoas.

A pesquisa obteve resultados satisfatórios em relação à utilização de linguagens gráficas para o aprendizado de programação e conseguiu estimular a resolução de problemas por meio de habilidades do PC, de diferentes formas, desde a escrita de algoritmos no papel até a programação de robôs. Como esse ciclo aconteceu com três séries diferentes, como trabalho futuro, pretende-se realizar a sequência didática proposta com turmas de cada série, primeiro, segundo e terceiro ano, garantindo que todas as séries recebam todas as atividades da sequência didática. Espera-se, com isso, analisar qual turma terá o melhor desempenho e maior grau de engajamento.

## 8. Agradecimentos

Nossos agradecimentos à FAPEAM (Fundação de Amparo à Pesquisa do Estado do Amazonas) pelo financiamento que permitiu a execução das atividades do “Projeto de indução à formação de recursos humanos em Tecnologia da Informação no Amazonas (PRO-TI UEA/EST)”. Ao ThinkTed Lab<sup>2</sup> e ao LUDUS Lab<sup>3</sup> pela execução das atividades na Universidade do Estado do Amazonas e ao GRAEST<sup>4</sup> pelo apoio e suporte para a realização das atividades.

---

<sup>2</sup> Laboratório de Pesquisa e Desenvolvimento em Tecnologias Educacionais

<sup>3</sup> Laboratório de Tecnologia Inovação e Economia Criativa

<sup>4</sup> Grupo de Robótica e Automação da EST.

## Referências

- 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". 25º Workshop sobre Educação em Computação (WEI 2017).
- Bezerra, G., Barbosa, M., Alves, S., & da Silva Coutinho, J. C. (2017). "Ensino de computação no ensino médio utilizando técnicas da Computação Desplugada: um relato de experiência". II Congresso sobre Tecnologias na Educação (Ctrl+ E 2017).
- Brackmann, C. P. (2017). "Desenvolvimento do pensamento computacional através de atividades desplugadas na educação básica" Universidade Federal do Rio Grande do Sul, Porto Alegre.
- Duncan, C., & Bell, T. (2015). "A pilot computer science and programming course for primary school students". Proceedings of the Workshop in Primary and Secondary Computing Education.
- Grover, S., & Pea, R. J. (2013). "Computational thinking in K–12: A review of the state of the field. Educational researcher", 42(1), 38-43.
- Liukas, L. (2015). Hello Ruby: adventures in coding (Vol. 1): Macmillan.
- Moreno-León, J., Robles, G., & Román-González, M. J. R. R. d. E. a. D. (2015). Dr. Scratch: Automatic analysis of scratch projects to assess and foster computational thinking. (46), 1-23.
- Papert, S. (1980). Mindstorms: Children, computers, and powerful ideas: Basic Books.
- Pereira, A. C., & Franco, M. E. (2018). "Desenvolvendo o pensamento computacional no ensino fundamental com Arduino e Scratch". 5º Encontro Nacional de Computação dos Institutos Federais (ENCompIF 2018).
- Piaget, J. (1976). The grasp of consciousness: Action and concept in the young child.(Trans by S. Wedgwood).
- Raabe, A., Brackmann, C., & Campos, F. (2018). Currículo de Referência em Tecnologia e Computação da Educação Infantil ao Ensino Fundamental: Centro de Inovação para a Educação Brasileira (CIEB).
- Resnick, M. (2017). Fulfilling Papert's Dream: "Computational Fluency for All". In Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education.
- Resnick, M., & Robinson, K. (2017). Lifelong kindergarten: Cultivating creativity through projects, passion, peers, and play. Cambridge, Massachusetts: MIT Press.
- Vicari, R. M., Moreira, A. F., & Menezes, P. F. B. (2018). Pensamento computacional: revisão bibliográfica. Lume repositório Digital UFRGS, 1.
- Weintrop, D., Hansen, A., Harlow, D., & Franklin, D. (2018). Bringing computer science into elementary school classrooms. American Educational Research Association.
- Wing, J. (2006). "Computational thinking". Communications of the ACM, 49(3), 33-35.
- Wing, J.(2008). "Computational thinking and thinking about computing". Philosophical Transactions of the Royal Society A: Mathematical, Physical Engineering Sciences