

Proposta de ensino de programação para crianças com Scratch e Pensamento Computacional

Humberto Augusto Piovesana Zanetti (FT/UNICAMP)¹

Marcos Augusto Francisco Borges (FT/ UNICAMP)²

Valéria Cristina Gomes Leal (FT/ UNICAMP)³

Igor Yukio Matsuzaki (FT/UNICAMP)⁴

Resumo

Este trabalho apresenta uma proposta para um conjunto de práticas pedagógicas, com o objetivo de explorar o conteúdo de Lógica de Programação e Pensamento Computacional, em uma abordagem motivadora e criativa. São propostas atividades de Programação para crianças de 10 a 13 anos de idade usando como ferramenta de programação o ambiente Scratch. Essas atividades podem ser aplicadas em um contexto formal de ensino, em escolas que tenham uma carga horária específica para ensino de Computação ou em atividades fora do contexto formal da escola, em ambientes diferenciados de aprendizado.

Palavras-chave: Ensino de programação; Scratch; Pensamento Computacional; Oficina.

Abstract

This paper presents a proposal for a set of programming classes, which explores the logic programming content and Computational Thinking in a motivating and creative approach. This paper proposes programming activities for children from 10 to 13 years old using as a programming tool Scratch Environment. These activities can be applied in a formal context of teaching in schools that have a specific time dedicated for teaching computational content or activities outside the formal context of the school, in different learning environments.

Keywords: Teaching programming; Scratch; Computational Thinking; Workshop

¹ Contato: hzanetti@gmail.com

² Contato: marcosborges@ft.unicamp.br

³ Contato: vckawauchi@hotmail.com

⁴ Contato: igor-matsuzaki@live.com

1. Introdução

As novas tecnologias permitem que o processo de aprendizagem se crie por meio da experimentação, estimulando as descobertas. Para isto, é necessária uma rede de informações na qual pode-se aprender conteúdos ilimitados a partir de um espaço limitado e com regras, como, por exemplo, em jogos (Thomas; Brown, 2011). O fazer, o saber e o brincar são comportamentos que interagem de forma natural nesses espaços.

Segundo Wing (2006), Pensamento Computacional (PC) é um método que tem como objetivo solucionar problemas, conceber sistemas e compreender o comportamento humano inspirado em conceitos da Ciência da Computação. No contexto da educação infantil, a utilização de PC pode ajudar crianças a aprender técnicas de resolução de problemas, que podem ser úteis não apenas no escopo de informática, mas para os vários desafios que essas crianças vão enfrentar ao longo da vida.

O trabalho de Stella (2016) mostra que só o uso da ferramenta pode não ser efetivamente motivador, sendo necessário um método mais estruturado para que se desperte a atenção e envolva os alunos. Este trabalho apresenta a oficina “Era uma Vez”, uma proposta para estruturar um conjunto de aulas de programação, que explore o conteúdo de lógica de programação, em uma abordagem motivadora. São propostas atividades de programação para crianças de 10 a 13 anos de idade usando como ferramenta de programação o Scratch⁵. Essas atividades podem ser aplicadas em um contexto formal de ensino, em escolas que tenham uma carga horária específica para ensino de conteúdo computacional ou em atividades fora do contexto formal da escola, tanto conduzidas em escolas ou em ambientes diferenciados de aprendizado. Todo o material que forma a oficina “Era uma Vez” está aberto ao uso público⁶.

Este documento está organizado da seguinte maneira: a seção 2 apresenta o embasamento teórico do trabalho explanando sobre o Pensamento Computacional. Na seção 3 são apresentados alguns trabalhos relacionados. A seção 4 expõe a metodologia utilizada e a temática para a proposta das aulas. A seção 5 introduz o plano de aulas desenvolvidas e a seção 6 apresenta a conclusão.

2. Pensamento Computacional no ensino de programação

Para Phillips (2009), o PC busca integrar “o poder do pensamento humano com as capacidades computacionais”. O PC inclui um conjunto de habilidades, como a escrita, a

⁵ Website oficial do Scratch, com maiores informações: < <https://scratch.mit.edu/>>

⁶ Todo o material referente ao plano de aulas está disponível em <<https://goo.gl/q5JrEU>>

leitura e a aritmética, que são importantes para qualquer pessoa. Essas habilidades auxiliam na resolução de problemas não somente restritos à área da computação (Wing, 2006).

Os conceitos e estratégias ligados ao PC habilitam o aprendiz a resolver problemas complexos, com a estratégia de dividir em pequenos problemas simples, planejar todas as etapas antes de confeccionar o programa, integrar todas as partes de forma lógica e organizar as ideias de forma estruturada. Desta forma, facilitam a transposição do pensamento para a linguagem do computador, o que pode apoiar o desenvolvimento do raciocínio lógico e abstrato no aprendiz (Alcantara, 2016).

Blikstein (2016) afirma que PC é saber usar o computador como instrumento do poder cognitivo e operacional humano, a fim de aumentar a produtividade, inventividade e criatividade. O trabalho de Mestre et al. (2015) justifica que as habilidades estimuladas pelo PC estão diretamente relacionadas à resolução de problemas, pois envolvem a capacidade de compreender as situações propostas e criar soluções através de modelos matemáticos, científicos ou sociais.

O desenvolvimento do PC em alunos por meio do ensino de programação apresenta resultados positivos. Brennan e Resnick (2012) mostram que uma abordagem baseada em PC pode ser utilizada no aprendizado de programação, principalmente para jovens, usando ferramentas que motivem e estimulem a criatividade. Para Wing (2006), PC é baseado em abstração e decomposição de problemas ou sistemas complexos, habilidades altamente recomendadas para alunos que querem aprender programação.

Pensar computacionalmente não é sinônimo de “programar”. No entanto, Blikstein (2016) aponta que uma das etapas fundamentais do PC é saber programar um computador para realizar tarefas cognitivas e de maneira automatizada, para que este conhecimento seja um suporte ao raciocínio humano. O trabalho de De Paula, Valente e Burn (2014) afirma que o PC apresenta uma maneira específica de se pensar e de analisar uma situação ou um artefato, sendo independente do uso de tecnologia, mas indica que por meio da programação é possível alinhar a prática e os conceitos teóricos.

Práticas usando PC buscam desenvolver maior habilidade de raciocínio e capacidade de análise crítica, capacidade de criação e comunicação no mundo digital. Desenvolver atividades baseadas em PC em alunos do Ensino Fundamental, ensinando conceitos de Lógica de Programação de uma forma lúdica e adequada a essa faixa etária, pode auxiliar o aluno em resoluções de problemas em várias áreas, não somente na informática.

3. Trabalhos relacionados

Teixeira et al. (2015) apresenta o projeto “Escola de Hackers” em Passo Fundo-RS, que reuniu mais de 300 crianças para o desenvolvimento de competências na área de programação de computadores e de raciocínio lógico matemático para alunos do 6º ao 9º ano do Ensino Fundamental, com idades entre 11 e 14 anos, utilizando a ferramenta Scratch. Segundo os colaboradores do projeto, há uma necessidade de maior envolvimento dos gestores das escolas e dos professores no projeto, observando uma demanda por formação dos docentes no uso de tecnologia em sala de aula.

O trabalho de Andrade, Silva e Oliveira (2013) aborda o diferencial de práticas que sejam lúdicas e motivadoras. No projeto foi proposto uma metodologia de unir o desenvolvimento de jogos digitais para o ensino de matemática, utilizando o ambiente Scratch. Como resultado, foi criada uma coleção de jogos desenvolvidos pelos 15 alunos do 9º ano do Ensino Fundamental, de uma escola estadual em Garanhuns-PE. Para o desenvolvimento dos jogos, os alunos foram conduzidos a representar os conceitos matemáticos aprendidos em seus projetos.

A disciplina de raciocínio lógico é obrigatória para os anos finais do Ensino Fundamental e o ensino médio das escolas de Mato Grosso do Sul. Batista et al. (2015) propõem um projeto para auxiliar os professores a trabalhar com Scratch e para desenvolver o raciocínio lógico dos alunos. Como muitos professores ainda não possuem o domínio das ferramentas necessárias para desenvolver esta capacidade nas crianças, se fez necessária a criação da oficina. Na oficina, serão apresentados a esses professores, o ambiente do Scratch, alguns conceitos de programação e as possibilidades de aplicar a ferramenta nas diferentes disciplinas, com grande enfoque na disciplina de raciocínio lógico.

O projeto de Santos et al. (2015) utiliza a metodologia de “Computação Desplugada” com objetivo de promover habilidades do PC através do ensino de algoritmos. O trabalho teve como inspiração a ferramenta Scratch, principalmente o modo de construção de algoritmos, utilizando blocos. Na atividade “Algoritmos Desplugados”, os comandos são representados por blocos de borracha EVA (Etil Vinil Acetato) de diferentes cores e tamanhos. Os blocos podem ser encaixados entre si, sendo que um conjunto de blocos devidamente encaixados representa um algoritmo. Para a execução dos algoritmos são utilizados diversos tabuleiros impressos em folhas de tamanho A4, além de itens coletáveis e desafios. O centro da atividade é o personagem Jack, representado por uma figurinha disposta no tabuleiro, capaz de executar os algoritmos desenvolvidos. A atividade “Algoritmos Desplugados” dispõe de uma série de tarefas com o objetivo de ensinar as estruturas básicas de programação, como repetição e decisão, através de algoritmos.

De Oliveira et al. (2014) apresenta uma pesquisa que mostra o uso de Scratch e “Computação Desplugada” como ferramentas pedagógicas auxiliares para a compreensão e desenvolvimento de algoritmos. O trabalho contou com 20 alunos do 9º ano do Ensino Fundamental de uma escola pública de Garanhuns-PE. As práticas foram direcionadas para o ensino de conceitos básicos de computação (como números binários) e programação, usando como contexto básico o desenvolvimento de jogos simples e animações. Como resultado, o estudo apontou que todos os participantes conseguiram entender mais sobre computação e se mantiveram motivados para continuar os estudos juntos às ferramentas apresentadas.

O estudo de caso apresentado por Von Wangenheim, Nunes e dos Santos (2014) reforçam a ideia de que é necessário ensinar a proficiência digital (*IT fluency*), incluindo PC e programação. A pesquisa propõe uma unidade instrucional com o objetivo de ensinar computação para o Ensino Fundamental de forma interdisciplinar e seguindo as diretrizes do modelo de currículo K-12⁷ (*Model Curriculum for K-12 Computer Science*), utilizando Scratch. O experimento foi realizado com 24 alunos do 1º ano do Ensino Fundamental de uma escola pública em Florianópolis-SC. Foram conduzidas práticas nas quais os aprendizes criaram produtos multimídias, como jogos e animações com narrativas (histórias).

4. Materiais e métodos

Esta seção apresenta os materiais criados para o desenvolvimento da ferramenta utilizada para a proposta de ensino, e o método adotado para o design e interatividade com o ambiente. A seção se divide em: a Seção 4.1, apresenta a ferramenta Scratch; a Seção 4.2 apresenta uma discussão sobre a plataforma Code.org e, por fim, a Seção 4.3 aborda o tema, os elementos gráficos e os cenários criados para a ferramenta.

4.1. A ferramenta Scratch

Scratch é um projeto do grupo *Lifelong Kindergarten*⁸ do Media Lab do MIT (*Massachusetts Institute of Technology*). A finalidade do Scratch é apoiar o processo de ensino-aprendizagem de programação. Esta ferramenta permite, compartilhar os projetos criados na comunidade Scratch (Brennan; Resnick, 2012). Há uma versão do ambiente de

⁷ Este modelo apresenta um conjunto de padrões de aprendizagem para o ensino de Ciência da Computação no ensino fundamental e médio nos Estados Unidos da América (K-12). Mais informações sobre o modelo de currículo e suas diretrizes estão disponíveis em CSTA (2011)

⁸ Website da Lifelong Kindergarten: < <https://llk.media.mit.edu/>>

desenvolvimento que não exige acesso à internet (*offline*) e pode ser instalada gratuitamente a partir da web.

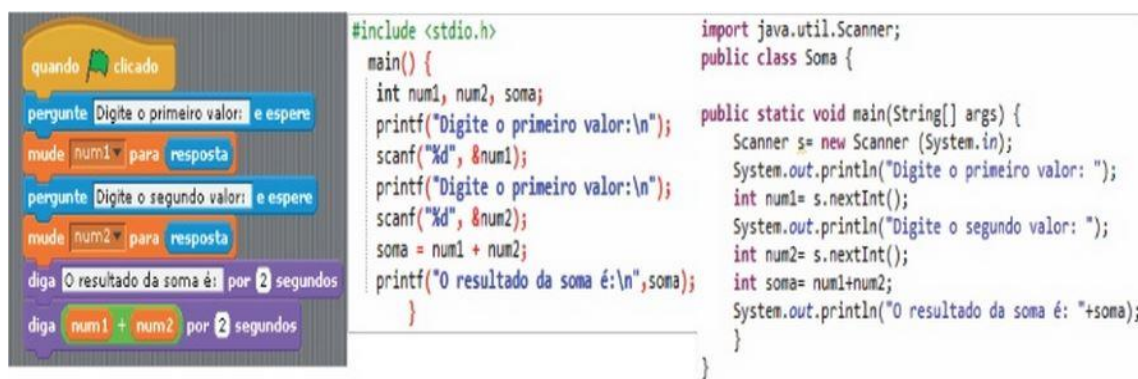
O ambiente Scratch busca apoiar o aprendizado da programação visual, que pode ser descrito como programar em blocos que se encaixam como um quebra-cabeça. Abre-se então espaço para um cenário, no qual, segundo Brennan (2011), existe a possibilidade de exploração de diversos conceitos, perspectivas e práticas computacionais de uma forma criativa, fazendo uso de uma abordagem de aprendizado baseado no conceito de design. O aprendizado baseado no conceito de design é fundamentado em três conceitos:

1. Concepção: tornar o aprendiz capaz de criar, ao contrário de apenas utilizar ou interagir;
2. Personalização: trazer significado pessoal e relevante para algo criado;
3. Reflexão: usar as práticas criativas de cada um como forma de repensar e rever.

No Scratch pode-se construir histórias animadas e jogos com personagens de maneira fácil, devido à importação ou criação de mídia de vários tipos (som, músicas, imagens). O Scratch permite que dentro de sua própria interface possam ser compartilhados os projetos desenvolvidos em uma página da Internet, possibilitando que os projetos desenvolvidos possam ser compartilhados, possibilitando que se receba avaliações (*feedback*) e que seja criada uma biblioteca de projetos, encorajando o aprendizado. (Resnick et al., 2009).

Os blocos do Scratch são divididos em nove classes: Movimento, Aparência, Som, Caneta, Controle, Operadores, Eventos, Sensores e Variáveis (SCRATCH, 2004). Os comandos possuem uma sintaxe simples, quando comparados com algoritmos construídos em linguagens tradicionais, como C e Java. A Figura 1 ilustra a diferença entre um programa nessas três plataformas. Este programa recebe dois valores de entrada que devem ser fornecidas pelo usuário e mostra o resultado da soma desses valores.

Figura 1 – Comparação entre Scratch, C e Java



Fonte: Autores

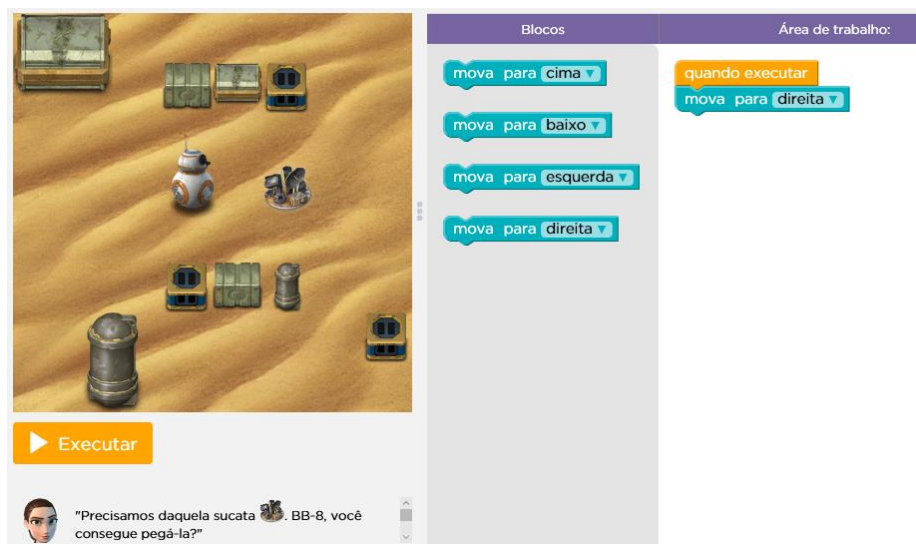
4.2. O método Code.org

Este trabalho foi inspirado no design e metodologia da iniciativa Code.org, que se apresenta como uma plataforma muito atrativa para crianças e jovens adolescentes, público-alvo do projeto. “A Code.org é uma organização sem fins lucrativos dedicada a expandir o ensino de ciência da computação, tornando-a disponível em mais escolas e incrementa-la ao currículo escolar” (CODE, 2015). O projeto Code.org tem como finalidade criar oportunidades de aprendizado de computação em todas as escolas. Desta forma, incrementar a ciência da computação ao currículo escolar.

O website⁹ da Code.org está disponível para alunos e professores. Os módulos de ensino para alunos usam o critério de faixa-etária, de forma que seja coerente a abordagem de conceitos de acordo com a série escolar. Para professores, os Planos e Recursos das aulas têm como objetivo auxiliar e descrever um roteiro para dirigir as aulas de programação.

Para aumentar a identificação com o público sugerido, a Code.org investe em artes gráficas com personagens e animações mundialmente conhecidas. Desta forma, os módulos desenvolvidos possuem uma abordagem gráfica, com vários temas, como o “Star Wars VII” na Figura 2, que atrai tanto crianças como jovens adolescentes para a tarefa de programação.

Figura 2 – Exemplo de atividade no Code.org



Fonte: CODE (2015)

⁹ Website oficial do Code.org: < <https://code.org/> >

5. Proposta de aplicação de oficina de ensino

Esta proposta objetiva a construção de uma oficina com base nos materiais e métodos descritos na seção 4. A oficina, segundo Volquind (2000), é uma prática de ensino que busca estimular os alunos a se aventurar na busca de novos conhecimentos, respeitando os processos mentais e cognitivos de cada sujeito, aproveitando a participação com atenção e, posterior intervenção adequada do professor. O formato de oficina não pressupõe apenas o “aprender fazendo”. Conforme Volquind (2000):

A proposta de oficinas de ensino para ser séria, gratificante e inovadora necessita criar um espaço para vivência, a reflexão e a construção [...] supõe principalmente o pensar, o sentir, o intercâmbio de ideias, a problematização, o jogo, a investigação, a descoberta e a cooperação (Volquind, 2000).

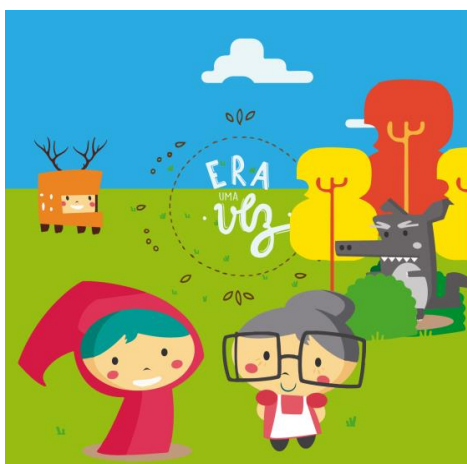
Este trabalho propõe um conjunto de práticas, sequenciais, a serem aplicadas em um formato de oficina, reunidas em uma coleção de planos de aulas, com o objetivo de proporcionar ao professor uma ferramenta didática para a aplicação de conceitos de programação e PC, em um ambiente inovador e lúdico. Além disso, esse instrumento possibilita criar critérios de avaliação através do acompanhamento do professor, sendo possível a validação do conhecimento dos alunos.

5.1. A temática da proposta

Usando a ferramenta Scratch e a preocupação com a motivação advinda do design baseado no Code.org, este projeto explorou como tema a estória da Chapeuzinho Vermelho. A oficina de ensino de programação Scratch proposta foi denominada “Era Uma Vez”, como mostra a Figura 3. A oficina, incluindo todo o material e documentação necessários para sua condução, está disponível na Internet¹⁰.

¹⁰ <http://www.ft.unicamp.br/liag/eraumavez/>

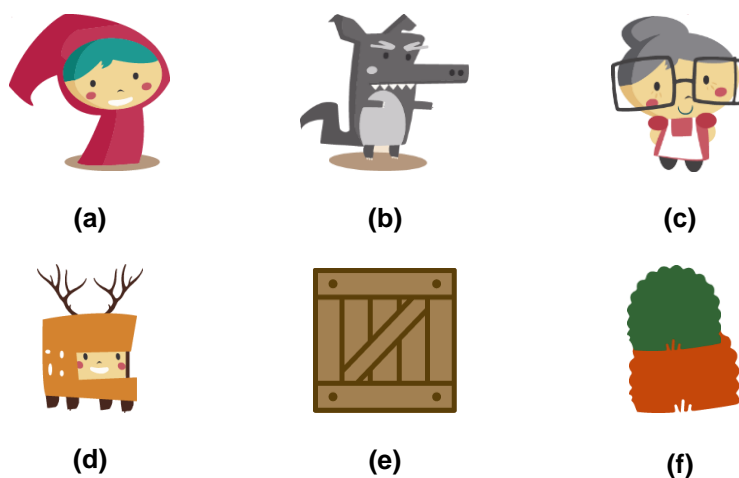
Figura 3 – Exemplo de atividade no Code.org



Fonte: Autores

A oficina “Era uma vez” é composta por um conjunto de desafios. Os desafios propostos aos participantes têm como objetivo guiar Chapeuzinho (Figura 4-a) através de labirintos para que a personagem encontre a personagem da Vovó (Figura 4-c), evitando o personagem Lobo Mau (Figura 4-b). O Cervo (Figura 4-d) poderá acrescentar pontos ao jogo em alguns labirintos. Os labirintos foram montados utilizando alguns elementos do cenário, composto por caixas (Figura 4-e) e arbustos (Figura 4-f). Os alunos devem desenvolver para cada labirinto um algoritmo, contendo elementos, ou funções, que serão abordados nas aulas. Foram criados labirintos com níveis de dificuldades crescentes conforme os alunos consigam resolver os desafios.

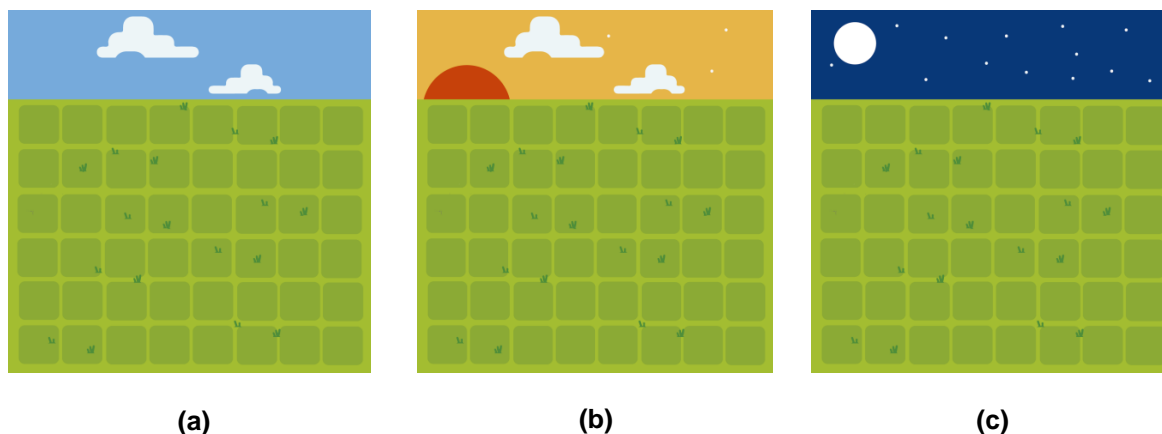
Figura 4 – Elementos do tema “Era uma vez”



Fonte: Autores

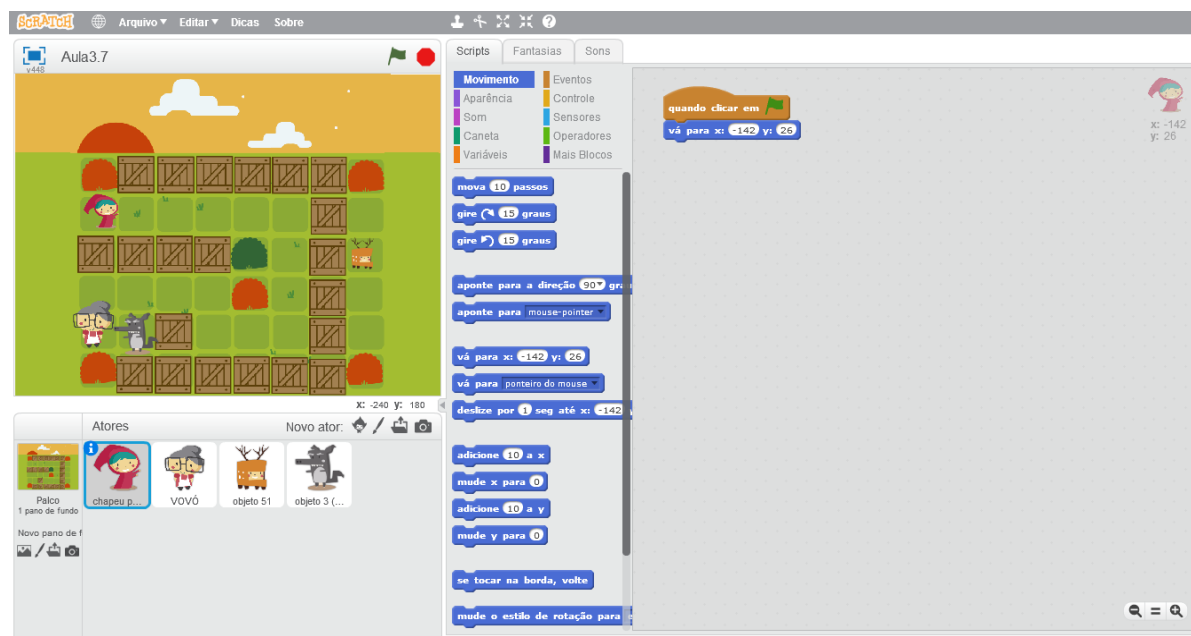
Para representar uma narrativa ao longo do tempo, foi desenvolvido um conjunto de cenários, que variam conforme o decorrer das etapas cumpridas. A Figura 5 apresenta os cenários “dia”, “tarde” e “noite” (respectivamente Figura 5-a, Figura 5-b e Figura 5-c). A Figura 6 apresenta um exemplo do cenário com todos os elementos visuais inseridos, formando um labirinto e a área de programação em blocos.

Figura 5 – Construção de cenário e exemplo de uma atividade



Fonte: Autores

Figura 6 – Exemplo de uma atividade



Fonte: Autores

5.2. Plano de aula

O plano de aula proposto é dividido em 6 aulas distintas, possuindo objetivos claros sobre as práticas a serem realizadas durante a oficina. A divisão das aulas busca uma aplicação gradativa dos conceitos teóricos de programação e desafios crescentes para manter a motivação dos alunos. A sequência visa trabalhar com as habilidades recorrentes no PC, criando pré-condições para que o aluno possa progredir nas tarefas, sendo mais independente e com maior confiança nas suas ações. As aulas estão divididas da seguinte forma:

- Aula 1: tem como objetivo apresentar o conceito de algoritmo, realizando uma dinâmica nomeada “Algoritmo no Papel”. A dinâmica usa um tabuleiro e uma folha de resposta. No tabuleiro, o aluno pode desenhar de forma livre, pintando os quadrados que desejar. Em seguida, o aluno deve preencher a folha de respostas com um algoritmo, uma “receita”, para reproduzir o desenho no tabuleiro. A turma deve trocar os tabuleiros entre si e tentar reproduzir um algoritmo que recrie o desenho do colega. Esta aula tem como objetivo apresentar o conceito de algoritmo como uma forma de “Receita”: por meio dessa receita, o cozinheiro, que faz o papel do computador, poderá ler e reproduzir os passos, ou seja, um “programa”;
- Aula 2: nessa aula são utilizados os mesmos conceitos da aula anterior, algoritmo e programa, com o objetivo de identificar no cotidiano a presença destes conceitos. O “Algoritmo do BIS” usa um dos alunos como voluntário. A turma passa comandos para esse voluntário, indicando as ações que ele deve realizar para abrir uma caixa de bombom. Outros exercícios para identificar os algoritmos nas atividades diárias são propostos para a turma descrever em uma folha de respostas;
- Aula 3: relaciona a ação de caminhar em um ambiente, com as coordenadas cartesianas. Faz-se uma analogia de apresentar os comandos de coordenadas para mover um aluno voluntário, com objetivo de chegar ao outro lado da sala. A partir deste momento, as atividades serão exercitadas dentro da ferramenta Scratch. Essas atividades de resolução de labirintos simples trabalha a ambientação dos alunos com a ferramenta e com os conceitos de coordenadas de um plano cartesiano;
- Aula 4: faz uso de um copo de plástico e uma superfície plana. A música “Cup Song” do filme *Pitch Perfect* (2012) é a utilizada para esta dinâmica com o copo. Um tutorial em vídeo está disponível como material auxiliar para o aplicador nesta dinâmica. O objetivo dessa dinâmica é apresentar atividades de repetição. As atividades na ferramenta Scratch exploram o uso dos comandos “Repita X vezes” e “Repita Sempre”;

- Aula 5: faz uso de questões de português e matemática, que estão respondidas de forma incorreta. O objetivo dessa dinâmica é explorar a capacidade do aluno de depuração, habilidade de identificar erros e propor a forma correta para cada questão. Algoritmos incompletos e errados estão dentro das atividades dos labirintos: os alunos têm o objetivo de encontrar os erros e corrigi-los;
- Aula 6: revisão de todos os conceitos e dinâmicas trabalhados durante a oficina, introduzindo o conceito de Condição. As atividades propostas para esta aula exploram o comando “Se-Então”.

O Quadro 1 apresenta os temas, que definem o conceito-chave de cada aula; as dinâmicas, que servem como uma metáfora do tema; os conceitos teóricos a serem apresentados e trabalhados durante a aula; e uma breve descrição da prática na qual os alunos farão atividades para exercitar os conceitos.

Quadro 1 – Descrição do plano de aulas

AULA	TEMA	DINÂMICA	TEÓRICO	PRÁTICA
1	Introdução	Algoritmo no Papel	Explicação dos termos: Algoritmo e Programa	Construir dentro de um tabuleiro um desenho e seu Algoritmo para desenhá-lo
2	Algoritmo no dia-a-dia	Algoritmo do BIS	Explicar a aproximação de conceitos abstratos para metáforas do mundo real	Encontrar e descrever as atividades do cotidiano em Algoritmos
3	Sequência de passos	Caminhar no espaço	Formalizar a forma de representar a sequência de passos descrita nos algoritmos	Apresentar os mecanismos de “Movimento” e “Controle” da ferramenta Scratch. Esses mecanismos serão utilizados para resolver os labirintos
4	Laços de repetições	Canção do Copo	Explicar que trabalhos repetitivos podem ser descritos em formas enxutas.	Apresentar os comandos de laços de repetição dentro da aba “Controle” e resolver os labirintos usando esse mecanismo.
5	Depuração	Achar o erro	Demonstrar que Algoritmo pode estar: errado ou incompleto. Neste caso, é importante saber encontrar o defeito.	Corrigir Algoritmos com defeitos nas atividades do labirinto usando o mínimo de trabalho possível, ou seja, não é interessante que seja refeito tudo desde o início.
6	Revisão e Condição	-	Apresentar o conceito de condição	Explorar o funcionamento dos comandos condicionais

Fonte: Autores

Esta proposta inclui três sugestões de avaliações para serem aplicadas com a oficina “Era Uma Vez”, quando colocada em prática com alunos. A primeira avaliação tem como principal objetivo avaliar o interesse do aluno por Tecnologia, Computação e Matemática; a

segunda, uma adaptação da avaliação proposta por Bradley e Lang (1994), vai coletar dos alunos o grau de entendimento, motivação e interesse de cada aula da oficina e a terceira será responsável por identificar se houve alguma mudança sobre o interesse dos alunos em tecnologia, computação e matemática. A primeira avaliação tem como objetivo verificar o conhecimento, quais são os interesses dos alunos sobre matemática, informática e tecnologia em geral. Com isso, é possível ver se houve um maior interesse ao final da aplicação das aulas. A segunda avaliação proporciona um instrumento de investigação da aceitação e entendimento dos alunos diante das práticas conduzidas ao longo da oficina. Essa avaliação oferece ao professor-aplicador recursos para verificar se há práticas que não tiveram os efeitos esperados, ou de menor aceitação por parte dos alunos, podendo o professor fazer correções, tanto na própria aula quanto no método de aplicação para futuras turmas de alunos. Por fim, a terceira avaliação reproduz as questões feitas na primeira avaliação, com o intuito de verificar se os alunos percebem que houve algum impacto em seus conhecimentos relacionados à matemática, informática e tecnologia. Com essa avaliação é possível verificar a eficácia da oficina.

5.3. Pensamento Computacional na proposta

No plano apresentado, as aulas foram propostas com o objetivo de promover as habilidades fundamentais de PC, conforme as premissas definidas pela *International Society for Technology in Education (ISTE)*¹¹, *Computer Science Teachers Association (CSTA)*¹² e *National Science Foundation (NSF)*¹³ (ISTE, 2016). Essas habilidades fundamentais são:

1. Coleção de dados: processo de recolher as informações adequadas;
2. Análise de dados: encontrar o sentido dos dados, encontrar padrões e tirar conclusões;
3. Representação de dados: representação e organização de dados em gráficos adequados, cartas, palavras ou imagens;
4. Decomposição do problema: quebrar tarefas em partes menores e gerenciáveis;
5. Abstração: reduzir a complexidade para definir ideia principal;
6. Algoritmos e procedimentos: série de passos ordenados e encadeados para resolver um problema ou atingir algum fim;
7. Automação: usar computadores ou máquinas para fazer tarefas repetitivas ou tediosas;
8. Simulação: representação ou modelo de um processo. Simulação envolve também as experiências em andamento utilizando modelos;

¹¹ ISTE - <http://www.iste.org/>

¹² CSTA - <http://www.csta.acm.org/>

¹³ NSF - <http://www.nsf.gov/>

9. Paralelização: organizar recursos para, simultaneamente, realizar tarefas para alcançar um objetivo comum.

A maior parte das habilidades acima foram distribuídas nas atividades que compõem as aulas. Na Aula 1 pode-se encontrar o exercício das habilidades de análise de dados, representação de dados e coleção de dados dentro da dinâmica Algoritmo no Papel. A Aula 2 trabalha a decomposição do problema na proposta de identificar os algoritmos em atividades do cotidiano. Na Aula 3 os alunos praticam com os labirintos os algoritmos e procedimentos e simulação. A Aula 4 apresenta o uso do laço de repetições como uma nova forma de representar comandos repetitivos (automação), visto na dinâmica “Cup Song” e nas atividades do labirinto. A Aula 5 trabalha a habilidade de coleção e análise de dados enquanto apresenta o conceito de depuração. Na Aula 6 os alunos têm a liberdade de criação e experimentação da ferramenta, podendo aplicar todas as habilidades trabalhadas nas aulas anteriores. A habilidade de paralelização não é trabalhada nesta oficina.

6. Conclusões e trabalhos futuros

Este trabalho desenvolveu uma proposta para estruturar um conjunto de aulas com o objetivo de explorar o PC com crianças do nível fundamental, além de trabalhar conceitos de lógica de programação em uma abordagem lúdica e motivadora. A dinâmica das aulas foi inspirada nas estratégias e no *design* do Code.org, usando como plataforma de programação a ferramenta Scratch. Todo o material da oficina “Era uma vez” está disponível ao público, para uso livre, de modo que professores possam aplicar em suas turmas.

O alinhamento das habilidades essenciais do PC com plano de aula proposta, apresenta uma maneira interessante de estimular a prática do raciocínio e da programação com os alunos. Com isso, é possível criar uma estratégia de aplicação de tarefas, servindo como um instrumental para o professor, tanto apresenta uma rotina de atividades, quanto meios de avaliação. Entretanto, a proposta ainda não esgota todas as possibilidades de abordagem das habilidades do PC, mas apresenta uma solução sistemática, que pode ser melhorada e adaptada, para vários temas e ambientes de ensino amparados pelo uso do computador.

Como trabalho futuro para este projeto está a aplicação sistemática de oficinas em escolas públicas de ensino fundamental, para conduzir uma análise do método em um ambiente de ensino, junto à professores e alunos. A análise desses resultados terá como objetivo verificar a eficiência da abordagem proposta e quais são as possíveis abordagens para melhorá-la. Após a condução da oficina, será analisado se o instrumento de avaliação proposto é suficiente para apresentar aos professores o retorno de satisfação e do conhecimento dos alunos. Com relação à estruturação do cenário das atividades, poderão

ser propostos novos temas, que possam motivar os alunos e trazer mais cenários lúdicos ao processo educativo. Por fim, em relação ao alinhamento do PC, após a aplicação das dinâmicas com alunos e criação de novos temas, será possível abranger mais habilidades, criando um *framework* mais completo e consistente às práticas de programação juntamente com PC.

7. Referências

ALCANTARA, L. C. Programação para crianças, será? **Robota Tecnologia Educacional**. Disponível em <<http://www.robota.com.br>>. Acesso em: 20 mai. 2016.

ANDRADE, M.; SILVA, C.; OLIVEIRA, T. Desenvolvendo games e aprendendo matemática utilizando o Scratch. **Anais do SBGames**, São Paulo/SP, 2013.

BATISTA, E. J. S. et al. Utilizando o Scratch como ferramenta de apoio para desenvolver o raciocínio lógico das crianças do ensino básico de uma forma multidisciplinar. **XXI Workshop de Informática na Escola – WIE**, 2015.

BLIKSTEIN, P. **O Pensamento Computacional e a Reinvenção do Computador na Educação**. Disponível em: <<http://bit.ly/1IXIbNn>>. Acesso em: 10 abr. 2016.

BRADLEY, M. M.; LANG, P.J. Measuring Emotion: The Self Assessment Manikin and the Semantic Differential, in: **Journal of Behavioral Therapy and Experimental Psychiatry**, 25, 49-59, 1994.

BRENNAN, K. **Creative computing: A design-based introduction to computational thinking**. 2011. Disponível em: < <http://scratched.gse.harvard.edu/sites/default/files/curriculumguide-v20110923.pdf> >. Acesso em: 29 ago. 2015.

BRENNAN, K.; RESNICK, M. New frameworks for studying and assessing the development of computational thinking. **American Educational Research Association (AERA)**, Vancouver, BC, Canada, 2012. Disponível em: <http://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf>. Acesso em: 29 ago. 2015.

CODE. **Anybody Can Learn**. Disponível em <<http://code.org/>>. Acesso em: 15 set. 2015.

CSTA – **Operational Definition of Computational Thinking**, 2011. Disponível em: <<http://csta.acm.org/Curriculum/sub/CurrFiles/CompThinkingFlyer.pdf>>. Acesso em: 26 fev. 2015.

DE PAULA, B. H.; VALENTE, J. A.; BURN, A. O uso de jogos digitais para o desenvolvimento do currículo para a Educação Computacional na Inglaterra. **Currículo sem Fronteiras**, v. 14, n. 3, p. 46-71, 2014.

DE OLIVEIRA, M. L. S. et al. Ensino de lógica de programação no ensino fundamental utilizando o Scratch: um relato de experiência. In: **Anais do Workshop de Informática na Escola**, Brasília–DF. 2014.

ISTE. **Computational thinking teacher resources**. Disponível em: https://csta.acm.org/Curriculum/sub/CurrFiles/472.11CTTeacherResources_2ed-SP-vF.pdf. Acesso em: 28 jan. 2016.

MESTRE, P. et al. Pensamento Computacional: Um estudo empírico sobre as questões de matemática do PISA. In: **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**. p. 1281, 2015.

PHILLIPS, P. **Computational Thinking: A problem-solving tool for every classroom**, Microsoft Corporation, Computer Science Teachers Association, 2009. Disponível em: <<http://www.csta.acm.org/Resources/sub/ResourceFiles/CompThinking.pdf>>. Acesso em: 01 set. 2015.

PITCH perfect. Direção: Jason Moore, Baseado no livro de Mickey Rapkin. Baton Rouge, Louisiana - EUA: Gold Circle Films, Universal Pictures, 2012. Formato: 35 mm (112 min), 3.068 m (6 bobinas), son., Legendado. Inglês.

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

SANTOS, G. et al. Proposta de atividade para o quinto ano do ensino fundamental: Algoritmos Desplugados. **XXI Workshop de Informática na Escola –WIE**, 2015.

SCRATCH. **Lifelong Kindergarten no Media Lab do Instituto de Tecnologia de Massachusetts**, 2004 – Disponível em: <http://scratch.mit.edu>. Acesso em: 01 set. 2015.

STELLA, A. L. **Utilizando o pensamento computacional e a computação criativa no ensino da linguagem de programação Scratch para alunos do ensino fundamental**. Dissertação, 2016 (Mestrado), Faculdade de Tecnologia – Universidade Estadual de Campinas. Limeira, 2016.

TEIXEIRA, A.C. et al. Programação de computadores para alunos do ensino fundamental: A Escola de Hackers. **XXI Workshop de Informática na Escola –WIE**, 2015.

THOMAS, D.; BROWN, J. S. A New Culture of Learning: Cultivating the Imagination for a World of Constant Change. **Createspace Independente Publishing Platform**, 2011.

VOLQUIND, L. Oficinas de ensino: o quê? por quê? como? **EDIPUCRS**, 2000.

VON WANGENHEIM, C. G.; NUNES, V. R.; DOS SANTOS, G. D. Ensino de computação com Scratch no ensino fundamental—um estudo de caso. **Revista Brasileira de Informática na Educação**, v. 22, n. 03, p. 115, 2014.

WING, J. M. Computational Thinking. **Communications of the ACM**, vol. 49, n° 3, mar. 2006. Disponível em: <<https://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf>>. Acesso em: 30 ago. 2015.