

Evolução de um ambiente para programação tangível e avaliação de seu uso em cenário educacional

Marleny Luque Carbajal

Universidade Estadual de Campinas
UNICAMP
Instituto de Computação
Av. Albert Einstein 1251
CEP 13083-852
Campinas/SP – Brasil
Fone: (+55) 19 3521- 5838
marleny.carbajal@students.ic.
unicamp.br

M. Cecília C. Baranauskas

Universidade Estadual de Campinas
UNICAMP
Instituto de Computação & NIED
Av. Albert Einstein 1251
CEP 13083-852
Campinas/SP - Brasil
Fone: (+55) 19 3521- 5838
cecilia@ic.unicamp.br

ABSTRACT

In recent years, the idea that the “computational thinking”, that is the thinking used while we program, is a key skill for everyone, not just for computer scientists, has revived. Based on this idea, several efforts have been made to introduce Computing concepts in the educational process of children. In this article, we present the evolution of TaPrEC, a tangible programming environment created for children to learn basic programming concepts. TaPrEC allows the physical construction of computer programs using colorful wooden blocks shaped as puzzle pieces that have a symbol embossed representing its functionality and a RFID tag for identification. The environment in its second version was put into use in a real educational setting with children and elementary school teachers. Results of use suggest that it is an easy programming environment to learn and use. The results also showed that children got a preliminary understanding of the basic concepts of programming treated.

RESUMO

Nos últimos anos tem ressurgido a ideia de que o “pensamento computacional”, isto é, o pensamento utilizado enquanto se programa, é uma habilidade fundamental para todos, não apenas para cientistas da computação. Com base nessa ideia, vários esforços têm sido feitos para introduzir conceitos da Computação no processo educativo de crianças. Neste artigo apresentamos a evolução de TaPrEC, um ambiente de programação tangível criado para que crianças aprendam conceitos básicos de programação. TaPrEC permite a construção física de programas de computador utilizando blocos coloridos de madeira em forma de peças de quebra-cabeças que possuem um símbolo em alto-relevo para representar sua funcionalidade e uma etiqueta RFID para sua identificação. O ambiente em sua segunda versão foi colocado em uso em um cenário educacional real com crianças e professores do ensino fundamental. Resultados desse uso sugerem que é um ambiente de programação fácil de aprender e usar. Os resultados também demonstraram que as crianças conseguiram um entendimento preliminar dos conceitos básicos de programação tratados.

Descritor de Categorias e Assuntos

K.3.1 [Computers and Education]: Computers Uses in Education.

H.5.2. [Information interfaces and presentation (e.g., HCI): User Interfaces.

Termos Gerais

Design, Experimentation, Human Factors.

Palavras Chaves

Tangible Programming; Computational Thinking, Education; Scratch

1. INTRODUÇÃO

Os benefícios do contato da criança com resolução de problemas via linguagem de programação foram discutidos originalmente por Seymour Papert ainda na década de setenta, formalizando sua teoria em 1980 no seu famoso livro “*Mindstorms Children, Computers, and Powerful Ideas*” [14]. Mais recentemente, Jeannette Wing [24] descreve o pensamento computacional como uma “habilidade fundamental para todos, não apenas para cientistas da computação”. Foi por isso que defendeu a ideia de que o pensamento computacional deveria ser incluído na formação de toda criança já que representa uma competência vital para a aprendizagem da ciência, tecnologia, engenharia e matemáticas.

A aprendizagem da programação é uma forma eficiente de desenvolver o pensamento computacional; no entanto, com a forma tradicional de programação, as crianças têm dificuldades não só na aprendizagem da sintaxe e símbolos rígidos, mas também no uso de ambientes de programação complexos [7]. Existem autores [9] que afirmam que as linguagens de programação baseadas nas TUIs (*Tangible User Interfaces*) [11] têm o potencial de facilitar a aprendizagem de sintaxes complicadas, de promover a colaboração, e facilitar aos professores manter um ambiente positivo de aprendizagem.

No entanto, a tecnologia tangível é considerada delicada, cara e não padronizada [8]. Este pode ser um problema em contextos educacionais onde o custo é um fator decisivo na hora de optar por determinada tecnologia. Outro fator importante a considerar é o processo de inclusão de ambientes de programação tangíveis já existentes num contexto educacional distinto para o qual foi pensado. Forçar uma tecnologia num contexto educacional novo pode ser um processo lento e complicado.

Considerando o exposto, projetamos, criamos e avaliamos o ambiente de programação tangível TaPrEC (*Tangible Programming Environment for Children*), envolvendo no processo de design as principais partes interessadas

(pesquisadores, professores e alunos) para garantir que a solução criada fizesse sentido para elas. Nesse processo, utilizamos o Modelo Semio-Participativo de Design [2] que articula ao mesmo tempo o desenvolvimento de sistemas interativos e práticas sociais com as partes interessadas.

TaPrEC permite às crianças criar programas de computador organizando objetos tangíveis, aplicando três conceitos básicos de programação: Sequências, Repetições e Procedimentos. A arquitetura do TaPrEC está composta pelo *Raspberry Pi* [17], a tecnologia RFID incorporada nos Blocos de Programação e um Software de Controle desenvolvido na linguagem de programação *Scratch* [5]. Almejamos que o ambiente proposto permita às crianças aprender conceitos básicos de programação, que seja uma alternativa de baixo custo para o ensino de programação nas escolas e possibilite uma transição mais suave para ambientes virtuais de aprendizagem e o mundo da programação de computadores.

Neste artigo apresentamos a evolução de nosso protótipo inicial e detalhamos os resultados do Estudo de Caso realizado no segundo semestre de 2015 com 26 participantes entre alunos e professores, num contexto formal de ensino para avaliar a segunda versão de TaPrEC, que chamamos de TaPrEC 2.0. Assim, o texto é organizado como segue: na próxima seção apresentamos uma síntese dos principais trabalhos relacionados ao projeto. Em seguida apresentamos a evolução do ambiente TaPrEC. Na sequência, detalhamos o Estudo de Caso realizado para avaliar TaPrEC 2.0. Depois fazemos a discussão dos resultados obtidos e finalizamos com a conclusão e próximos passos.

2. SÍNTESE DE TRABALHOS RELACIONADOS

O conceito de programação tangível, relativo ao uso de técnicas de interação tangível para construir programas de computador, nasceu na década de 1970 quando Radia Perlman criou o *Slot Machine* [15] que permitia a criação de programas físicos para a linguagem Logo. Desde então, diversos grupos de investigação criaram uma variedade de sistemas de programação tangível que suportam a construção física da estrutura de um algoritmo. Em cada um desses sistemas os usuários têm que conectar ou empilhar peças para criar um programa de computador. Alguns deles conseguem controlar objetos virtuais no monitor (*AlgoBlock* [21], *T-Maze* [22], *E-Block* [23]) e outros, objetos Lego e robôs (*Tangible Programming Bricks* [13], *Electronic Blocks* [25], *Tern* [10]).

O ambiente *AlgoBlock* permite aos usuários manipular blocos de alumínio que podem ser conectados entre si para construir um programa executável com o objetivo de deslocar um submarino virtual no monitor. *Tangible Programming Bricks* é um sistema de peças Lego empilháveis usadas para construir programas tangíveis. *Electronic Blocks* foi projetado para criar programas tangíveis empilhando blocos Lego que possuem circuitos eletrônicos. *Tern* é uma linguagem de programação tangível que combina materiais de baixo custo e visão computacional. *T-Maze* possui um conjunto de blocos de madeira interligados por ímãs e utiliza tecnologia de visão computacional para converter automaticamente os programas físicos em código digital. *E-Block* está composta por blocos de programação que têm incorporado microcomputadores SCM (*Single Chip Microcomputer*) e permitem criar programas tangíveis.

Nos ambientes apresentados distinguimos dois tipos de tecnologias: i) circuitos eletrônicos e microprocessadores; e, ii) visão computacional. Para criar um objeto tangível utilizando circuitos eletrônicos e/ou microprocessadores é preciso ter conhecimentos básicos de eletrônica. Para o caso dos ambientes que utilizam códigos de visão computacional (*Tern* e *T-Maze*) é necessário conhecimento na linguagem *Java* para customizar os objetos tangíveis. Então, apenas pessoas especialistas conseguirão configurar, consertar ou personalizar as peças tangíveis se necessário. Para o grupo que utilizam microprocessadores e circuitos eletrônicos, o investimento será maior por tratar-se de tecnologia mais sofisticada. Para o caso dos ambientes que utilizam visão computacional, o investimento se eleva se consideramos os computadores especializados necessários para processar o software de visão computacional e uma câmera com boa resolução para garantir a captura dos códigos.

Essas características complicam a extensão e customização desses ambientes. Pensando nessas limitações e com base nas contribuições dos trabalhos relacionados, propusemos e desenvolvemos o ambiente de programação tangível TaPrEC [6] com a ideia de usar tecnologia acessível a populações socioeconomicamente menos favorecidas sem fazer um enorme investimento. Com esse propósito, escolhemos utilizar um computador de placa única (*SBC - Single-Board Computer*) de baixo custo, a tecnologia RFID para os objetos tangíveis e a linguagem de programação *Scratch* para facilitar a customização do ambiente e sua apropriação por não especialistas. Neste artigo mostramos a evolução do ambiente, desde sua versão inicial, a partir de lições aprendidas no processo, e aprofunda na análise de seu uso em um Estudo de Caso ao longo de um semestre de práticas com crianças e professores.

3. EVOLUÇÃO DO AMBIENTE TAPREC

A visão do ambiente TaPrEC é tornar-se uma ferramenta que permita promover o desenvolvimento do pensamento computacional em estudantes de ensino fundamental por meio da aprendizagem de programação baseado nas Interfaces Tangíveis. Com base nos *feedbacks* das crianças e professores obtidos nas primeiras Oficinas Experimentais [5], fizemos mudanças no protótipo inicial para torná-lo o mais eficaz possível para a sua finalidade. Algumas das principais mudanças foram (Figura 1): i) criamos blocos que permitissem o deslocamento diagonal; ii) modificamos alguns símbolos dos blocos para uma melhor compreensão da funcionalidade que representam; iii) mudamos a cor dos blocos inspirados na linguagem *Scratch*.

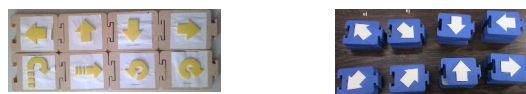


Figura 1. Evolução dos Blocos de Programação

3.1 Componentes do ambiente TaPrEC 2.0

TaPrEC está composto por um Hardware de baixo custo, os Blocos de Programação e o Software de Controle (Figura 2). O *Raspberry Pi* é o hardware principal do nosso ambiente. Para o protótipo inicial utilizamos o *Raspberry Pi* 1 Modelo B e para TaPrEC 2.0, utilizamos o modelo de segunda geração, *Raspberry Pi* 2 Modelo B. Trocamos de modelo de *Raspberry Pi* por causa das limitações (por exemplo de RAM) do primeiro para suportar o Software de Controle. A outra tecnologia utilizada no TaPrEC é a Identificação por Radiofrequência (RFID - *Radio-Frequency*

Identificação). O funcionamento dos sistemas RFID é simples: a etiqueta RFID, que contém os dados de identificação, gera um sinal de radiofrequência com esses dados. Esse sinal é detectado por um leitor RFID, responsável pela leitura da informação e o envio em formato digital para uma aplicação específica.

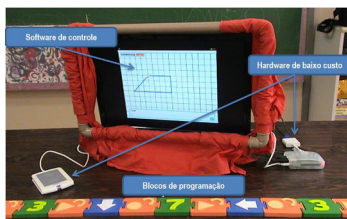


Figura 2. TaPrEC 2.0

Os Blocos de Programação são blocos coloridos de madeira semelhantes a peças de quebra-cabeças que contêm uma etiqueta RFID em um lado e em outro, um símbolo em alto-relevo. Criamos os símbolos em alto-relevo para possibilitar que uma pessoa com deficiência visual, por exemplo, possa construir programas tangíveis. As cores dos Blocos de Programação estão inspiradas na linguagem Scratch com a finalidade de permitir uma transição suave para esse ambiente de programação, posteriormente. Desenvolvido na linguagem de programação Scratch 1.4, o Software de Controle armazena os identificadores das etiquetas RFID dos Blocos de Programação. Quando o software recebe uma sequência de identificadores: i) verifica se o identificador existe na lista de códigos RFID; ii) executa o código Scratch associado a essa ação e; finalmente iii) mostra os resultados no ambiente Scratch. Para a segunda versão do software acrescentamos o deslocamento diagonal. Na Figura 3 apresentamos a arquitetura do ambiente TaPrEC e como seus três componentes interagem entre si.

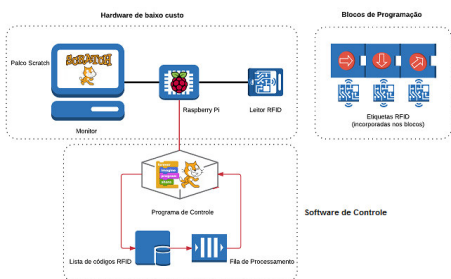


Figura 3. Arquitetura TaPrEC 2.0

Para construir um programa no TaPrEC se deve organizar os Blocos de Programação numa sequência específica: primeiro o “bloco de início”, depois os “blocos de ações” e finalmente o “bloco de fim” como ilustramos na Figura 4.

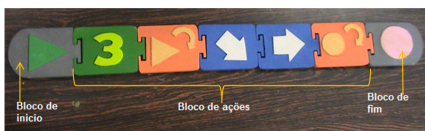


Figura 4. Ambiente proposto TaPrEC 2.0

Os blocos de controle indicam o início e o fim do programa. As informações do programa tangível são ingressadas no ambiente TaPrEC por meio do leitor RFID.

4. AVALIANDO TAPREC EM SUA EVOLUÇÃO

No segundo semestre de 2015 realizamos um Estudo de Caso para avaliação do TaPrEC 2.0 no Programa de Integração e Desenvolvimento da Criança e do Adolescente (PRODECAD), que oferece educação complementar às crianças de 6 a 14 anos. No espaço físico do PRODECAD, realizamos as atividades no espaço “ateliê”. As atividades foram agendadas formalmente pela coordenação no conjunto de atividades semanais das professoras e alunos envolvidos. O projeto teve a aprovação do Comitê de Ética em Pesquisa (CEP) da Universidade Estadual de Campinas (UNICAMP), Campinas, São Paulo.

4.1 Participantes

No total participaram 8 professores (31-54 anos) e 18 crianças (7-10 anos) da turma da tarde do PRODECAD.

4.2 Oficinas

Realizamos dois tipos de oficinas (Figura 5): Oficinas Experimentais (OE) e Oficinas Semio-Participativas (OSP). Nas OE, professores e alunos trabalharam separadamente as mesmas atividades. Ao finalizar as OE, realizamos uma OSP apenas com professores. No total realizamos 7 OE (4 com crianças e 3 com professores) e 1 OSP apenas com os professores. Filmamos as oficinas com a permissão da instituição, professores e adultos responsáveis por cada criança participante do projeto. Utilizamos os vídeos para aprimorar o ambiente, analisando comentários e sugestões dos participantes.



Figura 5. Oficinas: a) Experimentais; b) Semio-Participativas

4.3 Métodos

Na OSP trabalhamos com os artefatos do Modelo Semio-Participativo de Design com o objetivo de articular soluções para os problemas antecipados, encontrados durante a utilização de TaPrEC. Utilizamos o Diagrama de Partes Interessadas [12] para que os participantes identificassem os envolvidos direta ou indiretamente com o projeto. Os professores conversaram e identificaram as diferentes partes interessadas e passaram a localizá-las nas camadas do diagrama. Com o Quadro de Avaliação [1], os participantes anteciparam problemas e questões de cada parte interessada a respeito do uso do ambiente, sua utilização como ferramenta de apoio nas diferentes atividades da escola e as possíveis soluções. Com a Escada Semiótica [20] os participantes conseguiram entender as diferentes implicações do ambiente a partir do mundo físico até o mundo social.

Utilizamos a Autoavaliação de Emoções (SAM - *Self-Assessment Manikin*) [4] como método para que os participantes avaliassem sua resposta afetiva com relação às atividades realizadas em três dimensões afetivas: Satisfação, Motivação e Controle. Para analisar os resultados dos formulários da Autoavaliação de Emoções utilizamos uma escala de 9 pontos representados por círculos (Figura 6). Os participantes marcam com um “x” sobre qualquer círculo da escala. Determinamos a frequência de cada valor da escala em cada dimensão afetiva e calculamos o valor

porcentual destas frequências. Analisamos os resultados da seguinte maneira: os valores mais altos (7, 8 e 9) representam uma resposta afetiva positiva, os valores intermediários (4, 5 e 6) representam uma resposta afetiva neutra; e os valores mais baixos (1, 2 e 3) representam uma resposta afetiva negativa. Para determinar a resposta afetiva predominante em cada dimensão afetiva somamos separadamente as porcentagens dos valores positivos, neutros e negativos; e depois comparamos os resultados, sendo o maior dos valores a resposta afetiva predominante.

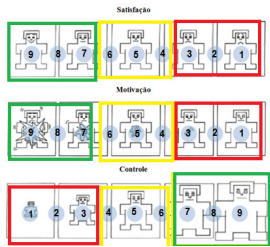


Figura 6. Método de análise da Autoavaliação de Emoções

4.4 Resultados

Os resultados estão organizados em: resultados das OE, resultados da OSP e resultados da Autoavaliação de Emoções.

4.5.1 Resultados das Oficinas Experimentais

Nas OE trabalhamos três estruturas algorítmicas: Sequências, Repetições e Procedimentos. Os participantes realizaram exercícios com o propósito de trabalhar esses conceitos de programação. Na Figura 7a) ilustramos o exercício proposto na oficina de Procedimento: desenhar as figuras geométricas utilizando os blocos de Procedimentos.

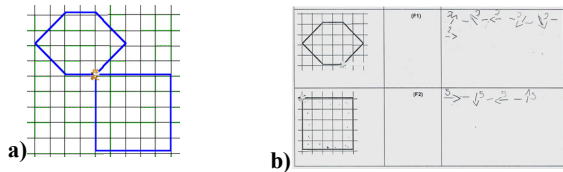


Figura 7. a) Exercício proposto; b) Estratégia de solução

Como estratégia de solução atribuímos um nome de procedimento (F1 e F2) para cada figura e os participantes deviam descrever as ações de cada procedimento utilizando inicialmente apenas blocos de deslocamento (Figura 7b). Depois continuaram com o planejamento da solução tangível utilizando a sintaxe de Procedimento de TaPrEC: definir as ações do procedimento e invocá-lo. Na Figura 8 e Figura 9 mostramos as soluções na folha de exercícios de crianças e professores respectivamente.

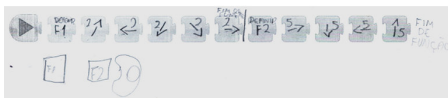


Figura 8. Solução crianças - oficina de Procedimentos



Figura 9. Solução professores - oficina de Procedimentos

Os participantes também realizaram exercícios chamados “desafios” que consistiam em analisar um programa tangível e corrigi-lo se necessário. Os desafios tinham como objetivo avaliar

principalmente as crianças na aprendizagem dos conceitos de programação e da sintaxe de TaPrEC. Esses exercícios eram apresentados na parte final da folha de exercícios de cada oficina. Na Figura 10 apresentamos o desafio da oficina de Sequências onde as crianças deviam analisar o programa tangível para criar um quadrado. Como observamos, uma equipe de crianças acrescentou e apagou blocos de deslocamento para corrigir o programa tangível.

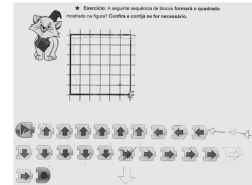


Figura 10. Desafio da oficina de Sequências – solução crianças

4.5.2 Resultados das Oficinas Semio-Participativas

Com a construção colaborativa do diagrama de Partes Interessadas (Figura 11a) os professores levantaram as partes que direta ou indiretamente influenciam ou sofrem a influência do projeto. Algumas partes interessadas levantadas pelos professores foram: os estagiários colocados na camada de Contribuição; o grupo de gestão escolar colocado na camada Fonte; os jogos pedagógicos em DVD colocados na camada de Mercado; e as ONGs colocadas na camada de Comunidade. Uma vez que os professores identificaram as partes envolvidas no projeto, os professores passaram a preencher o Quadro de Avaliação (Figura 11b). O grupo discutiu sobre questões/problemas específicas de cada uma das Partes Interessadas para a utilização de TaPrEC e as ideias ou soluções vislumbradas. Por exemplo, na camada de Operação levantaram a seguinte questão/problema: “Em que tipo de atividades se aproveitaria melhor o ambiente?”. A resposta/solução foi: “o ambiente poderia ser utilizado para ajudar nas matemáticas, alfabetização, socialização e desenvolvimento da motricidade”. Os resultados dessa discussão foram essenciais para esclarecer as demandas relativas a cada um dos envolvidos. Com a Escada Semiótica (Figura 11c), os diferentes níveis de abstração do projeto TaPrEC foram clarificados, desde o nível físico onde foram identificados e explicados cada um dos dispositivos utilizados no ambiente, até o mundo social onde por exemplo o desenvolvimento do pensamento computacional promovido por TaPrEC foi destacado.

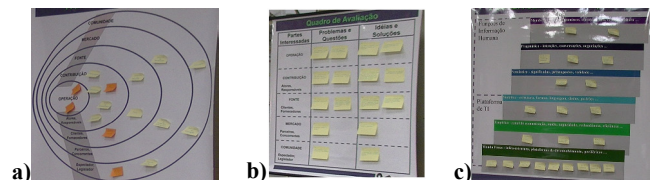


Figura 11. a) Diagrama de Partes Interessadas; b) Quadro de Avaliação; c) Escada Semiótica.

4.5.3 Resultados da Autoavaliação de Emoções

Nos resultados das respostas afetivas dos professores (Tabela 1) observamos que nas três dimensões afetivas a maioria dos professores teve uma resposta afetiva positiva durante as OE. Em particular, na oficina de Procedimentos nas dimensões de Satisfação e Motivação, 100% dos professores experimentaram uma resposta afetiva positiva.

Tabela 1. Resultados Autoavaliação de Emoções - Professores

Oficina	Satisfação	Motivação	Controle
Sequências	Positiva (75%)	Positiva (87,5%)	Positiva (62,5%)
Repetições	Positiva (71,4%)	Positiva (57,1%)	Positiva (56,6%)
Procedimentos	Positiva (100%)	Positiva (100%)	Positiva (66,7%)

Os resultados das crianças (Tabela 2) mostram que a resposta afetiva das crianças foi positiva para quase todas as OE e dimensões afetivas. Com relação à Satisfação as crianças tiveram uma resposta positiva em todas as OE, inclusive alcançando 100% na oficina de Repetições. A dimensão afetiva de Motivação foi avaliada positivamente em todas as OE, alcançando 100% nas oficinas de Repetições e Procedimentos. Para o caso do Controle, as oficinas de Sequências e Procedimentos foram avaliadas positivamente e a oficina de Repetições foi avaliada como neutra.

Tabela 2. Resultados Autoavaliação de Emoções - Crianças

Oficina	Satisfação	Motivação	Controle
Sequências	Positiva (88,9%)	Positiva (94,4%)	Positiva (76,5%)
Repetições	Positiva (100%)	Positiva (100%)	Neutra (75%)
Procedimentos	Positiva (88,9%)	Positiva (100%)	Positiva (100%)

5. DISCUSSÃO

Os resultados das equipes de trabalho mostram que os participantes aprenderam a lógica para construir um programa tangível no ambiente TaPrEC (início, sequência de ações e fim) e os conceitos de programação (Sequências, Repetições e Procedimentos). Observamos que a principal dificuldade dos participantes na oficina de Procedimentos foi entender que durante a definição do procedimento o programa tangível não executa nenhuma ação; somente quando o procedimento é invocado pelo nome executa as ações definidas previamente. Apesar dessa dificuldade, algumas equipes conseguiram resolver o Desafio dessa oficina demonstrando que entenderam o conceito e o funcionamento de Procedimento Um problema encontrado durante as OE foi que as crianças às vezes confundiam o símbolo “Fim Repetir” com o bloco de controle “Fim”. Isso provavelmente se deve ao fato de que ambos os blocos utilizam um círculo com símbolo.

As soluções (algoritmos) criadas pelos participantes deviam realizar uma ação específica (criar uma figura geométrica, percorrer o caminho certo no labirinto, etc.) e conseqüentemente observamos como os participantes encararam situações de correção de erros até alcançar o objetivo estabelecido. Algumas equipes conseguiam a solução correta na primeira tentativa, outras apresentavam erros e começavam o processo de correção. Primeiro revisavam se a solução do papel coincidia com a solução tangível. Também analisavam o programa tangível comparando sequencialmente os blocos com o resultado mostrado no monitor para descobrir o erro do programa. Durante esse processo de correção as crianças conversavam e davam sugestões entre elas, algumas sugeriam trocar um bloco por outro, outras queriam diminuir ou acrescentar a quantidade de um determinado bloco. Se depois de várias tentativas não encontravam o erro, a pesquisadora dava sugestões que ajudavam na identificação e correção do programa tangível. Quando as crianças demoravam muito tempo tentando encontrar o erro ficavam ansiosas, mas quando conseguiam resolver corretamente o exercício sentiam orgulho e alegria.

Os professores resolveram os mesmos exercícios que as crianças e quase não tiveram dificuldades. O objetivo de trabalhar

com os professores foi capacita-los na manipulação de TaPrEC para que num futuro próximo consigam criar seus próprios exercícios para trabalhar diferentes temas utilizando conceitos de programação com as crianças. Recebemos muitas sugestões dos professores que nos ajudaram a aprimorar o ambiente e a organização das OE, algumas delas foram: “*melhorar os desenhos dos blocos de programação para facilitar a interpretação*”, “*que o deslocamento na tela e o comando aconteçam ao mesmo tempo*”. Essa última sugestão foi muito importante porque implica a criação de um mecanismo de *debugging* que permita aos usuários realizar uma correção de erros mais rápida.

Incorporar o uso de uma nova ferramenta tecnológica na sala de aula não se limita a ensinar como se utiliza simplesmente; faz parte do processo trabalhar desafios relacionados a questões técnicas (como por exemplo, como manusear e dar manutenção para a tecnologia) e também a questões pragmáticas (o que fazer com a tecnologia em atividades relacionadas com o conteúdo escolar). Por isso TaPrEC não está apenas centrado na tecnologia utilizada para criar um ambiente de programação tangível, senão também no apoio às partes interessadas/envidadas.

A OSP serviu: i) para que os professores clarificassem como seria a inclusão de TaPrEC no cotidiano da escola como uma ferramenta de apoio em diferentes atividades desenvolvidas com as crianças; ii) para entender a influência das outras partes interessadas (por ex. estagiários, diretora, coordenadora); iii) e antecipar problemas e propor soluções. Decidimos realizar a OSP após finalizar as OE de modo que os professores tivessem uma visão mais ampla do funcionamento de TaPrEC e conseguissem vislumbrar melhor possíveis problemas no processo de incorporar o ambiente nas atividades com as crianças. Os professores imaginaram a utilização de TaPrEC como apoio no ensino de temas de matemática, orientação no espaço, lógica e alfabetização. Destacaram a oportunidade que tem as crianças de serem criadoras de seus próprios programas e conhecer o computador de outro ponto de vista.

Os resultados da Autoavaliação de Emoções foram quase todas positivas. Apenas na oficina de Repetições das crianças teve uma avaliação neutra. Uma possível explicação pode ser que nesse semestre a turma com a qual trabalhamos tinha crianças mais novas (7 anos) ainda em processo de alfabetização (com dificuldade para a etapa de planejamento em papel por exemplo). Já na oficina de Procedimentos, a turma foi dividida e trabalhamos apenas com as crianças maiores (8 e 9 anos) o que explicaria o excelente resultado obtido (100% positivo). Note-se que nas oficinas com crianças e professores não existe nenhuma avaliação negativa.

6. CONCLUSÃO

O ressurgimento da discussão sobre os benefícios do pensamento computacional para todos tem motivado a proposição de ambientes para a criança aprender conceitos básicos da computação. Neste trabalho apresentamos a evolução de TaPrEC, um ambiente para acesso de crianças a conceitos básicos de programação via Interface Tangível. O uso do ambiente em cenário educativo real foi experimentado em oficinas com alunos e professores ao longo de dois semestres.

Os resultados das OE demonstram que TaPrEC é fácil de aprender, uma vez que os usuários conseguiram criar corretamente programas tangíveis em pouco tempo; é eficiente no uso, isto é, os usuários conseguiram rapidamente proficiência na criação de

programas tangíveis. De acordo com os resultados da Autoavaliação de Emoções, os usuários mostraram uma resposta afetiva positiva ao usa-lo. Em termos de tempos de resposta podemos afirmar que o hardware utilizado na segunda versão do ambiente TaPrEC mostrou-se mais adequado para a programação tangível, uma vez que o tempo de resposta melhorou em comparação ao hardware inicial (de 4s para 1.5 aproximadamente). No caso dos professores, durante a OSP conseguiram antecipar questões e esclarecer suas dúvidas com relação à manutenção do ambiente e aos diferentes usos do ambiente TaPrEC como apoio às atividades cotidianas da escola. Alguns usos que os professores imaginaram estão dentro da área da matemática, também como ajuda na socialização e no desenvolvimento da motricidade e na alfabetização. Por meio da participação ativa dos professores foi possível também melhorar os Blocos de Programação inicialmente propostos, assim como sua funcionalidade. Como nosso principal próximo passo está o envolvimento das crianças e professores no desenho dos símbolos que representam as funcionalidades de Repetição e Procedimento.

AGRADECIMENTOS

Agradecemos à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), ao CNPq (#308618/2014-9) e ao Grupo Gestor de Benefícios Sociais (GGBS) pelo apoio financeiro concedido, que foi de fundamental importância para o desenvolvimento deste trabalho.

7. REFERÊNCIAS

- [1] Baranauskas, M. C. C., Schimiguel, J., Simoni, C. A. C., & Medeiros, C. M. B. (2005). Guiding the process of requirements elicitation with a semiotic approach. In 11th International Conference on Human-Computer Interaction (pp. 100-111).
- [2] Baranauskas, M. C. C., Martins, M. C., & Valente, J. A. (2013). Codesign de Redes Digitais: tecnologia e educação a serviço da inclusão social. Penso Editora.
- [3] Baranauskas, M. C. C., Martins, M. C., & de Assis, R. (2012). XO na escola e fora dela: uma proposta semi-participativa para tecnologia, educação e sociedade.
- [4] Bradley, M. M., & Lang, P. J. (1994). Measuring emotion: the self-assessment manikin and the semantic differential. *Journal of behavior therapy and experimental psychiatry*, 25(1), 49-59.
- [5] Carbajal, M. L., & Baranauskas, M. C. C. TaPrEC: Desenvolvendo um ambiente de programação tangível de baixo custo para crianças. CEP, 13083, 852.
- [6] Carbajal, M. L. Design e desenvolvimento de um ambiente de programação tangível de baixo custo para crianças. Dissertação de Mestrado. 2016
- [7] Cockburn, A., & Bryant, A. (1997). Leogo: An equal opportunity user interface for programming. *Journal of Visual Languages & Computing*, 8(5), 601-619.
- [8] Horn, M. S., & Jacob, R. J. (2007, February). Designing tangible programming languages for classroom use. In *Proceedings of the 1st international conference on Tangible and embedded interaction* (pp. 159-162). ACM.
- [9] Horn, M. S., Solovey, E. T., Crouser, R. J., & Jacob, R. J. (2009, April). Comparing the use of tangible and graphical programming languages for informal science education. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 975-984). ACM.
- [10] Horn, M. S., & Jacob, R. J. (2007, April). Tangible programming in the classroom with tern. In *CHI'07 extended abstracts on Human factors in computing systems* (pp. 1965-1970). ACM.
- [11] Ishii, H., & Ullmer, B. (1997, March). Tangible bits: towards seamless interfaces between people, bits and atoms. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems* (pp. 234-241). ACM.
- [12] Liu, K. (2000). *Semiotics in information systems engineering*. Cambridge University Press.
- [13] Mc Nerney, T. S. (1999). *Tangible programming bricks: An approach to making programming accessible to everyone* (Doctoral dissertation, Massachusetts Institute of Technology).
- [14] Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc.
- [15] Perlman, R. (1976). *How to Use the Slot Machine*. Logo Working Paper 43, Logo Laboratory, Massachusetts Institute of Technology, Cambridge.
- [16] Perlman, R. (1976). *Using computer technology to provide a creative learning environment for preschool children*.
- [17] Raspberry Pi. <https://www.raspberrypi.com>
- [18] Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., ... & Kafai, Y. (2009). Scratch: programming for all. *Communications of the ACM*, 52(11), 60-67.
- [19] ScratchJr. <https://www.scratchjr.org>
- [20] Stamper, R. (1994, July). Social norms in requirements analysis: an outline of MEASUR. In *Requirements engineering* (pp. 107-139). Academic Press Professional, Inc..
- [21] Suzuki, H., & Kato, H. (1993, August). AlgoBlock: a tangible programming language, a tool for collaborative learning. In *Proceedings of 4th European Logo Conference* (pp. 297-303).
- [22] Wang, D., Zhang, C., & Wang, H. (2011, June). T-Maze: a tangible programming tool for children. In *Proceedings of the 10th International Conference on Interaction Design and Children* (pp. 127-135). ACM.
- [23] Wang, D., Zhang, Y., Gu, T., He, L., & Wang, H. (2012, October). E-Block: a tangible programming tool for children. In *Adjunct proceedings of the 25th annual ACM symposium on User interface software and technology* (pp. 71-72). ACM.
- [24] Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- [25] Wyeth, P., & Wyeth, G. F. (2001). Electronic blocks: Tangible programming elements for preschoolers. In *IFIP TC. 13 International Conference on Human-Computer Interaction* (Vol. 1, pp. 496-503). IOC Press.