

Ensino do pensamento computacional em escola pública por meio de uma plataforma lúdica

Mauro Marcelo Mattos, Luciana Pereira de Araújo Kohler, Fabrícia Durieux Zucco, Karina Zendron da Cunha, Nelson Hein, Bruno F. F. Santos, Heitor U. C. da Silveira, Gian Carlo Giovanella, Leonardo Fronza, Lucas Eduardo Schlögl

¹ Laboratório de Desenvolvimento e Transferência de Tecnologias (LDTT)
Departamento de Sistemas e Computação (DSC)
Universidade Regional de Blumenau (FURB) - Blumenau - SC

{mattos,lpa, fabricia, karinazedron, nelson}@furb.br

Abstract. *This paper presents a platform for teaching and computational thinking, being to children from 8 years old. The platform is in development phase and tests, being already applied in approximately 60 workshops in schools of the region. Similar to game, the platform is divided in levels and is composed by exercises that involves sequential commands and repetition commands. There are some characters that interact on the scene. The platform has the goal to develop programming skills and computational thinking from the earliest stages of elementary school, thus helping the developing of cognitive issues.*

Resumo. *Este trabalho apresenta uma plataforma que é utilizada para o ensino da programação de computadores e do pensamento computacional, sendo voltada a crianças a partir de 8 anos. A plataforma ainda está em fase de desenvolvimento e testes em campo, sendo que já foi aplicada em aproximadamente 60 oficinas em escolas da região. Semelhante a um jogo, a plataforma é dividida em níveis e composta por exercícios que envolvem comandos sequenciais e de repetição. Ainda, existem alguns personagens que interagem no cenário. A plataforma tem o objetivo de desenvolver as habilidades da programação e do pensamento computacional de forma lúdica desde as primeiras fases do ensino fundamental, auxiliando assim a construção das questões cognitivas.*

1. Introdução

Conforme [Medeiros 2008] e [SBC 2006], reconhecendo o atraso tecnológico do Brasil, em 2006, a Sociedade Brasileira de Computação (SBC), com o objetivo de promover o planejamento das pesquisas de longo prazo, lançou cinco grandes desafios para as pesquisas em Ciência da Computação no Brasil para a década de 2006-2016. Destaca-se o quarto grande desafio, que trata do "Acesso participativo e universal do cidadão brasileiro ao conhecimento" reforçando a preocupação com as pessoas e a forma como a computação poderia colaborar para que os brasileiros pudessem não apenas ter acesso ao conhecimento, mas também contribuir como geradores de informação [Ferreira et al. 2017]. Passados mais de dez anos do lançamento do desafio, percebe-se que a situação ainda demanda esforços.

O Pensamento Computacional introduz uma nova abordagem para a área da Ciência Cognitiva e da Ciência da Computação, pois parte da premissa de que a inserção dos conceitos da Ciência da Computação na educação básica desenvolve uma habilidade de abstração diferente, a qual ajuda as crianças na resolução de problemas em todas as áreas da vida. A lista de conhecimentos e habilidades necessárias para o pleno exercício da cidadania no século XXI é extensa, incluindo o “Pensamento Computacional”, o qual pode relacionar-se com outras áreas [Wing 2006]. No intuito de disseminar este conceito na educação básica, diversas ações têm sido realizadas, destacando as iniciativas de introdução de conceitos de computação na Educação Básica em função do seu caráter transversal às demais áreas do conhecimento [de França et al. 2015, Bezerra and Dias 2014]. Contudo, no Brasil, o ensino de tal habilidade não integra o currículo escolar, o que amplia os desafios a serem superados para a sua adoção no ensino fundamental [França and Tedesco 2015]. Sabe-se que ainda há dificuldades em envolver os alunos na construção de conhecimento sobre lógica de programação e algoritmos [Amaral et al. 2016]. Tendo em vista que a sociedade está cada vez mais tecnológica deve haver uma conscientização para a inclusão das habilidades tecnológicas nos currículos escolares [Holanda and Bairral 2017].

Uma das maneiras de motivar estudantes é o uso da técnica de *gamification*, ou seja, a adoção de uma estratégia de interação com base na oferta de estímulos que favoreçam o engajamento de uma maneira lúdica. Dentre várias plataformas existentes para facilitar o aprendizado de lógica de programação, a plataforma desenvolvida foi concebida para minimizar as dificuldades de aprendizagem e ensino na lógica de programação por meio de um forte apelo à área de jogos, criando assim uma atmosfera facilitadora ao aprendizado. O elemento central do ambiente é a programação de um robô que vive em um mundo bidimensional junto de outros tipos de objetos, os quais também podem ser programados. Sobre esse mundo, o aluno desenvolve atividades de movimentação em quatro direções e também de detecção de obstáculos. Esses exercícios evoluem em grau de dificuldade conforme a evolução do estudante, a ponto de possibilitar o desenvolvimento de um jogo completo.

A plataforma desenvolvida tem como objetivo introduzir crianças à programação de forma intuitiva, assim como desenvolver sua capacidade de construção de algoritmos através de exercícios implementados dentro dela. Por ser uma plataforma de aprendizagem infantil é caracterizada como um jogo eletrônico para despertar interesse e ao mesmo tempo ensinar de forma descontraída conceitos básicos de programação, estimulando o raciocínio lógico.

Sendo assim, a plataforma desenvolvida é uma plataforma educacional voltada ao ensino da programação e do pensamento computacional de forma lúdica. A plataforma vem sendo melhorada em um projeto de extensão há dois anos e está sendo aplicada em turmas do segundo, terceiro, quarto e quinto ano do ensino fundamental, além de uma turma mista de alunos entre 9 e 15 anos. As atividades relacionam-se com as disciplinas de matemática, ao ensinar contas básicas; português, ao praticar a leitura e a interpretação dos objetivos a serem alcançados; e, geografia, ao praticar as direções em um tabuleiro.

O artigo segue dividido da seguinte forma. A seção 2 apresenta outras plataformas e trabalhos relacionados a apresentada neste artigo. A seção 3 descreve sobre a plataforma desenvolvida. A seção 4 discorre sobre as oficinas realizadas com a plataforma e os

resultados alcançados até o momento. Por fim, a seção 5 demonstra as considerações finais.

2. Trabalhos Relacionados

Essa seção apresente três trabalhos que relacionam-se a plataforma apresentada neste artigo, sendo o Code.Org [Code.org 2018], o LightBot [LightBot 2017] e o Made With Code [Google 2018].

Code.org é uma plataforma de uso livre que reúne espaços para alunos e educadores aprenderem ou ensinarem a programação. A plataforma fornece vários cursos de programação voltados a crianças a partir de 4 anos de idade até 18 anos, além de uma área denominada a hora do código, a qual não possui uma idade predeterminada. Nesses cursos, os alunos encontram atividades que ensinam a lógica e o pensamento computacional através de puzzles, quebra-cabeças, labirintos, até chegar a um nível de programação com blocos. Por meio da programação com blocos, o jogador recebe um desafio de deslocar algum personagem dentro de um cenário que normalmente é constituído por um labirinto. Na parte da programação com códigos, a medida em que o aluno vai passando a fase, ele é instruído com novos conceitos e comandos da programação. O ambiente fornece sempre um tutorial antes de iniciar a fase para que o aluno compreenda como jogá-la sem necessitar do auxílio de um usuário externo. Sempre após a conclusão das fases, o ambiente apresenta uma mensagem informando que a mesma foi concluída. Em algumas situações, quando o aluno não resolveu da melhor forma, ele é instruído a realizar a atividade novamente.

O LightBot pode ser jogado no navegador ou a partir de aplicativo Android ou iOS que ensina conceitos bases da programação a partir de um quebra-cabeças. O elemento central do LightBot é um robô que se movimenta sobre um labirinto utilizando os comandos ir para frente, girar para a esquerda, girar para a direita e finalizar, sendo que cada um deles é representado por um símbolo do quebra-cabeças. Para fazer o robô se movimentar, o jogador deve juntar essas peças lado a lado em um espaço pré-definido que comporta uma quantidade máxima de jogadas. Caso o jogador precise de mais movimentos do que o espaço comporta, ele precisa pensar em outra lógica de programação. Em certo momento do jogo, o jogador é instruído a respeito de subprogramas, de modo a poder repetir partes de seu quebra-cabeça construído. Sempre que o jogador finaliza, ele deve executar o seu quebra-cabeça para o robô percorrer o cenário. Todas as casas em que o robô pisa são acendidas por uma lâmpada no tom amarelo, indicando que o robô conseguiu passar pelo trajeto. Sempre que o robô atinge a última casa, a fase é finalizada e então é iniciada a próxima. Ainda, quando há comandos novos, o aplicativo apresenta um tutorial sobre seu uso.

Por fim, a plataforma Made With Code, da Google, reúne conteúdos relacionados a código-fonte, sendo fornecido um espaço para aprender conceitos básicos da programação. Nesse espaço, são fornecidos alguns jogos relacionados a alguma história ou conceito, como a Mulher Maravilha, caleidoscópio, avatar, entre outros. Cada um desses jogos, formado por algumas fases internas, tem o objetivo de ensinar conceitos da programação a partir da programação de blocos. O jogador recebe uma missão e deve completá-la utilizando uma quantidade limitada de blocos, sendo que passa para a próxima fase somente com a conclusão da lógica exata. A plataforma ensina comandos

condicionais, de repetição e sequenciais. Os códigos-fontes dos blocos são fornecidos somente no idioma em inglês, bem como as instruções de como jogá-lo.

Essas plataformas apresentadas, fundamentam a plataforma construída descrita neste artigo, sendo que ao final do mesmo, elas são relacionadas a plataforma desenvolvida.

3. Desenvolvimento

A plataforma apresentada neste artigo, denominada Furbot, é desenvolvida na linguagem de programação Java, no ambiente Eclipse IDE, utilizando recursos gráficos da biblioteca *swing* para componentes *desktop*. A equipe de desenvolvimento é interdisciplinar formada por alunos e professores dos cursos de Ciência da Computação, Sistemas de Informação, Publicidade e Propaganda, Letras e Matemática. Nesse contexto, cada membro tem seu papel fundamental, de modo que os membros dos cursos de Ciência da Computação e Sistemas de Informação são responsáveis pela programação da plataforma. Os membros do curso de Publicidade e Propaganda são responsáveis pelo leiaute da plataforma, tornando-a mais atrativa. Os membros do curso de Matemática são responsáveis pela construção de enunciados que envolvam práticas matemáticas. E por sua vez, membros da equipe de Letras trabalham com uma melhor interpretação dos enunciados, auxiliando no processo de letramento digital das crianças.

A plataforma atual é desenvolvida utilizando como base um *framework* Java existente há aproximadamente 10 anos que foi concebida para o ensino da programação aos alunos da graduação dos cursos de Ciência da Computação e Sistemas de Informação. Dessa forma, pode-se otimizar o processo de desenvolvimento da plataforma, tornando-a lúdica e atrativa se preocupando principalmente com a camada visual e de interação com o usuário. Por ser em Java, a plataforma é independente de Sistema Operacional e não demanda de hardwares potentes para sua execução. Ainda, por ser *desktop* não é necessário o uso da internet para que o jogo funcione.

A plataforma é composta por um jogo que ensina os conceitos iniciais de programação de forma lúdica, além de permitir que alunos criem seus próprios mapas. A Figura 1 apresenta a tela inicial do sendo que há três opções: Jogar, Editor e Sair. Conforme estudos realizados pela equipe de Publicidade e Propaganda do time, pode-se observar que as cores utilizadas para o leiaute são cores primárias e que chamam a atenção das crianças.



Figura 1. Tela Inicial

Ao clicar no botão Jogar, a criança é encaminhada para a tela com os níveis de dificuldade, sendo: fácil, médio e difícil. Por sua vez, ao selecionar um dos níveis, são exibidos os exercícios que a criança pode jogar conforme. Para cada nível há uma quantidade diferenciada de exercícios. Após selecionar um exercício, é apresentado o enunciado da atividade, sendo este o objetivo a ser cumprido. O objetivo dos exercícios é realizado pelo personagem principal da plataforma, sendo o robô denominado Furbot.

Ao iniciar o exercício, a criança visualiza o mundo bidimensional com os elementos posicionados por ele, conforme a atividade e o objetivo proposto. No mundo podem conter alienígenas, que são inimigos do Furbot, ou seja, o robô não pode passar sobre eles; anjos, que são os amigos do Furbot; paredes, utilizadas para criar labirintos e desvios físicos; tesouros, utilizados para pontuar, sendo normalmente a coleta destes o objetivo principal da atividade; e números, utilizados para problemas matemáticos nas fases mais avançadas. A Figura 2 demonstra um mundo populado com alguns desses elementos. É importante ressaltar que atualmente o tamanho máximo do mundo é um mundo quadrado 12x12. Todo o cenário construído para ser exibido no jogo é modelado em cima de um arquivo XML identificado por *tags* que definem aonde os personagens serão posicionados. Na parte lateral direita são apresentados contadores de objetos coletados pelo mundo. O valor é atualizado dinamicamente conforme o robô passa por cima desses objetos. Ainda, são apresentadas duas setas indicando qual é o lado esquerdo e qual é o lado direito, pois essa ainda é uma das dificuldades encontradas pelas crianças e é a forma com a qual o robô se move pelo mundo. Na parte inferior tem-se um botão com uma seta para voltar que retorna as fases do jogo, um botão com uma lupa que é utilizado para a resolução do exercício, um botão *play* que inicia o jogo e uma interrogação que apresenta o enunciado novamente.

Para iniciar a resolução da atividade deve-se clicar no botão *play*. Feito isso, a resolução ocorre conforme o nível de dificuldade selecionado. No nível fácil a criança é iniciada ao pensamento computacional, sendo instruída a pensar de forma sequencial. Nessa fase são utilizadas as setas do teclado para movimentar o robô pelo tabuleiro com o objetivo de completar o exercício. Ao finalizar a atividade, a criança deve pressionar o botão pause que é atribuído no local do *play*. Após realizar a ação ela visualizará um código-fonte gerado que corresponde as setas do teclado utilizadas para percorrer o cenário, conforme é visualizado na Figura 2. Pode-se visualizar na figura que o caminho em que o robô andou é colorido de amarelo, mostrando o rastro executado pelas setas. Ainda, o código que aparece demonstra a sequência de passos em forma de um programa de computador, acostumando a criança com o código-fonte.

Na fase médio a criança é instruída a resolver a atividade utilizando o código-fonte. Para percorrer o caminho com o robô, ela precisa utilizar os comandos `andarDireita()`, `andarEsquerda()`, `andarAcima()`; ou `andarAbaixo()`. Esses comandos são os mesmos apresentados como solução na fase fácil. Para o preenchimento dos comandos, deve-se clicar no botão de lupa que exibe a tela de código-fonte. Os comandos devem ser feitos sequencialmente, um abaixo ao outro. Dessa forma, além de continuar a pensar sequencialmente, a criança treina a prática inicial da programação. Isto é, ela é instruída a fazer chamadas de funções assim como em linguagens tradicionais. Após concluir o código, deve-se clicar no botão executar e então na tecla espaço do teclado para que o robô faça sua movimentação no cenário. O processo da movimentação ocorre exatamente

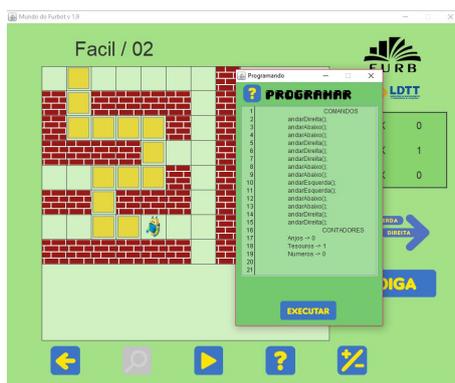


Figura 2. Exercício da fase fácil concluído

igual ao nível fácil. No entanto, ao realizar a execução é apresentada uma tela semelhante a um depurador de código-fonte, que apresenta a linha que o robô está executando em cada momento.

Por fim, no nível difícil, o aluno é instruído a lógica de repetição. Nessa fase é habilitada uma tela que contém o comando enquanto, junto com campos de seleção para que seja selecionado o que verificar e em qual direção. As opções de verificação são: enquanto não é vazio; enquanto é vazio; e, enquanto não é fim. Para todas as opções deve-se indicar qual a direção. Essa funcionalidade pode ser visualizada na Figura 3. Dessa forma, a criança começa a pensar sequencialmente e não precisa mais repetir tanto código-fonte como nas atividades iniciais. Todos os demais processos continuam iguais aos níveis anteriores.

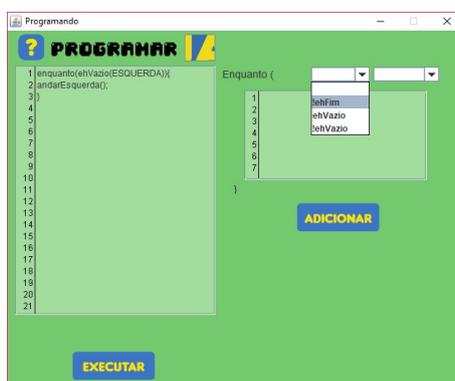


Figura 3. Comando de repetição enquanto

Em todos os casos, para completar a fase corretamente o aluno deve realizar a coleta de todos os tesouros disponíveis no mundo. Ainda, o arquivo XML que configura o exercício fornece a solução para o código-fonte. Dessa forma, o aluno sabe se atingiu o objetivo, quando coleta todos os tesouros, pois após a coleta deles pelo robô a fase é encerrada, apresentando uma mensagem parabenizando o jogador por completar a missão. Além disso, o jogador sabe o quão bem feito foi seu código-fonte com base na pontuação recebida após completar a fase. Em caso de produzir um código-fonte idêntico ao definido no arquivo de configuração, ele recebe uma bonificação maior. Esse código-fonte definido sempre está relacionado ao melhor código possível, ou seja, aquele que possui uma menor

quantidade de linhas de programação.

Além da opção de jogar, crianças e professores podem criar novas atividades. Essas atividades, no entanto, não são adicionadas de forma automática na plataforma. Para isso, por enquanto é necessário recompilar o arquivo com os novos exercícios gerados, sendo este também um trabalho futuro a ser desenvolvido. Para isso, deve-se clicar no botão editor apresentado na figura inicial. A tela do editor possui a área para construção do mundo como ponto central e vários botões no lado direito com os objetos a serem incluídos nele. Primeiramente deve-se clicar no botão Inserir Tabela e digitar a quantidade de linhas e colunas desejadas, bem como o enunciado da atividade. Em seguida, os botões são habilitados. Pode-se adicionar um robô por exercício e N elementos dos demais. Para colocar o elemento no mundo, basta clicar sobre ele uma vez e em seguida clicar no mundo. Para facilitar a inserção de múltiplos elementos, pode-se arrastar com o mouse pressionado sobre as células, que todas receberão o mesmo objeto. Com relação aos números, para alterar seu valor, basta clicar sobre o número após inserido no mapa. Caso seja necessário remover algum objeto, deve-se clicar na vassoura e em seguida nos objetos a serem apagados do mundo. Para salvar, deve-se clicar no símbolo de *check*. Como a plataforma ainda está em desenvolvimento, não há um link para uso livre da mesma. Contudo, os alunos que querem levar a plataforma para brincar em casa, recebem um link do Google Drive ou o aplicativo em pendrive.

4. Oficinas realizadas e resultados alcançados

Durante o ano de 2017, a plataforma desenvolvida foi testada com crianças de ensino fundamental do terceiro e quarto ano em uma escola estadual da cidade. Todas as semanas, as crianças tiveram uma hora de aula com a plataforma na qual trabalhava com atividades relacionadas a matemática, unindo a programação e o pensamento computacional. Ao todo foram 22 oficinas com cada turma, totalizando 44 oficinas. Nesse primeiro ano de teste da plataforma, utilizou-se do método de pesquisa-ação, o qual permitiu evoluir a plataforma a medida em que as necessidades eram encontradas em campo, unindo as crianças no processo de colaboração desse desenvolvimento. Essa pesquisa-ação encontra-se relatada em [Mattos et al. 2018].

Já no ano de 2018, continuou-se as oficinas com a plataforma na mesma escola, porém nas turmas de segundos, quartos e quintos anos do ensino fundamental, no qual alguns alunos já haviam utilizado a plataforma no ano anterior. Ainda, em uma segunda escola nos terceiros e quartos anos do ensino fundamental. Dessa forma, a faixa etária de uso variou entre 7 e 13 anos. Para a realização das oficinas, utilizou-se da metodologia de estudo de caso, sendo que as aulas foram sempre planejadas previamente, em seguida aplicadas e posteriormente analisadas. As oficinas ocorreram a cada 15 dias com cada uma das turmas, com duração de uma hora cada.

Em cada uma das oficinas realizadas foi apresentado o objetivo da aula, sendo que este é evoluído a cada novo encontro. Para nivelar a turma e ambientá-las na plataforma, inicialmente trabalhou-se com a lógica sequencial utilizando elementos e mapas simples e, com o passar das aulas, mapas mais elaborados e com uma maior quantidade de elementos. Ainda, por ser uma plataforma em desenvolvimento, a cada aula foram apresentados novos recursos, corrigidos problemas encontrados e coletados novos dados e funcionalidades a serem desenvolvidas para melhorar a habilidade e o engajamento no

jogo.

O desempenho das crianças era observado a cada encontro pelos participantes do projeto, sendo que os comportamentos eram anotados para uma posterior análise, bem como eram anotados a quantidade de exercícios realizados a cada encontro, assim como dificuldades encontradas. Ainda, para avaliar o conhecimento adquirido após cada encontro, eram realizadas perguntas simples a respeito da atividade executada no dia, como por exemplo “o que você aprendeu hoje?”.

Ao analisar esses dados, percebeu-se uma evolução com a maioria dos alunos. Muitos dos alunos não eram alfabetizados ainda, contudo sabiam identificar letras e copias. Ainda, alguns alunos nunca tinham utilizado um computador, logo, não sabiam digitar. Ainda assim, todos os alunos conseguiram realizar as atividades com sucesso. Para que entendessem a lógica do jogo foi necessário somente uma aula com o uso do nível fácil, ou seja, a partir das setas do teclado, com um total de 30 exercícios. Após essa aula, os alunos já conseguiram realizar o nível médio corretamente, demonstrando que compreenderam os princípios do pensamento computacional ao realizar a movimentação com o robô. Essa afirmação pode ser realizada, pois o pensamento computacional inclui o poder de abstração, análise e automação, de modo que o aluno conseguiu analisar o cenário para produzir o código-fonte correspondente que gerasse a automação para o robô caminhar pelo seu labirinto. A abstração está relacionada, neste caso, ao uso do código-fonte ou das setas de teclado para abstrair o caminhar do robô, além de que as direções são abstrações do mundo real. Acredita-se que a evolução não ocorreu para todos os alunos até o momento da aplicação das oficinas, pois cada indivíduo tem o seu próprio tempo de aprendizado e evolução, sendo que em torno de 10% dos alunos demoram mais para ir para o código-fonte e para conseguir resolver as atividades de forma individual.

Ao serem questionados a respeito do que aprenderam na aula, foram produzidas respostas com “aprendi que é necessário pensar antes de agir”, “aprendemos a utilizar nosso raciocínio”, “aprendemos que é importante ter paciência, para pensar e então resolver a atividade”. Esses relatos mostraram que a plataforma realmente alcançou o objetivo inicial de introduzir o pensamento computacional, sendo relatos não esperados previamente, ainda mais por crianças com essa faixa etária. De acordo com a observação da equipe durante as aulas, a nova interface projetada e os novos exercícios disponibilizados, fizeram com que as crianças progredissem aos poucos, compreendendo os comandos da programação, sendo mais fácil de completar as missões. Ainda, outra observação realizada, foi que crianças que já sabiam ler e já utilizaram o computador previamente, compreenderam que poderia ser realizado um comando de repetição para que o robô desse mais passos antes mesmo dessa instrução ser apresentada, concluindo-se mais uma vez que o objetivo do ensino do pensamento computacional foi alcançado.

Relacionando a plataforma desenvolvida com outras semelhantes encontradas no mercado e apresentadas na seção de Trabalhos Relacionados, esta plataforma possui o principal diferencial que é a digitação do código-fonte, ao invés do uso de estrutura de blocos. Ao digitar o código-fonte, o aluno está mais próximo do que realmente é um código de programação, acostumando-se com chamada de métodos e laços de repetição definidos por blocos de início e fim (no caso as chaves). Ainda, semelhante ao Code.org que ao finalizar uma tarefa a plataforma exibe a pontuação e se o jogador conseguiu completar da melhor forma a fase, a plataforma desenvolvida exibe a pontuação e apresenta

uma mensagem parabenizando por completar a missão. Relacionando ao ambiente LightBot, a plataforma introduz os conceitos iniciais e as direções a serem utilizadas por meio das setas do teclado, uma vez que o LightBot utiliza blocos com simbologia de setas. No entanto, esta plataforma não ensina conceitos de procedimentos, uma vez que o LightBot ensina. Por fim, relacionando a plataforma ao Made With Code, da Google, esta plataforma define a quantidade de comandos necessários para o aluno completar a tarefa, porém essa informação não é exibida de forma explícita ao aluno. A informação da quantidade de linhas de programação necessária fica no arquivo de configuração do XML e é utilizada somente para pontuar mais o jogador caso utilize somente aquela quantidade. Já a plataforma Made With Code obriga o jogador a resolver a atividade somente com aquela quantidade de comandos pré-definidos. Não se optou por realizar dessa forma, pois a plataforma é para ensinar o pensamento computacional, sendo que é possível resolver um mesmo problema de várias maneiras, embora sabe-se que possa não ser a melhor maneira. Para não desmotivar o aluno, decidiu-se permitir que o mesmo conclua a fase e receba bonificações extras ao alcançar a quantidade mínima de linhas de código.

5. Considerações finais

Este trabalho apresentou de forma geral a construção e uso de uma plataforma como objeto educacional para o ensino da programação de computadores e do pensamento computacional. Como citado no início do artigo, o pensamento computacional é uma área do saber que está relacionada a vida cotidiana das pessoas, pois está relacionado a resolução de problemas cotidianos e a tomada de decisões. Dessa forma, se ele for incluído na educação básica, as crianças poderão desenvolvê-lo com maior facilidade.

Por ser uma plataforma lúdica e em forma de um jogo, ela atrai a maioria das crianças que o utilizam. A partir das oficinas realizadas, pode-se notar que as crianças gostam dos desafios proporcionados por ela e sempre esperam atividades mais desafiadoras, a medida em que vão completando as fases.

Através de relatos ouvidos das professoras regentes das turmas, percebeu-se uma melhora na concentração de alguns alunos, além da melhoria em habilidades matemáticas devido a alguns problemas estarem relacionados a conteúdos da disciplina. Sabe-se que com a contribuição das pedagogas regentes, ainda é possível atrelar melhor as diferentes disciplinas a plataforma, pois os exercícios são configuráveis conforme a necessidade demonstrada.

Uma vantagem da plataforma em relação a outras ferramentas disponíveis no mercado é que ela é desktop e leve, rodando nas máquinas disponíveis pelas escolas estaduais, sendo com o sistema operacional Linux Educacional Multiterminal e, muitas vezes, sem acesso à internet. Ainda, por ser construída em português, os alunos conseguem compreender e treinar suas habilidades de interpretação de texto ao ler os enunciados fornecidos.

Como a plataforma ainda está em desenvolvimento, há várias melhorias a serem realizadas, sendo algumas já mencionadas durante o artigo. Pretende-se principalmente melhorar a jogabilidade, incluindo novas características de *gamification*.

Dessa forma, a relevância tecnológica dessa plataforma é ensinar conceitos de programação que permitem o desenvolvimento do pensamento computacional desde o ensino fundamental. Ainda, como relevância social, a plataforma trabalha com a inclusão

social, pois desmistifica o uso das tecnologias computacionais e desenvolve habilidades relacionadas ao uso do computador.

Referências

- Amaral, E. M. H., Medina, R. D., and Tarouco, L. M. R. (2016). Processo de ensino e aprendizagem de algoritmos integrando ambientes imersivos e o paradigma de blocos de programação visual. In *Anais dos Workshops do V Congresso Brasileiro de Informática na Educação (CBIE)*, pages 20–29.
- Bezerra, F. L. and Dias, K. L. (2014). Programação de computadores no ensino fundamental: experiências com logo e scratch em escola pública. In *Anais do XXII WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO*, pages 1515–1524.
- Code.org (2018). Todos os alunos, em todas as escolas, devem ter a oportunidade de aprender ciência da computação. disponível em: <https://code.org/>.
- de França, R. S., dos Santos Ferreira, V. A., de Almeida, L. C. F., and do Amaral, H. J. C. (2015). *A disseminação do pensamento computacional na educação básica: lições aprendidas com experiências de licenciandos em computação*. EDUPE, 1th edition.
- Ferreira, S. B. L., Sacramento, C., da Silva Alves, A., Leitão, C. F., Maciel, D. d. R., Matos, S. N., and Britto, T. C. P. (2017). Accessibility and digital inclusion: Utopia or a great challenge? In *Proceedings of the XVI Brazilian Symposium on Human Factors in Computing Systems, IHC 2017*, pages 68:1–68:6, New York, NY, USA. ACM.
- França, R. S. and Tedesco, P. C. A. R. (2015). Desafios e oportunidades ao ensino do pensamento computacional na educação básica no brasil. In *Anais do workshop de ensino em pensamento computacional*, pages 1464–1473.
- Google (2018). Things you love are made w/ code. disponível em: <https://www.madewithcode.com/>.
- Holanda, A. C. A. and Bairral, M. A. (2017). Possibilidades do e-learning em projetos interdisciplinares no ifac. In *Anais dos Workshops do VI Congresso Brasileiro de Informática na Educação (WCBIE)*, pages 409–413.
- LightBot (2017). Solve puzzles using programming! disponível em: <http://lightbot.com/>.
- Mattos, M. M., de Araújo Kohler, L. P., nad Lucas Eduardo Schlogl, H. U. C. S., Giovanella, G. C., Santos, B. F. F., Fronza, L., Zucco, F. D., Hein, N., de Oliveira, G. C., da Cunha, K. Z., and Sartori, A. (2018). Uma pesquisa-ação sobre o desenvolvimento do pensamento computacional com crianças. In *Anais do Workshop de Informática na Educação (WIE)*.
- Medeiros, C. B. (2008). Grand research challenges in computer science in brazil. *Computer*, 41(6):59–65.
- SBC (2006). Grandes desafios da pesquisa em computação no brasil 2006-2016. In *Relatório Sociedade Brasileira de Computação*. Sociedade Brasileira de Computação.
- Wing, J. M. (2006). Computational thinking. *Commun. ACM*, 49(3):33–35.