

DESENVOLVIMENTO DO PROTÓTIPO DE UM JOGO PARA O ENSINO DE LÓGICA BÁSICA DE PROGRAMAÇÃO NO ENSINO FUNDAMENTAL.

Carlos Lacerda Rios Neto¹
carloslacerda.92@gmail.com

Luciano Rodrigo Ferreira²
luciano.ferreira@uniube.br

Antônio Luiz Ferreira Junior³
antonio.ferreira@uniube.br

RESUMO

O trabalho a seguir apresenta o desenvolvimento do protótipo de um jogo para auxiliar o ensino de lógica básica de programação para alunos do ensino fundamental, já que não há uma disciplina que introduza, de maneira básica, a lógica de programação para esses alunos. Com o desenvolvimento do protótipo em questão, tem-se como objetivo despertar o interesse do aluno do ensino fundamental em áreas relacionadas à computação e promover um desenvolvimento a longo prazo em habilidades exigidas pela programação, como o raciocínio lógico-matemático, por exemplo. O projeto tem como foco apenas o desenvolvimento do protótipo.

Palavras-chave: Lógica de Programação. Ensino Fundamental. Protótipo. Jogo.

DEVELOPMENT OF A GAME PROTOTYPE TO TEACH BASIC PROGRAMMING LOGIC TO ELEMENTARY SCHOOL STUDENTS.

ABSTRACT

The following project presents the development of a game prototype to aid the teaching of basic programming logic to elementary school students, since there is no discipline that introduces, in a simple way, programming logic to these students. With the

¹ Graduando do curso de Engenharia de Computação da Universidade de Uberaba.

² Professor orientador da Universidade de Uberaba, graduado em Sistemas de Informação.

³ Professor coordenador da Universidade de Uberaba, graduado em Engenharia de Produção.

construction of the prototype, it is intended to arouse the student's interest in computing-related fields and promote a long term improvement in programming required skills, like logical reasoning, for instance. This project's focus is only the development of the prototype.

Keywords: Programming Logic. Elementary School. Prototype. Game.

1. INTRODUÇÃO

Inúmeras são as disciplinas que compõem o plano de ensino básico das escolas do país. Todas são, indubitavelmente, primordiais para o desenvolvimento intelectual do aluno. Entretanto, as opções de curso superior para os alunos, ao concluírem o ensino médio, também são vastas. Ao optar por uma graduação, o aluno pode se deparar com conteúdos parcial ou totalmente novos, o que pode gerar uma certa dificuldade em seu aprendizado. É o que pode ocorrer nos cursos relacionados à computação.

O principal fator de motivação para o desenvolvimento desse trabalho, é a ausência de uma disciplina que apresente a lógica de programação na grade curricular da educação básica que estimule o aluno a raciocinar de maneira sistematizada e abstrata. Essa falta, na maioria das vezes, resulta em dificuldade do aluno ingressante em um curso na área de computação em compreender os novos conteúdos exigidos, como é o caso da lógica de programação. O trabalho realizado por Campos (2009), mostra que, em quatro instituições de ensino superior, mais de 60% dos alunos da disciplina de lógica de programação ou similar eram reprovados a cada semestre. Desses, mais de 65% já haviam sido reprovados mais de uma vez.

Dessa maneira, surge o seguinte questionamento: como apresentar a lógica básica de programação no ensino fundamental de forma simples e atrativa?

Com o desenvolvimento de um protótipo de um jogo, busca-se despertar o interesse do aluno do ensino fundamental nas áreas relacionadas à computação e promover um aumento, à longo prazo, em sua taxa de aprendizado em campos que se referem ao raciocínio lógico.

2. FUNDAMENTAÇÃO TEÓRICA

Neste tópico, será feita uma revisão do conteúdo envolvido no desenvolvimento desse trabalho, tanto no campo da lógica e da programação, como da educação.

2.1. Lógica

De acordo com Forbellone e Eberspächer (2005), é possível relacionar a lógica com a correção do pensamento, pois uma de suas preocupações é determinar quais operações são válidas e quais não são, fazendo análises das formas e leis do pensamento. Também pode ser definida como a “arte de bem pensar”, que é a ciência das formas do pensamento. Visto que a forma mais complexa do pensamento é o raciocínio, a lógica estuda a correção do raciocínio. Pode-se ainda dizer que a lógica tem em vista a ordem da razão. Com isso, entende-se que a razão pode funcionar desordenadamente. Por isso, a lógica estuda e ensina a colocar “ordem no pensamento”.

Erroneamente, a lógica tem sido associada apenas à matemática. Porém, ela é aplicável em inúmeras áreas do conhecimento, e está sempre presente no nosso cotidiano. O simples ato de expressar o pensamento, seja através da fala ou da escrita, pode ser considerado uma prática de lógica, pois faz-se necessário estabelecer uma ordem de ideias, isto é, uma sequência lógica no pensamento, para que possamos expressarmos-nos corretamente.

Exemplos:

a. Igor é mais novo que Vitória.

Vitória é mais nova que João.

Logo, Igor é mais novo que João.

b. O carro está sem combustível.

Precisa-se usar o carro.

Portanto, primeiro deve-se abastecer o carro, para então usá-lo.

2.2. Lógica de Programação

O raciocínio, além de ser abstrato, pode ser considerado universal. Isso significa que, em qualquer local do globo, ele será o mesmo. A única mudança será a forma com a qual ele será expresso, seja verbal, seja escrita, pois os idiomas variam conforme a região em que um indivíduo se encontra.

Pode-se dizer o mesmo em relação à Lógica de Programação, pois, apesar de o raciocínio ser o mesmo, existem diversas linguagens de programação, que funcionam como ‘idiomas’ que traduzem o pensamento referente à programação computacional. Com isso, define-se como sendo:

O uso correto das leis do pensamento, da “ordem da razão” e de processos de raciocínio e simbolização formais na programação de computadores, objetivando a racionalidade e o desenvolvimento de técnicas que cooperem para a produção de soluções logicamente válidas e coerentes, que resolvam com qualidade os problemas que se deseja programar. (FORBELLONE; EBERSPÄCHER, 2005, p. 2)

2.3. Algoritmos

Segundo Sousa Filho e Alexandre (2014), há divergências sobre a origem da palavra “algoritmo”. A mais difundida provém do sobrenome, Al-Khwarizmi, do matemático, astrônomo, astrólogo, geógrafo e autor persa, Mohamed ben Musa. Suas contribuições abrangem os mais diversos campos, como a álgebra, astronomia e astrologia.

Pode-se definir algoritmo como uma sequência finita, ordenada e não ambígua de passos para solucionar determinado problema ou realizar uma tarefa (SOUSA; DIAS JÚNIOR; FORMIGA, 2014). Apesar de ser um conceito primordial em computação, a aplicação de algoritmos não se restringe apenas a essa área. É possível encontrá-los em exemplos corriqueiros do cotidiano, como nos exemplos abaixo, e na figura 1.

Exemplo 1: Ligar computador

- conectar o cabo de energia do estabilizador na tomada;
- ligar o estabilizador;
- ligar a CPU;
- ligar o monitor;
- usar o computador;

Exemplo 2: Ligar computador com tela de autenticação de usuário

- conectar o cabo de energia do estabilizador na tomada;
- ligar o estabilizador;
- ligar a CPU;
- ligar o monitor;
- se aparecer tela de autenticação de usuário, então
 - digitar usuário;
 - digitar senha;

- usar o computador;
- senão
 - usar o computador;

Figura 1 - Exemplo de algoritmo em linguagem C

```
1 #include <stdio.h>
2 #include <iostream>
3
4 main() {
5
6     int numA, numB, soma;
7
8     printf("Digite o Primeiro Numero a ser somado: ");
9     scanf("%i", &numA);
10
11    printf("\nDigite o Segundo Numero a ser somado: ");
12    scanf("%i", &numB);
13
14    soma=numA+numB;
15
16    printf("\nO resultado da soma eh: %i \n\n", soma);
17
18    system("pause");
19 }
```

Fonte: <http://www.teclati.com>

2.4. O Método de Educação Tradicional

Não é novidade que o ensino, principalmente da matemática, que é a matéria comumente mais relacionada ao raciocínio lógico, é visto, na maioria das vezes, como algo complexo e maçante. Pode-se relacionar esse fato à maneira com a qual as aulas das disciplinas em questão são ministradas.

A maioria das escolas do Brasil aplicam o método tradicional de ensino. Nessa metodologia, todo o ensino é centrado no professor, fazendo com que este assuma a figura de sujeito ativo, enquanto o aluno toma uma postura mais passiva, pois apenas recebe a informação.

Esse modelo de ensino foi difundido pelas escolas públicas francesas a partir do Iluminismo (séc. 18). Pretendiam universalizar o acesso ao conhecimento para formar cidadãos. É uma filosofia que valoriza a quantidade de conteúdo ensinada, e as escolas que a adotam são voltadas para o sucesso do aluno em provas como o Exame Nacional do Ensino Médio (Enem) e o vestibular. (ROSA, 2015, p. 1)

Na técnica em questão, o professor é visto como o proprietário do conhecimento. Cabe ao aluno, assimilar o conteúdo dado em sala de aula. Normalmente, não há espaço para muitos questionamentos sobre a matéria. Logo,

ele tem metas a cumprir dentro de determinados prazos, e é avaliado periodicamente por meio de avaliações.

Uma característica marcante desse sistema de ensino, é também uma de suas maiores falhas: enquanto o professor “narra” o conteúdo, o aluno é responsável por fixar, memorizar e repetir o que lhe é passado. Dessa forma, o verdadeiro significado das lições passa, na maioria das vezes, despercebido. Segundo Freire (1970), nesse tipo de educação, o educador faz “comunicados” que os educandos recebem pacientemente, memorizam e repetem.

Fica evidente que na metodologia abordada, os alunos não aprendem o conteúdo. O saber envolvido é mínimo, pois limita-se apenas ao que o professor apresentou em sala de aula. Ainda de acordo com Freire (1970), essa visão de ensino, anula o poder criador dos educandos ou o minimiza, estimulando sua ingenuidade e não seu senso crítico, além de satisfazer o interesse dos opressores: para estes, o fundamental não é o desnudamento do mundo, a sua transformação. Os alunos são vistos como seres da adaptação, do ajustamento.

Alguns autores sugerem que o professor faça um planejamento centrado no aluno, e não em sua própria figura. Ele deve buscar meios de tornar a aula mais interessante, usar de recursos que facilitem a compreensão da criança, estabeleçam relações entre imagens significativas de pessoas, objetos e lugares com o conteúdo que está sendo desenvolvido e estimulem a imaginação e a criatividade.

O professor deve cuidar de planejar os procedimentos de ensino, de forma a estimular no aluno a leitura, a criatividade, a observação, a escrita, a ser provocado a investigação e busca de soluções de problemas, com vista a uma participação ativa na construção de seus conhecimentos e transformações de si próprio e do meio. (PARONETO, 2010, p. 152)

Figura 2 – Alunos em sala de aula



Fonte: <http://sinprominas.org.br>

2.5 O Ensino Lógico-Matemático e a criança

O ensino da matemática, e, conseqüentemente, a aplicação do raciocínio lógico, no ensino básico, esteve sempre muito ligado apenas às quatro operações básicas. Não há, na maioria dos casos, durante a exposição do conteúdo, exemplos de sua aplicabilidade no dia a dia. De acordo com Gomes (2012), as noções matemáticas são construídas pelas crianças a partir das experiências proporcionadas pelas interações com o meio. Esses conhecimentos são adquiridos e aprimorados no convívio com outras pessoas e com o contato com materiais concretos. A relação entre as práticas educacionais coerentes e o cotidiano permite a formação de um indivíduo que não só conhece, mas sim que pratica e entende a realidade vivida por ele.

A matemática e a lógica estão presente em várias situações do cotidiano e a criança tem total liberdade para explorar o mundo ao seu redor, pois são nessas situações corriqueiras que é possível identificar seus conceitos e noções em sua totalidade e com maior clareza.

Quando falamos em conhecimento lógico-matemático, estamos nos referindo a uma ação do sujeito e a uma abstração reflexiva que permite a ele tomar consciência das relações que estabelece tanto com os objetos quanto com os eventos do seu cotidiano. O conhecimento lógico-matemático define-se como uma invenção. (BORGES, 2003, p. 57)

Figura 3 – A Criança e o Ensino Lógico-Matemático



Fonte: <http://www.vermelho.org.br>

2.6. A Ideia de Jogo

Há muito tempo o homem já jogava. Alguns pesquisadores registram que o primeiro jogo foi criado por Adão. Conta-se que este juntava pedras e separava as mesmas por cores. Então, observava os pássaros que voavam e separava uma pedra para cada cor que via nas aves. O processo então recomeçava.

Segundo Almeida (1998), desde os tempos remotos os jogos eram utilizados como forma de motivar os alunos. Para ele, os humanistas começaram a perceber o valor educativo dos jogos e, assim, os colocaram em prática nos colégios jesuítas. Na Grécia Antiga, Platão já afirmava que os primeiros anos de uma criança devem ser ocupados com jogos educativos numa prática matemática lúdica.

Os jogos desempenham uma função imprescindível no contexto educacional. O aluno, ao jogar, fixa o conteúdo já estudado, estimula o pensamento criativo e favorece a aquisição do conhecimento. As ações tomadas durante o jogo representam momentos de decisão importante para a criança, pois é onde ela irá aplicar todo o conhecimento que aprendeu.

O jogo é uma prática que auxilia o desenvolvimento infantil, constrói e potencializa o conhecimento. Ludicidade e jogo são a combinação ideal, pois por meio dessa parceria é dada a criança a possibilidade de uma participação ativa e natural, em que a natureza lúdica e prazerosa, inerente a diferentes tipos de jogos, tem servido de argumento para fortalecer essa concepção. (GOMES *et al*, 2012, p. 139)

Figura 4 – Alunos e jogos educacionais



Fonte: <http://www.colmagno.com.br>

2.7. Trabalhos Anteriores

Um trabalho interessante foi realizado por Oliveira *et al.* (2014). Trata-se de um relato das atividades de extensão que promoveram o ensino de lógica de programação para alunos do ensino fundamental no interior de Pernambuco, no município de Garanhuns. Foi utilizado o livro *Computer Science Unplugged*, do autor Tim Bell, publicado no ano de 2011, com o auxílio do ambiente *Scratch*⁴. Os autores contam que apesar da evasão de alguns alunos, os resultados foram otimistas, visto que o percentual de desistência foi de 31,25%, e os que concluíram conseguiram alcançar os objetivos propostos. Todos os que participaram das atividades, conseguiram entender mais sobre computação, e alguns com o interesse de dar continuidade estudando em casa com o auxílio da ferramenta.

Figura 5 - Scratch



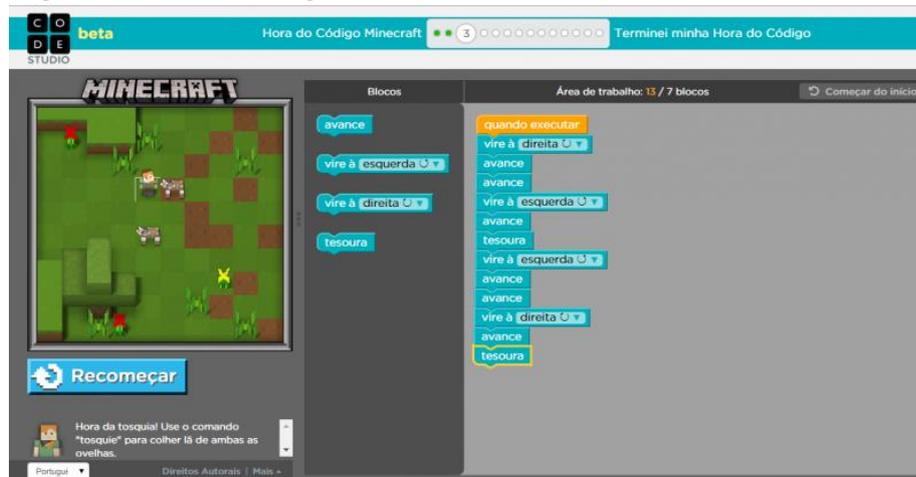
Fonte: <http://www.fabricadejogos.net>

Uma iniciativa de uma organização tem sido uma opção para levar as noções básicas da lógica de programação para crianças, idosos e qualquer pessoa com interesse em explorar tais conceitos de maneira simples e dinâmica.

Code.org, uma organização sem fins lucrativos, teve a iniciativa de divulgar e ensinar programação a pessoas de todas as idades. A instituição tem parcerias com vários gigantes da tecnologia, e algumas das aulas são ministradas por personalidades da área, como Mark Zuckerberg e Bill Gates, por exemplo. Traduzido como “A Hora do Código” no Brasil, o projeto liderado pela Fundação Lemann visa desmistificar a programação e apresentar material de qualidade para que pais, professores e alunos tenham acesso às matérias da área de forma amigável.” (COSTA, 2014, p.1).

⁴ Linguagem de programação criada em 2007 pelo Media Lab do MIT.

Figura 6 – Hora do Código Minecraft



Fonte: <https://www.blogmicrosoftbrasil.com.br>

3. MATERIAIS E MÉTODOS

Os tópicos a seguir abordarão a construção do protótipo, desde a concepção, materiais e metodologia utilizada, assim como todos os recursos empregados no seu desenvolvimento.

3.1 A Ideia

A área de Tecnologia da Informação, é uma das poucas áreas que estão sempre em alta no mercado. Logo, a demanda por profissionais é sempre elevada. Porém, são comuns os relatos de que falta mão-de-obra e há muitos profissionais mal capacitados, que não dominam os conceitos de sua área de atuação, tampouco os aplicam de maneira adequada e otimizada.

De acordo com um estudo realizado pela Softex, o mercado brasileiro de Tecnologia da Informação terá um déficit de 400 mil profissionais até 2022. Em 2009, o mercado do Paraná, por exemplo, tinha 1887 profissionais formados para 1231 vagas; em 2014, foram 2802 profissionais para 5178 vagas. (SUMARES, 2015, p.1).

Durante toda a sua vida escolar, o aluno tem contato com inúmeros conteúdos. Entretanto, uma disciplina que apresente os conceitos básicos de lógica de programação ao aluno, não compõe a grade curricular do ensino fundamental ou do ensino médio. Essa ausência é um dos fatores que contribuem para a estatística acima, pois além de possíveis dificuldades durante a graduação, no que se refere ao raciocínio lógico, os alunos não tomam conhecimento da profissão enquanto estão no

ensino básico e, portanto, dificilmente se interessam por esta. Em declaração à *Code.org*, Mark Zuckerberg⁵, deixou claro que as portas estão abertas: “ Nossa política no Facebook é, literalmente, contratar quantos engenheiros talentosos pudermos encontrar. Simplesmente não há um número suficiente de pessoas que são treinadas e possuem essas habilidades hoje.” (PEREIRA, 2013, p.1).

À primeira vista, a lógica de programação pode mostrar-se complexa e é necessário um certo nível de abstração de ideias para entender o seu funcionamento. Esse ponto é crucial, e muitos alunos dos cursos de computação apresentam dificuldade nele. Portanto, isso pode se tornar um grande desafio quando apresentado aos alunos do ensino fundamental, que são em sua maioria crianças.

Através do uso de jogos e atividades lúdicas, podemos ter mais compreensão, assimilação e fixação dos conteúdos pedagógicos. Seu uso permite ainda o desenvolvimento do raciocínio lógico-matemático, da autonomia e da observância às regras, além de estimular a competitividade construtiva e a criatividade (GOMES *et al.*, 2012, p.139).

Por essa razão, acredita-se que com o auxílio de uma ferramenta, nesse caso, uma aplicação desktop, similar a um jogo, é possível reduzir o nível de dificuldade de abstração de pensamento, através de exemplos (imagens, simulações do cotidiano, etc.) que ilustrariam de maneira mais clara a teoria das disciplinas de lógica básica de programação e sua aplicação prática.

3.2. O Projeto

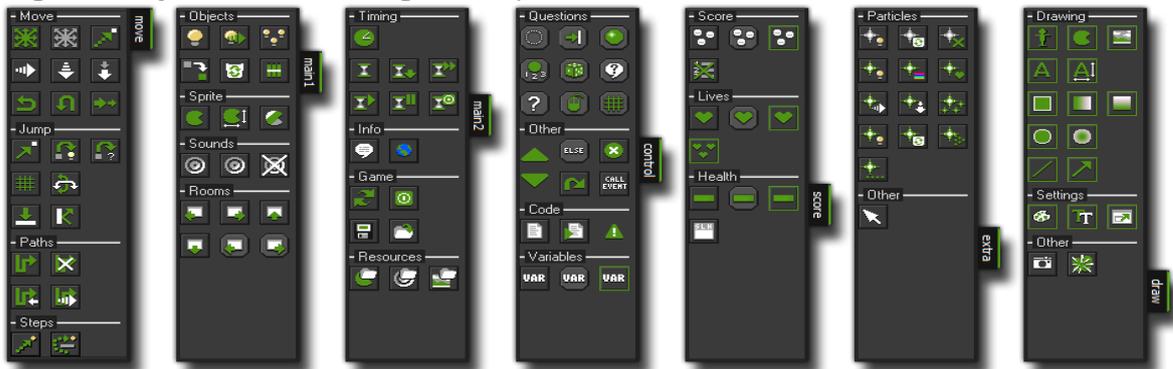
Para a construção deste protótipo, foi escolhido o motor de jogo *Game Maker Studio* (versão 1.4.1763), desenvolvido pela empresa *YoYo Games*. Criado em 15 de novembro de 1999 por Markus Hendrik Overmars, o projeto era chamado de *Animo* em suas primeiras versões, pois era um software de animação 2D. Posteriormente, como muitos de seus usuários o utilizavam com o intuito de criar jogos, Overmars renomeou o programa, e a primeira versão pública do *Game Maker* foi liberada em meados de 1999. Segundo Moura (2013), o programa foi desenvolvido em *Delphi*⁶, contendo mais de 40.000 linhas de código escritas por Markus.

⁵ Programador e empresário norte-americano, é um dos fundadores do Facebook.

⁶ Produzido atualmente pela Embarcadero, é um compilador e uma IDE, do inglês, Ambiente de Desenvolvimento Integrado.

O software faz uso do sistema *Drag and Drop*⁷, representado na figura 7, o que torna a construção dos jogos relativamente mais fácil, pois, dessa maneira, o desenvolvedor não tem que se preocupar em aprender uma linguagem de programação mais complexa. Entretanto, é possível criar *scripts*⁸ (figura 8) e funções personalizadas, através da *Game Maker Language* (GML), linguagem de programação suportada pelo *Game Maker*.

Figura 7 – Ações do sistema Drag and Drop



Fonte: <https://docs.yoyogames.com>

Figura 8 – Tela de script, utilizando a linguagem GML

```

Script: MovePlayer
Name: MovePlayer

MovePlayer
1 //MovePlayer(leftpressed, rightpressed)
2
3 var xx,yy,c1,c2;
4
5 // Apply gravity (and jumping)
6 y = y+grav;
7 grav+=0.4;
8 if( grav>=10 ) grav=10;
9
10 // If falling, check UNDER the player
11 if( grav<0 )
12 {
13     if( dir=1){
14         sprite_index = jump_right;
15     }else{
16         sprite_index = jump_left;
17     }
18     c2 = -1;
19     c1 = GetCollision(x,y);
20     if( (x&$1f)>0 ) {

```

Fonte: <https://docs.yoyogames.com>

⁷ Em tradução livre, Arrastar e Soltar. Sistema no qual clica-se em um objeto virtual, arrastando-o a uma nova posição e/ou sobre outro objeto virtual.

⁸ Programas escritos em uma determinada linguagem de programação.

Com relação à parte gráfica do protótipo, todos os personagens, objetos e cenários foram construídos utilizando a ferramenta *Photoshop*⁹. Foi adotado o estilo *Pixel Art*¹⁰, por ser um método mais simples, leve e rápido, além de possuir ótima estética e envelhecer bem com o passar dos anos.

A ideia do projeto é inserir a criança em um ambiente lúdico e agradável, sem a necessidade de linhas de código e de um conteúdo teórico extenso, no qual ela planejará suas ações antecipadamente, através da observação do local em que o personagem do jogo se encontra, e as organizará em uma sequência lógica para cumprir determinado objetivo.

Ao iniciar o jogo, o aluno irá se deparar com uma tela contendo alguns personagens, um cenário, botões e blocos de ação, além de um objetivo a ser concluído. Os blocos de ação são responsáveis por dar vida ao personagem principal, como, na figura 9, movimentá-lo nos quatro sentidos.

Figura 9 – Tela do primeiro nível do protótipo



Fonte: acervo do autor

⁹ Software editor de imagens.

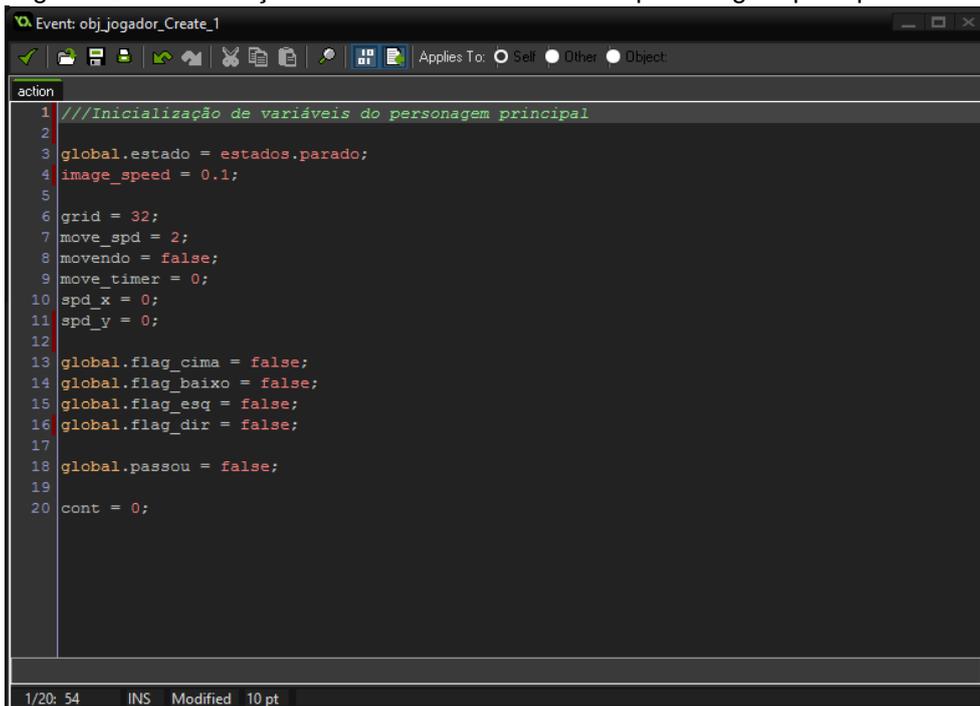
¹⁰ Forma de arte digital na qual as imagens são criadas ou editadas tendo como elemento básico os pixels, considerados os menores elementos em um dispositivo de exibição ao qual pode-se atribuir uma cor.

Na figura 9, tem-se os seguintes elementos:

- 1 - Personagem principal;
- 2 - Objetivo a ser concluído
- 3 - Botão de início do jogo;
- 4 - Botão para excluir os blocos indesejados;
- 5 - Bloco de inicialização das ações (o primeiro bloco de ação a ser executado deverá ser anexado a esse bloco, e os próximos, ao último anexado, estabelecendo então, uma sequência lógica para a realização das ações);
- 6 - Blocos de ação.

A cada ação de movimento executada, o personagem move-se na tela pela distância de uma célula. Para tal, foram inicializadas algumas variáveis referentes ao personagem principal, ilustradas na figura 10. Essas variáveis contêm informações como a distância a ser percorrida pelo personagem, sua velocidade, entre outras. Foram elaborados também cinco *scripts*, responsáveis pelos movimentos nos quatro sentidos (cima, baixo, esquerda e direita) e um último, que mantém o personagem parado na tela. Um exemplo de um dos cinco *scripts* encontra-se representado na figura 11.

Figura 10 – Inicialização das variáveis referentes ao personagem principal



```

Event: obj_jogador_Create_1
Applies To: Self Other Object

action
1  ///Inicialização de variáveis do personagem principal
2
3  global.estado = estados.parado;
4  image_speed = 0.1;
5
6  grid = 32;
7  move_spd = 2;
8  movendo = false;
9  move_timer = 0;
10 spd_x = 0;
11 spd_y = 0;
12
13 global.flag_cima = false;
14 global.flag_baixo = false;
15 global.flag_esq = false;
16 global.flag_dir = false;
17
18 global.passou = false;
19
20 cont = 0;
  
```

Fonte: acervo do autor

Figura 11 – Exemplo de script do protótipo (mover personagem para cima)

```

1 with (obj_jogador) {
2
3     global.flag_cima = scr_percorre_lista("movcima");
4
5     if (!movendo && global.flag_cima) {
6         scr_detecta_colisao("cima");
7     }
8
9     if (movendo) {
10        x += spd_x;
11        y += spd_y;
12
13        sprite_index = spr_mover_cima;
14        image_speed = 0.3;
15
16        move_timer -= move_spd;
17
18        if (move_timer == 0) {
19            movendo = false;
20            global.flag_cima = false;
21            global.estado = estados.parado;
22        }
23    }
24 }

```

Fonte: acervo do autor

Junto aos scripts de movimentação do personagem está embutida uma função destinada a detectar colisões com os objetos presentes no cenário. Um dos objetivos dessa função é determinar se o jogador concluiu o nível em questão ou não. Ela utiliza como argumento o sentido no qual o personagem está se movendo, pois é necessário que seja detectada a presença de objetos alguns pixels acima, abaixo, à esquerda, ou à direita do personagem, para que então a ação seguinte possa ser realizada. Um trecho dessa função está representado na figura 12.

Figura 12 – Trecho do script para detecção de colisões nos quatro sentidos

```

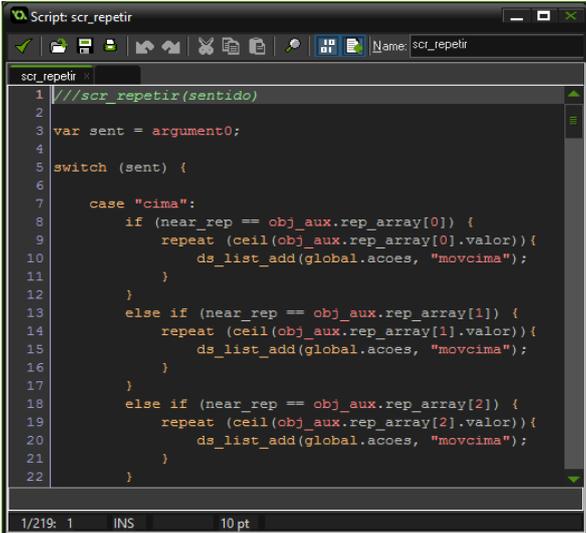
1 //scr_detecta_colisao(sentido)
2
3 var sentido = argument0;
4
5 switch (sentido) {
6
7     case "cima":
8         if (!place_meeting(x, y-10, all)) {
9             movendo = true;
10            move_timer = grid;
11            spd_x = 0;
12            spd_y = -move_spd;
13        }
14        else if (place_meeting(x, y-10, obj_velho)) {
15            movendo = false;
16            scr_passou_fase();
17        }
18        else {
19            movendo = false;
20            scr_gameover();
21        }
22        break;
23    }

```

Fonte: acervo do autor

Para o desenvolvimento do projeto, escolheu-se desenvolver, além dos blocos de movimentação, o bloco de repetição. Este funciona de maneira similar ao laço de repetição *while* (em português, enquanto). Ao colocar o bloco em questão dentro da sequência lógica desejada, um balão é criado ao lado do bloco. Este irá conter o número de vezes que uma determinada ação deverá ocorrer. Então, o jogador, após definir quantas vezes a ação ocorrerá quando o jogo iniciar, deverá posicionar o bloco de ação desejado dentro da estrutura de repetição.

Figura 13 – Trecho do script do bloco de repetição



```

1 //scr_repetir(sentido)
2
3 var sent = argument0;
4
5 switch (sent) {
6
7     case "cima":
8         if (near_rep == obj_aux.rep_array[0]) {
9             repeat (ceil(obj_aux.rep_array[0].valor)){
10                ds_list_add(global.acoes, "movcima");
11            }
12        }
13        else if (near_rep == obj_aux.rep_array[1]) {
14            repeat (ceil(obj_aux.rep_array[1].valor)){
15                ds_list_add(global.acoes, "movcima");
16            }
17        }
18        else if (near_rep == obj_aux.rep_array[2]) {
19            repeat (ceil(obj_aux.rep_array[2].valor)){
20                ds_list_add(global.acoes, "movcima");
21            }
22        }
23    }

```

Fonte: acervo do autor

Todas as ações realizadas pelo personagem são armazenadas em uma estrutura de dados do tipo lista, através de uma função pré-definida do próprio programa. Quando um bloco é anexado ao bloco de inicialização, o próximo ao último anexado, e assim sucessivamente, um dado do tipo *string*¹¹ é armazenado em sua respectiva posição na lista de ações. Esse dado faz referência a qual ação será tomada quando o jogo for executado. O processo está ilustrado na figura 14.

¹¹ Sequência de caracteres utilizados em programação para, normalmente, representar palavras ou frases.

Figura 14 – Trecho do script de uma ação, do tipo *string*, sendo adicionada à lista

```

Event: obj_bt_cima_Left Released_1
Applies To: Self Other Object

action
1 if (place_meeting(x, y, all) && !exec) {
2
3     near = scr_obj_mais_prox(all, self.x, self.y - 18, 2);
4
5     if (!near.original || near.object_index == obj_bl_iniciar_ws) {
6
7         if (near.object_index == obj_bt_repetir) {
8
9             if (self.y >= near.y - 40 && self.y <= near.y + 24) {
10
11                 self.x = near.x + 24;
12                 self.y = near.y + 9;
13
14                 near_rep = scr_obj_mais_prox(all, self.x, self.y - 18, 3);
15
16                 scr_repetir("cima");
17
18                 ds_list_add(global.lista_ids, self.id);
19             }
20         else {
21             self.x = near.x;
22             self.y = near.y + 53;
23
24             ds_list_add(global.acoes, "movcima");
25             ds_list_add(global.lista_ids, self.id);
26         }
27     }
28     else {
29         self.x = near.x;
30         self.y = near.y + 32;
31
32         ds_list_add(global.acoes, "movcima");
33         ds_list_add(global.lista_ids, self.id);
34     }
35 }

```

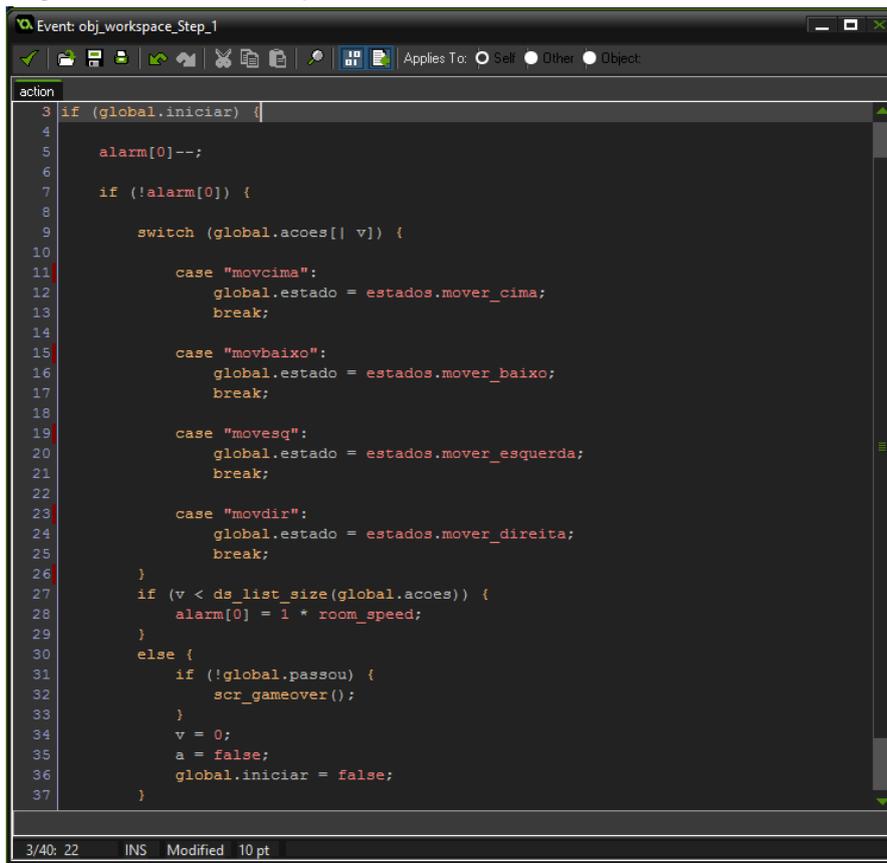
34/53: 10 INS Modified 10 pt

Fonte: acervo do autor

Após a montagem da sequência lógica, quando o jogador clicar no botão “Iniciar”, a lista de ações será percorrida e, para cada dado encontrado na lista, uma variável do tipo *enum*¹², referente ao personagem principal, assume um valor correspondente à informação previamente encontrada, que, então aciona o *script* da ação, movendo o personagem, como ocorre nas figuras 15 e 16, respectivamente.

¹² Tipo de dado abstrato, no qual os valores são atribuídos a um elemento de um conjunto finito de identificadores escolhidos pelo programador.

Figura 15 – Lista sendo percorrida



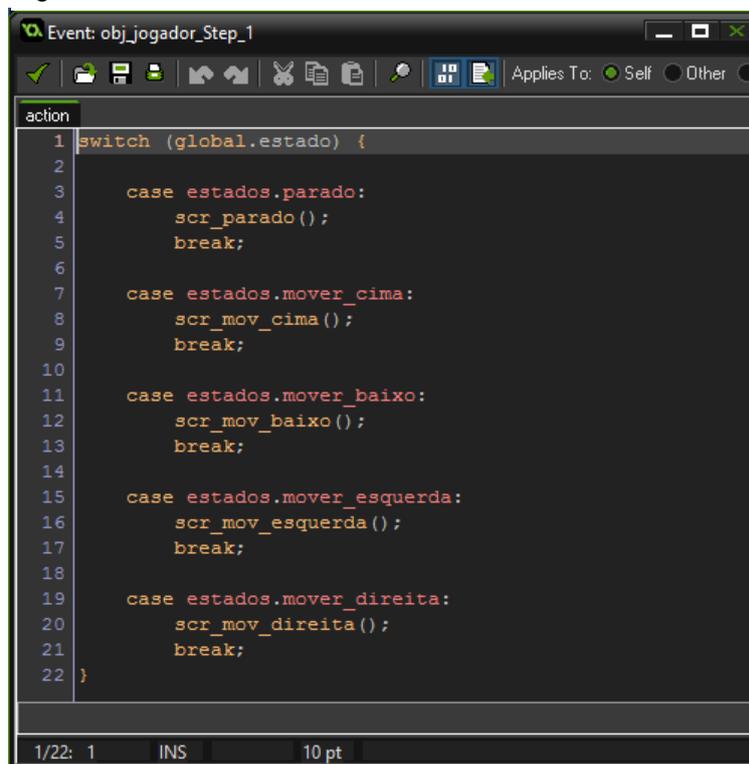
```

3  if (global.iniciar) {
4
5      alarm[0]--;
6
7      if (!alarm[0]) {
8
9          switch (global.acoes[| v]) {
10
11             case "movcima":
12                 global.estado = estados.mover_cima;
13                 break;
14
15             case "movbaixo":
16                 global.estado = estados.mover_baixo;
17                 break;
18
19             case "movesq":
20                 global.estado = estados.mover_esquerda;
21                 break;
22
23             case "movdir":
24                 global.estado = estados.mover_direita;
25                 break;
26
27             }
28             if (v < ds_list_size(global.acoes)) {
29                 alarm[0] = 1 * room_speed;
30             }
31             else {
32                 if (!global.passou) {
33                     scr_gameover();
34                 }
35                 v = 0;
36                 a = false;
37                 global.iniciar = false;
38             }
39
40         }
41     }
42 }

```

3/40: 22 INS Modified 10 pt

Fonte: acervo do autor

Figura 16 – Variável *enum* e seus diversos estados


```

1  switch (global.estado) {
2
3      case estados.parado:
4          scr_parado();
5          break;
6
7      case estados.mover_cima:
8          scr_mov_cima();
9          break;
10
11     case estados.mover_baixo:
12         scr_mov_baixo();
13         break;
14
15     case estados.mover_esquerda:
16         scr_mov_esquerda();
17         break;
18
19     case estados.mover_direita:
20         scr_mov_direita();
21         break;
22 }

```

1/22: 1 INS 10 pt

Fonte: acervo do autor

4. RESULTADOS E DISCUSSÕES

Com base na montagem deste artigo e na construção do protótipo, pode-se concluir que o ensino de lógica básica de programação para crianças traz como resultado inúmeros benefícios. Segundo o professor de informática Mateus Ferreira, o objetivo do ensino de lógica de programação para crianças e adolescentes é desenvolver certas habilidades que a programação exige dos alunos, como raciocínio lógico, causa e consequência, raciocínio matemático, linguística, ou seja, tudo o que está inserido no dia a dia¹³. Entretanto, não é possível obter um resultado satisfatório utilizando-se apenas da exposição do conteúdo, o que pode tornar a aula cansativa. É preciso fazer com que o aluno coloque em prática o conteúdo visto em sala de aula e sinta prazer em aprender. É nesse ponto que o protótipo apresentado neste artigo entra, servindo como uma ferramenta auxiliar no aprendizado dos alunos.

Durante cada nível do jogo, o aluno tem que planejar ações, prevendo possíveis falhas e, posteriormente, corrigi-las, além de organizar suas ações em uma sequência lógica que poderá leva-lo à solução do nível em questão, ideia que, na verdade, utiliza o mesmo princípio do conceito de algoritmos.

5. CONCLUSÕES

O trabalho em questão foi extremamente útil no que se refere à contribuição para o aprendizado do conteúdo visto durante a graduação, desde a concepção da ideia à finalização do protótipo. Durante o seu desenvolvimento, foi possível aplicar o conteúdo de diversas disciplinas, como Lógica, Programação e Estrutura de Dados. Além disso, a união entre a lógica de programação e arte/design, foi possível e bastante satisfatória, sendo a última, a construção dos personagens, cenários e demais objetos.

Com relação à exposição do projeto, espera-se poder apresentá-lo ao público alvo, mostrando que a lógica e as noções básicas de programação não apresentam a complexidade que a elas são, erroneamente, associadas.

¹³ Trecho de entrevista concedida ao portal Olhar Digital. Link: <https://olhardigital.com.br/noticia/escolas-defendem-ensino-de-programacao-a-criancas-e-adolescentes/35075>

6. IMPLEMENTAÇÕES FUTURAS

Para o futuro, pretende-se construir mais níveis, aumentando gradualmente a dificuldade destes. Dentro dos novos níveis, novas funções de programação seriam apresentadas através de novos blocos de ações, como uma função “SE”, por exemplo. Isso proporcionaria mais possibilidades ao aluno, além de estimular cada vez mais o seu raciocínio e aprendizado.

Além disso, há também a ideia de uma versão móvel, já que nos dias atuais, os aparelhos móveis e seus aplicativos estão em constante ascensão, devido a sua praticidade e o crescente aumento em seu poder de processamento. Segundo Lecheta (2015), pesquisas comprovam que mais de 3 bilhões de pessoas têm um telefone celular, e o mercado de aplicativos virou uma febre, rendendo bilhões todos os anos.

7. REFERÊNCIAS:

ALMEIDA, Paulo Nunes de. **Educação Lúdica; prazer de estudar**. Técnicas e jogos pedagógicos. Rio de Janeiro: Loyola, 1998.

BORGES, Teresa Maria Machado. **A criança em Idade Pré-Escolar: desenvolvimento e educação**. Rio de Janeiro: Editora Vitória, ed. Ver. e atual. 2003.

CAMPOS, Ricardo Luiz B. L. **ERM2: Uma proposta de metodologia para melhoria do ensino-aprendizado de lógica de programação**. Distrito Federal, 2009. Disponível em: <http://www.erm2c.com.br/CCESC_2009.pdf>. Acesso em 20 mai 2017.

COSTA, Ricardo. **Fundação Code.org ensina você a programar com conteúdo em português**. Disponível em: <<http://www.tecmundo.com.br/programacao/62979-fundacao-code-org-comeca-oferecer-conteudo-portugues.htm>>. Acesso em: 20 abr. 2017.

FORBELLONE, André Luiz Villar; EBERSPÄCHER, Henri Frederico. **Lógica de Programação: A construção de algoritmos e estruturas de dados**. 3. ed. São Paulo: Pearson, 2005. 197 p.

FREIRE, Paulo. **Pedagogia do Oprimido**. 17 ed. Rio de Janeiro: Paz e Terra, 1987. 107 p.

GOMES, Denise Cristina Ferreira; TOSTA, Cíntia Gomide; SCUSSEL, Denise Rodovalho; FREITAS, Fernanda Sousa; AVELAR, Mônica Corrêa; CUNHA, Neire Márcia da; COSTA, Selma Aparecida Ferreira da. **Necessidades educativas infantis e educação inclusiva. Desenvolvimento social e lógico-matemático da criança**. 1. ed. São Paulo: Pearson, 2012. 276 p.

LECHETA, Ricardo R. **Google Android: Aprenda a criar aplicações para dispositivos móveis com o Android SDK**. 5 ed. São Paulo: Novatec, 2015. 1068p.

MOURA, Daniel. **Desenvolvimento de jogos com Game Maker**. Disponível em: < https://books.google.com.br/books?id=R3qZ3rYFCNYC&printsec=frontcover&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false>. Acesso em 04 mai. 2017

OLIVEIRA, Millena Lauyse Silva de; SOUZA, Anderson Alves de; BARBOSA, Aline Ferreira; BARREIROS, Emanuel Francisco Spósito. **Ensino de lógica de programação no ensino fundamental utilizando o Scratch: um relato de experiência**. Pernambuco, 2014. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/wei/2014/0022.pdf>> Acesso em: 17 abr. 2017.

PARONETO, Glaura Moraes; BORGES, Maria Soledade Gomes; DAHER, Marília de Dirceu Cachapuz; SCUSSEL, Denise Rodovalho; ZAGO, Jaqueline Oliveira Lima; DALBÉRIO, Maria Célia Borges; LOYOLA, Maria Emília; SENE, Vânia Machado de. **O projeto político-pedagógico. Processos históricos da educação brasileira**. 1 ed. São Paulo: Pearson, 2010. 276 p.

PEREIRA, Leonardo. **Escolas defendem ensino de programação a crianças e adolescentes**. Disponível em: < <https://olhardigital.uol.com.br/noticia/escolas-defendem-ensino-de-programacao-a-criancas-e-adolescentes/35075>>. Acesso em: 26 abr. 2017.

ROSA, Renata Sguissardi. **Antes de escolher a escola de seu filho, conheça 9 metodologias de ensino**. Disponível em: <<http://www.semprefamilia.com.br/antes-escolher-a-escola-de-seu-filho-conheca-9-metodologias-de-ensino/>>. Acesso em 19 mai 2017.

SOUSA, Bruno Jefferson de; DIAS JÚNIOR, José Jorge Lima; FORMIGA, Andrei de Araújo. **Introdução à Programação**. 1. ed. João Pessoa: Editora da UFPB, 2014. 95 p.

SOUSA FILHO, Gilberto Farias de; ALEXANDRE, Eduardo de Santana Medeiros. **Introdução a Computação**. 2. Ed. João Pessoa. Editora da UFPB, 2014. 126 p.

SUMARES, Gustavo. **Mercado de TI no Brasil tem mais vagas que profissionais**. Disponível em: < <http://olhardigital.uol.com.br/pro/noticia/mercado-de-ti-no-brasil-tem-mais-vagas-que-profissionais/53548>>. Acesso em: 20 abr 2017.