



SERVIÇO PÚBLICO FEDERAL
UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE CIÊNCIAS EXATAS E NATURAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM MATEMÁTICA EM REDE NACIONAL

GIANCARLO SECCI DE SOUZA PEREIRA

**A LINGUAGEM DE PROGRAMAÇÃO EDUCATIVA SCRATCH NA PRODUÇÃO
DE CONTEÚDOS DIGITAIS PARA MEDIAÇÃO DA APRENDIZAGEM DE
MATEMÁTICA NA EDUCAÇÃO BÁSICA**

BELÉM/PA
2019

GIANCARLO SECCI DE SOUZA PEREIRA

**A LINGUAGEM DE PROGRAMAÇÃO EDUCATIVA SCRATCH NA PRODUÇÃO
DE CONTEÚDOS DIGITAIS PARA MEDIAÇÃO DA APRENDIZAGEM DE
MATEMÁTICA NA EDUCAÇÃO BÁSICA**

Dissertação apresentada ao Programa de Pós-Graduação em Matemática em Rede Nacional do Instituto de Ciências Exatas e Naturais da Universidade Federal do Pará para obtenção do título de Mestre em Matemática

Área de concentração: Educação Matemática

Orientadora: Prof.^a Dr.^a. Cristiane Ruiz Gomes

BELÉM/PA
2019

Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD
Sistema de Bibliotecas da Universidade Federal do Pará
Gerada automaticamente pelo módulo Ficat, mediante os dados fornecidos pelo(a) autor(a)

P436l Pereira, Giancarlo Secci de Souza
A linguagem de programação educativa Scratch na produção de
conteúdos digitais para mediação da aprendizagem de Matemática
na educação básica / Giancarlo Secci de Souza Pereira. — 2019.
126 f. : il. color.

Orientador(a): Prof^a. Dra. Cristiane Ruiz Gomes
Dissertação (Mestrado) - Programa de Pós-Graduação em
Matemática em Rede Nacional, Instituto de Ciências Exatas e
Naturais, Universidade Federal do Pará, Belém, 2019.

1. Educação Matemática. 2. Tecnologias Digitais. 3.
Programação Educativa. 4. Scratch. 5. SpriteBox. I. Título.

GIANCARLO SECCI DE SOUZA PEREIRA

**A LINGUAGEM DE PROGRAMAÇÃO EDUCATIVA SCRATCH NA PRODUÇÃO
DE CONTEÚDOS DIGITAIS PARA MEDIAÇÃO DA APRENDIZAGEM DE
MATEMÁTICA NA EDUCAÇÃO BÁSICA**

Dissertação apresentada ao Programa de Pós-Graduação em Matemática em Rede Nacional do Instituto de Ciências Exatas e Naturais da Universidade Federal do Pará para obtenção do título de Mestre em Matemática

Área de concentração: Educação Matemática

Orientadora: Prof.^a Dr.^a. Cristiane Ruiz Gomes

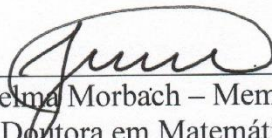
Defendido e aprovado em: 28 / 06 / 2019

Conceito: _____

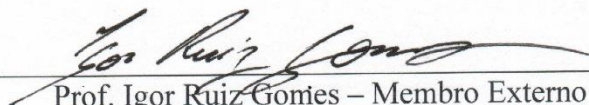
Banca examinadora:



Prof.^a Cristiane Ruiz Gomes - Orientador
Doutora em Engenharia Elétrica
Universidade Federal do Pará - FACMAT



Prof.^a Joeline Morbach – Membro Titular
Doutora em Matemática
Universidade Federal do Pará – FACMAT



Prof. Igor Ruiz Gomes – Membro Externo
Doutor em Engenharia Elétrica
Universidade Federal do Pará - FACOMP

*Dedico esta dissertação às minhas filhas
Isadora e Heloisa, à minha amada esposa
Nayra Maria e aos meus pais Carlos Rodrigues
Pereira e Maria Lucileide de Souza Pereira.*

AGRADECIMENTOS

Agradeço a todos que contribuíram direta ou indiretamente para que esta pesquisa fosse realizada com êxito.

Agradeço à minha esposa, que sempre me apoiou e me deu forças para ultrapassar cada um dos obstáculos durante minha trajetória no Mestrado, e as minhas filhas que me dão o amor necessário a restauração de minhas forças a cada dia.

À minha irmã Kelly Cristina deixo um agradecimento especial por me proporcionar conforto e aconchego em sua casa sempre que necessário.

Agradeço ao Professora Doutora Cristiane Ruiz Gomes por aceitar me orientar, por sua paciência, insistência, confiança e, principalmente, por seus valiosos ensinamentos acerca da pesquisa acadêmica.

Agradeço a Professora Doutora Joelma Morbach e ao Professor Doutor Igor Ruiz Gomes pela participação na banca de defesa e pelas grandes contribuições ao meu trabalho.

Agradeço a Escola Guaraci Mendes e aos sujeitos da pesquisa, meus alunos, por sua participação no minicurso de extensão e por sua colaboração no desenvolvimento das atividades realizadas durante a produção dos conteúdos digitais propostos.

Agradeço aos colegas e amigos do Programa de Pós-Graduação em Matemática em Rede Nacional (PROFMAT-Belém/PA) por seu apoio nos momentos de dificuldade e compartilhamento de experiência durante o Mestrado.

Agradeço à CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) pelo apoio financiamento prestado durante o período de curso e pesquisa.

*Ensinar exige compreender que a educação
é uma forma de intervenção no mundo.
(FREIRE, 2015)*

PEREIRA, G. S. S. A linguagem de programação educativa Scratch na produção de conteúdos digitais para mediação da aprendizagem de Matemática na educação básica. 127 f. Dissertação (Mestrado Profissional em Matemática) - Programa de Pós-Graduação em Matemática em Rede Nacional, Universidade Federal do Pará, Belém, 2019.

RESUMO

Este trabalho tem como principal objetivo apresentar as potencialidades e possibilidades de utilização da linguagem de programação educativa *Scratch* na produção de conteúdos digitais de matemática por alunos do 9º ano do ensino fundamental de uma escola pública do município de Tailândia no estado do Pará. Para isso, tomamos como pergunta norteadora: a linguagem de programação educativa *Scratch* e a produção de conteúdos digitais contribui para o processo de ensino e aprendizagem da Matemática? Em busca de nossos objetivos utilizamos como base teórica a teoria construcionista de Seymour Papert criador da linguagem LOGO, agregada as contribuições de Resnick com a criação e popularização da linguagem de programação *Scratch*, assim como as valorosas contribuições nacionais de Valente e Cysneiro. Para entendermos de que forma as Tecnologias Digitais da Informação e Comunicação começaram a ser inseridas nas instituições educacionais brasileiras, faremos uma reflexão sobre os documentos oficiais que fomentaram, fomentam e regulamentam o uso dessas tecnologias, como o Plano Nacional de Educação, os Parâmetros Curriculares Nacionais, as Diretrizes Curriculares para o Ensino Médio e a Base Nacional Comum Curricular. Para conhecermos o cenário nacional acerca das produções acadêmicas que utilizaram a linguagem de programação *Scratch* para a mediação do ensino e da aprendizagem da Matemática fizemos uma revisão da literatura utilizando como sistema de busca a Biblioteca Nacional de Teses e Dissertações chegando a resultados surpreendentes. Em nossa trajetória metodológica utilizamos o SpriteBox como recurso tecnológico facilitador a iniciação da programação educativa. Através do SpriteBox, do *Scratch* e da teoria construcionista de Seymour Papert propomos aos alunos o desenvolvimento de aplicações como conteúdos digitais de matemática que pudessem contribuir para a compreensão de tópicos desta disciplina.

Palavras-chave: Educação Matemática, Tecnologias Digitais, Programação Educativa, SpriteBox, *Scratch*.

PEREIRA, G. S. S. The Scratch educational programming language in the production of digital content for mediation of mathematics learning in basic education. 127 f. Dissertation (Professional Master in Mathematics) - Postgraduate Program in Mathematics in National Network, Universidade Federal do Pará, Belém, 2019.

ABSTRACT

This work has as main objective to present the potentialities and possibilities of using the Scratch educational programming language in the production of digital contents of mathematics by students of the 9th grade of a public school in the municipality of Tailândia in the state of Pará. We take as a guiding question: does the Scratch educational programming language and the production of digital content contribute to the teaching and learning process of Mathematics? In pursuit of our objectives we use as a theoretical basis the constructionist theory of Seymour Papert, creator of the LOGO language, adding Resnick's contributions to the creation and popularization of the Scratch programming language, as well as Valente and Cysneiro's valued national contributions. In order to understand how Digital Information and Communication Technologies began to be inserted in Brazilian educational institutions, we will reflect on official documents that have fostered, fostered and regulated the use of these technologies, such as the National Education Plan, the National Curricular Parameters, the Curriculum Guidelines for High School and the National Curricular Common Base. In order to know the national scenario about the academic productions that used the programming language Scratch for the mediation of the teaching and learning of Mathematics we did a review of the literature using as a search system the National Library of Theses and Dissertations reaching surprising results. In our methodological trajectory we use SpriteBox as a technological resource that facilitates the initiation of educational programming. Through SpriteBox, Scratch and the construction theory of Seymour Papert we propose to the students the development of applications as digital content of mathematics that could contribute to the understanding of topics of this discipline.

Keywords: Mathematics Education, Digital Technologies, Educational Programming, SpriteBox, Scratch.

LISTA DE SIGLAS

ANATEL – Agência Nacional de Telecomunicações
BDTD – Biblioteca Digital de Teses e Dissertações
BNCC – Base Nacional Comum Curricular
BPM – Batida por Minuto
CAI – Computer Aided Instruction
CAPES – Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
DCEM – Diretrizes Curriculares para o Ensino Médio
DCN – Diretrizes Curriculares Nacional
EDUCOM – Educação e computador
FNDE – Fundo Nacional de Desenvolvimento da Educação
IBICT – Instituto Brasileiro de Informações em Ciência e Tecnologia
IDEB – Índice de Desenvolvimento Básico da Educação
INEP – Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira
IPB – Interface de Programação em Bloco
LPE – Linguagem de Programação Educativa
MIT – Massachusetts Institute of Technology
NIED – Núcleo de Informática Aplicada a Educação
OCDE – Organização para a Cooperação e Desenvolvimento Econômico
OCEM – Orientações Curriculares para o Ensino Médio
OLPC – One Laptop per Child
PCN – Parâmetros Curriculares Nacionais
PNE – Plano Nacional de Educação
PROINFO – Programa Nacional de Informática na Educação
PRONINFE – Programa Nacional de Informática Educativa
PROUCA – Um Computador Por Aluno
SAEB – Sistema de Avaliação da Educação Básica
TDIC – Tecnologia Digital da Informação e Comunicação

LISTA DE FIGURAS

Figura 5.1 – Tela de boas-vindas	42
Figura 5.2 – Links de navegação para outros vídeos	42
Figura 5.3 – Breve apresentação do SpriteBox	43
Figura 5.4 – Habilidades desenvolvidas ao longo do jogo	43
Figura 5.5 – SpriteBox para todas as idades e links para lojas de Apps	44
Figura 5.6 – Links de acesso ao LightBot e SpriteBox Code Hour	44
Figura 5.7 – Tela inicial do SpriteBox Code Hour	45
Figura 5.8 – Questionário de feedback	46
Figura 5.9 – Tela inicial e botões	47
Figura 5.10.a – Modo bloco de comando	48
Figura 5.10.b – Modo linha de comando	48
Figura 5.11 – Tela secundária de escolha de fase	49
Figura 5.12 – SpriteBox no modo exploração	50
Figura 5.13 – SpriteBox no modo exploração	50
Figura 5.14 – Comando SET	51
Figura 5.15 – Comando MOVE	51
Figura 5.16 – Comando LOOP	52
Figura 5.17.a – Comando GET	52
Figura 5.17.b – Utilizando o comando GET	52
Figura 5.18 – Tela de boas-vidas do site do <i>Scratch</i>	54
Figura 5.19 – Tela inicial do estúdio de criação do <i>Scratch</i>	55
Figura 5.20 – Tutorial da tela inicial do estúdio de criação	55
Figura 5.21 – Página do botão Explorar	56
Figura 5.22 – Página do botão Ideias	56
Figura 5.23 – Tela inicial do estúdio de criação 3.0	57
Figura 5.24 – Tela inicial do estúdio de criação 2.0	58
Figura 6.1 – Proficiência em matemática em 2017	61
Figura 6.2 – Frente da Escola Guaraci Mendes	61
Figura 6.3 – Metas e IDEB observado da escola Guaraci Mendes	63
Figura 6.4 – Nível de enquadramento da escola Guaraci Mendes em 2015.....	64
Figura 6.5 – Aplicação Motivacional 01: Olá Mundo	66
Figura 6.6 – Aplicação Motivacional 02: Número Aleatório	66

Figura 6.7 – Aplicação Motivacional 03: Soma de dois números	67
Figura 6.8 – Aplicação Motivacional 04 (Labirinto) com parte dos códigos	67
Figura 6.9 – Tela de boas-vindas do SpriteBox	68
Figura 6.10 – Tela inicial do jogo SpriteBox	69
Figura 6.11 – Impossibilidade de prosseguir	69
Figura 6.12 – Janela de comandos	70
Figura 6.13 – Arrastando blocos de comando	70
Figura 6.14 – Comando MOVE	71
Figura 6.15 – Arrastando comandos	71
Figura 6.16 – Reorganização da Aluna M	72
Figura 6.17 – Último desafio do primeiro mundo	72
Figura 6.18 – Deletando comandos	73
Figura 6.19 – Mudando a direção	73
Figura 6.20 – Construindo uma ponte	74
Figura 6.21 – Algoritmo do Aluno H	74
Figura 6.22 – Construindo pontos de apoio	75
Figura 6.23 – Comando GET	75
Figura 6.24 – Utilização do comando GET	76
Figura 6.25 – Introdução do laço de repetição	76
Figura 6.26 – Introdução ao laço de repetição	77
Figura 6.27 – Utilização do laço de repetição	77
Figura 6.28 – Seções da tela inicial do estúdio de criação 3.0	78
Figura 6.29 – Alguns recursos do botão Adicionar Uma Extensão	78
Figura 6.30 – Comando posição pré-definida	79
Figura 6.31 – Ator antes e depois de aplicar um efeito pixelado	81
Figura 6.32 – Opção de gravação de áudio no <i>Scratch</i>	82
Figura 6.33 – Sensação natural de movimento	89
Figura 6.34 – Definição do novo bloco lógico: movimento natural	90
Figura 6.35 – Área de códigos	90
Figura 6.36 – Interação computador-usuário	91
Figura 6.37 – Cenários	92
Figura 6.38 – Aluno Z jogando SpriteBox	99
Figura 6.39 – Aluno K no terceiro desafio do segundo mundo do SpriteBox	99

Figura 6.40 – Pesquisador passando instruções as alunas sobre o SpriteBox	100
Figura 6.41 – Disposição da Sala de Informática	101
Figura 6.42 – Aluna N programando no <i>Scratch</i>	101
Figura 6.43 – Aluna M criando a aplicação Marcando Pontos	102
Figura 6.44 – Depoimento da aluna K	103
Figura 6.45 – Depoimento da aluna N	103

LISTA DE QUADROS

Quadro 1.1 – Etapas e atividades da pesquisa	20
Quadro 2.1 – Políticas públicas de inserção tecnológica na educação brasileira	23
Quadro 3.1 – Produções nacionais na BDTD relacionadas a linguagem <i>Scratch</i>	33
Quadro 5.1 – Opções de visualização dos comandos	47
Quadro 5.2 – Personagens do SpriteBox	48
Quadro 6.1 – Infraestrutura da escola Guaraci Mendes	62

SUMÁRIO

1	INTRODUÇÃO	17
1.1	Objetivo geral	19
1.2	Objetivos específicos	19
1.3	Metodologia	19
2	POLÍTICAS PÚBLICAS E O USO DAS TDIC NO BRASIL	21
2.1	A inserção dos computadores na educação brasileira	21
2.2	Programas governamentais de inserção tecnológica na educação	23
2.3	Metas e estratégias do PNE e a utilização da TDIC	26
2.4	As competências e habilidades da BNCC a luz da TDIC	28
3	REVISÃO DA LITERATURA	32
3.1	Produção nacionais ligadas ao <i>Scratch</i>	32
3.2	Produção nacionais ligadas ao Spritebox	36
4	REFERENCIAL TEÓRICO	38
5	DESBRAVANDO O SPRITEBOX E O SCRATCH	41
5.1	Explorando o SpriteBox e suas possibilidades	41
5.1.1	<i>Apresentação do SpriteBox</i>	41
5.1.2	<i>O site do SpriteBox</i>	41
5.1.3	<i>Versão web flash do SpriteBox</i>	46
5.1.4	<i>A tela inicial da versão web flash</i>	46
5.1.5	<i>Personagens do Jogo</i>	48
5.1.6	<i>Tela secundária para jogar via web</i>	49
5.1.7	<i>Modo exploração</i>	49
5.1.8	<i>Modo código</i>	50
5.1.9	<i>Comandos utilizados no SpriteBox</i>	51
5.2	Explorando a linguagem <i>Scratch</i> e suas possibilidades	53
5.2.1	<i>Apresentação do Scratch</i>	53
5.2.2	<i>O site do Scratch</i>	54
5.2.3	<i>Estúdio de criação Scratch</i>	57
6	CENÁRIO E APLICAÇÃO DO PROJETO LPE	59
6.1	Um breve relato sobre o município de Tailândia	59
6.2	A Escola Municipal Guaraci Mendes	61

6.2.1	<i>A escala SAEB</i>	63
6.3	Planejamento e aplicação do projeto LPE	64
6.3.1	<i>Aplicações motivacionais propostas</i>	65
6.4	Organização e aplicação do minicurso LEP	68
6.4.1	<i>Blocos de Comando (1)</i>	78
6.4.1.1	Movimento	79
6.4.1.2	Aparência	80
6.4.1.3	Sons	82
6.4.1.4	Eventos	83
6.4.1.5	Controle	84
6.4.1.6	Sensores	86
6.4.1.7	Operadores	86
6.4.1.8	Variáveis	88
6.4.1.9	Meus blocos	86
6.4.2	<i>Área de Códigos (2)</i>	90
6.4.3	<i>Palco (3)</i>	91
6.4.4	<i>Área dos Atores e Cenários (4)</i>	92
6.4.5	<i>Atividades Iniciais</i>	94
6.5	Registro das atividades	98
6.6	Alguns relatos das atividades	102
7	CONSIDERAÇÕES FINAIS	105
	REFERÊNCIAS	108
	APÊNDICE A – Projeto LEP	111
	APÊNDICE B – Plano de aula do projeto LPE	121
	APÊNDICE C – Códigos da aplicação Marcando Pontos LC	123

1 INTRODUÇÃO

No último meio século a Tecnologia Digital da Informação e Comunicação (TDIC) se tornou inerente ao processo de evolução da sociedade. Grande parte dos serviços públicos dependem da TDIC. Essa dependência é visivelmente observada quando ficamos sem o fornecimento de energia, por exemplo. Os bancos, os supermercados, os sistemas de transporte aéreo, os serviços de comunicação e outros dependem dessa tecnologia que a cada dia se torna mais dinâmica e portátil atingindo cada vez mais pessoas em todo o mundo.

Segundo dados da Agência Nacional de Telecomunicações (ANATEL) o Brasil registrou quase 236 milhões de linhas móveis em operação no mês de março de 2018, o que representa uma densidade de 112,87 celular/100 habitante. Apesar de a ANATEL apresentar cada linha móvel como um aparelho celular temos aparelhos que comportam até três chips, ou seja, poderíamos ter três linhas móveis em um único celular, o que não minimiza a massiva quantidade de equipamentos potenciais as práticas pedagógicas propostas neste trabalho.

Em 2017 o *Cetic.br*¹ realizou uma pesquisa sobre a utilização da TDIC nas escolas brasileiras, intitulada “TIC na Educação 2017”. Quando perguntado aos alunos das escolas públicas qual equipamento eles utilizavam para acessar a internet, 97% disseram utilizar o celular, 49% utilizam o computador de mesa, 51% utilizam o computador portátil, 37% disseram utilizar o tablet, 37% a televisão e 32% o videogame. Nesta pesquisa o somatório dos percentuais supera 100% pois os alunos poderiam escolher mais de uma das opções citadas.

Nesse sentido, mesmo não havendo consenso entre educadores e acadêmicos acerca do uso do celular nas salas de aula, não podemos ignorar a notória importância dessa tecnologia na vida dos alunos desta época, pois para Prensky (2001 apud Coelho 2012, p.90) os nativos digitais formam uma geração de indivíduos que experimentam e vivenciam múltiplas possibilidades oferecidas pela dinamicidade da tecnologia. Coelho (2012) afirma ainda que

Esse fascínio característico da geração Y pela descoberta e experimentação deve ser explorado pela escola, de forma a direcioná-la para um ensino e uma aprendizagem que dialoguem e interajam com os novos meios tecnológicos. (COELHO, 2012, p.3)

Este cenário reforça as ideias de Prensky (2001) de que nossos alunos hoje, os “nativos digitais”, formam as primeiras gerações a crescerem cercados de novos dispositivos tecnológicos, que a cada passo despontam como suas novas ferramentas e brinquedos. O termo

¹ Centro Regional de Estudos para o Desenvolvimento da Sociedade da Informação (Cetic.br). O Cetic.br é um departamento do Núcleo de Informação e Coordenação do Ponto BR (Nic.br), que implementa as decisões e projetos do Comitê Gestor da Internet do Brasil (Cgi.br).

“nativos digitais” surge a partir uma analogia que Prensky (2001) faz com pessoas nativas e imigrantes. Pessoas nativas aprendem, geralmente, sem dificuldades sua língua materna, já os imigrantes têm certa dificuldade para aprender uma língua diferente. Assim, os nativos digitais nascem inseridos em um mundo tecnológico, tendo praticamente nenhuma dificuldade para aprender seus princípios. Já os imigrantes digitais apresentam certa dificuldade ao lidarem com as novas tecnologias.

Além disso, as Orientações Curriculares para o Ensino Médio (OCEM), das Ciências da Natureza, Matemática e suas Tecnologias, versam em seu capítulo 3 que “é importante contemplar uma formação escolar nesses dois sentidos, [...] a Matemática como ferramenta para entender a tecnologia, e a tecnologia como ferramenta para entender a Matemática”. Quando se considera a tecnologia como ferramenta para entender a Matemática, foco do nosso trabalho, as OCEM destacam que

[...] há programas de computador (softwares) nos quais os alunos podem explorar e construir diferentes conceitos matemáticos, referidos a seguir como programas de expressão. Os programas de expressão apresentam recursos que provocam, de forma muito natural, o processo que caracteriza o “pensar matematicamente”, ou seja, os alunos fazem experimentos, testam hipóteses, esboçam conjecturas, criam estratégias para resolver problemas. (BRASIL, MEC, 2006, p. 88.)

Com isso, observa-se que não há coerência em dissociar o ensino da tecnologia. Logo, sendo os alunos de hoje nativos digitais, propomos a utilização de Interfaces de Programação em Bloco (IPB) para que estes construam aplicações digitais que venham trazer significativa contribuição ao seu processo de ensino e aprendizagem de Matemática. É importante ressaltar que Barcelos (2013) discorre que a falta do domínio adequado de conhecimentos matemáticos pelos alunos é um possível fator explicativo para a falta de interesse e evasão em cursos da área de TDIC, por exemplo. Nesse sentido, observa-se ser imprescindível que nossos alunos tenham conhecimentos e embasamentos matemáticos sólidos para que possam desenvolver atividades relacionadas a área da TDIC. Mas, e o processo inverso? Será que **a linguagem de programação educativa Scratch e a produção de conteúdos digitais contribui para o processo de ensino e aprendizagem da Matemática?**

Sendo assim, considerando os fatos citados e os questionamentos acerca a utilização da TDIC, com ênfase na Linguagem de Programação Educativa, e da busca por respostas embasadas no conhecimento científico traçamos para esta dissertação os seguintes objetivos (geral e específicos):

Objetivo geral

- Utilizar a linguagem de programação educativa *Scratch* para produzir conteúdos digitais e verificar quais suas contribuições ao processo de ensino e aprendizagem da Matemática para alunos da Educação Básica.

Objetivos específicos

1. Analisar métodos de ensino auxiliados por Tecnologia Digital da Informação e Comunicação em estudos anteriores;
2. Descrever as contribuições da utilização de interfaces de programação em bloco ao processo de ensino e aprendizagem da Matemática;
3. Utilizar o SpriteBox como facilitador no processo de iniciação a programação;
4. Apresentar as potencialidades da interface de programação em bloco *Scratch*;
5. Propor uma metodologia de ensino baseada na construção de aplicativos;
6. Produzir material didático - textos e vídeos - para mediar a iniciação a programação em bloco;
7. Mediar a produção de aplicações no *Scratch*;
8. Avaliar a metodologia proposta através de questionários, relatos e possíveis reflexões.

Metodologia

Este trabalho é uma pesquisa aplicada de cunho quali-quantitativo que envolve características de ação empírica. Os sujeitos da pesquisa serão agentes construtores de seus próprios conhecimento através da teoria construcionista de Seymour Papert. Os dados serão coletados a partir de questionários semiestruturados para que possamos conhecer a real situação dos alunos participantes desta pesquisa, além de entrevistas e observações que se estenderão ao longo dos seis encontros para que possamos avaliar o desenvolvimento dos mesmos acerca das competências e habilidades desenvolvidas tanto no que diz respeito a linguagem de programação quanto a Matemática.

Para alcançarmos nossos objetivos desenvolvemos a pesquisa em 5 (cinco) etapas (por exemplo: E1, E2, etc.), as quais são divididas em atividades (por exemplo: A11, referente a primeira atividade da primeira etapa e A23, referente a terceira atividade da etapa 2, assim sucessivamente). A seguir apresentamos a etapas e a descrição de suas respectivas atividades.

Quadro 1.1 – Etapas e atividades da pesquisa.

E1	ETAPA 1	Pesquisa Bibliográfica.
A11	Atividade 11	Pesquisa bibliográfica sobre o uso da Tecnologia da Informação e Comunicação na educação através do ensino de programação;
A12	Atividade 12	Pesquisa bibliográfica sobre a legislação educacional vigente e políticas governamentais para a informática educativa;
A13	Atividade 13	Pesquisa bibliográfica sobre os programas que serão apresentados nesse trabalho (<i>Scratch</i> e <i>Sprite Box</i>).
E2	ETAPA 2	Estudo das linguagens de programação educacionais.
A21	Atividade 21	Estudo do aplicativo <i>Sprite Box</i> ;
A22	Atividade 22	Estudo da linguagem <i>Scratch</i> ;
E3	ETAPA 3	Planejamento do minicurso de Linguagem de Programação Educacional
A31	Atividade 31	Elaboração das atividades;
A32	Atividade 32	Elaboração de apostila do minicurso.
E4	ETAPA 4	Aplicação do minicurso
A41	Atividade 41	Introdução a lógica de programação com o <i>Sprite Box</i>
A42	Atividade 42	Apresentação do <i>Sprite Box</i> como ferramenta auxiliar na aprendizagem de lógica de programação;
A43	Atividade 43	Treinamento e atividades de lógica de programação com o <i>Sprite Box</i> ;
A44	Atividade 44	Apresentação do <i>Scratch</i> como ferramenta na aprendizagem de programação;
A45	Atividade 45	Treinamento e atividades de programação em <i>Scratch</i> ;
A46	Atividade 46	Atividades de avaliação em <i>Scratch</i> ;
E5	ETAPA 5	Reflexões e Análises sobre o Minicurso.
A51	Atividade 51	Coleta de dados;
A52	Atividade 52	Registro de imagens, falas e comentários;
A53	Atividade 53	Análise dos resultados.

Fonte: Elaboração própria.

Diante disso, seguindo os objetivos e métodos traçados neste capítulo, em especial a Atividade 12 do Quadro 1.1, apresentaremos no capítulo seguinte um estudo reflexivo sobre as políticas públicas relacionadas a inserção das TDIC nas instituições educacionais brasileira, a partir da análise de alguns dos principais documentos oficiais que regem a educação no Brasil.

2 POLÍTICAS PÚBLICAS E O USO DAS TDIC NO BRASIL

O avanço das TDIC trouxe inúmeros benefícios para a sociedade nas últimas décadas, entre os quais se pode citar: a minimização no tamanho dos computadores com o criação dos transistores, ligações telefônicas 3G (terceira geração da internet móvel que hoje está na quinta geração, 5G), transações bancárias online, ensino à distância, entre outros. Esses avanços permitiram que pessoas que estão a quilômetros de distância possam se comunicar, que possamos realizar transações bancárias ou mesmo estudar sem a necessidade de deixar o conforto de nossas casas.

Para entender de que forma essas TDIC começaram a ser inseridas nas instituições educacionais destinamos este capítulo a uma reflexão sobre os documentos oficiais que fomentaram, fomentam e regulamentam o uso da TDIC na educação brasileira, como o Plano Nacional de Educação (PNE), os Parâmetros Curriculares Nacionais (PCN), a Diretrizes Curriculares para o Ensino Médio (DCEM) e a Base Nacional Comum Curricular (BNCC), principalmente.

2.1 A inserção dos computadores na educação brasileira

Apesar de a utilização de computadores como ferramenta auxiliar ao processo de ensino ter sido timidamente iniciada nas universidades em 1920, segundo Cuban (1987), “o interesse por ampliar e generalizar a incorporação e o uso pedagógico dos computadores nas escolas de modo mais intenso não se produziu até a década de 1980” (SANCHO, et al., 2006, p.153).

Para Valente (1999) as políticas públicas voltadas a implantação das TDIC nas escolas brasileiras têm apontado para uma mudança pedagógica, fazendo com que não haja consenso entre os educadores brasileiros sobre sua implantação, pois em outros países como Estados Unidos e França a inserção das TDIC tem se dado como um objeto com o qual o aluno deva se familiarizar e não um objeto de mudança da dinâmica pedagógica em sala de aula. Essa tentativa de mudança no panorama educacional do Brasil através da TDIC torna o processo mais complexo visto que na visão de Valente (1999) é necessário uma maior preocupação com a formação dos professores, alterações na dinâmica pedagógica em sala e modificações no currículo.

Embora o contexto mundial sobre o uso da Informática na Educação sempre tenha sido uma referência para as decisões que foram tomadas no Brasil, nossa caminhada é muito peculiar e difere daquilo que se faz em outros países. No entanto, se

compararmos os avanços pedagógicos conseguidos por intermédio da Informática no Brasil e em outros países, os resultados são semelhantes e indicam que ela praticamente não alterou a abordagem pedagógica. (VALENTE, 1999, p. 13).

Apesar dos resultados modestos, Valente (1999) ressalta que esses projetos governamentais têm buscado mudanças no cenário educacional brasileiro desde 1982, momento em que essas políticas públicas começam a ser delineadas a partir do I Seminários Nacional de Informática na Educação realizado em Brasília com a presença da encarregada da Área Internacional da Direção Geral das Indústrias Eletrônicas e de Informática da França, Françoise Faure e do diretor do Centro Nacional de Ensino de Informática do Ministério da Cultura e Educação da Argentina, Felix Kierbel.

No início dos anos 60, diversos softwares de instrução programada foram implementados no computador, concretizando a máquina de ensinar, idealizada por Skinner no início dos anos 50. Nascia a instrução auxiliada por computador ou o *Computer-Aided Instruction* (CAI), produzida por empresas como IBM, RCA e Digital e utilizada principalmente nas universidades. (VALENTE, 1999, p.14).

O CAI apresentava algumas limitações para sua plena implementação e divulgação, como o fato de os computadores da época serem de grande porte, o que os restringia as grandes universidades e dificultava sua disseminação nas escolas elementares e secundárias. O aparecimento dos microcomputadores nos anos 1980 parecia uma solução para alguns desses problemas, no entanto possibilitou o aparecimento de novas modalidades do uso do computador na educação aumentando as possibilidades de mudança na qualidade desta através da criação e aperfeiçoamento dos ambientes de aprendizagem. Dentre essas novas possibilidades do uso do computador na educação o LOGO foi a proposta mais marcante (VALENTE, 1999).

Segundo Silva (2018) a linguagem LOGO, desenvolvida na década de 1960 no laboratório de inteligência artificial do MIT por Seymour Papert e outros pesquisadores, pairava como uma ferramenta fundamental para facilitar e promover o aprendizado significativo do sujeito pois trazia certa autonomia na construção e reconstrução do conhecimento.

Valente (2011) ressalta que a

Linguagem de programação o Logo serve para nos comunicarmos com o computador. Entretanto, ela apresenta características especialmente elaboradas para implementar uma metodologia de ensino baseada no computador (metodologia Logo) e para explorar aspectos do processo de aprendizagem. Assim, o Logo tem duas raízes: uma computacional e a outra pedagógica. Do ponto de vista computacional, as características do Logo que contribuem para que ele seja uma linguagem de programação de fácil assimilação são: exploração de atividades espaciais, fácil terminologia e capacidade de criar novos termos ou procedimentos. (VALENTE, 2011, p. 7, documento eletrônico).

A Metodologia LOGO se trata de uma proposta de utilização dos computadores para medição do processo de ensino, todavia em uma perspectiva construcionista, onde o aluno tem total autonomia sobre o processo de criação e o professor atua apenas como mediador.

2.2 Programas de inserção tecnológica na educação

No Quadro 2.1 Prado (2017) apresenta um resumo dos programas governamentais brasileiro que objetivaram a inserção da TDIC nas escolas brasileira.

Quadro 2.1 – Políticas públicas de inserção tecnológica na educação brasileira.

Ano	Programa	Público alvo	Atualidade do programa
1983	EDUCOM – Educação e computador.	Foram selecionas 5 escolas de nível superior.	Extinto
1989	PRONINFE – Programa Nacional de Informática Educativa.	Formação de professores a partir dos Centros de Informática na Educação 1º e 2º graus; Centros de Informática na Educação Tecnológica (CIET); Centros de Informática na Educação Superior (CIES).	Extinto
1997	Proinfo – Programa Nacional de Informática na Educação.	Introdução a informática nas redes municipais e estaduais de educação. Inicialmente realizou aquisição de 100 mil computadores para distribuição entre os 27 estados da União.	Em execução
2006	PROUCA – Programa um computador por aluno.	Em 2007 pré-piloto - 5 escolas selecionadas em 5 estados diferentes; em 2010 o projeto foi ampliado e 300 escolas brasileiras receberam laptops.	1ª Etapa do projeto concluída.

Fonte: PRADO, *et al.*, 2017, p.182.

Segundo Valente (2006, documento eletrônico) o Projeto EDUCOM foi iniciado nos anos de 1981 e 1982 a partir dos Seminários Nacionais de Informática em Educação realizados respectivamente na Universidade de Brasília (UNB) e na Universidade Federal da Bahia (UFBA). O projeto foi implantado em cinco universidades que trabalharam com escolas públicas desenvolvendo atividades de pesquisa e formação, tendo como objetivo o uso da metodologia LOGO, desenvolvida por Seymour Papert, nas disciplinas de Matemática, Ciências (Física, Química e Biologia no 2º grau) e Português.

O Educom teve um papel fundamental nas atividades iniciais do Nied, contribuindo para entender como o Logo podia ser usado no contexto da escola pública. Porém, um dos subprodutos mais importantes, foi a formação da equipe, conhecimento esse que foi usado na capacitação de muitos pesquisadores e professores de escolas. (VALENTE, 2006, documento eletrônico).

O PRONINFE (Programa Nacional de Informática Educativa) foi lançado em 1989 pelo governo federal, “com o objetivo de desenvolver a informática educativa no Brasil, através de projetos e atividades apoiados em fundamentação pedagógica sólida e atualizada, assegurando a unidade política, técnica e científica” (TAVARES, S/D, p.6).

Este projeto

Buscava apoiar o desenvolvimento e a utilização da informática no ensino de 1º, 2º e 3º graus e na educação especial através da criação de núcleos (recomendados em número de 533) distribuídos geograficamente por todo o país e da formação de recursos humanos, particularmente os professores. Aliás, esse era o foco: capacitação contínua e permanente dos professores. (TAVARES, S/D, p.6).

Ademais, segundo Tavares (S/D) a estruturação e objetivos deste projeto serviram de base para a criação de seu sucessor: o PROINFO.

Com o objetivo de promover o uso da tecnologia digital como ferramenta de enriquecimento pedagógico no ensino público fundamental e médio, o Programa Nacional de Tecnologia Educacional (Proinfo) foi criado em 1997 por meio da Portaria nº. 522 sob o nome de Programa Nacional de Informática na Educação.

O programa leva às escolas computadores, recursos digitais e conteúdos educacionais. Em contrapartida, estados, Distrito Federal e municípios devem garantir a estrutura adequada para receber os laboratórios e capacitar os educadores para uso das máquinas e tecnologias. (BRASIL, 1997, documento eletrônico)

A estratégia utilizada pelo Proinfo é a distribuição de laboratórios de informática com dez computadores para as escolas públicas de ensino fundamental e médio. Como não havia recursos para distribuir equipamentos para todas, o corte inicialmente utilizado foi de escolas com mais de 250 alunos (BRASÍLIA, 2008).

Em 2007, por meio do Decreto nº. 6.300 de 12 de dezembro de 2007, o programa passa a se preocupar com toda a rede pública de educação básica, tendo como objetivos:

- I - promover o uso pedagógico das tecnologias de informação e comunicação nas escolas de educação básica das redes públicas de ensino urbanas e rurais;
- II - fomentar a melhoria do processo de ensino e aprendizagem com o uso das tecnologias de informação e comunicação;
- III - promover a capacitação dos agentes educacionais envolvidos nas ações do Programa;
- IV - contribuir com a inclusão digital por meio da ampliação do acesso a computadores, da conexão à rede mundial de computadores e de outras tecnologias digitais, beneficiando a comunidade escolar e a população próxima às escolas;
- V - contribuir para a preparação dos jovens e adultos para o mercado de trabalho por meio do uso das tecnologias de informação e comunicação; e
- VI - fomentar a produção nacional de conteúdos digitais educacionais. (BRASIL, 2007, documento eletrônico).

O Decreto nº. 6.300 também trata da forma de adesão, responsabilidades, financiamento e implantação do programa nas escolas.

Para fazer parte do Proinfo Urbano e /ou Rural, o município deve seguir três passos: a adesão, o cadastro e a seleção das escolas. A adesão é o compromisso do município com as diretrizes do programa, imprescindível para o recebimento dos laboratórios. Após essa etapa, deve ser feito o cadastro do prefeito em nosso sistema, que permitirá o próximo passo, que é a inclusão das escolas no Proinfo. (BRASIL, 1997, documento eletrônico)

Segundo o site do FNDE² em sequência ao Proinfo temos o Projeto Um Computador por Aluno (UCA), implantado com o objetivo de intensificar as TDIC nas escolas, por meio da distribuição de computadores portáteis aos alunos da rede pública de ensino. O projeto figura como uma complementação as ações do MEC referentes a tecnologias na educação, em especial os laboratórios de informática, produção e disponibilização de objetivos educacionais na internet dentro do Proinfo Integrado que promove o uso pedagógico da informática na rede pública de ensino fundamental e médio.

Vermieiro (2014) destaca que após visita a Seymour Papert, que estava iniciando seus trabalhos com a linguagem computacional LOGO, Alan Kay ficou maravilhado com a possibilidade de as crianças poderem utilizar os computadores na resolução de problemas matemáticos, culminando na ideia de que cada criança tivesse seu próprio computador.

No Brasil, o projeto (UCA) começa a ser pensado a partir da apresentação ao governo do projeto “Um Laptop Para Cada Criança (OLPC (*One laptop per child*))” durante o Fórum Econômico Mundial ocorrido em Davos na Suíça em 2005. O projeto passa a ser um programa do governo brasileiro com a aprovação em 11 de junho de 2010.

O Art. 7º. da Lei nº. 12.249/2010 que trata, entre outros assuntos, da criação do Programa Um Computador por Aluno (PROUCA) estabelece que

O Prouca tem o objetivo de promover a inclusão digital nas escolas das redes públicas de ensino federal, estadual, distrital, municipal ou nas escolas sem fins lucrativos de atendimento a pessoas com deficiência, mediante a aquisição e a utilização de soluções de informática, constituídas de equipamentos de informática, de programas de computador (software) neles instalados e de suporte e assistência técnica necessários ao seu funcionamento. (BRASIL, 2010, documento eletrônico).

² Fundo Nacional de Desenvolvimento da Educação. Disponível em: <https://www.fnde.gov.br/programas/proinfo/eixos-de-atuacao/projeto-um-computador-por-aluno-uca>. Acesso em 01.05.2019.

A legislação citada também incube o poder Executivo de relacionar os equipamentos de informática necessários a implantação do PROUCA e estabelecer processo produtivo básico específico, definindo etapas mínimas e condicionantes de fabricação dos mesmos.

A primeira fase do PROUCA, denominada Fase 1, ocorre de forma experimental com a doação de computadores pelas empresas Telavo, Intel e pela organização não-governamental (ONG) OLPC a cinco escolas públicas localizadas nas cidades de São Paulo/SP, Porto Alegre/RS, Palmas/TO, Piraí/RJ e Brasília/DF. A Fase 2 que pretendia comprar 150 mil computadores para serem distribuídos a 300 escolas públicas de até 500 alunos no início do ano letivo de 2008 foi adiada devido problemas no processo de compras sendo necessário a realização de um estudo de viabilidade do programa com base nos dados apurados na Fase 1 (BRASÍLIA, 2008).

A seguir temos uma análise de cada uma das metas do Plano Nacional de Educação (PNE) que podem contribuir direta ou indiretamente para o aumento dos fomentos relacionados a TDIC como ferramenta para o ensino e a aprendizagem.

2.3 Metas e estratégias do PNE e a utilização da TDIC

Para Monteiro (2014) “o Plano Nacional de Educação (PNE) aprovado em 2014 é a materialização histórica e articulada da intenção de um país em busca de garantir o direito à educação a todos os seus cidadãos”. Ainda segundo a autora, o PNE enfatiza que a articulação de práticas pedagógicas às tecnologias educacionais deve assegurar a melhoria da aprendizagem. Em sua análise Monteiro (2014) observa que há referências às TDIC em diversas partes do PNE, sendo que algumas dessas referências estão explícitas e outra não.

A **META 6** do PNE, por exemplo, objetiva “oferecer educação em tempo integral em, no mínimo, 50% (cinquenta por cento) das escolas públicas, de forma a atender, pelo menos, 25% (vinte e cinco por cento) dos (as) alunos (as) da educação básica” (BRASIL, PNE, 2014, p. 10), tendo como uma de suas estratégias institucionalizar e manter, em regime de colaboração, programa nacional de ampliação e reestruturação das escolas públicas, por meio da instalação de laboratórios, inclusive de informática. De acordo com o documento oficial que apresenta a linha de base dos indicadores selecionados pelo Ministério da Educação e pelo Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP) para o monitoramento do Plano Nacional de Educação 2014-2024

A educação em tempo integral se concretiza por meio do desenvolvimento de atividades de acompanhamento pedagógico, **experimentação e investigação**

científica, cultura e artes, esporte e lazer, **cultura digital**, educação econômica, **comunicação e uso de mídias**, meio ambiente, direitos humanos, práticas de prevenção aos agravos à saúde, promoção da saúde e da alimentação saudável, entre outras. (BRASIL, INEP, 2015, p. 98, grifo nosso).

Dentro desse contexto, a exigência estabelecida pela **META 6** pode fortalecer a proposta de inserção das TDIC através da Linguagem de Programação Educativa nas escolas públicas, pois aumenta as possibilidades de implantação de projetos que visam o desenvolvimento de habilidades relacionadas a perspectiva construcionista a partir do uso da LPE no ensino de Matemática, inclusive cita em sua estratégia 6.3 a ampliação e reestruturação de laboratórios de informática. Dessa forma, as escolas poderão destinar parte desse tempo – tempo integral – ao estudo da linguagem de programação com ênfase na construção de conteúdos digitais que propiciem uma aprendizagem Matemática dinâmica e efetiva.

A **META 7** que visa fomentar a qualidade da educação básica em todas as etapas e modalidades, com melhoria do fluxo escolar e da aprendizagem de modo a atingir as médias nacionais proposta para o Índice de Desenvolvimento da Educação Básica (IDEB) destaca em suas estratégias 7.12, 7.15 e 7.20 o papel fundamental da TDIC no processo de ensino e aprendizagem, o incentivo à práticas pedagógicas inovadoras e a universalização da internet. Essa universalização da internet é explicitada na estratégia 7.15:

Universalizar, até o quinto ano de vigência deste PNE, o acesso à rede mundial de computadores em banda larga de alta velocidade e triplicar, até o final da década, a relação computador/aluno (a) nas escolas da rede pública de educação básica, promovendo a utilização pedagógica das tecnologias da informação e da comunicação. (BRASIL, PNE, documento eletrônico)

A **META 9** tem como um de seus objetivos, reduzir em 50% (cinquenta por cento) a taxa de analfabetismo funcional até o final da vigência do PNE, tomando como uma de suas estratégias a “implementar programas de capacitação tecnológica da população jovem e adulta, direcionados para os segmentos com baixos níveis de escolarização formal e para os (as) alunos (as) com deficiência”, oferecendo a estes tecnologias assistivas que favoreçam a efetiva inclusão social e produtiva dessa população. (BRASIL, PNE, 2014, documento eletrônico).

Na **META 15**, apresenta-se a proposta de garantir no prazo de 1 (um) ano de vigência deste PNE, política nacional de formação dos profissionais da educação tendo como uma de suas estratégias

15.6) Promover a reforma curricular dos cursos de licenciatura e estimular a renovação pedagógica, de forma a assegurar o foco no aprendizado do (a) aluno (a), dividindo a carga horária em formação geral, formação na área do saber e didática

específica e incorporando as modernas tecnologias de informação e comunicação, em articulação com a base nacional comum dos currículos da educação básica, de que tratam as estratégias 2.1, 2.2, 3.2 e 3.3 deste PNE. (BRASIL, PNE, 2014, documento eletrônico).

Diante desse cenário, observa-se a notória preocupação do PNE com a inserção, manutenção e aprimoramento dos programas que visam a utilização das TDIC para a melhoria da educação básica, além da adequação dos currículos de formação inicial dos professores, para que estes tenham como um de seus focos de aprendizagem as modernas TDIC, que em seguida será estudada em face a BNCC ficada em suas habilidades e competências relacionadas a Matemática e suas tecnologias.

2.4 As competências e habilidades da BNCC a luz da TDIC

A versão final da BNCC com a inclusão da etapa do Ensino Médio foi homologada no dia 19 de dezembro de 2018, atingindo o objetivo de uma base para toda a Educação Básica brasileira, sendo peça fundamental na busca por uma educação pública de qualidade através da melhoria da aprendizagem, em especial do Ensino Médio “no qual os índices de aprendizagem, repetência e abandono são bastante preocupantes” (BRASIL, BNCC, 2018).

A Base Nacional Comum Curricular (BNCC) é um documento de caráter normativo que define o conjunto orgânico e progressivo de aprendizagens essenciais que todos os alunos devem desenvolver ao longo das etapas e modalidades da Educação Básica, de modo a que tenham assegurados seus direitos de aprendizagem e desenvolvimento, em conformidade com o que preceitua o Plano Nacional de Educação (PNE). Este documento normativo aplica-se exclusivamente à educação escolar, tal como a define o § 1º do Artigo 1º da Lei de Diretrizes e Bases da Educação Nacional (LDB, Lei nº 9.394/1996), e está orientado pelos princípios éticos, políticos e estéticos que visam à formação humana integral e à construção de uma sociedade justa, democrática e inclusiva, como fundamentado nas Diretrizes Curriculares Nacionais da Educação Básica (DCN). (BRASIL, BNCC, 2018).

Com a BNCC, busca-se garantir o conjunto de aprendizagens que a base entende como essenciais para o desenvolvimento integral dos estudantes brasileiros por meio de cruzamento de competências (mobilização de conhecimentos (conceitos e procedimentos)), apoiando suas escolhas para a concretização dos seus projetos de vida e a continuidade dos estudos, sendo um documento completo e contemporâneo, que corresponde às demandas do estudante desta época, preparando, assim, os estudantes para o futuro. (BRASIL, BNCC, 2018). No sentido de preparar os alunos desta época para o futuro, 40% das Competências Gerais (CG) da BNCC trazem referências diretas ou indiretas a tecnologia digital.

A **CG 01** traz a valorização e utilização da cultura digital, um dos conhecimentos construídos ao longo da vida, “para entender e explicar a realidade, continuar aprendendo e colaborar para a construção de uma sociedade justa, democrática e inclusiva”. Já a **CG 02** associa a tecnologia aos conhecimentos das diferentes áreas, evidenciando-a como uma das possibilidades para a criação de soluções na resolução de problemas.

A **CG 04** destaca a importância de se utilizar a linguagem digital para se expressar e compartilhar informações, enquanto que a **CG 05** vai além, alvitrando não apenas o compartilhamento de informações, mas a criação de TDIC capazes de produzir conhecimento, resolver problemas e evidenciar o aluno como protagonista e autor na vida pessoal e coletiva.

A área de Matemática e o componente curricular de Matemática em articulação com as Competências Gerais, devem garantir aos alunos o desenvolvimento de Competências Específicas (CE) (BRASIL, BNCC, 2018). Dessa forma, a BNCC traça 08 (oito) CE de Matemática para o Ensino Fundamental e 05 (cinco) CE de Matemática e suas Tecnologias para o Ensino Médio, sendo que das 08 (oito) CE traçadas para o Ensino Fundamental duas delas enfatizam a utilização das TDIC: as CE 05 e CE 06.

Na **CE 05**, de Matemática para o Ensino Fundamental, temos a utilização das TDIC como ferramenta matemática “para modelar e resolver problemas cotidianos, sociais e de outras áreas de conhecimento, validando estratégias e resultados” (BRASIL, BNCC, 2018, p.267). Já a **CE 06**, de Matemática para o Ensino Fundamental, propõe a utilização das TDIC como metodologias auxiliares no enfrentamento de situações-problema em diversos contextos “incluindo-se situações imaginadas, não diretamente relacionadas com o aspecto prático-utilitário, expressar suas respostas e sintetizar conclusões, utilizando [...] outras linguagens para descrever algoritmos, como fluxogramas, e dados” (BRASIL, BNCC, 2018, p. 267).

Para o Ensino Médio,

A BNCC da área de Matemática e suas Tecnologias propõe a consolidação, a ampliação e o aprofundamento das aprendizagens essenciais desenvolvidas no Ensino Fundamental. Para tanto, propõe colocar em jogo, de modo mais inter-relacionado, os conhecimentos já explorados na etapa anterior, a fim de possibilitar que os estudantes construam uma visão mais integrada da Matemática, ainda na perspectiva de sua aplicação à realidade. (BRASIL, BNCC, 2018, p. 527).

Corroborando com as previsões e proposições feitas pelas CE de Matemática para o Ensino Fundamental, todas as CE de Matemática e suas Tecnologias para o Ensino Médio apresentam, direta ou indiretamente, a TDIC como ferramenta para o desenvolvimento desta área do conhecimento.

A **CE 01** de Matemática e suas Tecnologias para o Ensino Médio, através de sua habilidade **EM13MAT101**, propõe a utilização das TDIC para auxiliar na “interpretação de situações econômicas, sociais e fatos relativos às Ciências da Natureza que envolvam a variação de grandezas, pela análise dos gráficos das funções representadas e das taxas de variação” (BRASIL, BNCC, 2018, p. 533).

A habilidade **EM13MAT202** da **CE 02** retorna a possibilidade de utilização das TDIC no planejamento e execução de pesquisas amostrais sobre questões relevantes, enquanto que a habilidade **EM13MAT203** vai além e propõe o uso de aplicativos e a construção de planilhas eletrônicas “para o controle de orçamento familiar, simuladores de cálculos de juros simples e compostos, entre outros” com o intuito de auxiliar na tomada de decisões.

Objetivando “utilizar estratégias, conceitos, definições e procedimentos matemáticos para interpretar, construir modelos e resolver problemas em diversos contextos” a BNCC Brasil (2018) em sua **CE 03** destaca ainda que

O uso de tecnologias possibilita aos estudantes alternativas de experiências variadas e facilitadoras de aprendizagens que reforçam a capacidade de raciocinar logicamente, formular e testar conjecturas, avaliar a validade de raciocínios e construir argumentações. (BRASIL, BNCC, 2018, p. 536).

Em suas habilidades **EM13MAT301**, **EM13MAT302**, **EM13MAT307** e **EM13MAT309** reforça-se a possibilidade da utilização das TDIC na resolução e elaboração de problemas do cotidiano, além da construção de “modelos empregando as funções polinomiais de 1º ou 2º graus, para resolver problemas em contextos diversos” (BRASIL, BNCC, 2018, p. 536). De modo análogo, a **CE 04** espera que o aluno seja capaz de “compreender e utilizar, com flexibilidade e precisão, diferentes registros de representação matemáticos (algébrico, geométrico, estatístico, computacional etc.), na busca de solução e comunicação de resultados de problemas” apoiando-se ou não nas TDIC, como é citado nas habilidades **EM13MAT403**, **EM13MAT404** e **EM13MAT406**.

Finalmente a **CE 05** traz a TDIC como ferramenta fundamental para a validação de conjecturas outrora investigadas e estabelecidas. Concordando com as habilidades citadas na **CE 04** de Matemática e suas tecnologias para, temos as habilidades **EM13MAT503** e **EM13MAT509**.

Essas reflexões dos documentos oficiais que regem a educação no Brasil reforçam a importância da TDIC nesse cenário, como ela pode ser aplicada e de que forma serão viabilizadas e implantadas para fins didáticos. Nesse sentido, para entendermos melhor de que

forma essas tecnologias digitais estão sendo utilizadas para mediar o ensino e aprendizagem da Matemática trazemos no próximo capítulo uma revisão literária acerca da utilização do *Scratch* no ensino e na aprendizagem da Matemática.

3 REVISÃO DA LITERATURA

A revisão da literatura além de nos fornece embasamento teórico, norteia a pesquisa através da análise de experimentações que resultaram em produtos ou proposta de aplicações da LPE, do *Scratch* ou do *Spritebox* (este último utilizado para a introdução à programação) na construção de conteúdos interativos para o ensino de Matemática. Para Pereira (2017) este levantamento se “trata do mapeamento realizado para o conhecimento das produções acadêmicas sobre o tema” ou sobre aspectos concernentes a pesquisa em questão.

3.1 Produção nacionais ligadas ao Scratch

Para o levantamento da produção acadêmica nacional acerca da utilização da LPE para a construção de conteúdos interativos para o ensino e aprendizagem da Matemática utilizamos como mecanismo de busca a plataforma do Instituto Brasileiro de Informações em Ciência e Tecnologia (IBICT), através da Biblioteca Digital de Teses e Dissertações (BDTD), utilizando as seguintes palavras-chaves: *Scratch*; *linguagem Scratch*; *Spritebox*.

Inicialmente se buscou na BDTD trabalhos relacionados apenas a palavra-chave *Scratch* com o filtro “todos os campos” ativada, tendo como resultado 208 (duzentos e oito) referências. Um dos motivos para esse grande número de trabalhos que contém a palavra *Scratch* é o fato de que a expressão portuguesa “a partir do zero” ou “do princípio” se traduz para o inglês como “*from scratch*” levando a busca eletrônica da BDTD a contabilizar essas expressões que, provavelmente, aparecem nos *abstract* de alguns desses trabalhos. Sendo inviável a análise das 208 pesquisas encontradas, mudamos a palavra-chave para *linguagem Scratch*, diminuindo demasiadamente o número de resultados encontrados para 23 (vinte e três), sendo 17 (dezessete) dissertações e 6 (seis) teses.

Com uma quantidade viável para analisar, partimos para a leitura dos resumos para identificarmos se de fato esses trabalhos encontrados trazia a linguagem *Scratch* como recurso metodológico para o ensino. A partir da análise dos desses 23 (vinte e três) trabalhos constatamos que apenas 13 (treze) (10 dissertações e 3 teses) utilizavam de fato a linguagem *Scratch*.

Dando sequência a análise da literatura fizemos uma breve leitura da trajetória metodológica de cada um desses trabalhos para relatarmos e verificarmos quais deles apresentavam a Linguagem de Programação Educativa *Scratch* como ferramenta auxiliar ao

processo de ensino e aprendizagem da Matemática, tendo como resultado o Quadro 3.1 apresentado a seguir.

Quadro 3.1 – Produções nacionais na BDTD relacionadas a linguagem *Scratch*.

Autor, ano	Título	Utiliza o Scratch com foco no ensino de Matemática	Tipo de produção
SILVA, Cristiane Samária Gomes da, 2019	Imersão nas tecnologias digitais para educação: uma experiência pedagógica no curso de Pedagogia da PUC-SP	Não	Dissertação
POLONI, Leonardo, 2018	Aprendizagem de programação mediada por uma linguagem visual: possibilidade de desenvolvimento do pensamento computacional	Não	Dissertação
SANTANA, Bianca Leite, 2018	Uma abordagem de ensino-aprendizagem de programação na educação superior	Não	Dissertação
ARAUJO, Luis Gustavo de Jesus, 2018	Uma Abordagem em Espiral para Disciplinas Iniciais de Programação na Educação Profissional em Informática	Não	Dissertação
CRUZ, Marcia Elena Jochims Kniphoff da, 2018	Produção didática do estudante de licenciatura em computação, epistemologia genética e neurociência cognitiva	Não	Tese
AZEVEDO, Greiton Toledo de, 2017	Construção do conhecimento Matemático a partir da produção de jogos digitais em um ambiente construcionista de aprendizagem: desafios e possibilidades	Sim	Dissertação
ROCHA, Helen Regiane Pará, 2017	Formação inicial de professores de ciências: construção de saberes docentes com a linguagem de programação visual Scratch	Não	Dissertação
ROCHA, Kátia Coelho da, 2017	Programação em Scratch na sala de aula de matemática: investigações sobre a construção do conhecimento de ângulo	Sim	Dissertação
RODEGHIERO, Carolina Campos, 2016	Aprendizagem conectada em modos de participação no Scratch: proposta de tipologia	Não	Tese
SILVA, Samantha Pinto da, 2016	O uso da lógica de programação para a Educação Matemática no Ensino Médio: experiências com o Scratch	Sim	Dissertação
RICARTE, Lidiany Teotonio, 2015	O uso da ferramenta Scratch na escola pública: multiletramentos, autoria e remixagem	Não	Dissertação
SANTOS, Ângelo Costa dos, 2014	Aprendizagem mediada por linguagens de autoria: o Scratch na visão de três pesquisadores	Sim	Dissertação
DALLA VECCHIA, Rodrigo, 2012	A modelagem matemática e a realidade do mundo cibernético	Sim	Tese

Fonte: Da pesquisa.

Silva (2018), tendo como sujeitos de sua pesquisa alunas do curso de Pedagogia, atuou como monitora na disciplina “As Novas Tecnologias na Educação: Comunidades de Aprendizagem” optando por uma linha de investigação de cunho qualitativo exploratório utilizando a linguagem *Scratch* em um dos momentos de seu trabalho com o objetivo de levar

experiência múltiplas sobre TDIC a essa turma. A base teórica da autora foram as teorias Construtivista e Construcionista de Jean Piaget e Seymour Papert, respectivamente.

Silva (2018) ministrou uma oficina na qual mostrou de forma criteriosamente organizada, a história da linguagem *Scratch*, a biografia de Seymour Papert e as funcionalidades e possibilidades da linguagem *Scratch*, tendo resultados satisfatórios em sua pesquisa, visto que as alunas participantes conseguiram se familiarizar com a linguagem *Scratch* e tiveram como atividade de avaliação, o desafio de produzir uma aplicação que simulasse uma história em quadrinhos para a apresentação de um determinado conteúdo.

Apoiando-se na teoria sócio-interacionista de Lev Vygotsky, Poloni (2018) objetivou analisar de que forma o *Scratch* pode mediar a aprendizagem de programação no Ensino Médio Técnico na perspectiva do desenvolvimento do pensamento computacional.

Através de uma pesquisa delineada por um estudo de caso, Poloni (2018) organizou uma oficina para introdução à linguagem de programação em uma instituição de ensino da Serra Gaúcha chegando à conclusão de que o ambiente *Scratch* tem potencial para mediar o aprendizado de programação, pois proporciona um ambiente agradável dinâmico e motivador levando o aluno a “desenvolver habilidades relacionadas ao pensamento computacional e interagir com autonomia” (POLONI, 2018).

Com o intuito de motivar e amenizar as dificuldades dos alunos *non-majors* em relação à programação de computadores, Santana (2018) combina o uso do ambiente *Scratch* a linguagem Python em um contexto de criação de jogos analisada a partir do modelo ARCS (Atenção, Relevância, Confiança e Satisfação). Buscando sanar dificuldades na aprendizagem de programação Araújo (2018) seguiu uma linha de pesquisa semelhante à de Santana (2018).

A pesquisa de Santana (2018) consistiu em dois estudos de caso com estudantes do curso de Engenharia Civil em uma disciplina introdutória de programação onde a utilização da linguagem *Scratch* associada a linguagem Python possibilitou a criação de programas que puderam auxiliar no estudo de funções, vetores e matrizes. Usando metodologia similar, Araújo (2018) três estudos de casos: um no ensino Fundamental, e dois em cursos técnicos de Informática, ambos no município de Feira de Santana, Bahia.

Santana e Araújo (2018, 2018) ainda ressaltam em seus resultados que essa abordagem com mídias se demonstrou potencial para a aprendizagem de programação, acreditando que os fatores práticos apresentado em suas pesquisas podem apoiar o design de disciplinas introdutórias de programação para alunos *non-majors*.

Azevedo (2017) almejou em sua pesquisa compreender o processo da construção do conhecimento matemático através da elaboração e desenvolvimento de jogos digitais com

alunos do Ensino Fundamental. Para isso o autor baseou todo seu estudo na linguagem de programação *Scratch* articulada com atividades exploratório-investigativas de matemática.

A análise dos dados permitiu observar que os resultados da pesquisa dão subsídios para a percepção do processo de construção do conhecimento matemático, além de mostrar que essa “construção se fundamenta pela produção quando há participação ativa do estudante no ambiente” (AZEVEDO, 2017).

Com o intuito de investigar que saberes futuros professores de Ciências poderiam desenvolver quando em contato com a linguagem de programação *Scratch*, Rocha (H., 2017) elaborou uma oficina investigativa denominada “Programação com o *Scratch* para aprender Ciências II”, aplicando-a a alunos do 5º e 7º período do curso de Licenciatura em Física do Instituto Federal de Educação, Ciência e Tecnologia do Amazonas (IFAM) na disciplina de Metodologia do Ensino de Ciências.

Para alcançar seus objetivos Rocha (H., 2017) teve como base teórica, para tratar do contexto programação de computadores como proposta a ser inserida na escola, bem como na formação de professores, autores como Papert (2008), Resnick (2007, 2013, 2014), Valente (1999, 2014), Prado (1996), Demo (2008). A partir de suas análises, Rocha (H., 2017) foi possível inferir que o uso da linguagem de programação *Scratch*, de fato, contribui para a construção de saberes relacionados a formação inicial de professores de Ciências.

Rocha (K., 2017) buscou em sua pesquisa quais as evidências de pensamentos matemáticos e como os alunos as expressam em atividades de programação com a linguagem *Scratch* envolvendo o conceito de ângulo. A pesquisadora teve como sujeitos da pesquisa, alunos do 6º ano do ensino fundamental de uma escola pública municipal de São Leopoldo, onde em 11 encontros coletou dados na tentativa de responder, apoiada na Teoria dos Campos Conceituais proposta por Gérard Vergnaud, a problemática de seu trabalho.

Após a fase de análise de dados, Rocha (K., 2017) chegou a apontamentos que evidenciam que durante as etapas de programação “os alunos vivenciaram diferentes situações em que o conceito de ângulo estava inserido, fazendo-os ativar, combinar, recombina e construir diversos esquemas, além de representar ângulo de formas variadas. (ROCHA, K., 2017).

Silva (2016) utiliza a lógica de programação para potencializar o ensino de Matemática quanto a resolução de problemas. Para isso, realizou um estudo de caso, em uma perspectiva socioconstrutivista, com alunos do 3º ano do ensino médio recorrendo a linguagem de programação visual *Scratch*, com a qual realizou experimentações que comprovaram que essa nova abordagem metodológica torna as aulas mais dinâmica e motivadoras fazendo com que os

alunos desta etapa escolar desenvolvam habilidades e competências necessárias a sua formação integral.

3.2 Produção nacionais ligadas ao Spritebox

Em relação a palavra-chave *Spritebox* não foram encontrados resultados nem na BDTD e nem no Catálogo de Teses e Dissertações da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CTD-CAPES), todavia encontramos 03 (três) referências a palavra quando fazemos a busca no *Google Acadêmico* com o filtro “*Pesquisar Páginas em Português*”.

Soares (2017) realiza em seu trabalho uma revisão sistemática dos jogos com o objetivo de identificar os jogos eletrônicos para dispositivos móveis capazes de auxiliar o ensino de programação. A pesquisa de Soares (2017) se justifica pelos elevados índices de reprovação e evasão nos cursos de computação causados, segundo alguns apontamentos feitos por pesquisas anteriores acerca das dificuldades dos alunos na aprendizagem de programação, por:

Dificuldades dos estudantes com a resolução de problemas, compreensão da natureza abstrata da programação e a sua sintaxe [25], as complexas atividades cognitivas envolvidas [19], falta de motivação pelo tema [7] e dificuldade em compreender o que ocorre ao programa enquanto ele é executado no computador [6]. (SOARES, 2017, p.1, documento eletrônico).

Soares (2017) parte de uma revisão literária e se fundamenta em metodologia específica gerando um protocolo como quatro itens, dos quais um deles trata dos critérios de seleção e exclusão dos jogos. Para o critério de seleção foram levantados os seguintes questionamentos:

- i. deve ser um jogo;
- ii. deve apresentar em sua jogabilidade conceitos de introdução à programação;
- iii. deve possuir versão para dispositivos móveis, *Android* ou *iOS*.

Após a tabulação dos dados coletados e análise dos resultados Soares (2017) identificou um quantitativo de 46 (quarenta e seis) jogos eletrônicos para dispositivos móveis que podem apoiar o ensino de programação, dentre eles podemos destacar o *LightBot*³ e o *Spritebox*, ambos dos mesmos criadores, sendo o *Spritebox* um dos objetos de pesquisa desta dissertação.

Para Soares (2017) o *Spritebox* apresenta em sua jogabilidade duas abordagens metodológicas de programação: a Linguagem de Programação (LP) e a utilização de Blocos

3

O LightBot é um jogo de quebra-cabeça baseado em codificação que ensina, de forma implícita, lógica de programação enquanto você joga! (LIGHTBOT, 2019, tradução nossa, documento eletrônico). Disponível em: <http://lightbot.com/>. Acesso em: 16.05.2019.

Visuais (BV). A partir dessas duas metodologias são explorados os conceitos de Pensamento Lógico (PL) e as Estruturas de Repetição (ER). Soares (2017) destaca ainda que “como qualquer ferramenta pedagógica, seu uso requer dos docentes uma avaliação quanto às suas adequações aos objetivos educacionais propostos”.

Nesse sentido, o *Spritebox* está entre os pouco mais de 17% dos jogos investigados que apresenta “referências pedagógicas que nortearam seus processos de desenvolvimento e/ou materiais que podem auxiliar nos seus usos” (SOARES, 2017).

Conhecendo o cenário nacional acerca do uso do SpriteBox como ferramenta auxiliar ao estudo da lógica de programação e da linguagem *Scratch* como recurso para mediação do ensino e aprendizagem de Matemática, assim como as propostas elencadas a partir das análises das produções acadêmicas no recorte temporal de 2012 a 2019 passamos ao próximo capítulo que será nossa base teórica.

4 REFERÊNCIAL TEÓRICO

Para compreender como a utilização da Linguagem de Programação Educativa (LPE) pode contribuir para o processo de ensino e aprendizagem da Matemática apresentaremos neste capítulo as categorias teóricas e seus respectivos autores.

No início da década de 1960 programar um computador era um grande feito para uma criança, visto que a ação era um privilégio de poucos adultos que possuíam um conhecimento especializado. Essa perspectiva foi possível a partir das ideias do americano Seymour Papert, um dos fundadores do MIT (Massachusetts Institute of Technology) e responsável pela criação da linguagem LOGO no final da década de 1960.

Apesar de Papert ser considerado um dos pioneiros quando falamos de TDIC na educação, vale ressaltar que a tentativa de inserção da TDIC na educação começa bem antes disso por volta da década de 1920. Segundo Cysneiro (1999)

Uma das principais referências nesta área é o trabalho de Larry Cuban [ii], professor de educação da Stanford University, intitulado *Professores e Máquinas: O Uso da Tecnologia na Sala de Aula desde 1920*. Cuban estudou a introdução do rádio, filme, TV e computador em escolas norte-americanas, abrangendo a literatura desde o início deste século até meados da década de oitenta. (CYSNEIROS, 1999, p. 2, grifo do autor).

Cysneiros (1999) salienta que esse apanhado histórico sobre a tentativa de inserção da TDIC na educação é fundamental para que possamos analisar que instrumentos e metodologias de fato contribuíram para a melhoria da aprendizagem matemática e quais figuram apenas como meros adereços. Passerino (2010, p. 64) corrobora com essa preocupação de Cysneiros quando enfatiza que os impactos diretos na aprendizagem e desenvolvimento do aluno causados pela inserção das TDIC nas salas de aula se trata de um mito.

Cysneiros in Papert (2008, p.10) salienta que a massiva introdução das TDIC nas escolas não pressupõe melhoria da aprendizagem. Para aprender de forma significativa e duradoura é necessário esforço, persistência e muita tolerância a frustração, pois está muito além da lastimável cultura de copiar da internet, colar, enfeitar e imprimir trabalhos acadêmicos ou escolares. Concordando com essas constatações Silva (2016) ao introduzi a ideia de Pensamento Computacional salienta que

O pensar computacionalmente não pode ser confundido com o simples uso de habilidades específicas ligadas à computação, como, por exemplo, criar arquivos de texto, imagens, slides ou utilizar quaisquer ferramentas informáticas. Há algo mais complexo com esse tipo de raciocínio, pois compreende o computador como

instrumento a nosso serviço, programável e destinado à execução de tarefas por intermédio do pensamento crítico e da criatividade para a resolução de problemas futuros. (SILVA, 2016, p. 35).

Valente (1999) destaca que

Quando o aluno usa o computador para construir o seu conhecimento, o computador passa a ser uma máquina para ser ensinada, propiciando condições para o aluno descrever a resolução de problemas, usando linguagens de programação, refletir sobre os resultados obtidos e depurar suas ideias por intermédio da busca de novos conteúdos e novas estratégias. (VALENTE, 1999, p. 12).

Papert (1993) corrobora com Valente (1999) quando deixa evidente sua insatisfação com modelos submissos de ensino que priorizam a instrução, ou seja, que acreditam que o computador é capaz de ensinar as crianças, sendo o caminho para melhorar a aprendizagem bastando apenas melhorar as instruções. Com isso, o Papert (1993) destina um capítulo inteiro de sua obra para tecer argumentos fundamentados na teoria do desenvolvimento cognitivo infantil de Jean Piaget para combater as concepções de que a máquina poderia nos ensinar.

Mesmo que modo mais abrangente, talvez não diretamente ligado as tecnologias digitais, Freire (2015, p. 16, grifos do autor) concorda com as ideias de Papert quando diz que “*formar é muito mais do que puramente treinar o educando*”.

Contrapondo as ideias instrucionistas Papert (2008) define o Construcionismo como uma filosofia de uma família de filosofias educacionais que nega a ideia de que o aperfeiçoamento da instrução, da escola, melhoraria o ensino, a meta do Construcionismo é ensinar de forma a produzir a maior aprendizagem com um mínimo de ensino. Papert ressalta que o intuito não é apenas reduzir a quantidade de ensino estando todo o resto inalterado. Ele utiliza um provérbio africano para enfatizar que precisamos ensinar as crianças a aprender, pois acredita que o aluno fará melhor descobrindo por si mesmo o conhecimento específico de que precisa, ou seja, construindo seu próprio conhecimento.

Segundo Papert (1980), ao programar o computador a criança adquire domínio sobre a máquina e aproxima-se das ideias concisas da matemática e da construção de modelos intelectuais. Essa ideia da construção do conhecimento a partir da programação de computadores

É construída na hipótese que as crianças serão melhores em encontrar por elas mesmas o conhecimento específico que elas precisam; educação organizada ou informal pode ajudar mais ou ter certeza que eles estão apoiados moralmente, psicologicamente, e intelectualmente em suas tentativas. (PAPERT, 1993, p. 139, tradução nossa).

Para Valente (1998) situações de aprendizagem construcionista trazem benefícios ao professor, que quando utiliza essa abordagem em sua formação inicial passa agir, refletir e depura seu conhecimento aumentando seu domínio sobre os diversos dispositivos e programas digitais e como esses podem contribuir para a mediação do conhecimento em sua prática em sala de aula.

Dessa forma, este construto teórico nos dá base para que possamos propor a utilização da TDIC para mediar o ensino e a aprendizagem de Matemática. Assim, o capítulo a seguir é dedicado a exploração do SpriteBox e da linguagem de programação *Scratch*.

5 DESBRAVANDO O SPRITEBOX E O SCRATCH

Neste capítulo apresentaremos uma visão geral dos programas utilizados na aplicação desta pesquisa. O SpriteBox com interface atrativa com um mix de Mário Bros e Minecraft e o *Scratch* com seus novos e surpreendentes recursos incorporados a partir de sua versão 3.0 lançada em 2019 após fase de experimentação com a versão beta.

5.1 Explorando o SpriteBox e suas possibilidades

5.1.1 Apresentação do SpriteBox

O SpriteBox é um novo jogo de quebra-cabeça que exige o raciocínio lógico do jogador para a construção, intuitiva ou em linha de código, de algoritmos que resultem no cumprimento dos desafios apresentados ao longo dos 4 mundos (fase do jogo na versão gratuita). O jogo é pago, todavia possui uma versão gratuita, que iremos utilizar, disponível tanto para dispositivos móveis quanto para computadores. O software foi desenvolvido pelos mesmos criadores do LightBot (<http://lightbot.com/>) que combina a diversão de jogos de 'plataformas' (como Minecraft e Super Mário Bros) com aprendizagem básica de programação.

O SpriteBox abrange algoritmos básicos, sequenciamento de instruções, instruções com parâmetro, loops simples a complexos e depuração. Sua interface simples e atrativa permite que o usuário crie seu próprio avatar, escolha um idioma preferido, além da opção modo de inserção de comandos.

Neste sentido, este material destina-se as pessoas que desejam conhecer o SpriteBox, aprender as ideias iniciais da linguagem de programação ou mesmo para professores que deseja utilizá-lo como ferramenta para este fim.

5.1.2 O site do SpriteBox

A tela de boas-vindas do site do Spritebox (<http://spritebox.com/>) possui algumas funcionalidades. Na Figura 5.1, temos o topo da página que contém um vídeo de apresentação do jogo. O vídeo tem 30 segundos e mostras alguns momentos do jogo com o Sprite em ação após o recebimento de algum comando. Logo abaixo do vídeo, temos os links para duas lojas de aplicativos: App Store e Google Play.

Figura 5.1 – Tela de boas-vindas.



Fonte: <http://spritebox.com>

Ao finalizar o vídeo de apresentação é mostrado dois links para navegação, um mostrará o jogo Spritebox versão gratuita (a esquerda) e outro mostrará o jogo Lightbot (a direita), conforme Figura 5.2. Caso você queira voltar a página inicial do SpriteBox basta pressionar a tecla ESC do seu teclado.

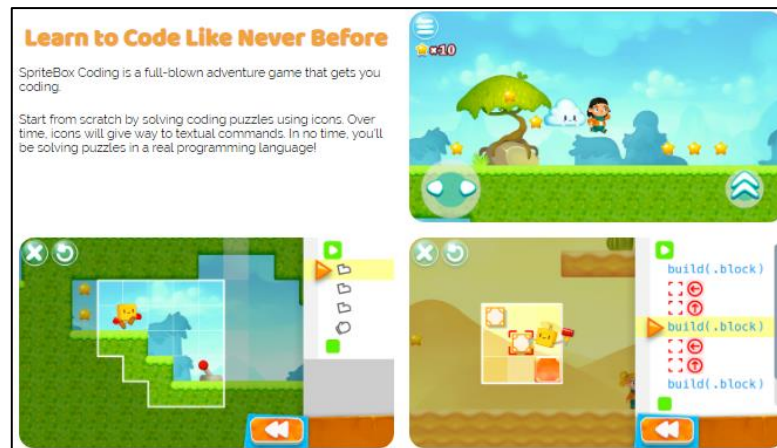
Figura 5.2 – Links de navegação para outros vídeos.



Fonte: <http://spritebox.com>

Na seção mediana da página do SpriteBox tece-se uma breve introdução sobre o jogo através de textos informativos que enfatizam a utilização do pensamento lógico dedutivo e códigos para a resolução de quebra-cabeças que serão propostos durante todo o percurso de cada fase. São mostradas, também, algumas imagens que retratam situações ocorridas durante as fases (chamadas de mundos). As imagens apresentadas na Figura 5.3 são mundos da versão paga do jogo.

Figura 5.3 – Breve apresentação do SpriteBox.

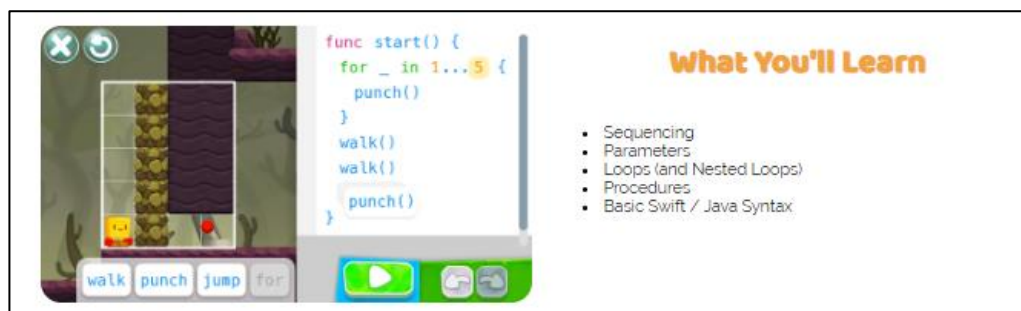


Fonte: <http://spritebox.com>

Um pouco mais abaixo, nessa mesma seção mediana da página, temos as habilidades que poderão ser desenvolvidas por um jogador que venha a concluir os desafios dos 4 mundos (Figura 5.4). Segundo os criadores do SpriteBox, o jogador aprenderá a trabalhar com:

- 🤖 Sequenciamento;
- 🤖 Parâmetros;
- 🤖 Laços de repetição (e laços animados);
- 🤖 Procedimentos (blocos de códigos para realizar determinada tarefa);
- 🤖 Sintaxes Basic Swift e Java (para os que optam por jogar com essas linhas de entrada de códigos).

Figura 5.4 – Habilidades desenvolvidas ao longo do jogo.



Fonte: <http://spritebox.com>

Na penúltima seção, temos uma chamada que desponta o jogo como amigável a pessoas de qualquer faixa etária (Figura 5.5) e destaca que o SpriteBox já foi jogado por mais de 20 milhões de crianças e usado por dezenas de milhares de professores em todo o mundo. Além disso, traz novamente os links para as duas lojas de aplicativos que disponibilizam o jogo.

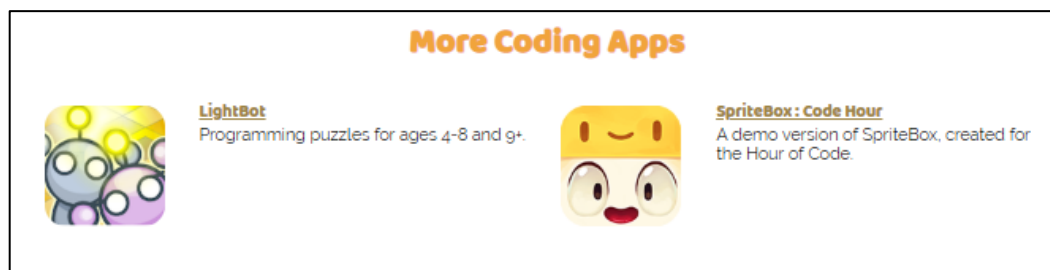
Figura 5.5 – SpriteBox para todas as idades e links para lojas de Apps.



Fonte: <http://spritebox.com>

Na última seção da página oficial do SpriteBox são mostrados dois links para que possamos acessar o site do LightBot, precursor do SpriteBox, e a plataforma online que nos permite jogar a versão demonstrativa do SpriteBox via web, denominada SpriteBox: Code Hour.

Figura 5.6 – Links de acesso ao LightBot e SpriteBox Code Hour.



Fonte: <http://spritebox.com>

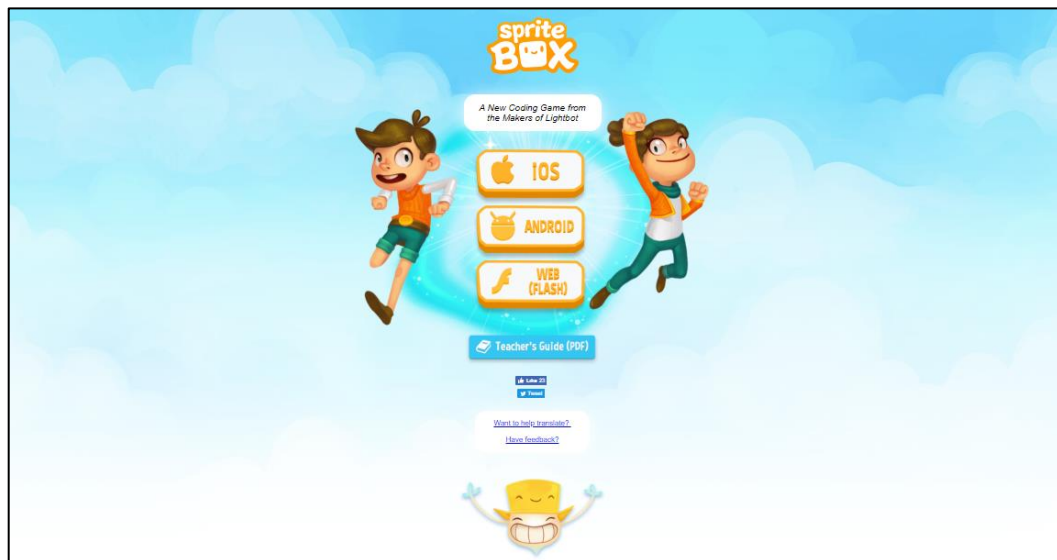
Clicando no link a direita (Figura 5.6) você será redirecionado a página inicial do jogo Spritebox, onde teremos, além da versão Web Flash (versão gratuita) que possibilita que você jogue online caso esteja conectado à internet, outros links que te levarão, por exemplo, as lojas de aplicativos de dispositivos móveis que utilizam sistemas iOS⁴ e Android⁵ (Figura 5.7).

⁴ iOS (antes chamado de iPhone OS) é um sistema operacional móvel da Apple Inc. desenvolvido originalmente para o iPhone, também é usado em iPod touch e iPad. A Apple não permite que o iOS seja executado em hardware de terceiros.

⁵ Android é um sistema operacional (SO) baseado no núcleo Linux e atualmente desenvolvido pela empresa de tecnologia Google. Com uma interface de usuário baseada na manipulação direta, o Android é projetado principalmente para dispositivos móveis com tela sensível ao toque como smartphones e tablets; com interface específica para TV (Android TV), carro (Android Auto) e relógio de pulso (Android Wear).

Um guia em formato PDF é disponibilizado para professores (*Teacher's Guide (PDF)*) que desejam utilizar o SpriteBox com seus alunos, que inclusive foi a base para a construção dessas orientações, já que o material original se encontra em língua inglesa. Além disso, temos outros três links importantes: *compartilhamento* em redes sociais, *ajuda para tradução*, *feedback*.

Figura 5.7 – Tela inicial do SpriteBox Code Hour.



Fonte: <http://spritebox.com/hour.html>.

No link de *compartilhamento* é possível compartilhar em suas redes sociais (twitter e facebook) essa nova descoberta. Se você conheceu o SpriteBox porque ouvir alguém falar ou compartilhar em mídias sociais, então faça o mesmo, compartilhe e fale para as pessoas sobre este jogo que torna as ideias iniciais de programação divertida e prazerosa.

O link *ajuda para tradução* o direciona ao gerenciador de email do seu computador, onde você poderá encaminhar suas dúvidas diretamente aos criadores do SpriteBox por meio do endereço eletrônico <contact.lightbot@gmail.com>.

O *feedback* permite que você relate sua experiência com o SpriteBox através de um Formulário Google (*Google Forms*⁶). É necessário deixar um email para confirmação de seu envio. Neste formulário respondemos as seguintes perguntas: “Que versão você jogou?” (Android ou Web Flash); “Como você avalia sua experiência jogando o SpriteBox?” (atribuir

⁶ O Formulários Google é um aplicativo de administração de pesquisas incluído na suíte de escritório do Google Drive. Disponível em: https://en.wikipedia.org/wiki/Google_Forms. Acesso em: 20.mai.2019.

nota de 1 a 10); “Há alguma característica que gostaria de ver?” (alago que pudesse ser acrescentado ao jogo); “Você tem alguma crítica ou bug a relatar?”; “Algum comentário final?”.

Figura 5.8 – Questionário de feedback.

SpriteBox - Alpha Feedback

We hope you enjoyed playing the alpha version of SpriteBox at spritebox.com/hour.html. We really appreciate all the input you can give us, so we ask that you fill in as many answers as you can below. We will then do our best to make SpriteBox that much better with your feedback. Thank you for you help!

*Obrigatório

Which version(s) did you play? *

☒ Android

☒ Web/Flash

How would you rate your experience playing SpriteBox?

1 2 3 4 5 6 7 8 9 10

Disliked ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☒ Loved

Are there any features that you'd like to see?

Yes, I'd like to see a comparison between my solve and the shortest solve.

Do you have any critiques or bugs reports?

No, I don't.

Any final comments?

The SpriteBox is a fantastic game to start the programing learn.

ENVIAR

Fonte: <https://docs.google.com/forms/d/e/1FAIpQLSeSeGQciMI7I6-qnkXXOg6odKECfBd1lbgD0aqereozLhm3yg/viewform>

5.1.3 Versão web flash do SpriteBox

A versão Web Flash pode ser jogada em qualquer navegador, desde que o *flash player* esteja atualizado, caso contrário o jogo não iniciará e será mostrada uma mensagem de erro.

Caso necessário, podemos instalar a versão gratuita do Adobe Flash Player disponível em: <<https://get.adobe.com/br/flashplayer/>>. Para isso, basta acessarmos esse link e seguirmos o passo a passo mostrado na tela.

5.1.4 A tela inicial da versão web flash

A tela inicial do SpriteBox, versão gratuita para web, possui oito botões responsáveis pela configuração do avatar, língua padrão, opção de visualização dos códigos, opção de tela, habilitar e desabilitar sons, jogar e apagar. Na Figura 5.9 é apresentado um esquema que destaca cada um desses botões, em seguida é feito a descrição das funcionalidades de cada botão.

Figura 5.9 – Tela inicial e botões.



Fonte: <http://spritebox.com/codehour>

1. **Tela Cheia:** utilizado para maximizar a tela de visualização do jogo;
2. **Cor do Avatar:** permite que você escolha uma cor para a pele do seu avatar;
3. **Avatar:** mostra três opção para escolha do avatar;
4. **Apagar:** utilizado para resetar o jogo, zerando todas as fases;
5. **Idioma:** permite que você escolha a língua padrão entre português e inglês;
6. **Código:** permite que o usuário escolha a forma de visualização dos códigos, que pode ser feito de três maneiras, conforme Quadro 5.1.
7. **Som:** utilizado para habilitar ou desabilitar o som do jogo;
8. **Play:** botão utilizado para dar início ao jogo.

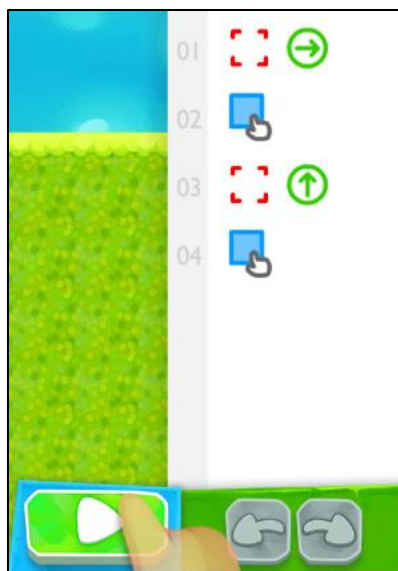
Quadro 5.1 – Opções de visualização dos comandos.

	Modo bloco de comando	Blocos com comandos pré-definidos que permitem que o usuário iniciante na programação visualize uma imagem que representa determinado código.
	Modo linha de comando	Ambos os modos permitem que o usuário com mais experiência possa visualizar os comandos em linhas de códigos.
	Modo linha de comando	

Fonte: Elaboração própria.

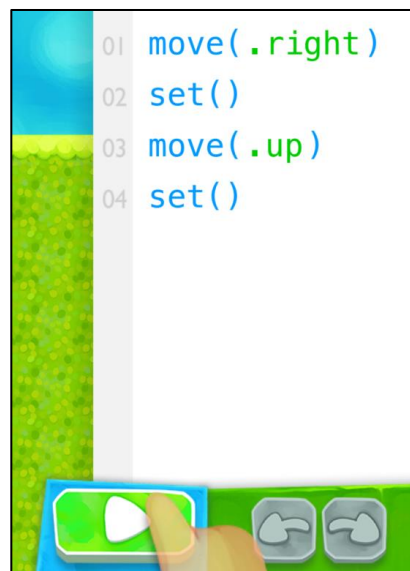
Podemos observar nas Figura 5.10.a e Figura 5.10.b a diferença na visualização dos códigos apresentados no jogo quando optamos pelo **modo bloco de comando** ou pelo **modo linha de comando**.

Figura 5.10.a – Modo bloco de comando.



Fonte: Do jogo SpriteBox.

Figura 5.10.b – Modo linha de comando.







Fonte: Do jogo SpriteBox.

Neste projeto utilizaremos o *Modo bloco de comando*. Fica a cargo do usuário explorar os outros modos de visualização dos códigos.

5.1.5 Personagens do jogo

O SpriteBox versão demonstrativo possui três personagens: o avatar, o spritebox e o sprite. Veja no Quadro 5.2.

Quadro 5.2 – Personagens do SpriteBox.

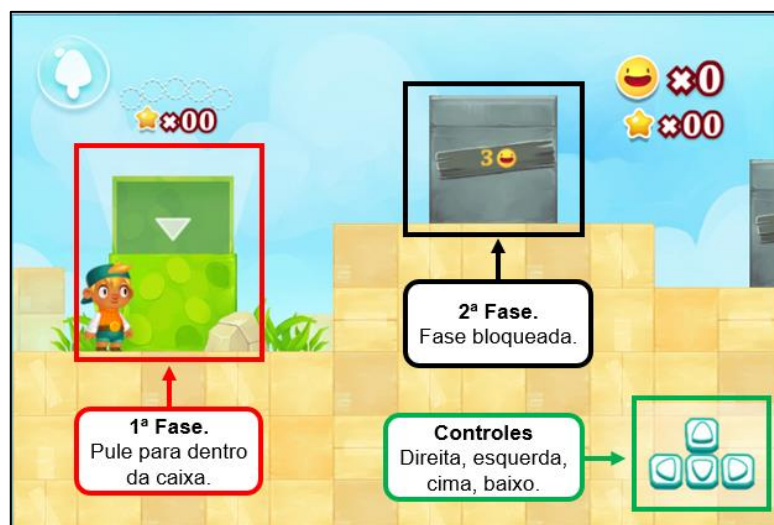
	 	
Avatar Personagem responsável por explorar os mundos.	spritebox Personagem responsável por ajudar a executar as instruções fornecidas pelo avatar.	sprite Personagens aprisionados que aguardam para serem libertados pelo avatar.

Fonte: Elaboração própria.

5.1.6 Tela secundária para jogar via web

Na tela secundária de escolha de fase temos quatro caixas, as quais representam os quatro mundos ou fases. A primeira caixa é a única desbloqueada (Figura 5.11), que se abrirá assim que o usuário posicione o avatar a frente dela. Cada fase é iniciada quando o avatar pula para dentro da respectiva caixa. As outras caixas começam bloqueadas e são liberadas sempre que você obtém êxito no nível anterior com a coleta de mais Sprites.

Figura 5.11 – Tela secundária de escolha de fase.



Fonte: Adaptado do jogo SpriteBox.

Além dos mundos, temos os botões de controle: correr para frente (direita), para traz (esquerda), pular ou subir (cima) e descer (baixo), conforme Figura 4.1. O controle do avatar é feito exclusivamente com as setas direcionais.

Para começar basta corre para baixo da caixa verde, quando ela abrir pule para dentro dela para começar o primeiro mundo ou primeira fase.

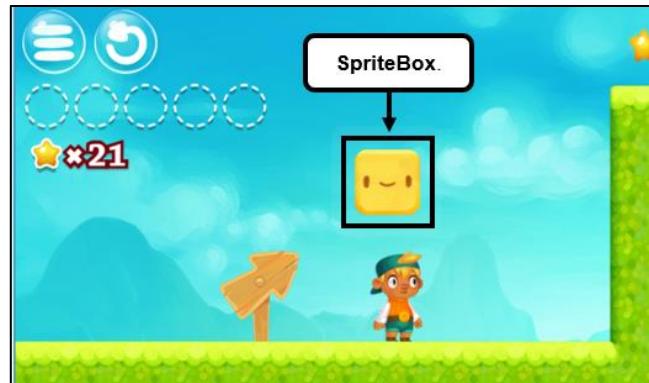
O Jogo alterna a entre dois momentos, o da exploração e o da codificação. A seguir falaremos sobre cada um desses modos.

5.1.7 Modo exploração

No modo exploração o jogador pode correr, pular e subir, com o objetivo de encontrar Sprites e coletar estrelas. Durante a exploração os jogadores descobrirão que algumas plataformas estão fora de alcance ou algumas seções são intransitáveis. Nesse momento, o

jogador deve procurar por uma spritebox (caixa amarelo) (Figura 5.12), pular e bater de baixo dele, ativando o modo ajudante do spritebox que, inicialmente mostrará um tutorial e tentará ajudá-lo a construir o caminho mais adequado para salvar os Sprites.

Figura 5.12 – SpriteBox no modo exploração.



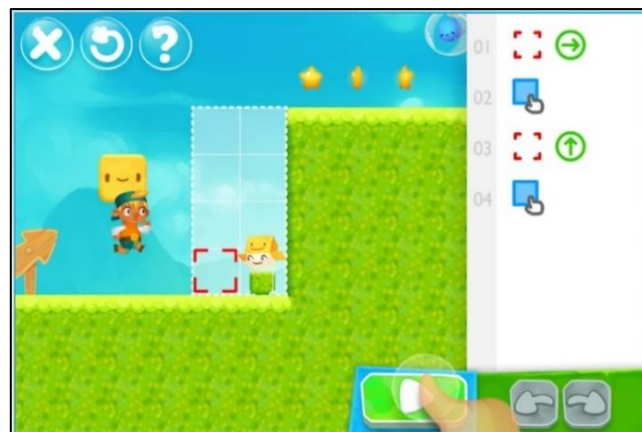
Fonte: Adaptado do jogo Spritebox.

5.1.8 Modo código

Os jogadores podem controlar o spritebox ajudante, um pequeno personagem fantasma que está usando um chapéu de caixa, usando o código. O spritebox pode então ajudar o jogador a progredir, modificando o ambiente do jogo e tornando as seções anteriormente intransitáveis acessíveis.

O spritebox irá apresentar várias instruções ao longo de alguns níveis, onde aparece uma instrutiva animação de mão, a qual os jogadores devem tentar copiar, fazendo o que ela faz, seja clicando em um botão ou arrastando uma instrução.

Figura 5.13 – SpriteBox no modo exploração.



Fonte: Do jogo SpriteBox.

5.1.9 Comandos utilizados no SpriteBox

Em todas os mundos do jogo são utilizados apenas 4 comandos: SET, MOVE, LOOP e GET. A seguir faremos uma breve exposição das funcionalidades de cada um desses comandos.

O comando SET, representado pela Figura 5.14, é utilizado para dar uma ordem de execução ou construção. Um detalhe importante é que o spritebox sempre construirá o que estiver em suas mãos.

Figura 5.14 – Comando SET.



Fonte: Do jogo SpriteBox

O comando MOVE é essencial para o desenvolvimento do jogo, pois permite que o spritebox se desloque para cima, para baixo, para frente ou para trás (Figura 5.15) quando este está a realizar alguma instrução passada pelo jogador.

Figura 5.15 – Comando MOVE.



Fonte: Do jogo SpriteBox.

Em algumas situações é necessário que repetir um determinado comando duas, três ou até mais do que três vezes. Quando isso acontece é viável se utilizar o comando LOOP (Figura 5.16) que possibilita que um determinado comando seja repetido uma quantidade de vezes pré-determinada pelo usuário. Esse comando também permite que o usuário repita um bloco de repetição, que é útil em situações com *loops* superiores a sete repetições.

Figura 5.16 – Comando LOOP.



Fonte: Do jogo SpriteBox.

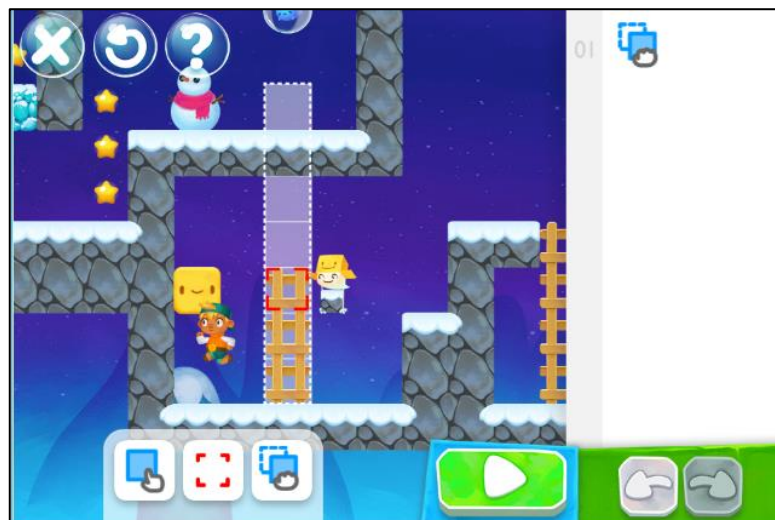
O comando GET (Figura 5.17.a) permite que o SpriteBox copie um determinado elemento do jogo. Na Figura 5.17.b, por exemplo, é necessário que a construção da escada seja concluída, para isso o spritebox deverá copiar, com o comando GET, o elemento escada, para que posteriormente com o comando SET construa o que falta da mesma.

Figura 5.17.a – Comando GET.



Fonte: Do jogo SpriteBox.

Figura 5.17.b – Utilizando o comando GET.



Fonte: Do jogo SpriteBox.

5.2 Explorando a linguagem Scratch e suas possibilidades

5.2.1 Apresentação do Scratch

O *Scratch* é uma linguagem de programação gratuita que utiliza blocos lógicos, itens de som e imagem, para o desenvolvimento de histórias interativas, jogos e animações que permite o compartilhamento de maneira online de suas criações. O *Scratch* é um projeto do grupo Lifelong Kindergarten no Media Lab do MIT (Instituto de Tecnologia de Massachusets), tendo como idealizador Mitchel Resnick. Vale ressaltar que a linguagem *Scratch* precede a linguagem LOGO, criada no final da década de 1960 pelo também construcionista Seymour Papert, cofundador do MIT.

A capacidade de escrever programas de computador é uma parte importante da alfabetização na sociedade de hoje. Quando as pessoas aprendem a programar usando o Scratch, elas aprendem estratégias importantes para a resolução de problemas, projetos de design e comunicação de ideias. (SCRATCH BARSIL, 2014, documento eletrônico).

Segundo Lévy (1999) a grande velocidade das inovações tecnológicas, as decorrentes mudanças no mundo do trabalho, a disseminação de novas formas de conhecimento consolidam o que o autor chama de ciberespaço, onde o saber articula-se à nova perspectiva de educação, em função das novas formas de construção do conhecimento, que contemplam a democratização do acesso à informação, os novos estilos de aprendizagem e a emergência da inteligência coletiva, pairando como evidências que acabam por questionar os modelos tradicionais de ensino, que enfatizam a transmissão dos saberes.

Apesar de ter sido projetado para crianças com faixa etária entre 8 e 16 anos, a linguagem *Scratch* é utilizada por milhões de pessoas de todas as faixas de idade em uma grande variedade de configurações, inclusive casas, escolas, museus, bibliotecas e centros comunitários.

Numa palestra ministrada em novembro de 2012 no TED⁷, Mitchel Resnick relata sua surpresa ao se deparar com cartões interativos para o dia das mães. Ressalta que quando criou a linguagem *Scratch* esperava que os jovens usuários da linguagem pudessem criar algo criativo, que propiciasse a expressão e o compartilhamento de suas ideias.

⁷ TED é uma organização sem fins lucrativos dedicada ao lema “ideias que merecem ser compartilhadas”. Começou há 26 anos como uma conferência na Califórnia, e, desde então, o TED tem crescido para apoiar ideias que mudam o mundo através de múltiplas iniciativas. Disponível em: <https://www.tedx-dantealighierischool.com.br/o-que-e-tedx/>. Acesso em: 18.mai.2019.

Resnick compara a fluência em programação com a fluência em uma língua estrangeira. Quando se domina uma outra língua, pode-se escrever um diário, contar piadas, escrever uma carta para um amigo. De modo análogo ocorre com a fluência em programação, quando os jovens passam a dominar e se apropriam das novas tecnologias para expressar e compartilhar suas ideias.

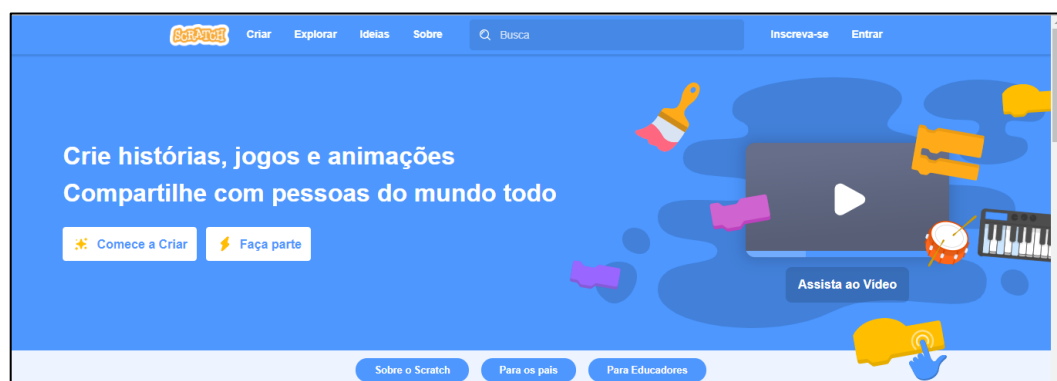
“A habilidade de escrever programas de computador é uma parte importante da alfabetização na sociedade. Quando as pessoas aprendem a programar no *Scratch*, elas aprendem estratégias importantes para resolver problemas, desenvolver projetos e comunicar ideias.” (SCRATCH, 2019, documento eletrônico). Além disso, o *Scratch* é utilizado em mais de 150 países diferentes e está disponível em mais de 40 idiomas. Para mudar o idioma, basta clicar no botão especial em forma de lista suspensa e escolher o idioma desejado.

De maio de 2009 até maio de 2019, mais de 27000 educadores que apoiam o aprendizado com a linguagem de programação *Scratch* compartilharam 4749 postagens de discussão, 1027 recursos e 354 histórias aqui na Comunidade Online do ScratchEd (SCRATCHED, 2019, tradução nossa, documento eletrônico).

5.2.2 O site do Scratch

A tela de boas-vindas do site do *Scratch* (<https://scratch.mit.edu/>) possui cores alegres e vibrantes com figuras que retratam a ideia de uma linguagem de programação sem linhas de código, estando dividida em 5 seções. Na Figura 5.18, temos a visualização do topo da página, onde podemos observar em destaque 3 dessas seções: barra de menu (composta de 7 botões e o campo de busca); Cabeçalho (logo abaixo, dividido em duas colunas, composta de 3 botões) e; botões de informação (sobre o *Scratch*, para pais e para educadores).

Figura 5.18 – Tela de boas-vidas do site do *Scratch*.



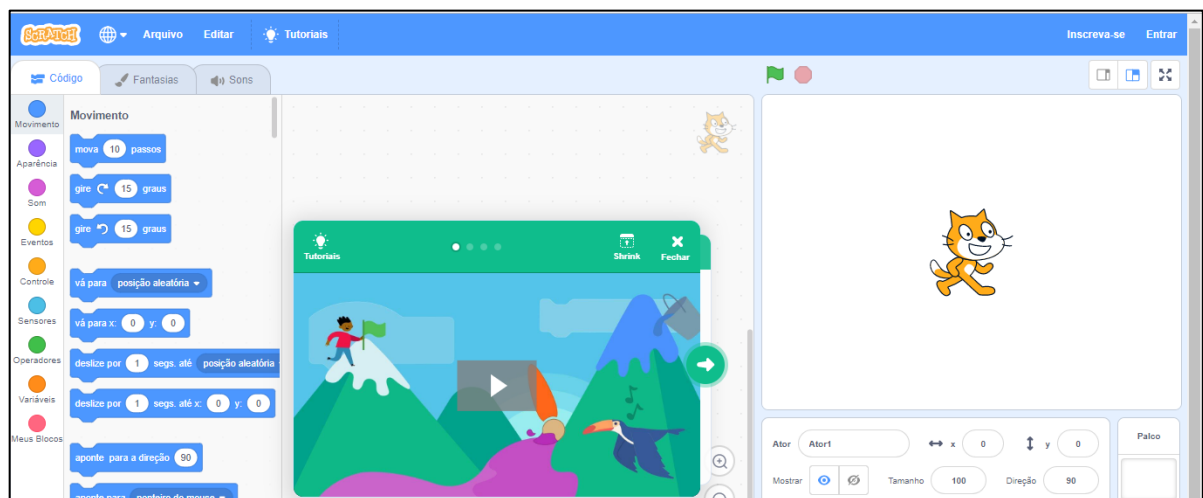
Fonte: <https://scratch.mit.edu/>

Na barra superior temos os botões: Scratch; Criar; Explorar; Ideias; Sobre; Inscreva-se; Entrar e a o campo de busca.

O botão *Scratch* tem a função de um botão página inicial, redirecionando-o para a tela de boas-vindas sempre que clicado.

O botão Criar nos direciona ao estúdio de criação online do *Scratch*, mostrando a tela inicial da interface de programação (Figura 5.19).

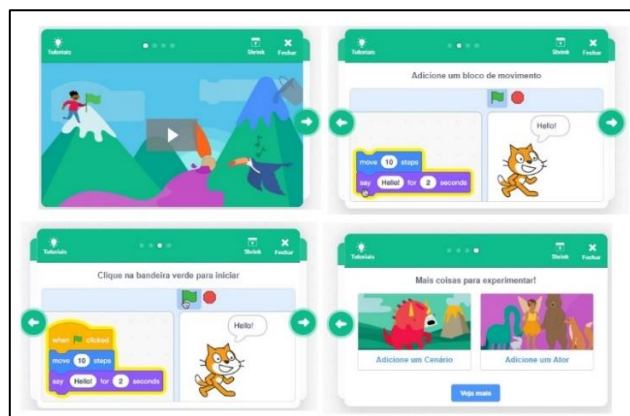
Figura 5.19 – Tela inicial do estúdio de criação do *Scratch*.



Fonte: <https://scratch.mit.edu/projects/editor/?tutorial=getStarted>

Nesta versão do *Scratch* (versão 3.0) a tela inicial do estúdio de criação traz um pequeno tutorial (Figura 5.20) em uma tela verde que contém um vídeo de 52 segundos; dois passos a passos para movimentar e adicionar sons a um ator e; outros links tutoriais que nos ajudam a adicionar um novo cenário ou adicionar um novo personagem (ator).

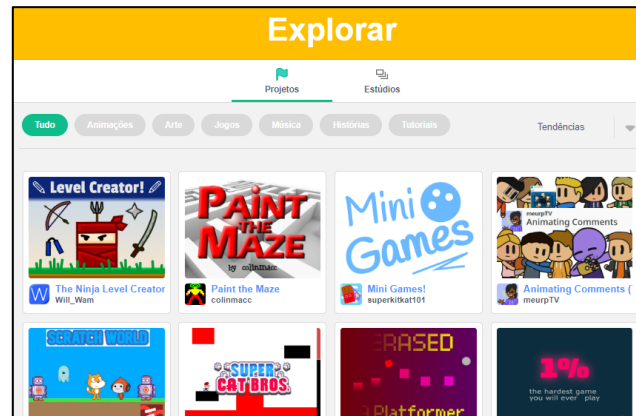
Figura 5.20 – Tutorial da tela inicial do estúdio de criação.



Fonte: <https://scratch.mit.edu/projects/editor/?tutorial=getStarted>

O botão Explorar da barra de menu (Figura 5.21) nos fornece milhares de projetos construídos com a linguagem *Scratch*. O usuário pode alternar entre as opções de exibição de Projetos e Estúdios, além de poder refinar sua busca clicando em algum dos seis temas disponíveis: Animações, Arte, Jogos, Música, Histórias e Tutoriais.

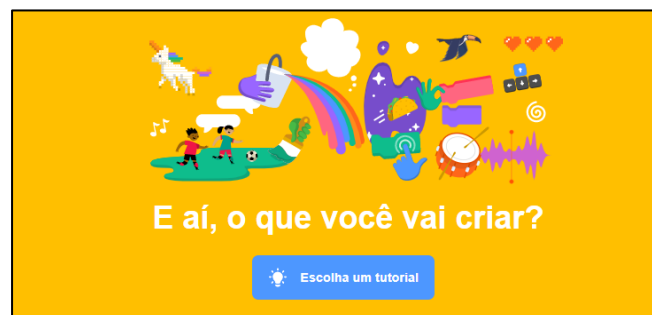
Figura 5.21 – Página do botão Explorar.



Fonte: <https://scratch.mit.edu/explore/projects/all/>.

O botão Ideias (Figura 5.22) direciona o usuário a uma página repleta de projetos e tutorias que servirão de base para suas primeiras criações com o *Scratch*. Nessa página estão disponíveis botões como: Escolha um tutorial, Experimente (primeiros passos), além de um guia de atividades que nos oferece um conjunto de cartões *Scratch*, disponíveis para download, que nos auxiliarão em nossas criações.

Figura 5.22 – Página do botão Ideias.



Fonte: <https://scratch.mit.edu/ideas>.

O botão Sobre direciona o usuário a uma página repleta de informações acerca do *Scratch*, desde sua criação até sua utilização na educação. Logo em sua vista inicial temos um vídeo de 1 minuto que nos mostra algumas das aplicações construídas com a linguagem *Scratch*.

Explorando mais a páginas encontramos informações para pais ou responsáveis, para educadores, faixa etária destinada o projeto inicial, quem usa o *Scratch*, *Scratch* nas escolas, pesquisas, citações, etc.

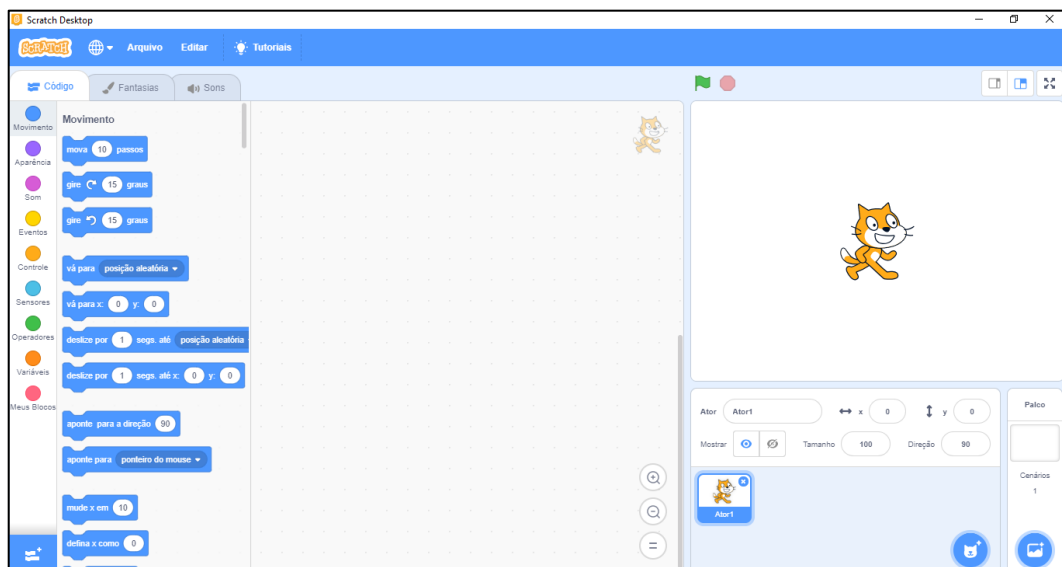
5.2.3 Estúdio de criação Scratch

O estúdio de criação do *Scratch* pode ser acessado online através do endereço eletrônico <<https://scratch.mit.edu/projects/editor/?tutorial=getStarted>> ou descarregado no computador do usuário através do link *Editor Offline* disponível no rodapé da página inicial do *Scratch*, sendo o terceiro item na lista pertencente ao campo *Suporte*. A tela inicial do estúdio de criação da versão 3.0 (Figura 5.23) chamada de *Scratch Desktop* tem um layout mais atraente e mais limpo em relação a versão 2.0 (Figura 5.24).

Na versão 3.0 as abas Códigos, Fantasia e Sons encontram-se organizados na lateral esquerda, a área destinada a inserção dos códigos foi deslocada para o centro e o palco (área onde as ações são executadas pelos atores) está fixa a direita podendo ser reduzida a partir de uma segunda opção de layout de tela disponível.

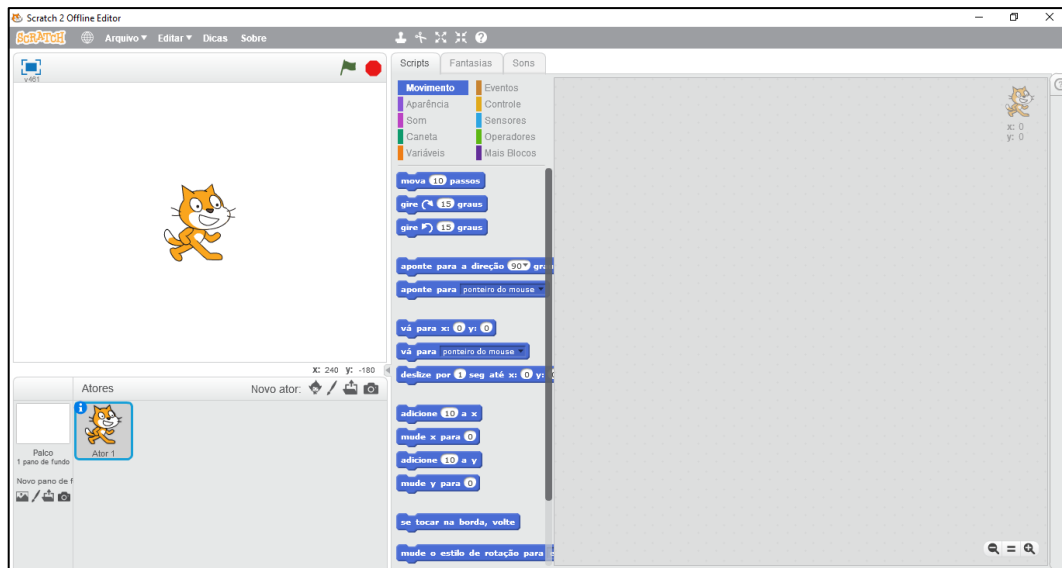
A aba Códigos está distribuída em 9 categorias de blocos lógicos principais: movimento, aparência, sons, eventos, controle, sensores, operadores, variáveis e mais blocos. A novidade fica por conta do botão *Adicionar Uma Extensão*, que abarca recurso antigos como a Caneta e outros modernos e interativos como o detector de vídeo, conexão com LEGO e MicroBit, etc.

Figura 5.23 – Tela inicial do estúdio de criação 3.0.



Fonte: <https://scratch.mit.edu/projects/editor/?tutorial=getStarted>.

Figura 5.24 – Tela inicial do estúdio de criação 2.0.



Fonte: <https://scratch.mit.edu/projects/editor/?tutorial=getStarted>.

Alguns recursos inerentes ao ator, que na versão 2.0 ficavam escondidos, como o tamanho do ator, a direção, a opção de mostrar ou não um ator, na versão 3.0 ficam visíveis e de fácil acesso. As opções de escolha de um palco ou ator ganharam novos ícones e uma lista suspensa que facilitam a identificação da ação a ser executada pelo usuário.

6 DA APLICAÇÃO DO PROJETO LEP

Neste capítulo apresentaremos os sujeitos, os materiais e métodos e o cenário de realização da aplicação da pesquisa, a qual se deu a partir de um projeto de extensão elaborado por este pesquisador e colocado em apreciação da coordenação e equipe gestora da Escola Guaraci Mendes, tendo como sujeitos alunos do 9^a ano do Ensino Fundamental. Os encontros do projeto ocorreram na Sala de Informática da Escola Municipal de Ensino Fundamental (EMEF) Guaraci Mendes situada no município de Tailândia, sudeste do estado do Pará.

6.1 Um breve relato sobre o município de Tailândia

Tailândia/PA é um município brasileiro do estado do Pará, pertencente à Mesorregião do Nordeste Paraense. O município teve sua emancipação estabelecida pela Lei 5.452 de 10 de maio de 1988 sendo desmembrada do Município de Acará/PA. O município é atravessado de norte a sul pela PA-475, que lhe concede acesso a todo estado do Pará.

Segundo o Instituto Brasileiro de Geografia e Estatística (IBGE) Tailândia/PA tinha uma estimativa populacional de cerca de 103 664 habitantes para o ano de 2018, distribuídas em um território de 4 430,477 km² gerando uma densidade demográfica de aproximadamente 23,40 hab/km².

O município tem um Índice de Desenvolvimento Humano (IDH) de 0,588. Dos 5570 municípios brasileiros Tailândia/PA encontra-se na posição 4772 quando o assunto é saneamento básico, visto que possui apenas 5,4% de esgotamento sanitário adequado e pouco mais de 1% de vias públicas urbanizadas.

Quando falamos em economia, Tailândia/PA ocupa a posição 4 299, com renda per capita de R\$ 8.711,62, sendo 92% de sua receita oriunda de fontes externas. (IBGE, 2010).

De acordo com o site oficial da Prefeitura Municipal de Tailândia/PA (PMT),

Quanto à educação, pelos dados do PAD, 48% da população tinha apenas o curso primário incompleto, 24,22% analfabetos, 23% não se encontravam em idade escolar, 2,45% ginásio completo, 0,35% colegial (ensino médio) e 0,4% com escolaridade de nível superior. Enquanto isso, a carência educacional se refletia no número de escolas em 1980, que eram de 14 na colônia (zona rural) e 01 na vila (perímetro urbano), ou seja, a Escola de 1º Grau Prof. Gabriel Lage da Silva, com dois professores de ensino médio e outros educadores. Tailândia ainda não possuía uma escola de ensino médio. (PMT, 2019, documento eletrônico).

Segundo dados do site QEDU, com base nos resultados da Prova Brasil 2017, foi possível calcular a proporção de alunos com aprendizado adequado em Matemática nos anos finais do Ensino Fundamental da escola municipal Guaraci Mendes. O cálculo revela que de cada 100 alunos, apenas 19 conseguiram alcançar o aprendizado adequado na disciplina de matemática, ou seja 19%. A situação geral do município é um pouco mais preocupante, pois a proporção de alunos com aprendizado adequado é de 13%.

Na Tabela 6.1 e no gráfico da Figura 6.1 é possível verificar a distribuição dos alunos do 9º ano do Ensino Fundamental do município de Tailândia/PA em relação ao aprendizado em matemática na última avaliação nacional em 2017 em quatro níveis de proficiência:

- **Avançado:** alunos com pontuação igual ou acima de 350 pontos em Resolução de Problemas de matemática na Prova Brasil;
- **Proficiente:** alunos com pontuação de 300 a 349;
- **Básico:** alunos com pontuação de 225 a 299;
- **Insuficiente:** alunos com pontuação de 0 a 224.

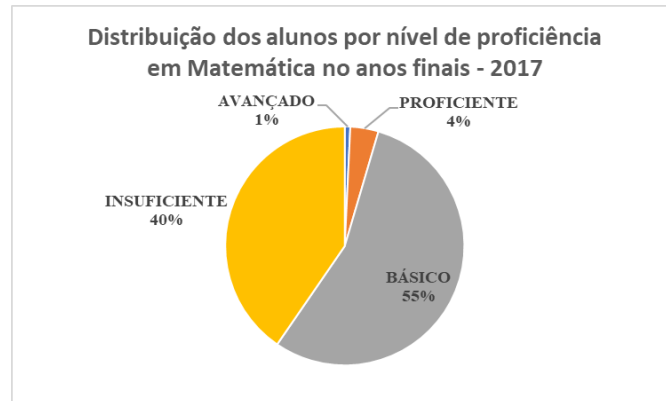
Tabela 6.1: Proficiência em matemática em 2017.

NÍVEL DE PROFICIÊNCIA	QUANTIDADE DE ALUNOS
AVANÇADO	6
PROFICIENTE	31
BÁSICO	448
INSUFICIENTE	329
TOTAL DE ALUNOS	814

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

Analisando a Tabela 6.1 podemos notar que 329 alunos da rede municipal de Tailândia/PA apresentam pouquíssimo aprendizado na resolução de problemas em matemática, sendo necessária, nesse sentido, a recuperação de conteúdos, que pode se dá a partir de projetos pedagógicos extraclasse. A partir de técnicas estatísticas é possível inferir que 40% dos alunos matriculados se enquadram no nível de aprendizagem considerado insuficiente, como mostra no gráfico da Figura 6.1.

Figura 6.1 – Proficiência em matemática em 2017.



Fonte: Elaboração própria.

Quando comparamos os resultados de proficiência em matemática do município de Tailândia/PA com o Brasil constatamos uma diferença de 10 pontos percentuais, visto que em 2017 a proporção de alunos que tiveram o nível de aprendizado esperado foi de 15%. Já em comparação ao estado do Pará, Tailândia/PA apresenta a mesma proporção de alunos com nível proficiente na resolução de problemas, 5% (QEDU, 2019, documento eletrônico).

6.2 A Escola Municipal Guaraci Mendes

A Escola Municipal, inicialmente de ensino infantil e fundamental, Guaraci Mendes (Figura 6.2) atende hoje alunos dos anos finais do ensino fundamental. Sediada na Avenida Rio Branco, nº. 139, Bairro Novo, Tailândia/PA, a escola funciona sobre a resolução de autorização Decreto nº. 001/2009.

Figura 6.2 – Frente da Escola Guaraci Mendes.



Fonte: Projeto Político Pedagógico (PPP) da Escola.

Com uma grande demanda de alunos das proximidades da escola e também de alunos oriundos da zona rural do município, a escola foi fundada no dia 02 de janeiro de 2009, em caráter de emergência, por meio do Decreto-Lei nº. 001.

O Quadro 6.1, com as características estruturais da escola, mostra um ambiente educacional com boa infraestrutura e propício ao desenvolvimento deste projeto.

Quadro 6.1 – Infraestrutura da escola Guaraci Mendes.

QUANTIDADE	DEPENDÊNCIAS
01	Arquivo
06	Banheiros
01	Almoxarifado
01	Coordenação
14	Salas de aula
01	Sala dos professores
01	Sala de recursos multifuncional-AEE
01	Secretaria
01	Depósito para guarda de gêneros alimentícios
01	Refeitório
01	Direção
01	Sala de informática
01	Sala de leitura
01	Cozinha
01	Deposito de materiais de limpeza;
01	Área de serviço
01	Sala no prédio em cima para beneficiar o programa mais educação
01	Deposito de merenda
01	Quadra Poliesportiva
02	Depósito de materiais esportivos de educação física
01	Depósito para guardar ferramentas e utensílios para a limpeza da piscina.
04	Quiosque na área externa.
01	Área de piscina, fechada com grade para a segurança do aluno.

Fonte: Projeto Político Pedagógico (PPP) da Escola.

Quando analisamos a situação da escola Guaraci Mendes a luz das avaliações nacionais, como a Prova Brasil, observamos que esta desponta como destaque no município de Tailândia/PA, visto que em 2015 obteve nota 4,6, com um aprendizado em matemática de 248,61. Este resultado deixou a escola 0,4 pontos acima da meta estabelecida pelo INEP para 2015, que era de 4,2 (Figura 6.3), podendo essa boa infraestrutura ter contribuído de forma direta ou indireta para o mérito em questão.

Figura 6.3 – Metas e IDEB observado da escola Guaraci Mendes.

8ª série / 9º ano																
Escola	Ideb Observado							Metas Projetadas								
	2005	2007	2009	2011	2013	2015	2017	2007	2009	2011	2013	2015	2017	2019	2021	
E M E F GUARACI MENDES				3.7		4.6	4.2				3.9	4.2	4.5	4.7	5.0	

Fonte: <http://ideb.inep.gov.br/resultado/resultado/resultado.seam?cid=1905201>

6.2.1 A escala SAEB

O Sistema de Avaliação da Educação Básica (Saeb) é composto por um conjunto de avaliações externas em larga escala que permitem ao Inep realizar um diagnóstico da educação básica brasileira e de alguns fatores que possam interferir no desempenho do estudante, fornecendo um indicativo sobre a qualidade do ensino ofertado (INEP, SAEB, 2019, documento eletrônico).

Por meio de provas e questionários, aplicados periodicamente pelo Inep, o Saeb permite que os diversos níveis governamentais avaliem a qualidade da educação praticada no país, de modo a oferecer subsídios para a elaboração, o monitoramento e o aprimoramento de políticas com base em evidências (INEP, SAEB, 2019, documento eletrônico).

Para entender o significado ou mesmo em que nível de aprendizado de matemática a escola Guaraci Mendes se encontra é necessário conhecer a escala SAEB. Segundo o site QEDU, “na Prova Brasil, o resultado do aluno é apresentado em pontos numa escala (Escala SAEB). Essa escala foi reformulada pelo Inep e agora é única para cada disciplina e ano” (QEDU, 2019, documento eletrônico).

Com um aprendizado em matemática avaliado em 248,61 no ano de 2015, a escola Guaraci Mendes se enquadra no nível 2 de proficiência da escala SAEB (Figura 6.4). Apesar de ser considerado um bom resultado, pois gera um IDEB acima da meta, ainda está longe de um aprendizado adequado.

Figura 6.4 – Nível de enquadramento da escola Guaraci Mendes em 2015.

<p>Nível 2 Desempenho maior ou igual a 225 e menor que 250</p>	<p>Além das habilidades anteriormente citadas, os estudantes provavelmente são capazes de:</p> <p>NÚMEROS E OPERAÇÕES; ÁLGEBRA E FUNÇÕES</p> <p>Reconhecer a fração que corresponde à relação parte-todo entre uma figura e suas partes hachuradas.</p> <p>Associar um número racional que representa uma quantia monetária, escrito por extenso, à sua representação decimal.</p> <p>Determinar uma fração irredutível, equivalente a uma fração dada, a partir da simplificação por três.</p> <p>TRATAMENTO DE INFORMAÇÕES</p> <p>Interpretar dados apresentados em um gráfico de linha simples.</p> <p>Associar dados apresentados em gráfico de colunas a uma tabela.</p>
---	---

Fonte:

http://download.inep.gov.br/educacao_basica/prova_brasil_saeb/escala/escala_proficiencia/2018/MT_9EF.pdf

6.3 Planejamento e aplicação do projeto LPE

O projeto Linguagem de Programação Educativa (LPE) teve seus encontros realizados na sala de informática da escola Guaraci Mendes, o qual é proveniente do Programa Nacional de Informática na Educação (PROINFO) do Governo Federal e conta com sete estações de trabalho em pleno funcionamento.

Inicialmente elaboramos um projeto pedagógico de iniciação a Linguagem de Programação Educativa (Apêndice A) intitulado “LINGUAGEM DE PROGRAMAÇÃO EDUCATIVA COM O SCRATCH: formando o cidadão do futuro”, com o objetivo de “Utilizar o *Scratch* na introdução da linguagem de programação educativa para a construção de jogos interativos relacionados a Matemática”.

Com a etapa de construção do projeto finalizada, enviamos a Coordenação e a Gestão Escolar para apreciação. Após sua aprovação iniciamos a etapa de seleção dos alunos. Para esta etapa, não houve critérios específicos para a seleção, contudo priorizamos os alunos do 9º ano, informando sobre nossa proposta pedagógica e aguardamos a apresentação dos interessados, que totalizaram 09 (nove) ao término do período de inscrição no projeto.

Para iniciarmos os encontros do projeto passamos ao planejamento das atividades. Para cada um dos seis encontros construímos um planejamento que trazia em seu escopo cinco pontos principais: Objetivos, Conteúdos, Procedimentos Metodológicos, Recursos Didáticos e Avaliação, conforme Apêndice B.

No dia 25 de setembro de 2018 se iniciou a aplicação do minicurso, buscando seguir de forma fidedigna o que fora planejado no projeto LPE e nos planos de aula construídos para cada encontro. O minicurso teve sua estrutura organizacional composta de seis encontros com duas

aulas de 45 minutos na forma de um minicurso com carga horária total de 12 horas aulas. As aulas foram divididas em três momentos: *motivação*, *aprendizado*, *construção* e *reflexão*.

No momento *motivação*, importantíssimo para o desenvolvimento deste trabalho, foram apresentados projetos construídos com linguagem de programação *Scratch* que pudessem despertar a atenção dos alunos, envolvê-los e trazer benefícios significativos ao seu aprendizado, tanto da linguagem de programação quanto de conceitos matemáticos apresentados de forma direta ou indireta no processo de construção dessas aplicações.

Após cada momento *motivação* era proposto aos alunos que tentassem construir aplicações semelhante, cada um com as suas particularidades. Em seguida discutíamos cada uma das construções para verificarmos que elementos foram adicionados ou alterados e, principalmente, a funcionalidade de cada aplicação. Este momento agregava respectivamente a construção, aprendizado e a reflexão.

6.3.1 *Aplicações motivacionais propostas*

Durante as quatro das seis semanas do Projeto LEP foram apresentadas algumas aplicações construídas com a linguagem *Scratch* com o intuito de motivar os alunos e mostrar que a programação de computadores não é algo tão distante e inalcançável. Além disso, nas duas primeiras semanas planejamos uma atividade mais que motivacional, apresentando aos alunos o Spritebox, disponível online na plataforma do Spritebox⁸ e nas lojas de aplicativos para dispositivos que utilizam os sistemas Android e iOS. Este jogo desponta como mais uma das ferramentas voltadas a aprendizagem de programação e destinada principalmente aos nativos digitais, com uma interface amigável e a combinação da diversão de jogos de 'plataformas' (como Minecraft e Super Mario Bros) com a aprendizagem básica de programação.

Na terceira semana (Aula 03), como é comumente utilizado na introdução de qualquer linguagem de programação segundo Guanabara (2015), mostramos a Aplicação Motivacional 01 (APMOT-01) onde um personagem (ator) diz “Olá Mundo”. Pode parecer simples ou mesmo estranho introduzir um minicurso de programação educativa com essa aplicação, mas há quem diga, no mundo da programação de computadores, que se caso não iniciarmos com o “Olá Mundo” (Figura 6.5) corremos o risco de não aprendermos tal linguagem. Por isso, achamos melhor não arriscar.

⁸ <http://spritebox.com/>

Figura 6.5 – Aplicação Motivacional 01: Olá Mundo.

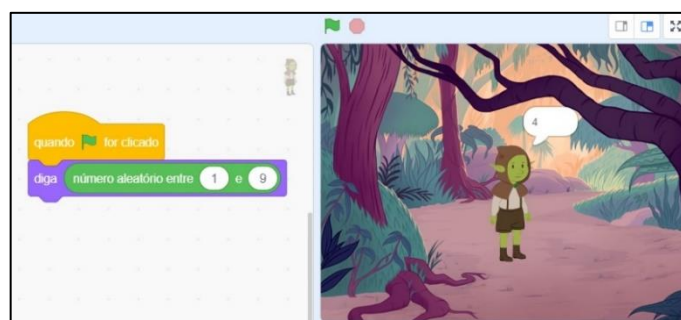


Fonte: Elaboração própria.

A aplicação “Olá Mundo!”, tida como lenda segundo Guanabara (2015) foi uma das maneiras divertidas de mostrar que a programação de computadores, hora vista como um privilégio de poucos adultos que possuíam um conhecimento especializado, agora está ao alcance de qualquer pessoa e pode ser praticada utilizando-se alguma das inúmeras ferramentas, ou Interfaces de Desenvolvimento Estruturado (IDE), disponíveis gratuitamente em plataformas online ou offline, como: *Scratch*⁹, *Code*¹⁰, *Python*¹¹, *App Inventor*¹².

Em sequência, ainda na terceira semana, apresentamos a Aplicação Motivacional 02 (APMOT-02) (Figura 6.6) onde um personagem mostra ao usuário um número aleatório entre 1 e 9. É importante mencionar que foi exibido na tela de projeção, na sala de informática, apenas a parte funcional das aplicações motivacionais, sem os códigos, para que posteriormente os alunos pudessem ter a oportunidade de criar suas próprias versões de cada uma dessas aplicações.

Figura 6.6 – Aplicação Motivacional 02: Número Aleatório.



Fonte: Elaboração própria.

⁹ <https://scratch.mit.edu/>

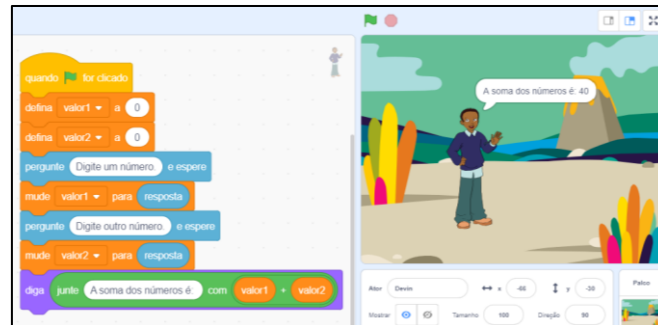
¹⁰ <https://code.org/>

¹¹ <https://www.python.org/>

¹² <http://ai2.appinventor.mit.edu/>

No encontro da quarta semana (Aula 04) mostramos aos alunos a Aplicação Motivacional 03 (APMOT-03). Nesta aplicação o usuário interage com o personagem que solicita dois números e em seguida mostra a soma destes (Figura 6.7).

Figura 6.7 - Aplicação Motivacional 03: Soma de dois números.

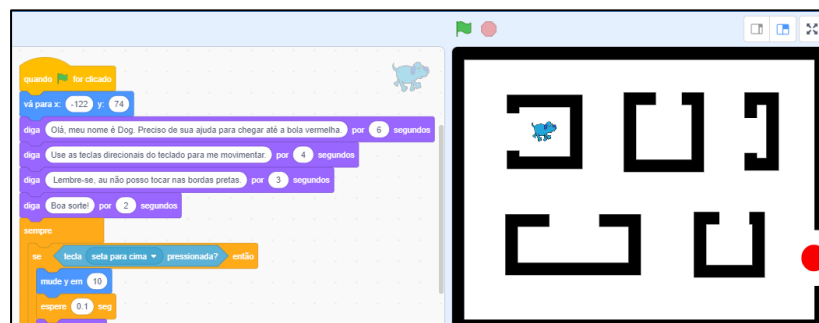


Fonte: Elaboração própria.

A APMOT-03 apresenta alguns comandos e ideias que exigem maior exploração da linguagem *Scratch*. O uso de variáveis, a ideia de armazenamento e descarga de informações e a utilização de sensores para a captação de interação com o usuário são fundamentais para o funcionamento efetivo desta aplicação.

Na quinta semana (Aula 05) apresentamos o jogo *Labirinto* (Figura 6.8) como Aplicação Motivacional 04 (APMOT-04), nossa última APMOT, visto que a última semana (Aula 06) fora destinada a avaliação final do projeto LPE. Assim, com uma estrutura de códigos mais elaborada com o uso de blocos de controle condicional, o jogo *Labirinto*, que objetiva levar o personagem (ator) até a bola vermelha, tem uma interface simples, porém embarca o pensamento condicional em três momentos fundamentais do jogo: quando o usuário utiliza os botões direcionais do teclado para movimentar o ator; quando o ator toca em alguma das bordas pretas; quando o ator toca na bola vermelha.

Figura 6.8 – Aplicação Motivacional 04 (Labirinto) com parte dos códigos.



Fonte: Elaboração própria.

6.4 Organização e aplicação do minicurso LEP

O minicurso seguiu a seguinte estrutura:

- Aula 1: Apresentação e utilização do SpriteBox como ferramenta de introdução a linguagem de programação;
- Aula 2: Aplicando ideias intuitivas de linguagem de programação no SpriteBox;
- Aula 3: Introdução a linguagem *Scratch*;
- Aula 4: Desenvolvendo aplicações no *Scratch*;
- Aula 5: Desenvolvendo aplicações matemáticas no *Scratch*;
- Aula 6: Avaliação: desenvolvimento de uma aplicação matemática (Plano Cartesiano);

Aula 1: Apresentação e utilização do SpriteBox como ferramenta de introdução a linguagem de programação.

Para a primeira aula utilizamos as estações de trabalho da sala de informática da escola Guaraci Mendes para acessar o SpriteBox via navegador, ressaltamos que para introdução e iniciação ao SpriteBox poderiam ser utilizados dispositivos móveis com sistema *Android* ou *iOS*.

No primeiro momento tivemos dificuldade com a atualização do *plugin* do *flash*, pois a versão para *web* do SpriteBox necessita da presença desse *plugin*. Quando acessamos o site oficial do jogo (<http://spritebox.com/hour.html>) visualizamos a tela de boas-vindas Figura 6.9.

Figura 6.9 – Tela de boas-vindas do SpriteBox.



Fonte: <http://spritebox.com/hour.html>

Na tela de boas vidas selecionando a opção *WEB (FLASH)* somos direcionados para a tela inicial do jogo Figura 6.10.

Figura 6.10 – Tela inicial do jogo Spritebox.



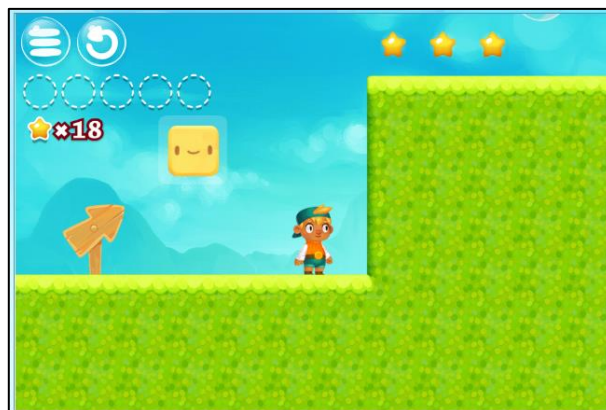
Fonte: <http://spritebox.com/flash.html>

Em alguns navegadores, como o Google Chrome, é necessário uma permissão do usuário para que esta tela de boas-vindas (Figura 6.10) seja carregada. Em seguida fizemos o reconhecimento dessa tela, especificando cada um de seus ícones e suas respectivas funções.

Primeiro mundo

Com as configurações iniciais já realizadas começamos o jogo. Logo surge o primeiro desafio, onde o avatar não consegue prosseguir devido seu salto ser insuficiente para transpor a parede.

Figura 6.11 – Impossibilidade de prosseguir.



Fonte: Do jogo SpriteBox

Nas proximidades dessas situações do jogo terá sempre uma **spritebox** (caixa amarela) piscando (Figura 6.11), que quando acionada libera o **spritebox ajudante**, mostra uma **região de execução** (região formada por pequenos quadrados delimitada por um retângulo pontilhado branco) e a **região de comandos**, destinada a organização e execução dos comandos (Figura 6.12), que nessa fase serão apenas tutoriais. Para continuar basta seguir as instruções passadas pelo SpriteBox ajudante e clicar no botão executar (em verde).

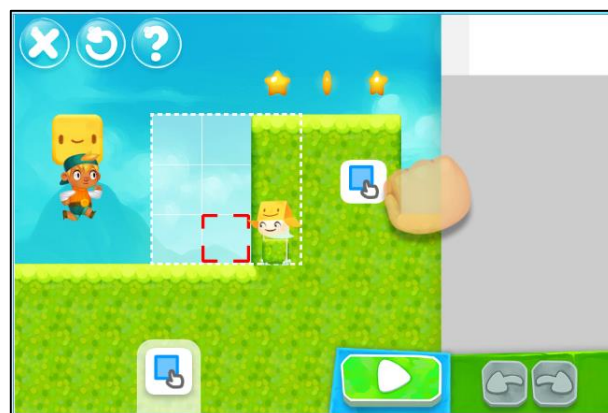
Figura 6.12 – Janela de comandos.



Fonte: Do jogo SpriteBox

No segundo desafio o jogo começa a ficar mais interativo, visto que é solicitado que o usuário arraste os blocos de instrução SET para a região de organização e execução dos comandos (Figura 6.13). Para continuar, basta seguir as instruções e clicar no botão executar.

Figura 6.13 – Arrastando blocos de comando.



Fonte: Do jogo SpriteBox

No terceiro desafio (Figura 6.14) é apresentado o comando MOVE, utilizado para mover o **spritebox** pela região de execução para a execução de um determinado comando. Para

prosseguir os alunos deveriam arrastar o bloco SET para a área de comandos e clicar no botão executar.

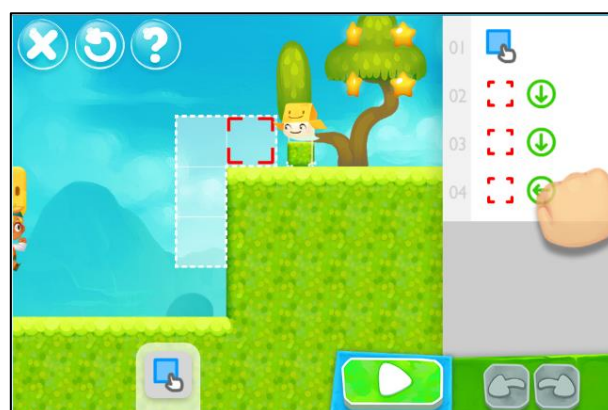
Figura 6.14 – Comando MOVE.



Fonte: Do jogo SpriteBox

No desafio quatro do jogo os alunos aprenderam que poderiam arrastar os blocos e reorganiza-los na região de comandos (Figura 6.15). Dessa forma, o jogo solicita que os blocos pré-fixados na área de comandos sejam reorganizados para que tenha sentido lógico. Neste momento os alunos foram questionados sobre qual configuração os blocos de comandos deveriam se encontrar para que o spritebox executasse a ação correta. Então, pedimos para que os alunos fizessem um rascunho em papel da organização dos comandos.

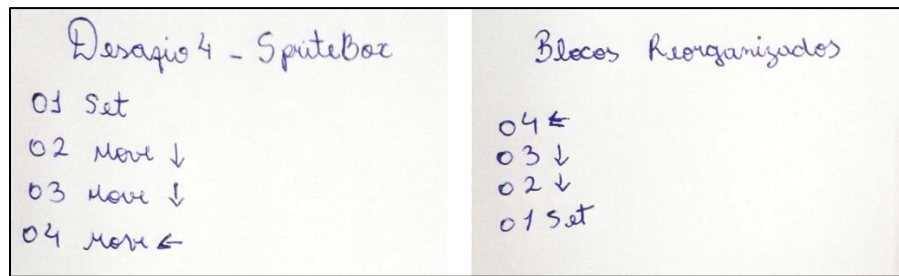
Figura 6.15 – Arrastando comandos.



Fonte: Do jogo SpriteBox

Nenhum dos alunos teve dificuldade em realizar essa tarefa e um desses resultados é apresentado na Figura 6.16.

Figura 6.16 – Reorganização da Aluna M.



Fonte: Da pesquisa.

Apesar de a Aluna M ter levado os números que marcam as linhas de código junto com os blocos de comandos, sua configuração está correta e quando executada levará o spritebox a construir um bloco “sólido” num local adequado e que possibilitará que o avatar consiga transpor esta etapa do jogo. Quando questionada sobre o motivo que a fez levar os números que marcam as linhas de códigos junto com os comandos, a Aluna M respondeu:

– *Na verdade eu ia colocar apenas os números, mas como achei que o senhor não ia entender, coloquei os comandos do lado.*

O quinto desafio retoma a habilidade anteriormente construída. O aluno, agora sem nenhuma pista por parte do jogo, tentará reorganizar os blocos de comando (Figura 6.17) para que o spritebox ajude-o a passar por mais essa etapa. Então, antes de colocar suas ideias de reorganização em prática, pedimos para que os alunos rascunhassem em papel o que fora pensado para a organização dos comandos.

Figura 6.17 – Último desafio do primeiro mundo.

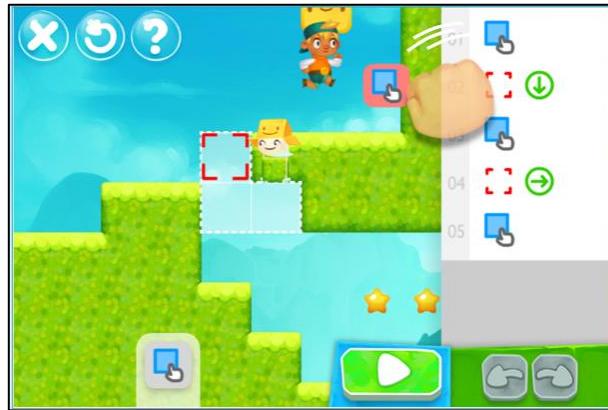


Fonte: Do jogo SpriteBox

No último desafio nesse mundo (fase do jogo) o SpriteBox nos ensina a deletar um comando caso este não seja necessário para a execução de uma determinada tarefa. Para isso,

basta arrastar o comando que desejamos deletar para fora da área de comandos, conforme Figura 6.18.

Figura 6.18 – Deletando comandos.



Fonte: Do jogo SpriteBox

Neste primeiro mundo os alunos não necessitaram de orientações acerca dos desafios, pois esta primeira fase do jogo paira apenas como um tutorial.

Segundo mundo

Aula 2: Aplicando ideias intuitivas de linguagem de programação no SpriteBox.

A segunda fase do SpriteBox é o Mundo Gelo, que logo no início já apresenta o primeiro desafio. Ainda em caráter introdutório, o SpriteBox orienta o aluno sobre a possibilidade de mudar a direção de um comando MOVE (Figura 6.19).

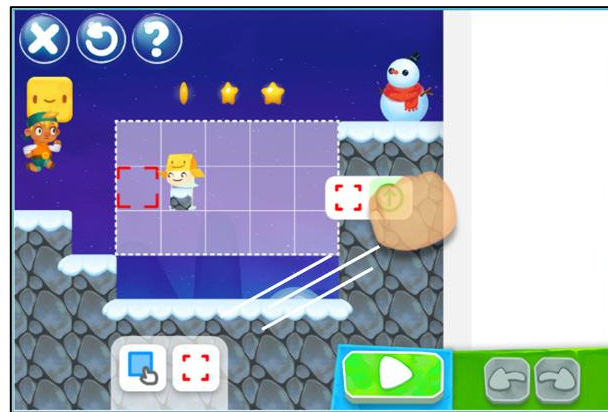
Figura 6.19 – Mudando a direção.



Fonte: Do jogo SpriteBox

No segundo desafio o aluno deverá construir uma ponte para que possa prosseguir. Inicialmente o aluno recebe como dica (Figura 6.20), arrastar o bloco MOVE para a área de comandos. Daí em diante é com o aluno, que deverá elaborar um algoritmo para a construção dessa ponte usando os comandos MOVE e SET.

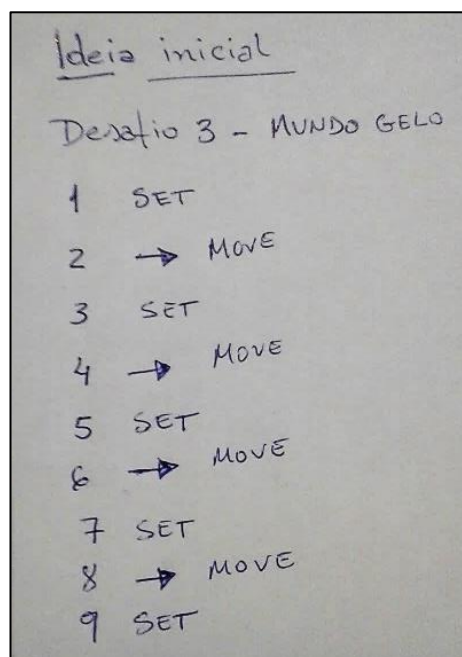
Figura 6.20 – Construindo uma ponte.



Fonte: Do jogo SpriteBox

Sendo assim, os alunos fizeram novamente em papel os rascunhos que descreviam suas ideias iniciais da organização dos comandos. Nenhum dos alunos teve dificuldade na realização dessa tarefa e um desses resultados obtidos é mostrado na Figura 6.21.

Figura 6.21 – Algoritmo do Aluno H.

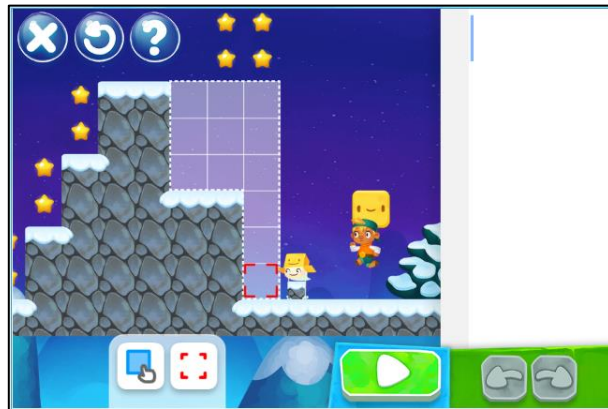


Fonte: Da pesquisa.

O Aluno H conseguiu construir um algoritmo organizado, com sintaxe correta que permitiu, após inserção dos blocos na área de comandos e execução, o prosseguimento do jogo.

Sem mais dicas, no terceiro desafio do segundo mundo (Figura 6.22) o aluno deverá construir dois pontos de apoio em forma de escada para que o avatar consiga transpor esta etapa. A combinação dos comandos MOVE e SET em disposição correta possibilitará que o avatar consiga alcançar o topo da parede e prosseguir no jogo.

Figura 6.22 – Construindo pontos de apoio.



Fonte: Do jogo SpriteBox

No desafio quatro temos um novo bloco de comando, o GET. Com a função de copiar aquilo que está selecionado pelo spritebox, o comando GET trabalha em conjunto com o comando SET, pois tudo que é copiado deverá de alguma forma ser construído posteriormente. Nesse desafio o aluno deverá utilizar o comando GET para copiar a escada e construí-la até que seja possível acessar a parte que se encontra bloqueada por um bloco de pedra (Figura 6.23).

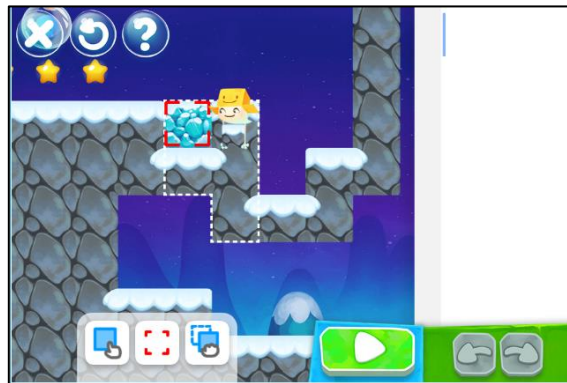
Figura 6.23 – Comando GET.



Fonte: Do jogo SpriteBox

Como último desafio do segundo mundo (Figura 6.24), o jogo retoma a ideia de copiar (GET) e construir (SET). O aluno deverá notar que a passagem se encontra totalmente bloqueada por blocos de pedra (não quebráveis) e que o spritebox está com sua seleção posicionada em um bloco de gelo (quebrável). A solução, nesse caso, será preencher toda a área de execução com blocos de gelo que em seguida poderão ser quebrados pelo avatar.

Figura 6.24 – Utilização do comando GET.



Fonte: Do jogo SpriteBox

E começa o terceiro mundo. Logo no início o aluno deparar-se-á com o primeiro desafio, no qual será introduzido o um novo comando: o laço de repetição. Esta introdução é feita como todas as outras, através de um pequeno diálogo com o spritebox (Figura 6.25).

Figura 6.25 – Introdução do laço de repetição.

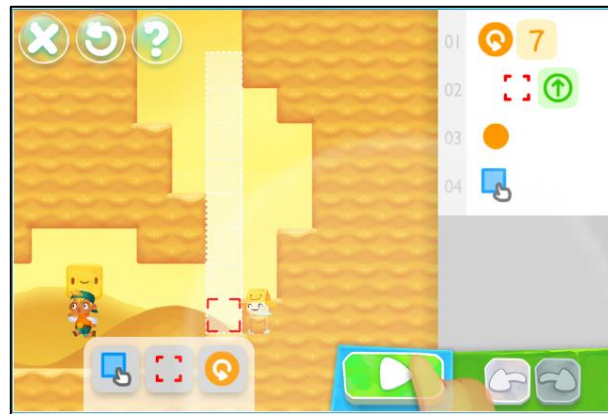


Fonte: Do jogo SpriteBox

Neste desafio o aluno deveria perceber que para alcançar o sétimo quadrado da região de execução ele poderia inserir o comando MOVE (para cima) 7 vezes, no entanto poderia

utilizar o novo comando apresentado para simplificar seu algoritmo. Com o laço de repetição o aluno precisaria utilizar o comando MOVE (para cima) uma única vez, conforme Figura 6.26.

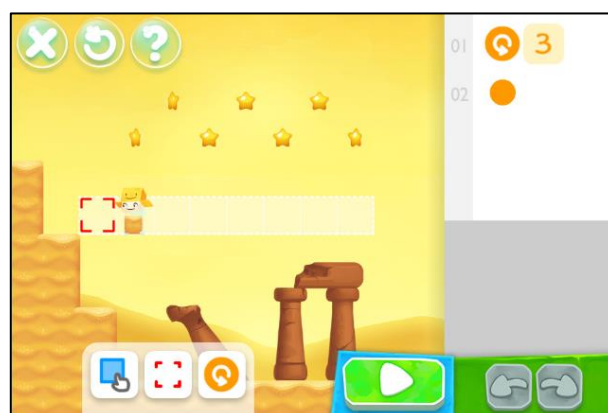
Figura 6.26 – Introdução ao laço de repetição.



Fonte: Do jogo SpriteBox

O segundo desafio (Figura 6.27) retoma a ideia de construção de uma ponte feita no segundo desafio do segundo mundo (Figura 6.20). Porém, aqui, o aluno pôde utilizar a ideia de repetição para evitar a utilização de um mesmo comando ou conjunto de comandos por definidas vezes. Ideias semelhantes seguiram os desafios seguintes solicitando ao aluno a construção de escadas e pontes. Como o intuito na utilização do SpriteBox era conhecer e aplicar ideias intuitivas relacionadas a iniciação a linguagem de programação, deixamos os dois mundos restantes a cargo dos alunos, visto que nenhuma nova função seria apresentada.

Figura 6.27 – Utilização do laço de repetição.

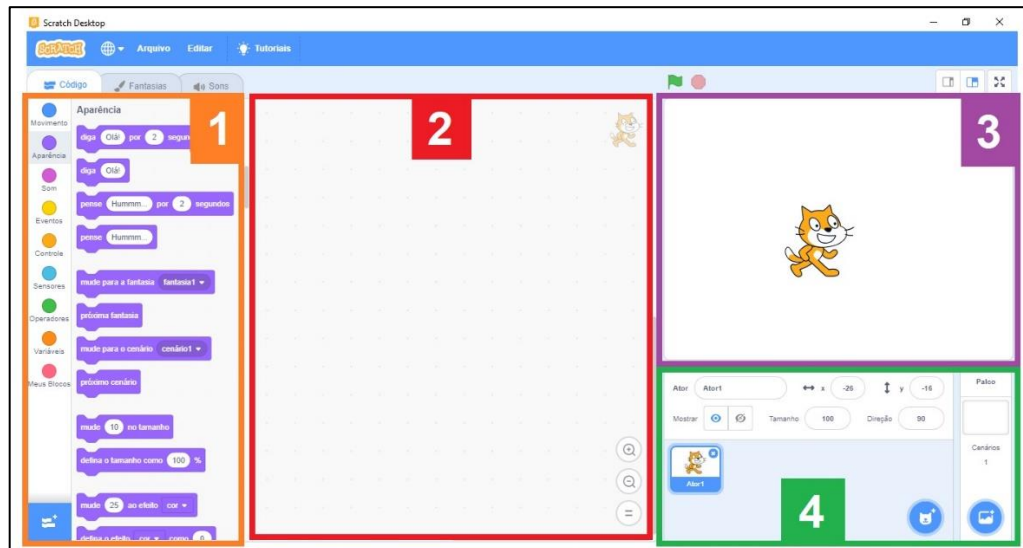


Fonte: Do jogo SpriteBox

Aula 3: Introdução a linguagem Scratch;

A tela inicial do estúdio de criação do *Scratch* é composta basicamente de 4 seções: Blocos de comando (1), Área de códigos (2), Palco (3) e Área dos atores e cenários (4), conforme Figura 6.28.

Figura 6.28 – Seções da tela inicial do estúdio de criação 3.0



Fonte: <https://scratch.mit.edu/projects/editor/?tutorial=getStarted>

6.4.1 Blocos De Comandos (1)

Nesta seção encontram-se os oito principais blocos lógicos: **Movimento**, **Aparência**, **Sons**, **Eventos**, **Controle**, **Sensores**, **Operadores**, **Variáveis** e **Mais Blocos**, além do botão *Adicionar Uma Extensão* que possui outra 9 opções, que embarcam recursos como: Música, Caneta, Detector de Vídeo, Texto Para Falar, Traduzir, Makey Makey, Micro:bit, LOGO Mindstorms EV3 e LEGO WeDo 2.0, conforme Figura 6.29.

Figura 6.29 – Alguns recursos do botão *Adicionar Uma Extensão*.



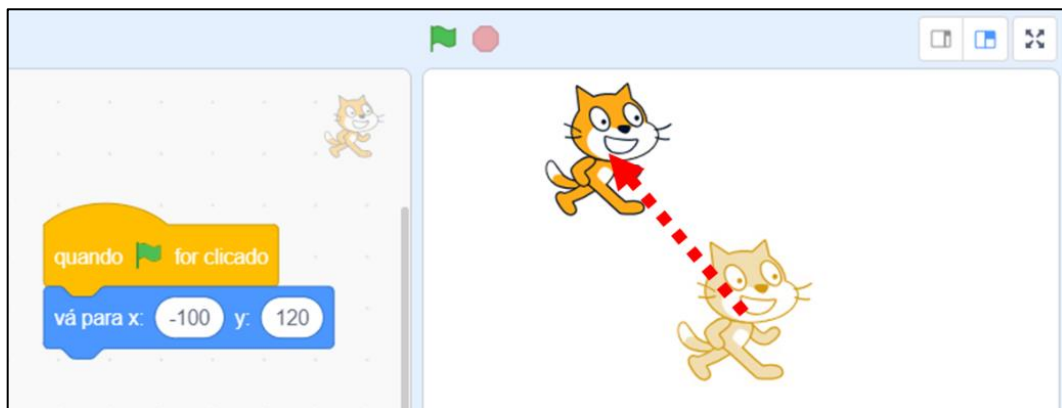
Fonte: <https://scratch.mit.edu/projects/editor/?tutorial=getStarted>

6.4.1.1 Movimento

Os comandos da aba Movimento dão ao usuário um elevado grau de liberdade para mover seus personagens por todo o Palco. A matemática por trás do Palco está alicerçada no sistema de coordenadas cartesianas desenvolvido no Séc. XVII por René Descarte, com a colaboração independente de Pierre de Fermat na construção da Geometria Analítica.

Essa estrutura permite que o usuário determine a posição exata de ator ou mesmo envie-o para uma determinada posição do Palco com a utilização de apenas um bloco de comando, como na Figura 6.30.

Figura 6.30 – Comando posição pré-definida.



Fonte: Elaboração própria.

Comandos de movimento e suas respectivas funções

BLOCO DE COMANDO	FUNÇÃO
	Faz o personagem se deslocar de sua posição para outra em 10 passos.
	Gira o personagem em torno de si mesmo 15° no sentido horário .
	Gira o personagem em torno de si mesmo 15° no sentido anti-horário .
	Assume a posição designada pelo ponteiro do mouse ou uma posição aleatória .
	Muda a posição do objeto para o par ordenado estabelecido.

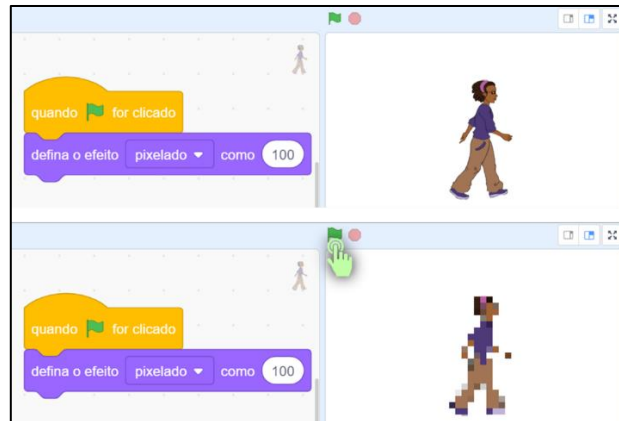
	O objeto deslizará da sua posição inicial para posição designada pelo ponteiro do mouse ou uma posição aleatória no tempo desejado.
	O objeto deslizará da sua posição inicial para o novo ponto no tempo desejado.
	O objeto será voltado para a direção que for colocada (para baixo, cima, esquerda ou direita).
	Muda a direção do objeto para posição designada pelo ponteiro do mouse .
	O valor de x (abscissa) mudará para o valor escolhido.
	O valor de y (ordenada) mudará para o valor escolhido.
	O valor de x mudará para o valor escolhido.
	O valor de y mudará para o valor escolhido.
	Ao tocar na borda o personagem retorna no sentido contrário.
	Define a forma como o personagem rotaciona (esquerda-direita, não rotaciona, rotação completa).
	Exibe o valor da coordenada x do personagem.
	Exibe o valor da coordenada y do personagem.
	Exibe o valor da direção do personagem.

6.4.1.2 Aparência

Os blocos de comando da aba Aparência proporcionam ao usuário recursos que tornam as aplicações em linguagem *Scratch*, dinâmica, interativas e envolventes, ingredientes necessários para prender a atenção dos alunos e interagir com os mesmos. Com apenas um bloco de comando é possível interagir com o usuário através de diálogos estáticos ou temporizados.


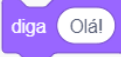
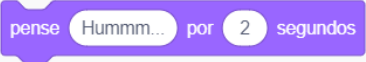
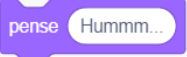
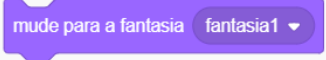

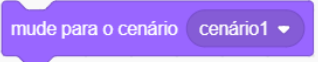
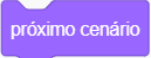
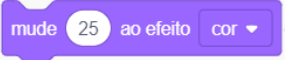
Pode-se também, com apenas um bloco de comando, alternar as fantasias de um ator, do cenário, aplicar efeitos de cores, mudar o tamanho, esconder ou mostrar um personagem, como na Figura 6.31.

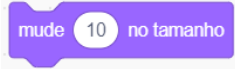
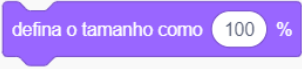
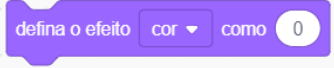


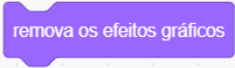
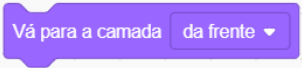
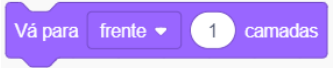
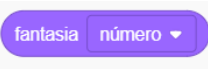
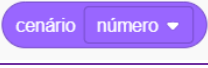
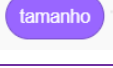
Figura 6.31 – Ator antes e depois de aplicar um efeito pixelado.



Fonte: Elaboração própria.

Comandos de aparência e suas respectivas funções

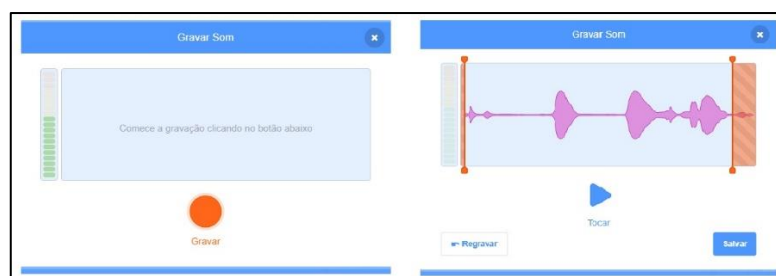
BLOCO DE COMANDO	FUNÇÃO
	Possibilita gerar um diálogo para o personagem por um tempo determinado.
	Possibilita gerar um diálogo para o personagem.
	Possibilita gerar um pensamento para o personagem por um tempo determinado.
	Possibilita gerar um pensamento para o personagem.
	Muda o personagem para uma determinada aparência.
	Muda para a próxima fantasia da lista.
	Possibilita mudar para um determinado cenário.
	Muda para o próximo cenário da lista.
	Muda o efeito gráfico de um personagem estabelecendo a textura.

	Possibilita mudar o tamanho de um personagem a uma quantidade de pontos determinada.
	Possibilita mudar o tamanho de um personagem em termos percentuais em relação ao original.
	Define o efeito gráfico de um personagem estabelecendo a textura.
	Mostra um objeto ou personagem que está escondido.
	Esconde um objeto ou personagem que está visível.
	Possibilita a remoção de efeitos adicionados anteriormente.
	Possibilita mandar um objeto ou personagem para a camada da frente ou para a camada de trás.
	Possibilita avança ou recuar um objeto ou personagem um determinado número de camada.
	Indica o número do traje de um personagem.
	Indica o número de um cenário.
	Indica o tamanho percentual (%) de um personagem.

6.4.1.3 Sons

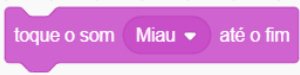
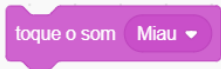
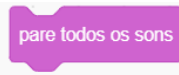
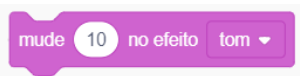
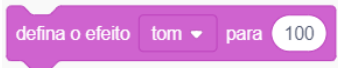
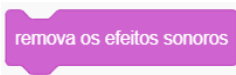
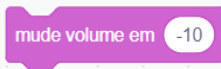
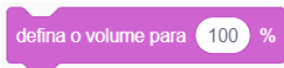
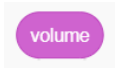
Na construção de conteúdos interativos, games, histórias e animações é possível adicionar músicas ou sons aos personagens, podendo o usuário optar por uma música ou som presente no banco de dados do *Scratch*, no computador ou na internet, além da possibilidade de o usuário gravação e editar seu próprio som através da aba **Sons** (Figura 6.32).

Figura 6.32 – Opção de gravação de áudio no *Scratch*.



Fonte: <https://scratch.mit.edu/projects/editor/?tutorial=getStarted>

Comandos de sons e suas respectivas funções

BLOCO DE COMANDO	FUNÇÃO
	Toca o som selecionado até que termine e então vai para o próximo bloco.
	Toca o som selecionado no menu e depois vai para o próximo bloco (mesmo que o som ainda esteja tocando).
	Para a execução de todos os sons.
	Permite aumentar ou diminuir progressivamente as Batidas por Minuto (BPM) de um determinado som a cada execução.
	Permite aumentar ou diminuir as Batidas por Minuto (BPM) de um determinado som.
	Remove todos os efeitos adicionados a um som.
	Permite aumentar ou diminuir o volume de um determinado som.
	Possibilita controlar o volume numa escala de 0 a 100%.
	Mostra o volume utilizado no momento.




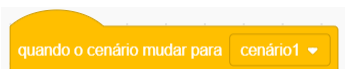

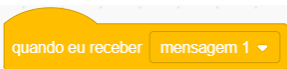


6.4.1.4 Eventos

Para que você possa estar visualizando este texto (caso esteja utilizando um dispositivo tecnológico) foi necessário que você clicasse ou tocasse sobre o ícone que o representa em seu dispositivo, certo?

Então, quando você faz essa ação (clicar ou tocar no ícone) seu dispositivo recebe essa mensagem e aciona um aplicativo capaz de fazer a leitura desse arquivo. Analogamente funcionam os blocos de comando da aba Eventos. A funcionalidade desses blocos depende de uma ação externa.

Fazendo uma alusão a Física poderíamos pensar em algo como a Primeira Lei de Newton: *um corpo em repouso permanecerá em repouso, a menos que uma força resultante recaia sobre ele*. Ou seja, um bloco da aba **Eventos** não fará sentido sem que haja uma ação externa.

Blocos de eventos e suas respectivas funções

BLOCO DE COMANDO	FUNÇÃO
	Iniciará um conjunto de comandos (script) que a ele esteja conectado sempre a bandeira verde for clicada.
	Iniciará um conjunto de comandos que a ele esteja conectado sempre uma tecla específica for clicada.
	Iniciará um conjunto de comandos que a ele esteja conectado sempre um ator a ele vinculado for clicada .
	Iniciará um conjunto de comandos que a ele esteja conectado sempre que tivermos um cenário específico .
	Iniciará um conjunto de comandos que a ele esteja conectado sempre que detectar um ruído superior ao especificado .
	Iniciará um conjunto de comandos que a ele esteja conectado sempre que receber uma mensagem específica .
	Envia uma mensagem a todos do projeto.
	Envia uma mensagem a todos do projeto e espera uma resposta.

6.4.1.5 Controle

Se eu tivesse dinheiro, viajaria pelo mundo. Se eu fosse alto, jogaria basquete. Se eu soubesse programar, criaria meus próprios conteúdos digitais. Note-se que para cada condição estabelecida há uma consequência, ações que podem ou não ocorrer. Essa ideia é fundamental em programação, pois evita que um determinado script (algoritmo) fique verificando uma condição pré-estabelecida que nunca acontece ou que não possua uma ação consequente.

Por isso, para evitar situações como essas existem os blocos lógicos de comandos da aba **Controle**. Dentre esses blocos os mais utilizados para verificar condições são o “se...então” que verifica certa condição e executa uma ação definida e o “sempre” que permite a verificação de determinada condição até que esta aconteça.





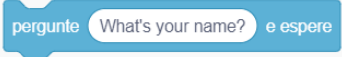




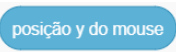



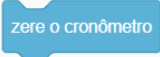
Comandos de controle e suas respectivas funções

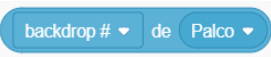
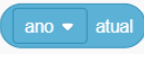
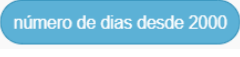

BLOCO DE COMANDO	FUNÇÃO
	Pausa o conjunto seguinte de comandos por determinados segundos .
	Repete tudo que por ele estiver envolvido quantas vezes necessário.
	Repetirá infinitamente um conjunto de comandos que por ele esteja envolvido.
	Se uma determinada condição for satisfeita , executar-se-á o que por ele estiver envolvido.
	Se uma determinada condição for satisfeita , executar-se-á o que primeiro estiver envolvido, caso contrário será executado o que estiver envolvido por segundo.
	Estabelece uma espera até que uma determinada condição seja satisfeita.
	Repete tudo que por ele estiver envolvido até que uma determinada condição seja satisfeita.
	Para todos os códigos, um conjunto específico de códigos ou apenas o conjunto de códigos a que pertence.
	Avisa um clone sobre os comandos a serem seguidos após sua criação.
	Cria um clone de um ator específico.
	Apaga um clone específico.

6.4.1.6 SENSORES

Quando utilizamos um bloco de Controle para estabelecer uma condição e uma ação consequente, necessitamos de **Sensores**. Geralmente no formato hexagonal, os Sensores são responsáveis por estabelecer uma determinada condição a um bloco de Controle, detectar a interação com o usuário, entre seus personagens ou com o cenário.

Blocos de sensores e suas respectivas funções

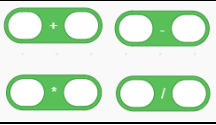
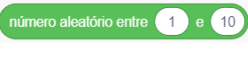






BLOCO DE COMANDO	FUNÇÃO
	Verifica se o personagem está tocando no ponteiro do mouse (ou na borda) (verdadeiro) ou não (falso).
	Verifica se o personagem está tocando em algum lugar com a cor estabelecida.
	Verifica se algum objeto com a cor estabelecida primeiro está tocando em algum lugar com a cor estabelecida por segundo.
	Retorna a distância do personagem ao ponteiro do mouse.
	Faz uma pergunta e espera o retorno e confirmação usuário.
	Retorna a resposta dada a uma determinada pergunta.
	Verifica se alguma tecla foi pressionada.
	Verifica se o mouse foi pressionado.
	Retorna o valor da coordenada x do ponteiro do mouse.
	Retorna o valor da coordenada y do ponteiro do mouse.
	Permite habilitar ou desabilitar o modo de arrasto.
	Retorna o valor do ruído detectado pelo microfone.
	Retorna o valor registrado pelo cronômetro .
	Zera o cronômetro.






	Retorna valores característicos do Palco , como: número do cenário, nome do cenário, volume e valor de uma variável.
	Retorna o segundo, minuto, hora, dia da semana, data mês ou ano.
	Retorna o número de dias desde o ano 2000.
	Retorna o nome de usuário.

6.4.1.7 Operadores

Para introdução desta subseção pode-se usar um tema muito debatido entre pesquisadores e educadores: o uso da calculadora em sala de aula. Dessa temática lançamos a seguinte questão: o que poder-se-ia fazer um aluno com uma calculadora caso não conhecesse as operações fundamentais da matemática? E a resposta desse pesquisador é: nada. Analogamente, sem os operadores lógicos apresentados nessa subseção não seria possível a construção de praticamente nenhuma das aplicações em *Scratch* ligadas a Matemática.



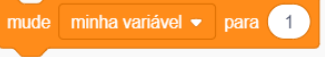



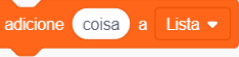

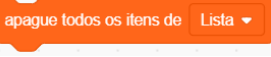


Blocos operadores e suas respectivas funções







BLOCO DE COMANDO	FUNÇÃO
	Operadores utilizados para somar , subtrair , multiplicar ou dividir os números que estão dentro dos círculos brancos.
	Retorna um número aleatório dentro dos limites estabelecidos.
	Verifica se um número é maior , menor ou igual a outro.
	Verifica se as duas sentenças são ambas verdadeiras .
	Verifica se pelo menos uma das sentenças é verdadeira .
	Verifica se a negação de uma sentença é verdadeira ou falsa.
	Junta uma frase (ou resposta) com outra.
	Retorna a letra que ocupa determinada posição de uma dada palavra.

	Retorna a quantidade de letras de uma determinada palavra.
	Verifica se uma dada palavra possui uma letra desejada.
	Retorna o resto da divisão entre dois números dados.
	Retorna o arredondamento de um número dado.
	Retorna cálculos além das operações básicas fundamentais. Exemplo: Calcula a raiz quadrada de um número, potências de base 10, valores trigonométricos, etc.

6.4.1.8 VARIÁVEIS

Blocos de variáveis e suas respectivas funções

BLOCO DE COMANDO	FUNÇÃO
	Retorna o valor de uma variável.
	Define um valor para uma variável.
	Muda o valor de uma variável para um valor desejado.
	Mostra uma determinada variável.
	Esconde uma determinada variável.
	Retorna uma determina lista.
	Adiciona um determinado valor ou palavra a uma lista.
	Apaga um item desejado de uma lista.
	Apaga todos os itens de uma lista.
	Inseri um valor determinado na posição desejada.
	Substitui um determinado item da lista por um determinado valor.

	Retorna um determinado item da lista.
	Retorna a posição de um item da lista.
	Retorna o tamanho da lista.
	Verifica se a lista contém um valor determinado.
	Mostra uma lista.
	Esconde uma lista.

6.4.1.9 MEUS BLOCOS

Quando o pedreiro constrói uma casa, por muitas vezes ele utiliza a mesma mistura de cimento e areia, ou seja, por várias vezes ele produz a mesma liga de concreto. Por essa razão, o pedreiro sabe quais as quantidades exatas de areia e cimento são necessárias para aquele serviço.

Com os blocos de comando não é diferente. Se na construção de algum projeto em *Scratch* você necessite utilizar a mesma estrutura de códigos várias vezes, então tem-se a possibilidade da criação de um novo bloco de códigos que contenha aquele script por vezes utilizado. Para construirmos um novo bloco lógico basta acessarmos a aba **Meus Blocos**.

Para construir uma animação que dê a sensação de movimento natural a um ator é necessário que este tenha mais de uma fantasia relacionada a movimento e a utilização de três blocos de comandos, conforme Figura 6.33.

Figura 6.33 – Sensação natural de movimento.



Fonte: Elaboração própria.

Caso seja necessário a utilização do código de movimento natural (Figura 6.33) por várias vezes podemos criar um novo e único bloco de comando para representar esta ação de movimento. Inicialmente criamos uma definição para o script, conforme Figura 6.34.

Figura 6.34 – Definição do novo bloco lógico: movimento natural.

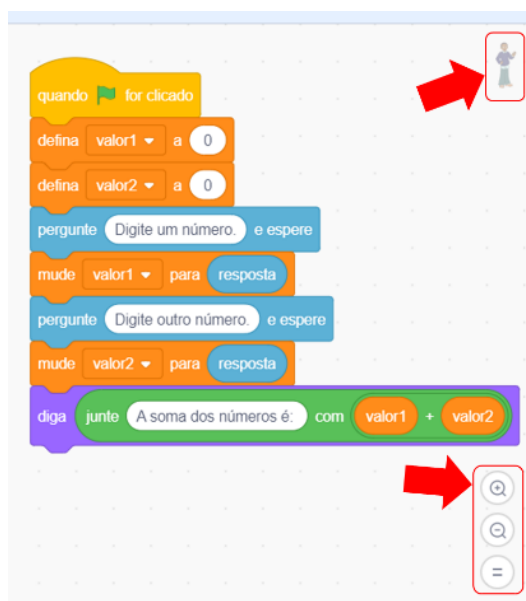


Fonte: Elaboração própria.





6.4.2 Área De Códigos (2)

Na área de códigos (Figura 35) existem três botões importantes para o processo de programação: o zoom +, zoom -, redimensionamento padrão e o avatar dos atores.

Figura 6.35 – Área de códigos



Fonte: Elaboração própria.

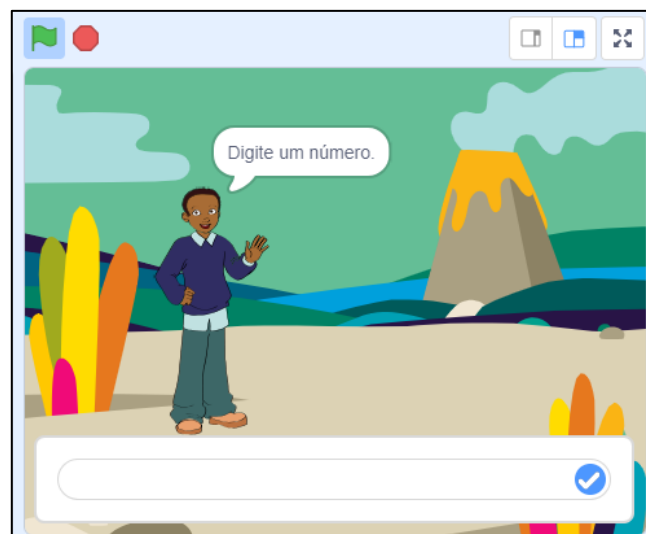
	Aumenta o tamanho de visualização dos códigos.
	Diminui o tamanho de visualização dos códigos.
	Redimensiona padrão do tamanho de visualização dos códigos.
	Retrata as configurações atuais do personagem.

Vale ressaltar que para que os códigos de um determinado conjunto funcionem em sintonia é necessário que estejam de alguma forma conectados, isto é, o fato de os blocos estarem na área de códigos não garante seu pleno funcionamento.

6.4.3 Palco (3)






O Palco é uma das seções mais importante do *Scratch*, pois toda entrada e saída de informações, a interação usuário-computador-usuário ocorre sob seu auspício. Na Figura 6.36 temos um exemplo de interação computador-usuário-computador, que irá se consolidar a partir da prestação de informação e posteriormente da confirmação dos dados de entrada.

Figura 6.36 – Interação computador-usuário.



Fonte: Elaboração própria.

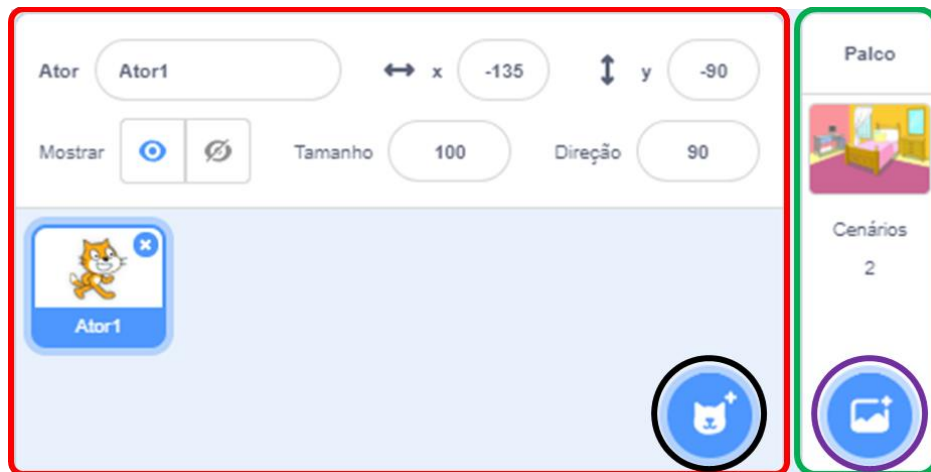
O usuário tem autonomia para mudar o plano de fundo (cenário) do Palco sempre que julgar conveniente. Para tanto, basta seguir as orientações descritas na seção **ÁREA DOS ATORES E CENÁRIOS**. Além disso, o usuário pode alternar entre três possibilidades de exibição do palco (layout), disponíveis na barra superior do palco, que contém outros dois botões, conforme a seguir.

				
Permite iniciar uma aplicação.	Permite parar uma aplicação.	Exibe o Palco em 4% da tela.	Exibe o Palco em 17% da tela.	Exibe o Palco e tela cheia.

6.4.4 Área Dos Atores E Cenários (4)





A área dos atores e cenários está dividida em duas partes, conforme Figura 6.37.

Figura 6.37 – Cenários

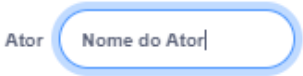



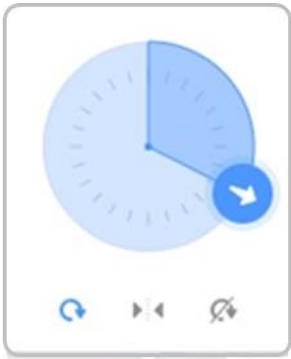






Fonte: Elaboração própria.

A região destacada em verde (Figura 6.37) destina-se a listagem de todos os cenários utilizado na elaboração de aplicações no *Scratch*. Na parte inferior desta região temos o botão **Selecionar Cenário** (destacado em lilás) que nos mostra uma lista suspensa, dando-nos quatro opções de escolha quando queremos adicionar um novo cenário ou alterar o cenário atual, bastando para isso passar o cursor do mouse sobre ele. Os botões disponíveis nessa lista suspensa estão abaixo com suas respectivas funcionalidades.

	Carregar Cenário: possibilitar carregar um cenário diretamente do armazenamento computador.
	Surpresa: permite a seleção aleatória de um cenário.
	Pintar: permite que o usuário crie seu próprio cenário.
	Selecionar Cenário: permite que o usuário selecione uma das centenas de cenário disponíveis na biblioteca do <i>Scratch</i> .

A região destacada em vermelho (Figura 6.37) é o local onde todos os atores utilizados na elaboração de aplicações no *Scratch* serão listados. Nesta região temos ainda algumas características do ator selecionado (sempre que um ator ou palco estiver selecionado seu avatar ou miniatura estará envolta por uma borda **azul**), conforme mostrado a segui.

	Nome do Ator: permite nomear um ator como desejar. Muito útil em projetos com muitos atores.
	Posição do Ator: permite posicionar um Ator em um ponto específico sem a necessidade da utilização de um código.
	Mostra/Esconder: permite que um Ator seja escondido ou mostrado sem a necessidade da utilização de um código.
	Tamanho: permite aumentar ou diminuir um Ator em termos percentuais, onde 100 representa o tamanho original.
	Direção: permite mudar a direção de um Ator em todo de seu ponto de ancoragem (ponto central). Em caso de movimento, o Ator seguirá essa nova direção.
	Rotação em torno do ponto central do Ator.
	Rotação reflexiva no sentido vertical.
	Bloqueia a rotação do Ator (direção padrão).
	Possibilita rotacionar o Ator livremente com o uso do mouse. Basta clicar e buscar o ângulo desejado.

Na parte inferior direita desta região destacada em vermelho (Figura 6.37) temos o botão **Selecionar Ator** (destacado em preto) que, também, nos mostra uma lista suspensa, dando-nos as mesmas quatro opções de escolha presentes no botão Selecionar Cenário.

Quando queremos adicionar um novo ator ou alterar o ator atual, basta passar o cursor do mouse sobre o botão Selecionar Ator e seguir as mesmas orientações de utilização do botão Selecionar Cenário, já que as quatro opções mostradas na lista suspensa têm os mesmos ícones e as mesmas funcionalidades, todavia relacionadas aos atores.

Aula 4: Desenvolvendo aplicações no Scratch;

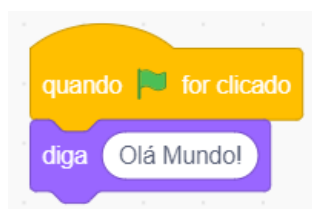
6.4.5 Atividades Iniciais

As atividades iniciais de programação em *Scratch* têm o objetivo de levar ao usuário, que a partir de então passaremos a chamar de Programador em *Scratch* ou simplesmente Programador, uma ideia intuitiva do processo de programação com a utilização de blocos lógicos de comando.

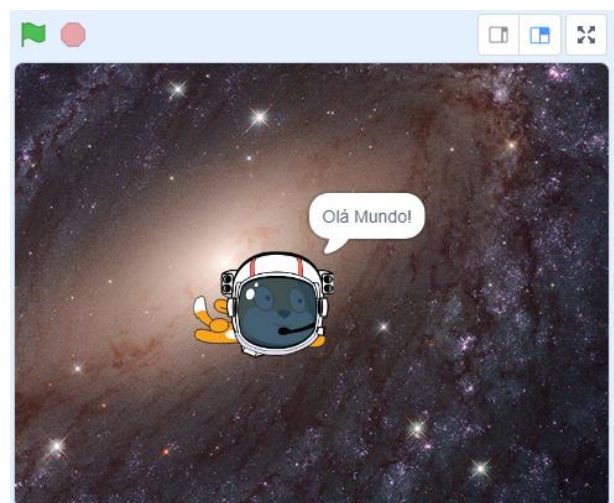
ATIVIDADE 01 (Olá Mundo!)

Construa uma aplicação em que um Ator diga “Olá Mundo!”

Blocos de comando necessários

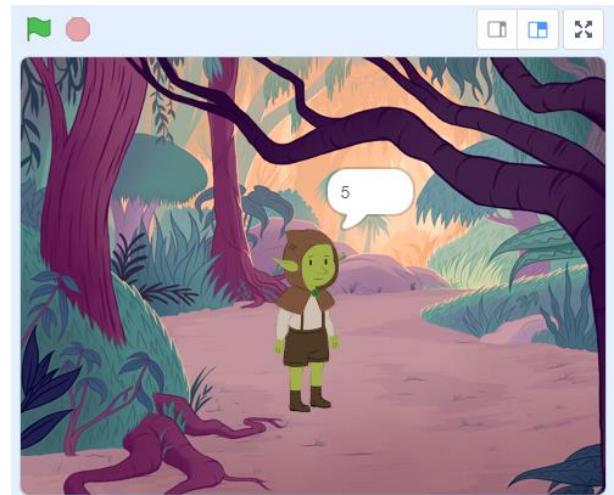


Resultado mostrado no palco

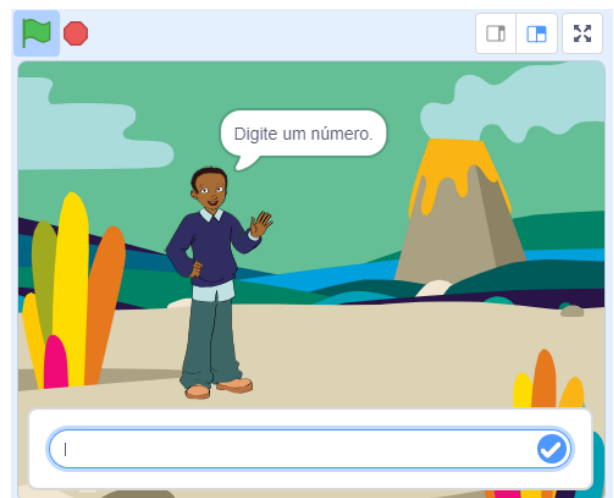


ATIVIDADE 02 (Sorteador)

Construa uma aplicação em que um Ator diga um número aleatório entre os valores 1 e 9.

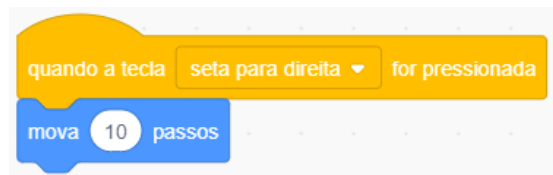
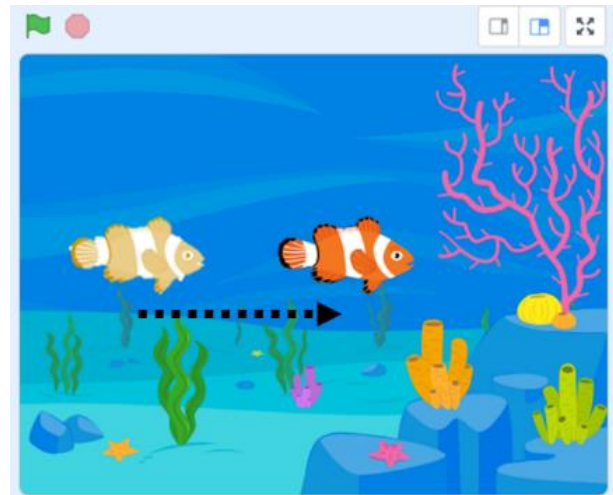
Blocos de comando necessários**Resultado mostrado no palco****ATIVIDADE 03** (Soma de dois números)

Construa uma aplicação em que um Ator solicite dois valores ao usuário e diga a soma desses valores por meio de um diálogo.

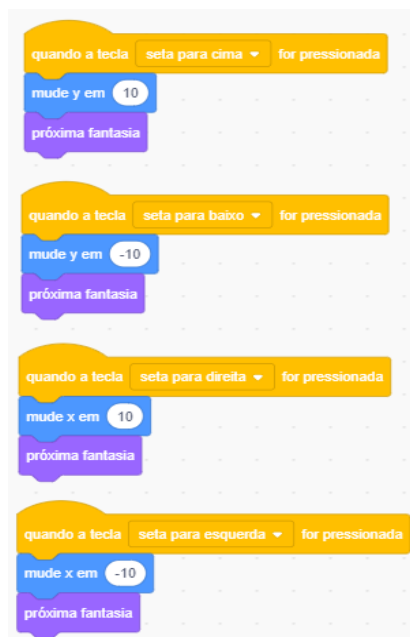
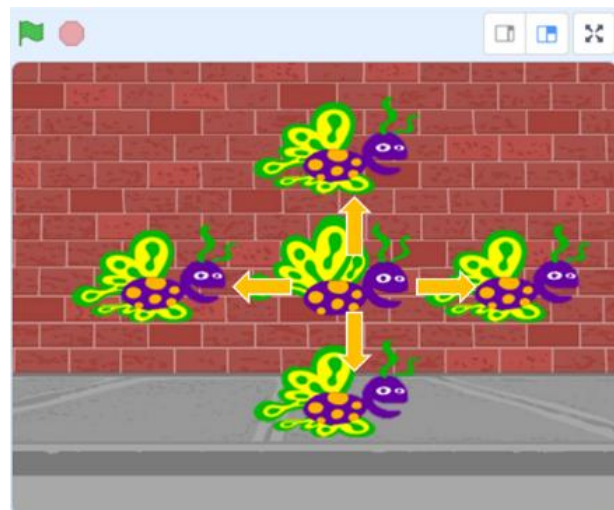
Blocos de comando necessários**Resultado mostrado no palco**

ATIVIDADE 04 (Controle direcional 01)

Construa uma aplicação em que o programador possa utilizar a tecla direcional do teclado (direita) para fazer um Ator andar para frente sempre que esta tecla (direita) for clicada.

Blocos de comando necessários**Resultado mostrado no palco****ATIVIDADE 05** (Controle direcional 02)

Construa uma aplicação em que o programador possa utilizar as teclas direcionais do teclado para controlar um Ator, podendo mudar sua posição para cima, para baixo, para direita ou para esquerda. Além disso, mude a fantasia do Ator sempre que ele mudar de posição.

Blocos de comando necessários**Resultado mostrado no palco**

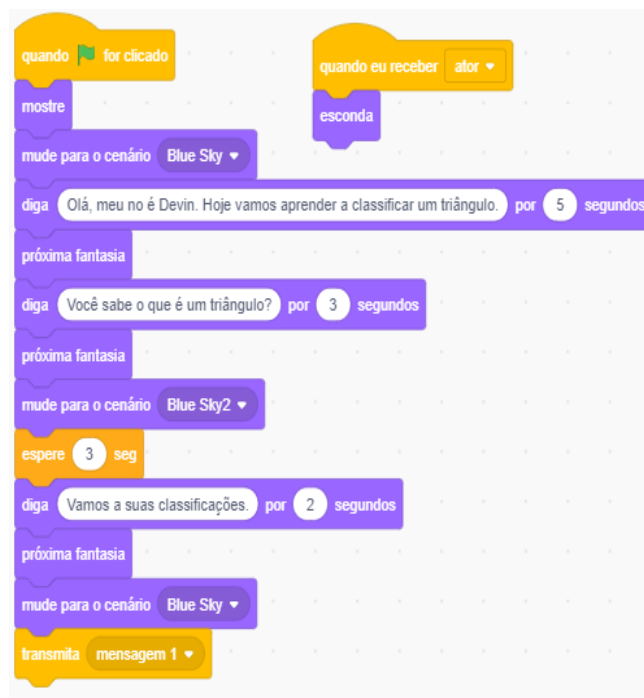
Aula 5: Desenvolvendo aplicações matemáticas no Scratch;

ATIVIDADE 06 (Tipos de triângulos)

Construa uma aplicação em que o programador possa utilizar o clique do mouse para interagir e navegar entre as opções de classificação dos triângulos enquanto a característica de seus lados.

Blocos de comando necessários

Resultado mostrado no palco



Aula 6: Avaliação: desenvolvimento de uma aplicação matemático (Plano Cartesiano);

ATIVIDADE 07 (Marcando pontos)

Construa uma aplicação em que o programador possa utilizar as teclas direcionais do teclado para controlar um Ator por um sistema de coordenadas de eixos cartesianos, podendo mudar sua posição para cima, para baixo, para direita ou para esquerda.

Resultado mostrado no palco via web



6.5 Registros das atividades

Neste tópico são apresentadas algumas das imagens registradas durante a aplicação do minicurso, junto da qual constará um breve relato sobre o momento, além de alguns dos relatos dos alunos sobre sua avaliação e satisfação pela participação no projeto, bem como suas visões acerca da aprendizagem e conhecimentos matemáticos desenvolvidos ao longo das aulas.

A Figura 6.38 mostra o aluno Z iniciando o segundo mundo do SpriteBox. Nesta fase o jogo começa a exigir mais dos alunos, visto que as apresentações tutoriais que auxiliam o usuário na realização das tarefas passa por um processo de transição tornando o jogador mais independentes nas tomadas de decisão.

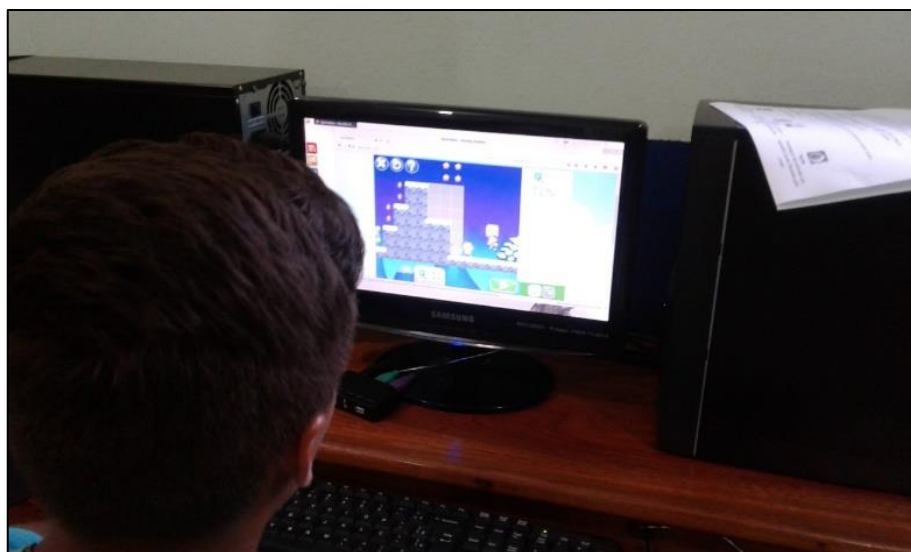
Figura 6.38 – Aluno Z jogando SpriteBox.



Fonte: Da pesquisa.

Durante o desenvolvimento do segundo mundo começam a surgir desafios menos instrutivos, nos quais os alunos precisam organizar suas ideias e projetar situações que culminem em soluções apropriadas para a evolução de seus avatares no jogo. Na Figura 6.39 o aluno K tenta elaborar um algoritmo necessário a construção de dois pontos de apoio para a progressão de seu avatar no jogo, cumprindo o terceiro desafio deste mundo.

Figura 6.39 – Aluno K no terceiro desafio do segundo mundo do SpriteBox.



Fonte: Da pesquisa.

Na Figura 6.40, em atividades de desenvolvimento das ideias de lógica de programação com a utilização do Spritebox, este pesquisador passa orientações duas alunas sobre o motivo pelo qual seus algoritmos referentes ao desafio quatro do segundo mundo do jogo não foi totalmente eficiente para que seus avatares prosseguissem.

Nesse desafio, que está ilustrado na Figura 6.23, é introduzido um novo comando, o GET. Para que o ajudante spritebox possa copiar determinado elemento (bloco de pedra sólido, quebrável ou escada) é necessário a utilização do comando GET. Porém, as alunas utilizaram apenas o comando SET (construir) em seu algoritmo fazendo com que o ajudante construísse blocos de pedra sólidos no lugar da escada. A sequência lógica para resolução do problema deveria trazer em sua parte inicial o comando GET para copiar a escada, a qual seria construída posteriormente com a utilização do comando SET.

Figura 6.40 – Pesquisador passando instruções as alunas sobre o SpriteBox.



Fonte: Da pesquisa.

Um dos fatores que podem contribuir para o melhor desenvolvimento e interação nas aulas nas salas de informática é a sua disposição estrutural. A Figura 6.41 traz o *layout* (dentro dos padrões estabelecidos pelo PROINFO) da sala de informática da escola Guaraci Mendes utilizada para os encontros do minicurso. Esta sala possui sete estações em pleno funcionamento e dois professores (um deles licenciado em Informática) que se revezam nos turnos da manhã e da tarde proporcionando atendimento aos alunos e auxílio aos professores que utilizam a sala. A sala é refrigerada e avaliada pelos alunos como confortável, mesmo sendo suas cadeiras de madeira e não estofadas.

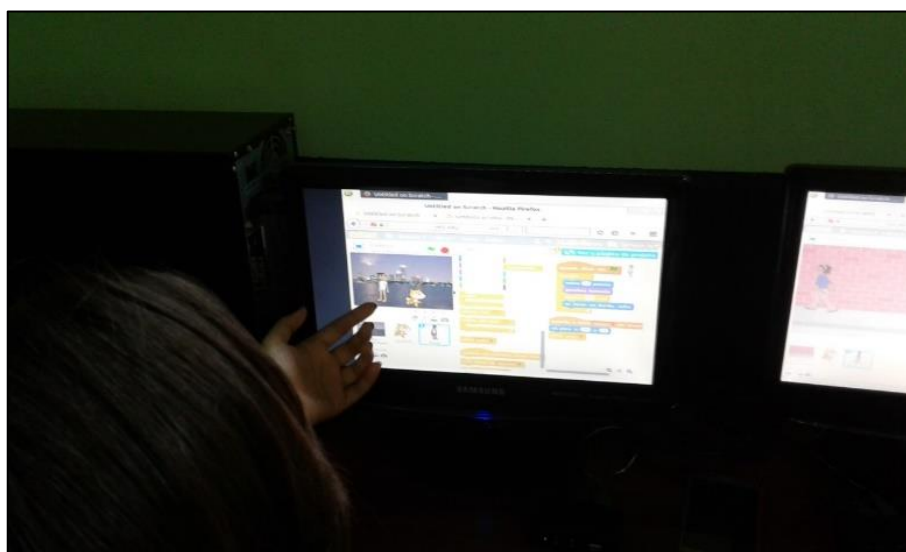
Figura 6.41 – Disposição da Sala de Informática.



Fonte: Da pesquisa.

A Figura 6.42 retrata um dos momentos em que os alunos estavam aprendendo os primeiros conceitos e funções dos blocos lógicos do *Scratch*. Nesse momento os alunos testavam a utilização de dois blocos lógicos: o bloco “quando a tecla (alguma) for pressionada”, um evento responsável pela ativação ou desativação de determinada ação e o bloco “vá para (x,y)” que leva o personagem a um determinado ponto do plano cartesiano que se estende por todo o palco.

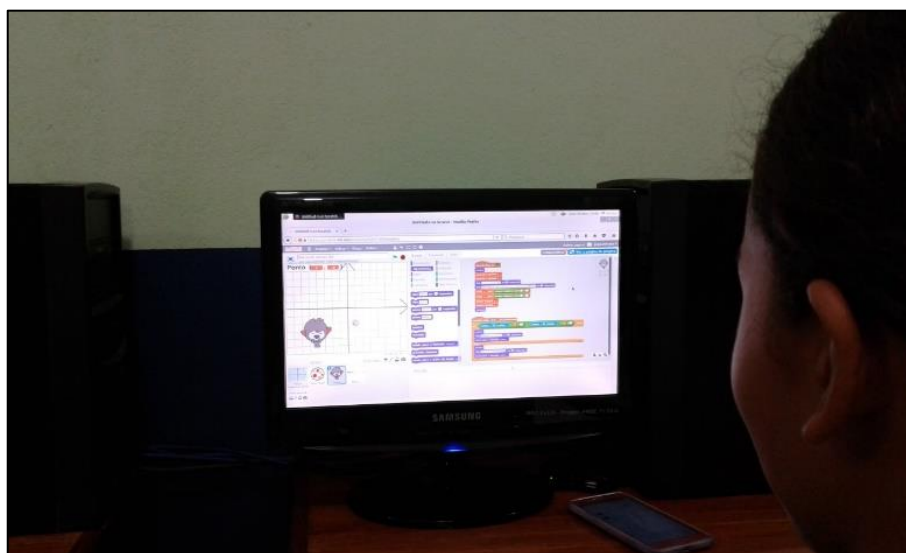
Figura 6.42 – Aluna N programando no *Scratch*.



Fonte: Da pesquisa.

A última atividade do minicurso está registrada na Figura 6.43, onde a aluna M aparece na parte de depuração dos códigos dos personagens referente a aplicação Marcando Pontos, produto deste trabalho disponível na plataforma do *Scratch*.

Figura 6.43 – Aluna M criando a aplicação Marcando Pontos.



Fonte: Da pesquisa.

6.6 Alguns relatos das atividades

Além das fotografias tiradas durante as atividades, temos alguns depoimentos que podem mostrar de que forma os alunos receberam a ideia do minicurso e qual suas perspectivas relacionadas a aprendizagem de programação e principalmente que conceitos matemáticos poderiam ser aprendidos durante essas atividades. Quando perguntados sobre como avaliariam suas experiências com o SpriteBox e o *Scratch* os alunos tiveram a oportunidade de expor opiniões, de que forma aprenderam e o que aprenderam.

Na Figuras 6.44 o depoimento da aluna K está dividido em dois parágrafos. No primeiro ele descreve parte da estrutura de desenvolvimento do SpriteBox, como seus desafios e os níveis de dificuldade em cada fase do jogo. No segundo parágrafo a aluna descreve sua visão sobre a linguagem de programação *Scratch* citando a presença da matemática, conceitos matemáticos como localização e coordenadas e finaliza mostrando sua satisfação com o projeto destacando que todos os benefícios a aprendizagem utilizando o *Scratch* se dão de forma prazerosa a base de muita diversão.

Figura 6.44 – Depoimento da aluna K.

O jogo é composto de desafios que vão geralmente voltados ao número de casas que se deve construir para que a personagem do jogo alcance a linha de chegada, onde esses desafios vão aumentando gradativamente, de acordo com a fase ou nível que você se encontra.

Ele proporciona ao jogador diversos desafios que o farão pensar em como se desvencilhar de determinado obstáculo, além de trabalhar também com a matemática, levando o jogador a aprender ou até mesmo relembrar alguns conceitos matemáticos como, o plano cartesiano e localização, além de muita diversão.

Fonte: Da pesquisa.

Analogamente, a aluna N (Figura 6.45) dividiu seu relato em dois parágrafos. Logo no início de seu primeiro parágrafo ela associando o SpriteBox a conceitos matemáticos e a diversão. A aluna segue citando os blocos e seus encaixes (característica marcante do modo utilizado no projeto para a introdução da lógica de programação utilizando o SpriteBox) necessários ao cumprimento das metas estabelecidas para a sequência do jogo. Ainda em seu primeiro parágrafo a aluna N relaciona o SpriteBox ao plano cartesiano, pois para que spritebox execute as tarefas proposta é necessário que se desloque pela área de construção, seguindo as direções definidas pelo usuário na área de códigos.

Figura 6.45 – Depoimento da aluna N.

Sprite Box é um jogo que envolve conceitos matemáticos e diversão, são usados blocos e feitos as colocações deles para que a personagem alcance sua meta. O principal conceito usado nesse jogo é o plano cartesiano, precisamos saber quais setas colocar exatamente nos comandos para que o Sprite possa se mover com precisão e isso se assemelha muito ao plano cartesiano. Em cada mundo existem vários desafios e os nós nos que controlamos os personagens, isso nos lembra joguinhos antigos como Sonic, Mario Bros, entre outros.

Foi uma boa experiência, jogos educativos sempre geram muitos conhecimentos, as crianças deveriam se interessar mais por eles, os métodos que são utilizados hoje, os meios de comunicações, são ótimas maneiras para se aprender cada vez mais.

Fonte: Da pesquisa.

No segundo parágrafo a aluna N avalia ter sido uma boa experiência o trabalho com jogos educativos, pois estes sempre geram muitos conhecimentos. Ela ressalta que as crianças deveriam se interessar mais por jogos educativos e utilizar a tecnologias digitais da informação e comunicação para aprender cada vez mais.

A valorosa coleta de dados possibilitou que os alunos participantes do projeto pudessem externar seus anseios, perspectivas e opiniões acerca da proposta metodológica utilizada na busca de maior interação, enriquecendo a relação ALUNO-TDIC e produzindo conhecimentos lógicos e matemáticos. A análise desses dados foi fundamental para que se pudesse entender de que forma os alunos desta época pensam as tecnologias digitais, sobretudo de que maneira elas podem contribuir para a aprendizagem da Matemática em sala de aula.

CONSIDERAÇÕES FINAIS

A aprendizagem da Matemática no Brasil vem apresentado singelo crescimento nos últimos anos de acordo com relatório da OCDE apresentado em 2012. O desempenho dos brasileiros em Matemática passou de 356 para 391 pontos no período entre 2003 e 2012 (INEP, 2012). Todavia, há muito o que melhorar. Com isso, esta dissertação pretendeu apresentar as potencialidades e possibilidades que a utilização da linguagem de programação educativa *Scratch* aliada a produção de conteúdos digitais trazem ao processo de aprendizagem da Matemática na perspectiva construcionista. Essa proposta corrobora com a notoriedade do avanço das TDIC, que por hora trouxe inúmeros benefícios para a sociedade nas últimas décadas. Para entendermos de que forma essas TDIC começaram a ser inseridas nas instituições educacionais fizemos uma profunda reflexão sobre os documentos oficiais que fomentaram, fomentam e regulamentam o uso da TDIC na educação brasileira, como o PNE, os PCN, a DCEM e a BNCC.

Durante este levantamento observou-se que a incorporação e o uso pedagógico dos computadores nas escolas começaram de modo mais intenso a partir dos anos 1980 com o aparecimento dos microcomputadores. Todavia, no Brasil os computadores acabam não sendo tão bem aceito por parte da academia, visto que seu propósito vislumbrava uma mudança radical na dinâmica pedagógica, na tentativa de melhoria da qualidade do ensino, o que em outros países como EUA e França tem se dado como um objeto com o qual o aluno deva se familiarizar. Em consonância montamos uma linha do tempo e a partir do Quadro 2.1 investigamos as políticas públicas relacionadas aos principais programas governamentais de incentivo ao uso das TDIC nas escolas, dos quais podemos destacar: o EDUCOM, PRONINFE, Proinfo e o PROUCA.

Na análise do PNE, observou-se ainda o interesse do governo em inserir as TDIC na educação, tanto que as Metas 6, 7, 9 e 15 versam de forma direta ou indireta sobre a inserção, manutenção e aprimoramento dos programas que visam a utilização das TDIC para a melhoria da educação básica no Brasil. Essa preocupação e interesse em tecnologia digital torna-se muito mais evidente quando estudamos a BNCC, visto que 40% das Competências Gerais (CG) da BNCC trazem referências diretas ou indiretas a tecnologia digital. Verificou-se ainda que em articulação com as Competências Gerais, a área de Matemática e o componente curricular de Matemática devem garantir aos alunos o desenvolvimento de Competências Específicas. Dessa forma, a BNCC traça 08 (oito) CE de Matemática para o Ensino Fundamental e 05 (cinco) CE

de Matemática e suas Tecnologias para o Ensino Médio, sendo que das 08 (oito) CE traçadas para o Ensino Fundamental duas delas enfatizam a utilização das TDIC: as CE 05 e CE 06.

Para se conhecer as demandas do cenário nacional acerca da produção de trabalhos acadêmicos que utilizaram a linguagem de programação *Scratch* indireta ou diretamente voltada à mediação do processo de ensino e aprendizagem da matemática foi feita uma revisão da literatura utilizando a Biblioteca Nacional de Teses e Dissertações (BNTD) chegando a resultados surpreendentes. Com a utilização das palavras-chaves *Scratch*; linguagem *Scratch*; Spritebox foram encontrados 211 trabalhos em língua portuguesa, sendo 208 relacionados ao *Scratch* e apenas 3 relacionados ao SpriteBox. No entanto, esse grande número de trabalhos encontrados, deu-se pelo fato de as expressões “a partir do zero” ou “do princípio” quando traduzidas para o inglês seja representada por “*from scratch*” levando a busca eletrônica da BNTD a contabilizar essas expressões, o que culminou no refinamento do filtro para a palavra-chave *linguagem Scratch*, que trouxe como resultados 23 trabalhos, sendo 17 dissertações e 6 teses. A partir da análise dos desses 23 (vinte e três) trabalhos foi constatado que apenas 13 (10 dissertações e 3 teses) utilizaram de fato a linguagem *Scratch*. Dos 13, apenas 5 utilizaram o *Scratch* com o intuito de mediar a aprendizagem de matemática.

A trajetória metodológica desta pesquisa utilizou o SpriteBox como recurso tecnológico facilitador no processo de iniciação a linguagem de programação educativa, que pelos depoimentos dos alunos tornou o processo atrativo e divertido. Com a combinação do SpriteBox, do *Scratch* e da teoria construcionista de Seymour Papert, buscou-se que os alunos desenvolvessem aplicações com conteúdos digitais de matemática que pudessem contribuir para a compreensão de tópicos desta disciplina relacionados ao seu nível escolar. Com o domínio dessas TDIC e sua utilização como ferramenta no ensino de linguagem de programação educativa foram construídas várias aplicações ao longo desta pesquisa. Para se avaliar a evolução dos alunos participantes do projeto foi proposto a construção de uma aplicação em *Scratch* na forma de conteúdo digital com o objetivo de mediar o desenvolvimento das habilidades iniciais necessárias para a marcação de pontos no plano cartesiano. Como resultado foi desenvolvida a aplicação “Marcando Pontos LC” (Apêndice C).

A partir das bases teórica investigadas e das interações apresentadas por cada um dos sujeitos, concluímos que quando o aluno é levado a refletir sobre a organização de comandos (algoritmo) para a realização de uma determinada tarefa, passa a perceber que esta tarefa poderá ser executada mesmo que o algoritmo não esteja completamente adequado ao problema, podendo otimizá-lo a cada execução até que se chegue ao modelo mais apropriado possível. Na resolução de uma equação do 1º grau, por exemplo, onde terá que seguir determinados passos,

o aluno poderá proceder de modo análogo, que poderá chegar ao resultado final ainda que seu algoritmo não esteja completamente adequado, refinando-o até a obtenção de modelo de solução cada vez mais completo. A partir da construção e consolidação dessa habilidade o aluno passará aplicá-la na resolução de outros problemas que tragam em sua essência similar contexto.

É importante ressaltar que o intuito deste trabalho foi maximizar as potencialidades de cada alunos para que estes pudessem construir seus próprios conteúdos digitais a partir da linguagem *Scratch*, desenvolver seu raciocínio organizacional (lógico) e oferecer a professores e gestores uma alternativa metodológica baseada na tendência educacional ligada as tecnologias digitais da informação e comunicação.

REFERÊNCIAS

ANATEL – Agência Nacional de Telecomunicação. Brasil registra redução de 2,88% no número de acessos em operação na telefonia móvel em 12 meses. Disponível em: <<http://www.anatel.gov.br/dados/destaque-1/283-brasil-tem-236-2-milhoes-de-linhas-moveis-em-janeiro-de-2018>>. Acesso em 16.jun.2018.

AZEVEDO, G. T. **Construção de conhecimento matemático a partir da produção de jogos digitais em um ambiente construcionista de aprendizagem: possibilidades e desafios**. 2007, 233 f. Dissertação (Mestrado em Educação em Ciências e Matemática) - Programa de Pós-Graduação em Educação em Ciências e Matemática, Universidade Federal de Goiás, Goiânia, 2017.

BARCELOS, T. S. **Relações entre o Pensamento Computacional e a Matemática através da construção de jogos digitais**. XII SBGames – São Paulo – SP – Brasil, Outubro 16-18, 2013. Disponível em: <http://www.sbgames.org/sbgames2013/proceedings/cd/_Thiago%20Schumacher%20Barcelos_CD_2013.pdf>. Acesso em 19.jun.2018.

BIANCONCINI DE ALMEIDA, M. E. **Tecnologias na Educação: dos caminhos trilhados aos atuais desafios**. Boletim de Educação Matemática, 2008, 21. Disponível em: <http://www.redalyc.org/articulo.oa?id=291221870006>. Acesso em: 07.fev.2019.

BRASIL, Ministério da Educação. Portaria nº 522 de 9 de abril de 1997. Disponível em: <http://www.dominipublico.gov.br/download/texto/me001167.pdf>. Acesso em: 22.mar.2019.

BRASIL, Ministério da Educação. Decreto nº 6.300 de 2007. Disponível em: http://www.planalto.gov.br/ccivil_03/_ato2007-2010/2007/decreto/d6300.htm. Acesso em: 22.mar.2019.

BRASIL. Ministério da Educação. Secretaria de Articulação com os Sistemas de Ensino (MEC/SASE). Planejando a Próxima Década: conhecendo as 20 metas do plano nacional de educação. Brasília, DF: MEC, 2014. Disponível em: http://pne.mec.gov.br/images/pdf/pne_conhecendo_20_metas.pdf. Acesso em: 01.fev.2019.

BRASIL. Base nacional comum curricular. Brasília, DF: MEC, 2016. Disponível em: http://basenacionalcomum.mec.gov.br/wp-content/uploads/2018/12/BNCC_19dez2018_site.pdf. Acesso em: 19.dez.2018.

BRASIL. MEC. SEMT. Parâmetros Curriculares para o Ensino Médio. Brasília, 2000. Disponível em: <http://portal.mec.gov.br/seb/arquivos/pdf/blegais.pdf>. Acesso em: 30.abr.2016.

BRASIL, MEC. Ciências da natureza, matemática e suas tecnologias / Secretaria de Educação Básica. Brasília: Ministério da Educação, Secretaria de Educação Básica, 2006. 135 p. (Orientações curriculares para o ensino médio; volume 2).

COELHO, P. M. F. Os nativos digitais e as novas competências tecnológicas. **Texto Livre: Linguagem e Tecnologia**, n. 2, v. 5, 2012. Disponível em: <<http://periodicos.letras.ufmg.br/index.php/textolivre>>. Acesso em 16.jun.2018.

CYSNEIROS, P. G. (1999). PAPERT, Seymour. **A máquina das crianças**: repensando a escola na era da informática.

FREIRE, P. **Pedagogia da autonomia**: saberes necessários à prática educativa. 51ª ed. Rio de Janeiro: Paz e Terra, 2015.

INEP – Instituto Nacional de Pesquisas Educacionais Anísio Teixeira. Brasil registra maior avanço em matemática, aponta OCDE. 2013. Disponível em: <http://www.publicacoes.inep.gov.br/artigo/-/asset_publisher/B4AQV9zFY7Bv/content/brasil-registra-maior-avanco-em-matematica-aponta-ocde/21206>. Acesso em 17.jun.2018.

_____. **SAEB**. Disponível em: <http://provabrasil.inep.gov.br/web/guest/educacao-basica/saeb> . Acesso em 18.mai.2019.

LÉVY, P. Ciberultura. 1ª reimpressão, Rio de Janeiro, editora 34, 1999.

PAPERT, S. **Mindstorms: Children, computers, and powerful ideas**. Basic Books, Inc., 1980.

_____. **Constructionism: a new opportunity for elementary science education**. A proposal to the National Science Foundation. Cambridge: Massachusetts Institute of Technology, Media Laboratory, Epistemology and Learning Group, 1986.

_____. **The children's machine: rethinking school in the age of the computer**. Nova Iorque: BasicBooks, 1993.

_____. **A Máquina das Crianças**: repensando a escola na era informática. Porto Alegre: Artes Médicas, 2008.

PEREIRA, A. B. C. **Uso de jogos digitais no desenvolvimento de competências curriculares da matemática**. 2015. Tese - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2017.

PRENSKY, M. Digital Native, digital immigrants. Digital Native immigrants. On the horizon, MCB University Press, Vol. 9, N.5, October, 2001. Disponível em: <<https://www.marcprensky.com/writing/Prensky%20-%20Digital%20Natives,%20Digital%20Immigrants%20-%20Part1.pdf>>. Acesso em 16.jun.2018.

QEDU. **Escala SAEB**. Disponível em: <https://academia.qedu.org.br/prova-brasil/454-2/>. Acesso em: 18.mai.2019.

RESNICK, M. (2012, November). **Let's teach kids to code**. [arquivo de vídeo]. Disponível em: https://www.ted.com/talks/mitch_resnick_let_s_teach_kids_to_code. Acesso em: 18.mai.2019.

SANCHO, J. M. et al. Tecnologias para transformar a educação. Tradução Valério Campos. Porto Alegre: Artmed, 2006.

SCRATCH. **Aprenda a Programar, Programe para Aprender**. Disponível em: <https://scratch.mit.edu/about>. Acesso em: 18.mai.2019.

SCRATCHED. **Welcome to the ScratchEd Online Community Archive!**. Disponível em: <http://scratched.gse.harvard.edu/>. Acesso em: 18.mai.2019.

SILVA, S.P. O uso da lógica de programação para a Educação Matemática no Ensino Médio: experiências com o Scratch. 2016. 133f. Dissertação (Mestrado em Ensino de Matemática). Programa de Pós-Graduação em Ensino de Ciências e Matemática, Faculdade de Educação, Universidade Federal de Pelotas, Pelotas, 2016.

SOARES, L. *et al.* Aprendizagem Móvel de Conceitos Introdutórios de Programação: Uma Revisão Sistemática dos Jogos. Nov.2017. ResearchGate. Disponível em: https://www.researchgate.net/publication/324584137_Aprendizagem_Movel_de_Conceitos_Introdutorios_de_Programacao_Uma_Revisao_Sistematica_dos_Jogos/download. Acesso em: 15.05.2019.

VALENTE, J. A. A telepresença na formação de professores da área de informática em educação: implantando o construcionismo contextualizado. In: CONGRESSO RIBIE, 4., 1998, Brasília. Anais...Brasília: 1998, p. 1-16.

VALENTE, J. A. Informática na Educação no Brasil: análise e contextualização histórica. In: Valente, J. A. (Org.). O Computador na sociedade do conhecimento. Campinas: Unicamp/Nied, 1999. p.11-30.

VALENTE, J. A. **Diferentes usos do Computador na Educação**. Documento eletrônico, 2011. Disponível em: http://www.pucrs.br/ciencias/viali/doutorado/ptic/aulas/aula_3/Valente_Jose_2.pdf. Acesso em: 02.jun.2019.

APÊNDICE A

Projeto LPE



SERVIÇO PÚBLICO MUNICIPAL
PREFEITURA MUNICIPAL DE TAILÂNDIA
SECRETARIA MUNICIPAL DE EDUCAÇÃO

E. M. E. F. GUARACI MENDES



PROJETO LINGUAGEM DE PROGRAMAÇÃO EDUCATIVA COM O SCRATCH: formando o cidadão do futuro.

1 Coordenador do Projeto: Giancarlo S. S. Pereira
--

2 Definição da modalidade:

() Curso Aberto	(X) Minicurso Fechado	()
Minicurso Comunitário		

3 Dados de Identificação:

Proponentes: Giancarlo S. S. Pereira

Área: Matemática e Tecnologia da Informação

Subárea: Programação educativa

Público alvo (definição): Alunos da escola.

Nº de vagas: 15 alunos

Carga horária: 2 h por semana (Terças-feiras de 15 às 16h45)

Período de realização: 2º semestre de 2018

Equipe responsável pelo Projeto: Direção, Coordenação e Professores responsáveis.

4 Introdução

No último meio século a Tecnologia da Informação e Comunicação (TIC) se tornou inerente ao processo de evolução da sociedade. Grande parte dos serviços públicos dependem da TIC. É fácil observar essa dependência quando, por motivos de força maior, ficamos sem o fornecimento de energia, por exemplo. Os bancos, os supermercados, os sistemas de transporte aéreo, os serviços de comunicação e outros, dependem dessa tecnologia, que a cada dia se torna mais dinâmica e portátil, atingindo cada vez mais pessoas em todo o mundo. Todavia, Barcelos (2013) ressalta que a falta do domínio adequado de conhecimentos matemáticos pelos alunos é um possível fator explicativo para a falta de interesse e evasão em cursos da área de TIC, por exemplo.

Nesse sentido, observa-se ser imprescindível que nossos alunos tenham conhecimentos e embasamentos matemáticos prévios sólidos para que possam desenvolver atividades relacionadas a área da TIC. Mas, e o processo inverso? **Será que as habilidades e competências em TIC têm influência no desenvolvimento de competências e habilidades em Matemática?** Neste trabalho pretendemos explorar as relações entre a utilização de interfaces de programação em bloco e o processo de ensino-aprendizagem da Matemática. Buscaremos, através da utilização de ambientes de programação em bloco como o *Scratch* e o *App Inventor 2*, ambos do *Massachusetts Institute of Technology* (MIT), que os alunos desenvolvam competências e habilidades que possam facilitar sua compreensão nos conteúdos de matemática da Educação Básica.

5 Justificativa

Segundo dados da Agência Nacional de Telecomunicações (Anatel) o Brasil registrou quase 236 milhões de linhas móveis em operação no mês de março de 2018, o que representa uma densidade de 112,87 celular/100 habitante, ou seja, mais de um aparelho por habitante.

Em 2016 o Cetic.com realizou uma pesquisa sobre o a utilização da TIC nas escolas brasileiras, intitulada “TIC na Educação 2016”. Nessa pesquisa, quando perguntado aos alunos das escolas públicas qual equipamento eles utilizavam para acessar a internet, foram obtidos os seguintes resultados:

Tabela 1 - Alunos, por equipamentos utilizados para acessar a internet

Equipamento	Percentual
Celular	93%
Computador de mesa	50%
Computador portátil	43%
Tablet	31%
Televisão	26%
Videogame	24%
Outros	0%

Fonte: <https://cetic.br/tics/educacao/2016/alunos/B15/>

Nesse sentido, é notório que, mesmo não havendo consenso entre educadores e acadêmicos acerca do uso do celular nas salas de aula, não podemos ignorar a importância dessa tecnologia na vida dos nossos alunos, pois para Prensky (2001 apud Coelho 2012, p.90) os nativos digitais formam uma geração de indivíduos que experimentam e vivenciam múltiplas possibilidades oferecidas pela dinamicidade da tecnologia. Coelho (2012) afirma ainda que

Esse fascínio característico da geração Y pela descoberta e experimentação deve ser explorado pela escola, de forma a direcioná-la para um ensino e uma aprendizagem que dialoguem e interajam com os novos meios tecnológicos.

Este cenário reforça as ideias de Prensky (2001) de que nossos alunos hoje, os “nativos digitais”, formam as primeiras gerações a crescerem cercados de novos dispositivos

tecnológicos, que a cada passo despontam como suas novas ferramentas e brinquedos. O termo “nativos digitais” surge a partir uma analogia que Prensky (2001) faz com pessoas 3nativas e imigrantes. Pessoas nativas aprendem, geralmente, sem dificuldades sua língua materna, já os imigrantes têm certa dificuldade para aprender uma língua diferente. Assim, os nativos digitais nascem inseridos em um mundo tecnológico, tendo praticamente nenhuma dificuldade para aprender seus princípios. Já os imigrantes digitais apresentam certa dificuldade ao lidarem com as novas tecnologias.

Além disso, as Orientações Curriculares para o Ensino Médio (OCEM), das Ciências da Natureza, Matemática e suas Tecnologias, versa em seu capítulo 3 que “é importante contemplar uma formação escolar nesses dois sentidos, [...] a Matemática como ferramenta para entender a tecnologia, e a tecnologia como ferramenta para entender a Matemática”. Quando se considera a tecnologia como ferramenta para entender a Matemática, foco do nosso trabalho, as OCEM destacam que

Há programas de computador (softwares) nos quais os alunos podem explorar e construir diferentes conceitos matemáticos, referidos a seguir como programas de expressão. Os programas de expressão apresentam recursos que provocam, de forma muito natural, o processo que caracteriza o “pensar matematicamente”, ou seja, os alunos fazem experimentos, testam hipóteses, esboçam conjecturas, criam estratégias para resolver problemas. (BRASIL, MEC, 2006, p. 88.)

Com isso, observa-se que não há coerência em dissociar o ensino da tecnologia. Logo, sendo os alunos de hoje nativos digitais, propomos a utilização de interfaces de programação em bloco para que estes construam aplicações digitais que venham trazer significativa contribuição ao seu processo de ensino-aprendizagem de Matemática.

6 Objetivos

OBJETIVO GERAL

Utilizar o Scratch na introdução da linguagem de programação educativa para a construção de conteúdos interativos de Matemática.

OBJETIVOS ESPECÍFICOS

- Analisar métodos de ensino auxiliados por Tecnologia da Informação e Comunicação;
- Investigar a relação entre a utilização de interfaces de programação em bloco e o processo de ensino-aprendizagem da Matemática.
- Apresentar as potencialidades das interfaces de programação em bloco, *SpriteBox*, *Scratch*.
- Propor uma metodologia de ensino baseada na construção de aplicativos;
 - Produzir sequências didáticas assistidas por TIC.
 - Produzir material didático - textos e vídeos - para mediar a iniciação a programação em bloco;
- Aplicar a metodologia proposta numa sala de aula do Ensino Básico;
 - Utilizar o *Sprite Box* como facilitador no processo de iniciação a programação;
 - Mediar a produção de aplicações no *Scratch*;
- Avaliação da metodologia proposta através de questionários, relatos e possíveis reflexões.

7 Do Projeto

O projeto será desenvolvido pelo Prof. Esp. Giancarlo S. S. Pereira, com carga horária de 2h semanais. A princípio, o projeto atenderá a demanda dos alunos da manhã, que frequentarão o projeto as terças-feiras no horário de 15 às 16h45. Os conteúdos que serão ministrados servirão de embasamento teórico e prático necessário para que o aluno desempenhe e construam atividades básicas utilizando a linguagem de programação educativa com o Scratch.

As tendências metodológicas apontadas nas Diretrizes Curriculares de Matemática para o ensino fundamental e médio sugerem encaminhamentos metodológicos e de aporte

teórico para os conteúdos propostos, que deverão ser abordados articuladamente, contemplando os conteúdos relacionados a Tecnologia da Informação e Comunicação. O professor deverá orientar os alunos a partir do processo dialético que possa servir como instrumento eficaz na aprendizagem das propriedades e relações da programação, bem como as diferentes representações e conversões através da linguagem e operações simbólicas, formais e técnicas.

Objetivando uma formação científica geral, os procedimentos e estratégias a serem desenvolvidas pelo professor poderá garantir ao aluno o avanço em estudos posteriores, na aplicação dos conhecimentos matemáticos em atividades tecnológicas, cotidianas, das ciências e da própria ciência matemática.

Em relação aos métodos de abordagens, destaca-se a aplicação e execução dos algoritmos básicas, assim como suas aplicações na análise e interpretação crítica e resolução de problemas apresentados no processo de programação.

8 Critérios Para Seleção dos Alunos

Os critérios apresentados abaixo poderão ser analisados e/ou omitidos.

- Estar regulamente matriculado na Escola Guaraci Mendes;
- Ter o encaminhamento da coordenação pedagógica para ingresso no projeto, através de inscrição prévia.

9 Responsabilidades

Escola M. E. F. Guaraci Mendes: a responsabilidade da Instituição limita-se às dependências da **Escola**.

10 Metodologia

Este projeto propõe uma pesquisa aplicada quali-quantitativa que envolve características de ação empírica. O trabalho será desenvolvido em 7 (sete) etapas (por exemplo: E1, E2, etc.), as quais são divididas em a

E1: Pesquisa Bibliográfica

A11: Pesquisa bibliográfica sobre o uso da Tecnologia da Informação e Comunicação na educação através do ensino de programação;

A12: Pesquisa bibliográfica sobre a legislação educacional vigente e políticas governamentais para a informática educativa;

A13: Pesquisa bibliográfica sobre os programas que serão apresentados nesse trabalho (Scratch, SpriteBox).

E2: Estudo das linguagens de programação educacionais

A21: Estudo do aplicativo Sprite Box;

A22: Estudo da linguagem Scratch.

E3: Planejamento do minicurso de Linguagem de Programação Educacional

A31: Elaboração das atividades;

A32: Elaboração de apostila do minicurso.

E4: Aplicação do minicurso

A41: Introdução a lógica de programação com o SpriteBox

A42: Apresentação do SpriteBox como ferramenta auxiliar na aprendizagem de lógica de programação;

A43: Treinamento e atividades de lógica de programação com o SpriteBox;

A44: Apresentação do Scratch como ferramenta na aprendizagem de programação;

A45: Treinamento e atividades de programação em Scratch;

A46: Atividades de avaliação em Scratch;

E5: Reflexões e Análises sobre o Minicurso

A51: Coleta de dados;

A52: Registro de imagens, falas e comentários;

A53: Análise dos resultados.

E6: Elaboração de trabalhos para publicação em eventos e periódicos relacionados.

10. Descrições de Necessidades Específicas

A definir

11.1. Material de Consumo
A definir
11.2. Material Didático Pedagógico
<ol style="list-style-type: none"> 1. Datashow; 2. Sala de informática; 3. Material Apostilado; 4. Acesso à internet para desenvolvimento da pesquisa; 5. Kit multimídia.
12. Critérios e instrumentos de avaliação da aprendizagem do aluno:
O critério para o acompanhamento e avaliação da aprendizagem do aluno será feito através de portfólio, avaliação diagnóstico e avaliação qualitativa.

13. Critérios e instrumentos de avaliação do minicurso:
Autoavaliação dos Alunos;
Verificação do aprendizado através de testes avaliativos teóricos e práticos ;

14. Referências
<p>ANATEL – Agência Nacional de Telecomunicação. Brasil registra redução de 2,88% no número de acessos em operação na telefonia móvel em 12 meses. Disponível em: <http://www.anatel.gov.br/dados/destaque-1/283-brasil-tem-236-2-milhoes-de-linhas-moveis-em-janeiro-de-2018>. Acesso em 16.jun.2018.</p> <p>BARCELOS, T. S. Relações entre o Pensamento Computacional e a Matemática através da construção de jogos digitais. XII SBGames – São Paulo – SP – Brasil, Outubro 16-18, 2013. Disponível em: <http://www.sbgames.org/sbgames2013/proceedings/cd/_Thiago%20Schumacher%20Barcelos_CD_2013.pdf>. Acesso em 19.jun.2018.</p> <p>BRASIL, MEC. Ciências da natureza, matemática e suas tecnologias / Secretaria de Educação Básica. Brasília : Ministério da Educação, Secretaria de Educação Básica, 2006. 135 p. (Orientações curriculares para o ensino médio ; volume 2).</p> <p>INEP – Instituto Nacional de Pesquisas Educacionais Anísio Teixeira. Brasil registra maior avanço em matemática, aponta OCDE. 2013. Disponível em: <</p>

http://www.publicacoes.inep.gov.br/artigo/-/asset_publisher/B4AQV9zFY7Bv/content/brasil-registra-maior-avanco-em-matematica-aponta-ocde/21206>. Acesso em 17.jun.2018.

MCLUHAN, M. Understanding Media: The extensions of Man. New York: New American Library, 1964. Disponível em: <http://robynbacken.com/text/nw_research.pdf>. Acesso em 16.jun.2018.

PRENSKY, M. Digital Native, digital immigrants. Digital Native immigrants. On the horizon, MCB University Press, Vol. 9, N.5, October, 2001. Disponível em: <<https://www.marcprensky.com/writing/Prensky%20-%20Digital%20Natives,%20Digital%20Immigrants%20-%20Part1.pdf>>. Acesso em 16.jun.2018.

15. Parecer do Conselho Escolar:

16. Parecer da Assessoria Técnico Pedagógica da Escola:

17. Parecer da Direção da Escola:

APÊNDICE B
Plano de aula do projeto LPE



Universidade Federal do Pará (UFPA)
Instituto de Ciências Exatas e Naturais (ICEN)
Programa de Pós-graduação em Matemática

Planejamento da Aula 1

Dados de Identificação	
Professor:	Giancarlo Pereira
Disciplina:	Minicurso de Linguagem de Programação Educativa
Tema:	Introdução a lógica de programação com o SpriteBox
Turma:	Alunos do 9º ano – Matutino.
Data:	25 de setembro de 2018
Horário:	15h às 16h45

1 Objetivos

1.1 Geral

Apresentar o programa SpriteBox e motivar os alunos mostrando que as habilidades adquiridas no aplicativo possibilitará o desenvolvimento do raciocínio lógico.

1.2 Específicos

- Apresentar o SpriteBox;
- Discutir a interface do aplicativo;
- Trabalhar as funções movimentar e fazer;
- Apresentar a estrutura de uma sequência lógica e laços de repetição.

2 Conteúdos

Introdução a lógica de programação: sequência lógica;
Comandos: movimento, faça e laços de repetição.

3 Procedimentos metodológicos

A apresentação do SpriteBox será realizada de forma expositiva e dialogada. Inicialmente apresentaremos as possibilidades de utilização do aplicativo (smartphone ou PC). Nesta aula utilizaremos a versão para web que necessitará de acesso à internet.

Durante esta fase do minicurso as intervenções do professor serão minimizadas pela estrutura didática do aplicativo, que apresenta ao aluno instruções prévias antes de cada novo comando introduzido.

4 Recursos didáticos

Computador, multimídia e material apostilado sobre o Spritebox.

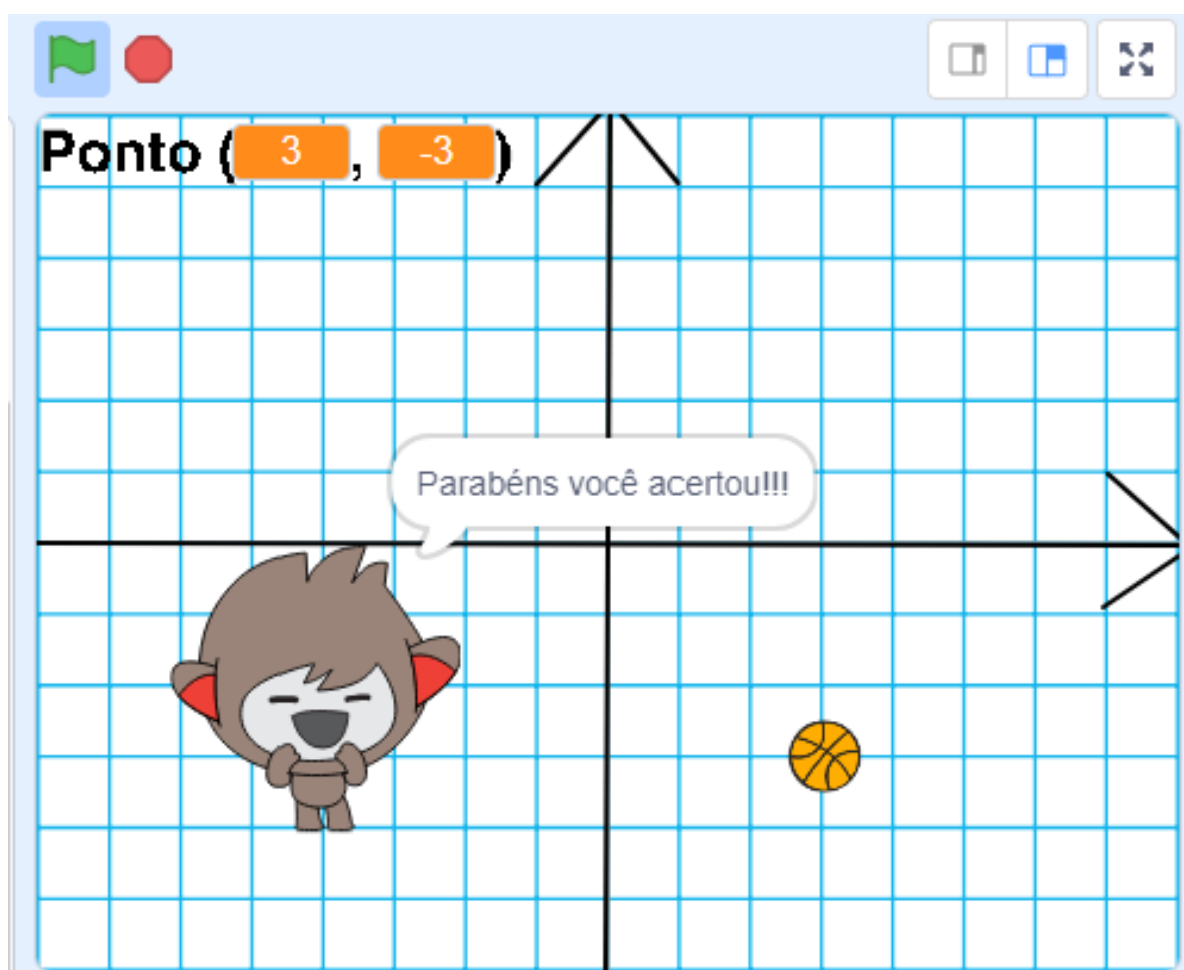
5 Avaliação

Os alunos serão avaliados durante toda o processo de aprendizado, sendo observados suas dificuldades e questionamentos, além da análise do seu desenvolvimento lógico a partir dos diferentes níveis dos desafios apresentados pelo jogo.

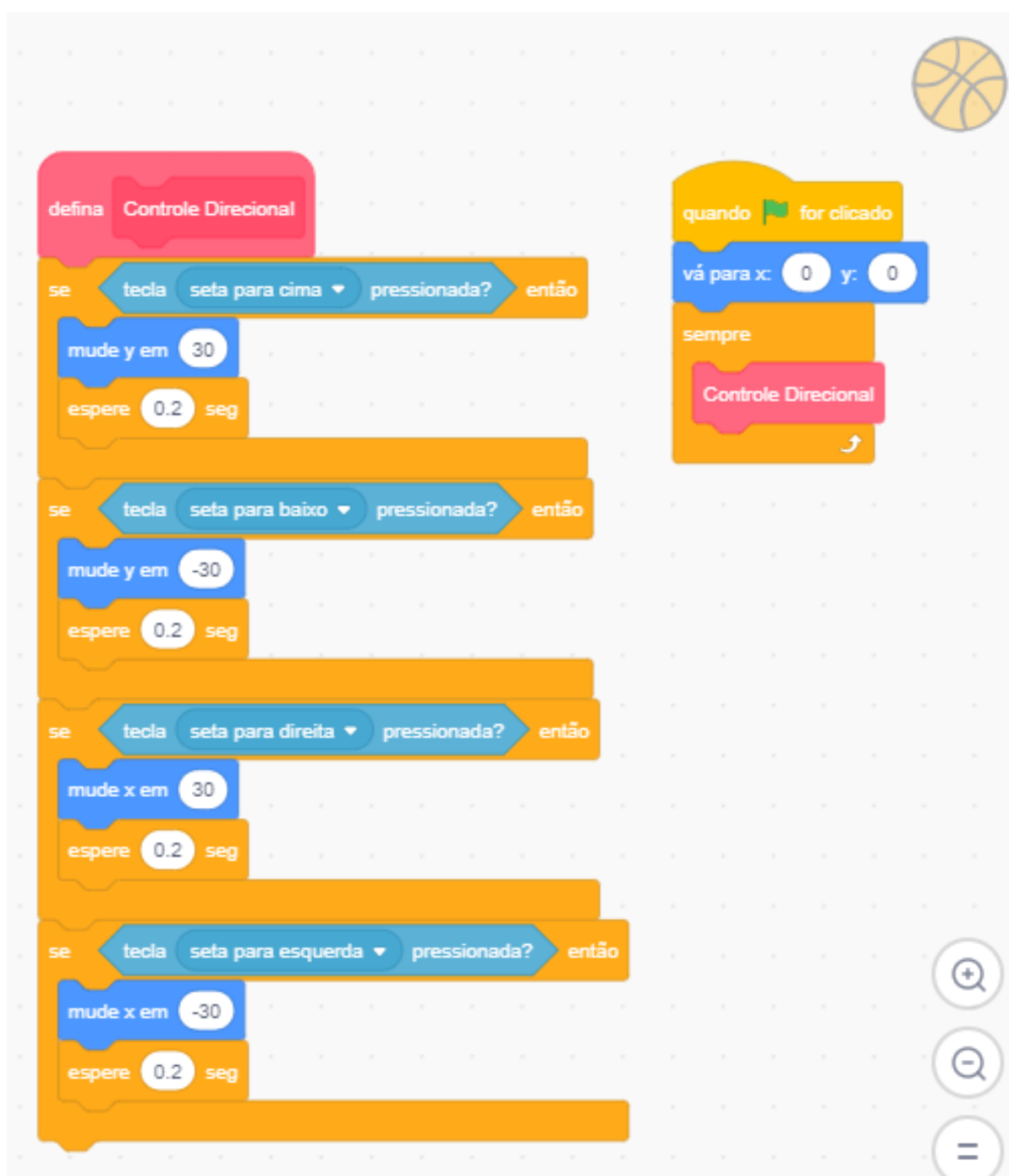
APÊNDICE C

Códigos da aplicação Marcando Pontos LC

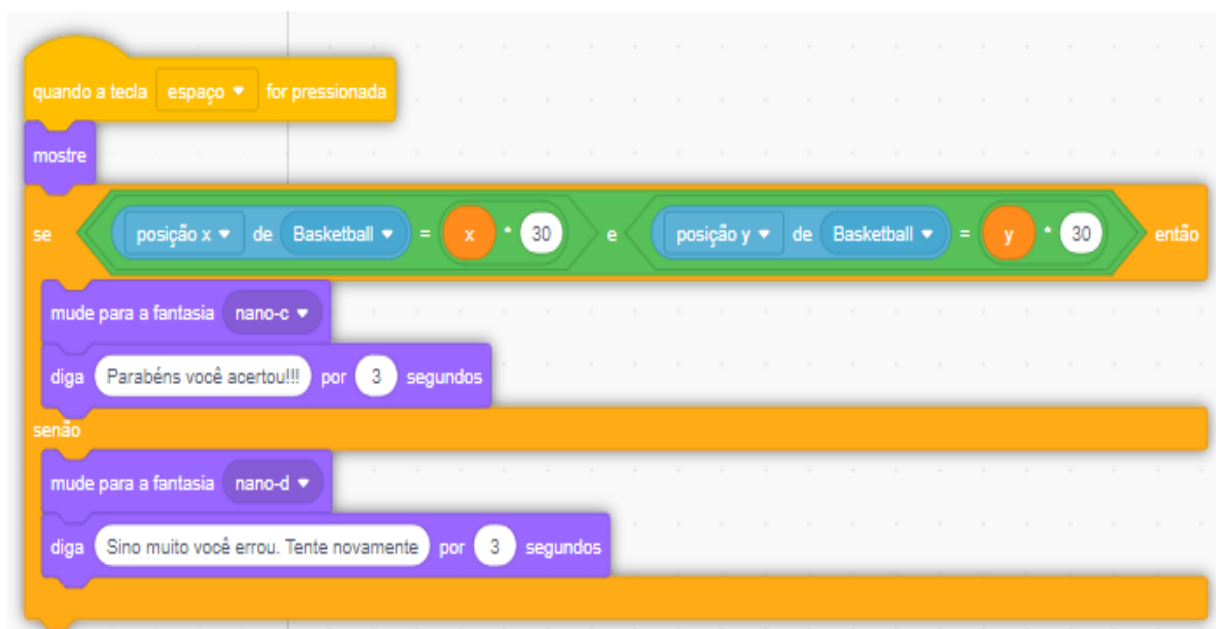
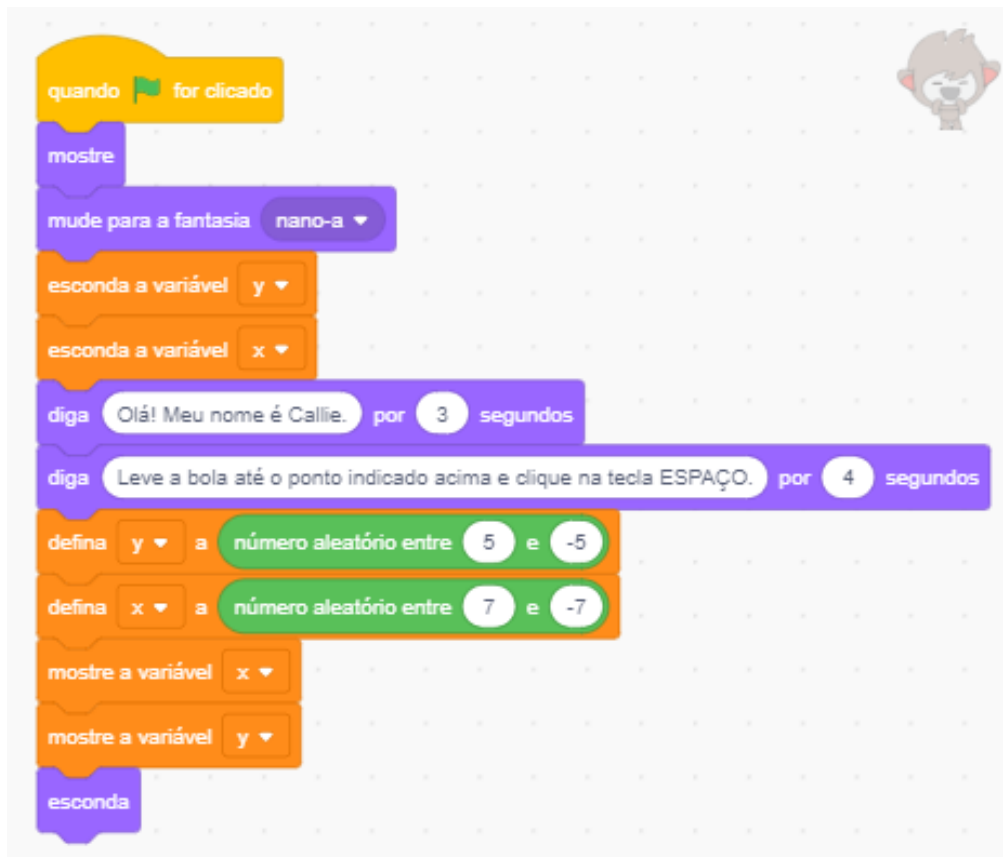
Aplicação: Marcando Pontos LC



Códigos da Bola



Códigos do Callie



Instruções da aplicação

INTRODUÇÃO

Este jogo tem por objetivo auxiliar o aluno no processo de marcação de pontos no plano cartesiano.

Cada vez que uma das teclas direcionais for pressionada (\uparrow \downarrow \rightarrow \leftarrow) a bola se moverá no plano cartesiano uma unidade (para a direita, esquerda, cima ou baixo). Sempre que quiser jogar novamente, pressione a BANDEIRA VERDE.

ORIENTAÇÕES

- ▶ Os pontos aparecerão na parte superior esquerda da tela do jogo;
- ▶ Para movimentar a bola utilize as setas direcionais do teclado;
- ▶ Após levar a bola até o local indicado pressione a tecla ESPAÇO para verificar se você acertou ou não.

Créditos

Projeto de Linguagem de Programação Educativa - LPE

Escola: Guaraci Mendes.

Orientador: Prof. Esp. Giancarlo Pereira

Alunos(as): Ana Clara Marques Pinto; Larissa Araújo Queiróz.