



O Pensamento Computacional na educação

Apresentação e análise de ferramentas de apoio

Victor Coelho Pereira

JUIZ DE FORA

JULHO, 2016

O Pensamento Computacional na educação

Apresentação e análise de ferramentas de apoio

VICTOR COELHO PEREIRA

Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Bacharelado em Ciência da Computação

Orientador: Liamara Scortegagna

JUIZ DE FORA

JULHO, 2016

O PENSAMENTO COMPUTACIONAL NA EDUCAÇÃO
Apresentação e análise de ferramentas de apoio

Victor Coelho Pereira

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.

Aprovada por:

Liamara Scortegagna
Doutora em Engenharia de Produção (UFSC)

Luciana Conceição Dias Campos
Doutora em Engenharia Elétrica (PUC - Rio)

Victor Ströele de Andrade Menezes
Doutor em Engenharia de Sistemas e Computação (UFRJ)

JUIZ DE FORA
27 DE JULHO, 2016

À toda minha família, por sempre me apoiar.

Resumo

O Pensamento Computacional pode ser considerado como uma das habilidades intelectuais básicas de um ser humano, comparada à leitura, fala e realização de operações. Dessa forma é preciso discutir o aumento da abrangência do ensino deste, passando-se a considerá-lo como uma ciência básica. Este trabalho tem como objetivo descrever a utilização do Pensamento Computacional e apresentar tecnologias para seu desenvolvimento, a fim de divulgá-lo como possibilidade para melhorar o cenário educacional e expor os desafios que estão sendo enfrentados em sua implementação. Para a concretização, utilizou-se a metodologia de pesquisa exploratória a fim de estabelecer a base teórica necessária à análise e apresentação das tecnologias de apoio. Com isso, são apresentadas algumas definições relevantes, as principais metodologias aplicadas no processo de ensino e aprendizagem, bem como apresentadas e analisadas quatro ferramentas que auxiliam no desenvolvimento do Pensamento Computacional.

Palavras-chave: Pensamento Computacional, Ciência da Computação, Ambientes Visuais de Programação

Abstract

The Computational Thinking can be considered as one of the basic intellectual abilities of a human being, compared to reading, speaking and performing arithmetic operations. Thus it is necessary to discuss the increased scope of the teaching of this, going to consider it as a basic science. This paper aims to describe the use of Computational Thinking and present the technologies for its development in order to promote it as a possibility to improve the educational environment and expose the challenges being faced in its implementation. For the concretion, it used the exploratory research methodology to establish the theoretical basis necessary for the analysis and presentation of assistive technologies. So, it presents some relevant definitions, the main methodologies applied in the teaching and learning as well as presented and analyzed four tools that assist in the development of computational thinking.

Keywords: Computational Thinking, Computer Science, Visual Programming Environments.

Agradecimentos

A todos os meus parentes, por sempre me apoiarem, principalmente nas horas mais complicadas.

A professora Liamara Scortegagna pela paciência, compreensão e pelas valiosas orientações que me ajudaram a trilhar o caminho da pesquisa e da elaboração desse trabalho.

A todos os professores do Departamento de Ciência da Computação pelos seus ensinamentos.

E principalmente à minha namorada Luciana, por estar sempre comigo, me apoiando e motivando.

“Se a educação sozinha não transforma a sociedade, sem ela tampouco a sociedade muda.”

Paulo Freire

Conteúdo

Lista de Figuras	7
Lista de Tabelas	8
Lista de Abreviações	9
1 Introdução	10
1.1 Justificativa	10
1.2 Objetivos	11
1.3 Metodologia	12
1.4 Estrutura do trabalho	12
2 Pensamento Computacional	14
2.1 Conceito	14
2.2 Metodologias de ensino do Pensamento Computacional	18
3 Apresentação e análise das ferramentas de apoio ao Ensino do Pensamento Computacional	21
3.1 Scratch	21
3.2 RoboMind	23
3.3 App Inventor	25
3.4 SuperLogo	27
4 Considerações finais	29
Referências Bibliográficas	32

Lista de Figuras

3.1	Tela inicial do Scratch - Fonte: site	22
3.2	RoboMind	24
3.3	App Inventor Designer	25
3.4	App Inventor: Janela Blocks Editor	26
3.5	SuperLogo - Janela Gráfica	28
3.6	SuperLogo - Janela de Comandos	28

Lista de Tabelas

3.1	Instruções para movimentação do robô	24
3.2	Comandos básicos para movimentação da tartaruga no SuperLogo	27

Lista de Abreviações

CSTA	Computer Science Teachers Association
ISTE	International Society for Technology in Education
MIT	Massachusetts Institute of Technology
NSF	National Science Foundation
SBC	Sociedade Brasileira de Computação

1 Introdução

A Computação constitui uma área de conhecimento que abrange inúmeras atividades humanas, de forma que não se pode imaginar uma sociedade sem computadores e suas tecnologias (França et al, 2014). A combinação do pensamento crítico com os fundamentos da Computação define uma metodologia para resolver problemas, que Wing (2006) denominou Pensamento Computacional.

Pensamento Computacional é o processo de pensamento envolvido na formulação de problemas e das suas soluções, tendo como base os fundamentos da Ciência da Computação (Andrade et al, 2013). Esta visão defende que a cultura do computador ajuda a sociedade não somente a aprender, mas especialmente oferece uma nova maneira de aprender a aprender. De forma geral, habilidades comumente utilizadas na criação de programas computacionais para resolver problemas específicos são utilizadas como uma metodologia para resolver problemas nas mais diversas áreas (Nunes, 2011).

Dessa forma, Nunes (2011) afirma que a introdução do Pensamento Computacional na educação básica provê os recursos cognitivos necessários à resolução de problemas, transversal a todas as áreas do conhecimento. Assim, inclusão da Ciência da Computação na educação básica brasileira é algo acontecerá em breve (França et al, 2014).

1.1 Justificativa

A lista de conhecimentos e habilidades necessária para o pleno exercício da cidadania no século XXI é extensa e inclui o Pensamento Computacional (Blikstein, 2008), que apoia e se relaciona com diversas outras áreas de conhecimento. De acordo com Wing (2006), o Pensamento Computacional pode ser colocado como uma das habilidades intelectuais básicas de um ser humano, comparada à ler, escrever, falar e fazer operações aritméticas.

As soluções computacionais estão presentes em problemas de diversas áreas, promovendo oportunidades de emprego e inserção de jovens em um mercado de trabalho globalizado e de alta competitividade (França et al, 2014).

Dessa forma discute-se o aumento da abrangência do ensino da Ciência da Computação, passando-se a considerá-la uma ciência básica (Barcelos; Silveira, 2012). Um subconjunto de competências e habilidades básicas relacionadas a esta área de conhecimento deveria ser desenvolvido pelos estudantes desde o ensino básico. Segundo Wing (2006), esse subconjunto de competências e habilidades, denominado como Pensamento Computacional, envolve as competências relacionadas à abstração e decomposição de problemas de forma a permitir sua resolução utilizando recursos computacionais e estratégias algorítmicas.

Empresas multinacionais apoiam o ensino do Pensamento Computacional e se empenham na disseminação de sua metodologia. Em 2007, a Microsoft e a Universidade de Carnegie Mellon criaram o Centro de Pensamento Computacional (Carnegie Mellon, 2013). O Google, por sua vez, tem se empenhado em promover esta metodologia em todo o currículo do ensino primário e secundário nos Estados Unidos (Google, 2016).

No Brasil, podemos destacar dois projetos voltados à introdução do Pensamento Computacional no âmbito do ensino básico: o Game Design Brasil, vinculado a Pontifícia Universidade Católica do Rio de Janeiro, que tem como objetivo o desenvolvimento de uma tecnologia brasileira para estimular o ensino e aprendizado de Computação em escolas de ensino fundamental e médio (Puc-Rio, 2016) e o Computação Desplugada, que tem como meta propagar os fundamentos da Ciência da Computação sem o uso de computador para alunos de escolas públicas (Scaico et al, 2012b).

1.2 Objetivos

Com este trabalho, busca-se pesquisar e descrever a utilização do Pensamento Computacional nas escolas e apresentar tecnologias para sua concretização, a fim de divulgá-lo como possibilidade para melhorar o cenário educacional e expor os desafios que estão sendo enfrentados em sua implementação.

Como objetivos específicos, podem ser citados:

1. Definir o conceito de Pensamento Computacional e os assuntos tratados por essa metodologia;

2. Pesquisar e apresentar métodos para o ensino do Pensamento Computacional propostos na literatura;
3. Apontar as tecnologias disponíveis e apresentar como elas auxiliam no processo de ensino da metodologia do Pensamento Computacional.

1.3 Metodologia

Com o intuito de alcançar os objetivos descritos, foi realizada uma pesquisa exploratória sobre o Pensamento Computacional, a fim de estabelecer a base teórica necessária para a análise e apresentação de ferramentas voltadas para a Educação. Com isso, busca-se auxiliar as escolas na escolha de uma dessas ferramentas para a inclusão da metodologia do Pensamento Computacional em suas instituições.

Esse tipo de pesquisa, segundo Gil (2002), busca trazer mais informações sobre um tema através de pesquisas bibliográficas. Em geral, esse tipo de pesquisa envolve:

1. Um levantamento bibliográfico, que utiliza artigos, monografias, dissertações, teses, livros, periódicos e pesquisas na Internet como fontes de pesquisa;
2. A análise de exemplos para auxiliar no entendimento do tema.

1.4 Estrutura do trabalho

Esse trabalho está estruturado da seguinte forma:

No primeiro capítulo, de introdução, é realizada a contextualização do tema e apresentada justificativa para realização do trabalho e seus objetivos e a metodologia utilizada na pesquisa e a maneira como o trabalho está estruturado.

Os conceitos e definições sobre o Pensamento Computacional são descritos no segundo capítulo, bem como as principais metodologias para ensino do Pensamento Computacional.

No terceiro capítulo, é feita uma apresentação e análise sobre as ferramentas de apoio ao ensino do Pensamento Computacional. As seguintes ferramentas são analisadas neste capítulo: Scratch, RoboMind, App Inventor e SuperLogo.

Por fim, o quarto capítulo apresenta as considerações finais a respeito do tema abordado neste trabalho e algumas propostas para trabalhos futuros.

2 Pensamento Computacional

2.1 Conceito

A Computação apresenta mecanismos singulares de raciocínio para resolução de problemas, cujas aplicações ultrapassam as fronteiras da Computação em si (Barcelos; Silveira, 2012). O Pensamento Computacional é o método que utiliza fundamentos e técnicas da Ciência da Computação para solução de problemas (Wing, 2006). Esse método tem como proposta aplicar habilidades utilizadas para desenvolvimento de programas computacionais como metodologia para resolução de problemas gerais (Andrade et al, 2013).

De acordo com Andrade et al (2013), existem três pilares básicos que fundamentam o Pensamento Computacional: a abstração, a automação e a análise.

A abstração é a habilidade de extrair apenas as características importantes de um problema para o solucionar, levando em consideração que as características secundárias já foram solucionadas. Problemas com elevado grau de complexidade tornam-se difíceis de serem resolvidos caso não seja feita uma abstração adequada.

A automação é a utilização de um meio eletrônico na substituição do trabalho manual. Um computador é considerado um meio eletrônico capaz de substituir o trabalho de um ser humano. Entretanto, o computador só será útil se forem dadas instruções corretas para a execução de determinada tarefa.

A análise é o estudo dos resultados obtidos pela automação. Se eles não saírem como o esperado, existe a possibilidade de não ter sido escolhido um nível de abstração adequado ou até mesmo de não ter sido planejada a solução automatizada corretamente.

Wing (2006) define o Pensamento Computacional através das seguintes características:

Conceituar ao invés de programar. O Pensamento Computacional significa mais do que saber programar. Aplicar o Pensamento Computacional para a resolução de problemas significa reduzir os problemas, aparentemente insolúveis, em problemas menores e mais simples de serem solucionados. Isso exige a capacidade de pensar de forma abstrata

e não a mera aplicação de técnicas de programação.

É uma habilidade fundamental e não repetitiva. Uma habilidade fundamental é algo que todo ser humano precisa conhecer para viver na sociedade moderna. O Pensamento Computacional não é uma habilidade mecânica ou repetitiva, mas algo que permite a resolução de problemas diversos utilizando um recurso de fácil acesso na sociedade atual – os computadores – e por isso deveria ser desenvolvido por todos os estudantes.

É a maneira na qual pessoas pensam, e não os computadores. O Pensamento Computacional não é uma redução do raciocínio para simular o processamento do computador, ao contrário disso. Pessoas são inteligentes e criativas. Dessa forma, o computador é apenas um facilitador para a resolução de problemas através do Pensamento Computacional, aplicando um tratamento específico para um problema, de forma que ele possa ser resolvido por computadores.

Complementa e combina a Matemática e a Engenharia. O Pensamento Computacional utiliza de forma inerente o pensamento matemático, uma vez que os seus fundamentos formais são baseados na matemática. Além disso, o Pensamento Computacional também utiliza de forma inerente o pensamento de engenharia, uma vez que são construídos sistemas que interagem com o mundo real. As limitações dos computadores forçam a utilização do pensar computacionalmente, não apenas matematicamente. Ser livre para construir mundos virtuais permite projetar sistemas para além do mundo físico.

Gera ideias e não artefatos. O Pensamento Computacional não irá necessariamente produzir um *software* ou *hardware*. O Pensamento Computacional envolve conceitos computacionais que usamos para abordar e resolver problemas, gerenciar nossas vidas diárias, e se comunicar e interagir com outras pessoas.

Para todos, em qualquer lugar. Por fim, o Pensamento Computacional é útil para todas as pessoas, em diversas aplicações.

Além da definição de Pensamento Computacional, Wing (2006) ainda defende a introdução do Pensamento Computacional no ensino básico de forma a despertar o interesse dos alunos pela área de Computação, mostrando a versatilidade e relevância na solução de problemas do mundo atual.

Isbell et al (2010) apresentam o resultado de um debate sobre a natureza da

Computação com o objetivo de orientar o currículo de cursos de graduação. O resultado desse debate foi uma visão alternativa sobre o Pensamento Computacional. Na visão dos autores, o conceito de Computação está baseado na produção de modelos representativos de um determinado domínio. Esses modelos devem ser expressos em uma determinada linguagem e podem ser manipulados, interpretados e modificados por uma máquina. Nesse caso, o Pensamento Computacional se baseia nos seguintes conceitos: modelos; abstração; interpretação de dados ou de padrões; escala e limites para o processamento; simulação de sistemas a partir de modelos e possibilidade de automação dos modelos. A partir desses conceitos, os autores enumeram algumas habilidades necessárias para desenvolver modelos baseados no Pensamento Computacional. Essas habilidades são apresentadas a seguir:

- Analisar um problema para entender o contexto e requisitos;
- Desenvolver uma solução para esse problema e implementar essa solução usando ferramentas e técnicas adequadas;
- Verificar se a solução - que pode ou não pode ser um programa de computador - funciona corretamente e validar que corresponde aos resultados esperados;
- Identificar e corrigir os componentes com problemas;
- Desenvolver documentos sobre a solução para permitir que outras pessoas entendam quais foram as razões para as decisões tomadas durante o desenvolvimento;
- Gerenciar o processo de desenvolvimento, incluindo ser capaz de informar estimativas sobre as dificuldades relacionadas ao desenvolvimento do modelo.

Blikstein (2008) afirma que o Pensamento Computacional é saber utilizar o computador como instrumento de aumento do poder cognitivo e operacional humano. O autor define duas etapas para o que ele chama de “pensar computacionalmente”. A primeira etapa é identificar tarefas cognitivas que podem ser realizadas de forma mais rápida e eficiente por um computador. A segunda etapa é saber programar um computador para realizar essas tarefas cognitivas, em outras palavras, transferir aquilo que não é essencialmente humano para um computador. Dessa forma, o autor defende o uso de

computadores e redes de computadores para aumentar a produtividade, inventividade e criatividade. Blikstein (2008) ainda critica a forma como as escolas ensinam as novas tecnologias, argumentando que os alunos estão aprendendo apenas a recombina informações já existentes em mecanismos de busca e não estão aprendendo a criar conhecimento novo por meio da tecnologia.

Com base no princípio de que todos os alunos devem demonstrar competência nas habilidades básicas do Pensamento Computacional na conclusão do ensino médio, as organizações CSTA, ISTE e a NSF propuseram um conjunto com nove conceitos da área da Computação considerados fundamentais de serem trabalhados para o desenvolvimento do Pensamento Computacional (CSTA et al, 2011). Esses conceitos são enumerados a seguir.

1. Coleta de dados: é o processo de reunir dados de forma apropriada.
2. Análise de dados: é o passo que objetiva tornar os dados coletados coerentes, encontrando padrões e tirando conclusões a partir destes dados.
3. Representação de dados: é o processo de organizar apropriadamente as informações por meio de tabelas, gráficos, palavras, imagens ou qualquer outro recurso disponível.
4. Decomposição de problemas: é capacidade de divisão das tarefas em partes menores e manuseáveis.
5. Abstração: é a redução da complexidade de um problema para focar na questão principal.
6. Algoritmos e procedimentos: são definidos como uma possível série organizada de passos para resolver um problema ou atingir algum objetivo.
7. Automação: é a utilização de computadores ou máquinas para fazer tarefas repetitivas ou tediosas.
8. Simulação: é a representação ou a modelagem de um processo e a sua execução.

9. Paralelismo: é a forma de organizar recursos para simultaneamente desenvolver tarefas que atinjam um objetivo em comum.

Dessa forma, de acordo com CSTA et al (2011), o Pensamento Computacional é um processo de resolução de problemas que inclui, mas não se limita, algumas características como formular problemas de modo que seja possível utilizar um computador para auxiliar a resolvê-los; organizar os dados logicamente de modo que, no futuro, seja possível analisá-los; representar dados através de abstrações, tais como modelos e simulações; criar formas de automatizar as soluções através do pensamento algorítmico; permitir identificar, analisar e implementar soluções possíveis, com o objetivo de conseguir a combinação mais eficiente e eficaz de etapas e recursos; generalizar e transferir este processo de solução de problemas a uma variedade de outros problemas.

2.2 Metodologias de ensino do Pensamento Computacional

O interesse em ensinar a Computação na educação básica é um tema que vem sendo tratado em vários locais no mundo. Em alguns países já se utiliza um currículo mínimo para o ensino da Ciência da Computação, como em Israel (Gal-Ezer, 1995), no Canadá (MEO, 2008) e nos Estados Unidos, onde o modelo de currículo investe no ensino de conteúdos baseados na chamada Educação Imperativa, a qual mais importante do que aprender temas ligados às tecnologias está a capacidade de desenvolver o Pensamento Computacional (Scaico et al, 2012b). Um dos projetos com maior aceitação e mundialmente conhecido por favorecer o ensino do Pensamento Computacional é denominado Computação Desplugada (Scaico et al, 2012b).

A Computação Desplugada é baseada no livro “Ensinando Ciência da Computação sem o uso do computador”, escrito por Bell et al (2011). Este livro apresenta uma série de atividades lúdicas desenvolvidas com o objetivo de ensinar os fundamentos da Ciência da Computação sem a necessidade de computadores. A grande vantagem dessa abordagem é a independência de recursos de hardware ou software. Dessa forma, as atividades propostas no livro podem ser aplicadas em qualquer lugar, até mesmo em escolas

sem a disponibilidade de computadores ou dispositivos tecnológicos. Além disso, as atividades podem ser ministradas por uma pessoa que não é especialista em Computação. Os conteúdos abordados nas atividades propostas baseiam-se em conceitos fundamentais da Ciência da Computação, o que torna seu uso abrangente e evita a defasagem do conteúdo no tempo (Sousa et al, 2011).

O site *Computer Science Unplugged*¹ disponibiliza o livro gratuitamente em diversos idiomas. Este livro está estruturado em três partes: (i) Representando as Informações, (ii) Algoritmos, (iii) Representando Procedimentos. A Representação das Informações possui atividades que ilustram as formas utilizadas pelos computadores na representação dos dados tratando de temas como armazenamento e representação da informação (números binários, texto e imagens) e compressão de dados. A parte sobre Algoritmos aborda métodos computacionais de uso frequente no cotidiano, tais como os algoritmos de ordenação e de busca de informação. A última parte, Representação de Procedimentos, apresenta conceitos mais avançados, a exemplo dos autômatos de estados finitos, grafos e linguagens de programação (Scaico et al, 2012a). A Computação Desplugada vem sendo empregada em diversos países, como Estados Unidos, França e Japão (Sousa et al, 2011). No Brasil, há relatos do uso dessa metodologia, evidenciado por Sousa et al (2011) e Scaico et al (2012a).

A interdisciplinaridade do Pensamento Computacional também tem sido considerada em ações para disseminar os conhecimentos em Computação. Nessa iniciativa, os alunos e professores são envolvidos em projetos interdisciplinares, nos quais a Computação age como meio para a construção de soluções. Além disso, existe a possibilidade de desenvolver jogos digitais que apoiam o ensino de áreas como a Matemática. O desenvolvimento de tais artefatos é realizado por meio de ambientes visuais de programação. Essa iniciativa apoia a aprendizagem de conceitos associadas às disciplinas contempladas, além de possibilitar o desenvolvimento do raciocínio lógico dos alunos (França et al, 2014).

Outra proposta de trabalho que vem sendo adotada refere-se ao ensino de programação como forma de desenvolvimento do raciocínio lógico. O desenvolvimento de algoritmos é a base para todas as áreas relacionadas com a Computação, no entanto, o

¹<http://csunplugged.org/>

estudo de algoritmos não deveria interessar apenas de estudantes de Computação (Scaico et al, 2013).

Aprender a programar permite desenvolver outras habilidades que contribuem para aprimorar o raciocínio lógico. O desenvolvimento de algoritmos envolve a habilidade de solucionar problemas. Se o problema for grande, será preciso o exercício de outras habilidades, como dividir o problema em subproblemas e criar uma solução central. Entretanto, aprender uma linguagem de programação é uma tarefa desafiadora. Sendo assim, é preciso tornar o ensino de programação acessível para um maior número de indivíduos, estimulando capacidades cognitivas e possibilitando a aplicação de técnicas utilizadas na programação para a resolução de diversos outros tipos de problemas, nas mais distintas profissões (Scaico et al, 2013).

Dessa forma, devido às dificuldades na aprendizagem de programação apresentadas por iniciantes, desenvolveu-se o *ensino de programação orientado ao design* (Scaico et al, 2013). Essa proposta utiliza ambientes visuais de programação que possibilitam introduzir tal conceito de maneira simples e fácil. Desse modo, o aluno foca apenas na lógica de funcionamento do projeto, sem preocupar-se com a sintaxe de uma linguagem de programação específica (França et al, 2014).

3 Apresentação e análise das ferramentas de apoio ao Ensino do Pensamento Computacional

As ferramentas de apoio facilitam o processo de ensino da Ciência da Computação, de forma lúdica e dinâmica. A utilização dessas ferramentas no âmbito escolar, propõem mais que o ensino dos conceitos de Computação. Através delas os alunos se sentem desafiados e desenvolvem a criatividade. Além disso, as ferramentas facilitam a resolução de problemas habituais, construindo assim o Pensamento Computacional (França et al, 2014).

A seguir, serão apresentadas algumas ferramentas para auxílio do ensino do Pensamento Computacional.

3.1 Scratch

O Scratch é uma linguagem de programação visual, desenvolvida pelo Media Lab do MIT, que possibilita a criação de histórias interativas, animações, simulações e jogos, e o compartilhamento dessas criações na Web.

O ambiente Scratch apresenta uma linguagem que auxilia a aprendizagem de programação através de um conceito inovador de desenvolvimento de código orientado ao design. Esse conceito privilegia a *Computação Criativa*. De acordo com Scaico et al (2013), a computação criativa é uma expressão utilizada para assumir que o conhecimento e as práticas que os jovens precisam adquirir para criar um *software* devem ser provenientes dos seus interesses pessoais. Além disso, a utilização do Scratch possibilita ensinar as principais estruturas de uma linguagem de programação e praticar técnicas utilizadas na construção de algoritmos, como é o caso do uso da abstração, depuração de erros, testes e melhoria de algoritmos simples (França e Tedesco, 2015).

Dessa maneira, o Scratch possibilita ao aluno concentrar-se no exercício do Pensamento Computacional e na criatividade durante o desenvolvimento das soluções, sem

precisar se preocupar com questões que são importantes, mas acessórias à programação, como é o caso da correção de problemas de sintaxe. Isso incentiva o aluno iniciante a se empenhar mais na criação dos códigos (Scaico et al, 2013).

As aplicações utilizando o Scratch também podem interagir com o mundo real, através de informações obtidas de uma câmera ou de um microfone. Assim, o comportamento que pode ser produzido é algo semelhante ao que acontece em jogos modernos (Scaico et al, 2013).

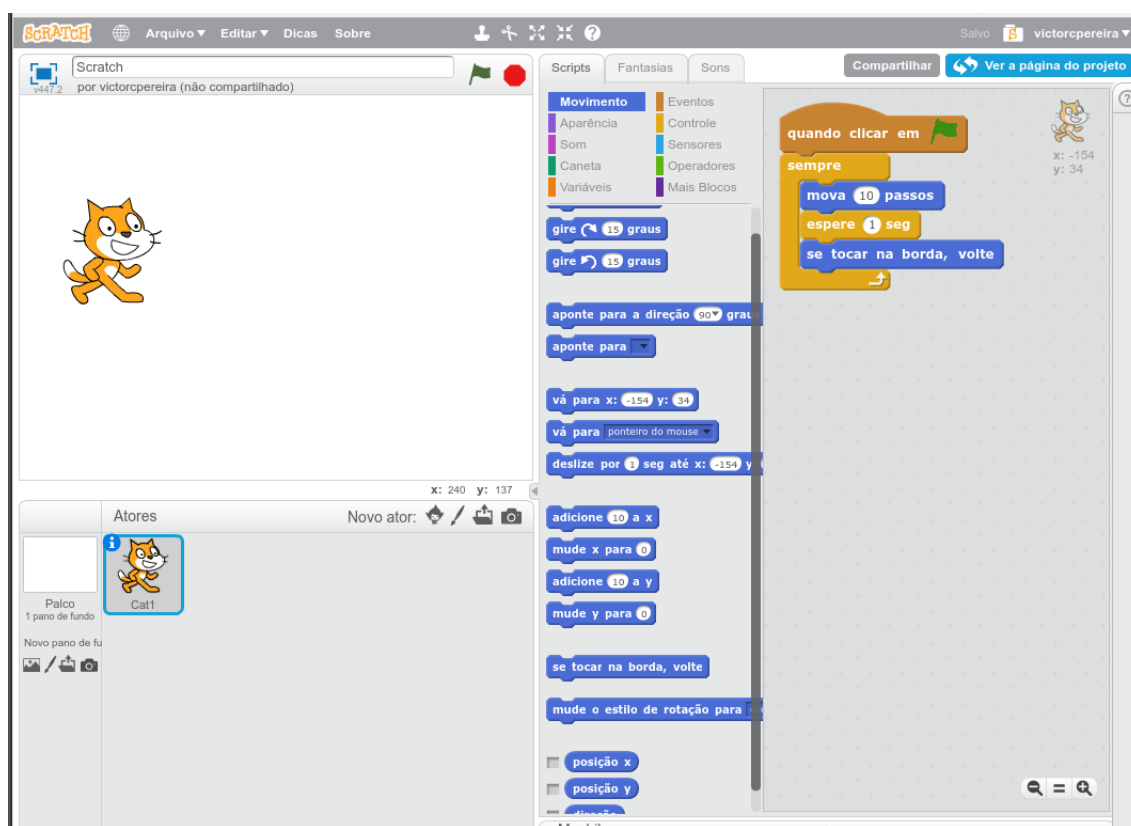


Figura 3.1: Tela inicial do Scratch - Fonte: site

Um projeto no Scratch é formado por um palco e por vários objetos. O palco é o plano de fundo estático onde os objetos podem executar as suas ações. Para executar o projeto e definir o comportamento de cada um dos objetos, os usuários associam a estes objetos sons, imagens e variáveis que podem ser manipulados pelos diferentes tipos de comandos fornecidos pela linguagem de programação do ambiente. Os comandos do Scratch precisam somente ser arrastados e soltados no espaço de comandos, onde, posteriormente, podem ser encaixados uns nos outros. Os comandos no Scratch assemelham-se a peças de quebra-cabeça e indicam com sua forma como podem ser combinados corretamente. Desta

maneira, os comandos quando combinados formam programas que estão sempre sintaticamente corretos. Sem a preocupação com erros sintáticos, os usuários focam apenas na criação da lógica de funcionamento de seus projetos (Aureliano e Tedesco, 2012).

A principal limitação que o Scratch apresenta é a falta de um feedback passo a passo sobre o que o aluno produz na ferramenta. Esse feedback poderá ajudar o aluno a entender qual parte do algoritmo está sendo executado e qual apresenta um erro, por exemplo. A diversidade de objetos lúdicos: jogos, animações, e histórias interativas, que se pode construir na ferramenta, e a linguagem em blocos são os pontos positivos que se destacam na ferramenta (Silva et al, 2014).

3.2 RoboMind

RoboMind é um ambiente de desenvolvimento criado pela Universidade de Amsterdam, com o intuito de facilitar no ensino de conceitos iniciais de programação de computadores (Silva et al, 2014). Esse ambiente oferece uma linguagem de programação simples para movimentação de um robô. Esse robô precisa ser programado para realizar algumas tarefas dentro de um mundo bidimensional. Dessa forma é possível ensinar conceitos de lógica de programação e ter uma noção inicial das áreas de Inteligência Artificial e Robótica.

A interface do ambiente de desenvolvimento é composta de três partes: à esquerda encontra-se a área de edição do programa; à direita o mundo em que o robô se movimenta de acordo com o programa; e na parte inferior está o controle de execução do programa junto de uma área de mensagens, que é usada para apresentar erros sintáticos ou situações em que o robô tem problemas durante a execução, como por exemplo, quando o robô bate numa parede (Benitti et al, 2009).

O robô contém sensores que conseguem enxergar se existem objetos ao seu redor, por exemplo, `vazioEsquerda()`, `temObstáculoEsquerda()`, `vazioFrente()`, `temObstáculoFrente()`, e assim por diante. Além disso, a linguagem utilizada pelo RoboMind oferece um conjunto de instruções para movimentação do robô, apresentado na tabela 3.1.

A linguagem também apresenta comandos para estrutura de seleção (se-senão) e repetição contendo três versões: `(repetir(n))`, repete a instrução *n* vezes; `(repetir())`, repete continuamente a instrução; e `(repetirEnquanto(condição))`, repete enquanto a condição for

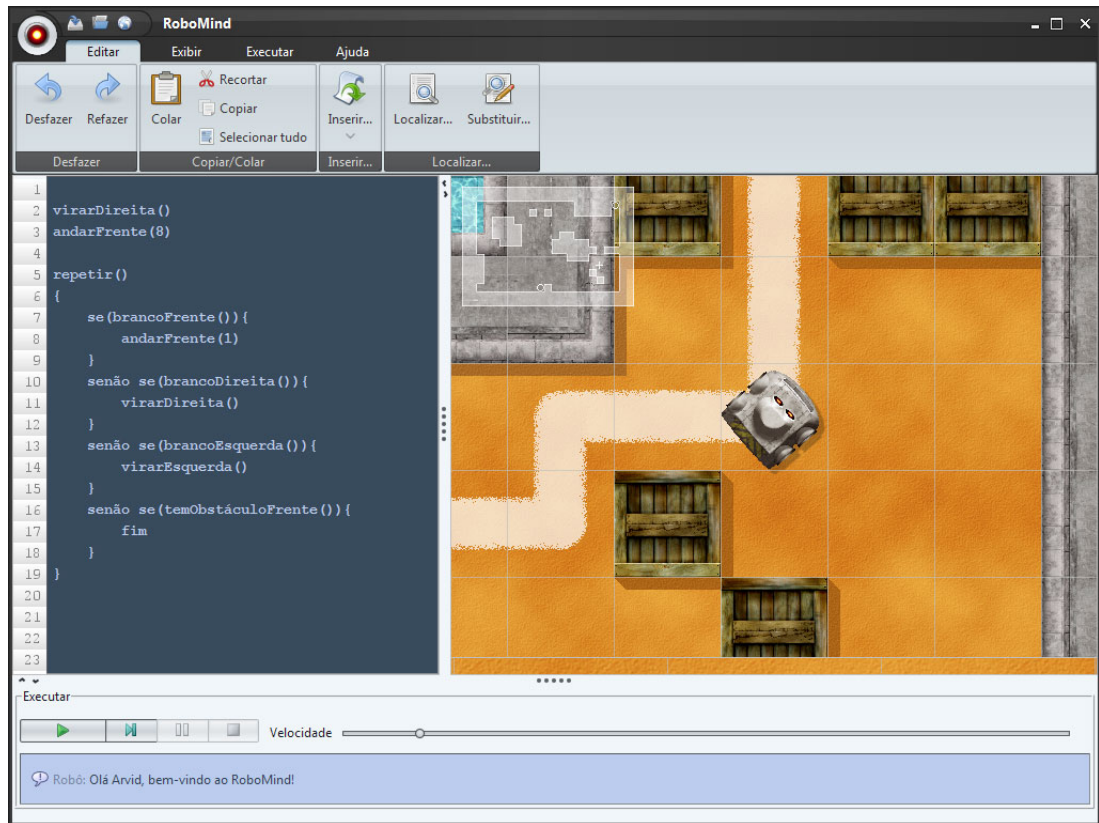


Figura 3.2: RoboMind

satisfeita. Dessa forma, é possível utilizar o RoboMind para o ensino de conceitos básicos de programação associado aos conceitos de expressões lógicas, estruturas de decisão e estruturas de repetição (Silva et al, 2014).

Tabela 3.1: Instruções para movimentação do robô

Comando	Descrição
andarFrente(n)	faz o robô se movimentar para frente
andarTrás(n)	faz o robô se movimentar para trás
andarNorte(n)	faz o robô se movimentar para o norte
andarSul(n)	faz o robô se movimentar para o sul
andarLeste(n)	faz o robô se movimentar para o leste
andarOeste(n)	faz o robô se movimentar para o oeste
virarEsquerda()	move o robô 90° para a esquerda
virarDireita()	move o robô 90° para a direita
pegar()	pegar objetos
soltar()	soltar objetos

O RoboMind foi desenvolvido para permitir a internacionalização tanto da interface do ambiente, quanto a própria linguagem de programação. Atualmente, o RoboMind está disponível nos idiomas árabe, alemão, chinês, francês, holandês, inglês, português do Brasil e sueco (Benitti et al, 2009).

Uma limitação do RoboMind é que não é possível abordar todos os conteúdos sobre a lógica de programação, mesmo que a linguagem utilizada consiga utilizar vários conceitos importantes como estruturas de decisão e repetição. Além disso o RoboMind não apresenta explicitamente os conceitos de variáveis e constantes, que são importantes para um melhor entendimento da lógica de Computação (Silva et al, 2014).

3.3 App Inventor

O App Inventor é um ambiente visual de programação em blocos, que permite a criação de aplicativos para dispositivos móveis *Android* de uma forma simples, principalmente se comparada às linguagens de programação tradicionais. Seu ambiente gráfico possibilita o ensino de conceitos de lógica de programação de uma maneira atraente e motivadora para estudantes do ensino médio e superior (Gomes e Melo, 2013).

O App Inventor foi desenvolvido conjuntamente pelo Google e MIT. O seu diferencial baseia-se na possibilidade dos usuários criarem aplicações que incorporem serviços baseados na web, interação com redes sociais, leitura de códigos de barra, interação com sensores de orientação e geolocalização, e de funcionalidades como text-to-speech e reconhecimento de fala (MIT, 2016).

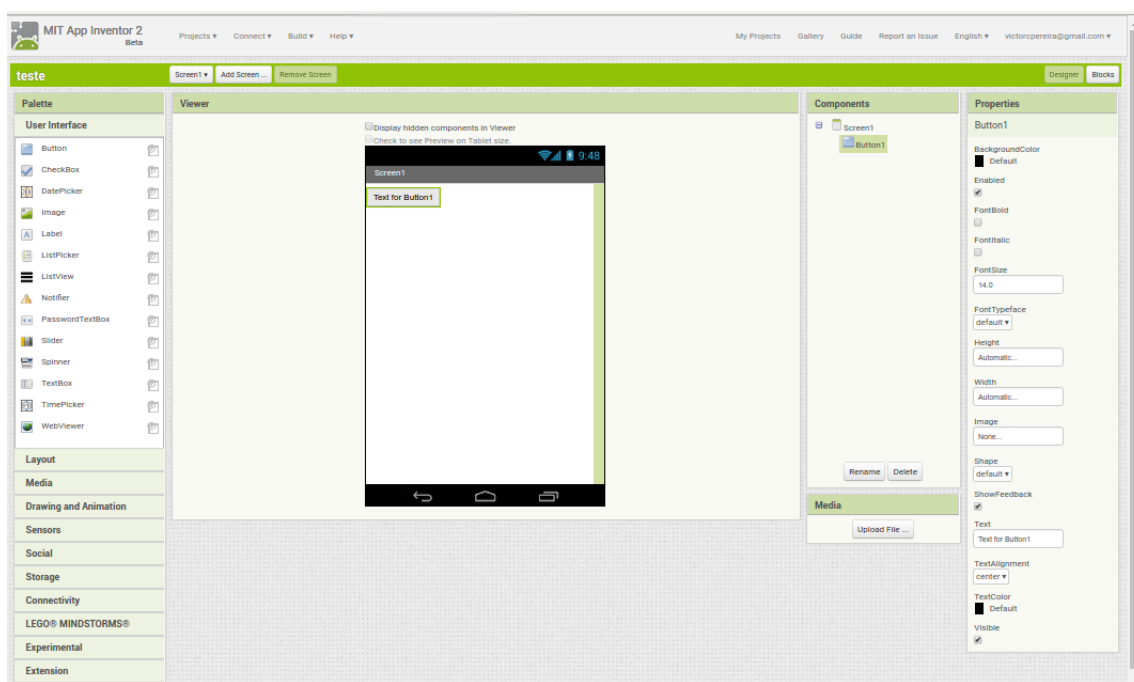


Figura 3.3: App Inventor Designer

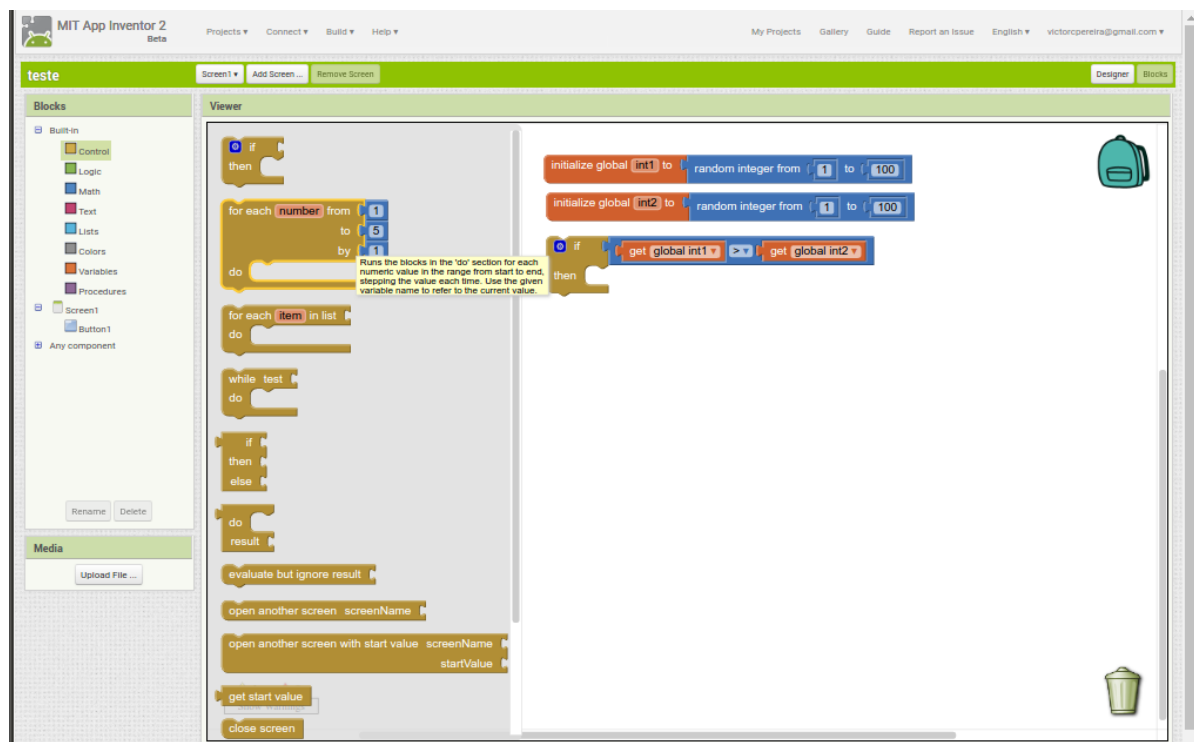


Figura 3.4: App Inventor: Janela Blocks Editor

Inspirado em linguagens como o Scratch, o App Inventor apresenta um ambiente de aprendizagem que permite aos estudantes criarem aplicações à medida que descobrem e exercitam sua criatividade, tornando o aprendizado mais lúdico. O desenvolvimento de aplicativos utilizando o App Inventor é bastante intuitivo e não exige conhecimento prévio avançado em programação. Além disso, os aplicativos criados podem ser utilizados em qualquer dispositivos com a plataforma Android. O App Inventor é baseado em componentes, os quais são manipulados por eventos, métodos e propriedades. Por exemplo, um *botão* é considerado um componente, que tem a propriedade *cor*, que dispara o método *trocarCor()* quando o evento **click** acontece.

O desenvolvimento de uma aplicação nesta ferramenta é realizado através de duas janelas: App Inventor Designer e Blocks Editor. A janela App Inventor Designer permite criar visualmente a interface do usuário, ao clicar e arrastar os componentes da Palette, tais como botões, caixas de texto, figuras, animações, sons, entre outros, para o Viewer. A janela Blocks Editor por sua vez, permite controlar o comportamento dos componentes definidos na App Inventor Designer. Nesta janela, o usuário encontra blocos conectáveis, que podem ser eventos ou métodos, em uma interface do tipo arrastar e soltar, de forma

muito similar ao Scratch.

3.4 SuperLogo

A linguagem Logo foi desenvolvida nos EUA, no MIT na década de 60. Essa linguagem é utilizada no ambiente SuperLogo. O SuperLogo tem como premissa básica despertar o interesse do aluno à construção de conceitos geométricos (Motta, 2008).

Ao iniciar o programa aparecerão duas janelas: a Janela Gráfica (Figura 3.5) e a Janela de Comandos (Figura 3.6). Na janela gráfica encontra-se a figura da tartaruga, que representa o cursor gráfico. Através da execução de comandos, movimenta-se o cursor resultando na construção de desenhos. Na janela de comandos, o usuário digita os comandos a serem executadas pela tartaruga e aciona os botões do ambiente.

Os comandos básicos movimentam a tartaruga na janela gráfica, ou seja, anda-se para frente ou para trás, indicando o valor do deslocamento ou gira-se a tartaruga para a direita ou para esquerda indicando o valor, em graus, do giro. (Motta, 2009).

Tabela 3.2: Comandos básicos para movimentação da tartaruga no SuperLogo

Comando	Descrição
para frente n	Desloca n passos para frente
para trás n	Desloca n passos para trás
para esquerda n	Gira n graus no sentido antihorário
para direita n	Gira n graus no sentido horário

Além dos comandos básicos, existem diversos outros comandos no SuperLogo, tais como: desabilitar o lápis da tartaruga, apagar segmentos, habilitar o lápis, trazer a tartaruga para o centro, desenhar em três dimensões, escrever na tela, efetuar operações aritméticas e lógicas, traçar círculos e arcos, colorir desenhos, etc (Motta, 2008).

A linguagem Logo permite criar estruturas de repetição através do comando *repita n [comandos]*. Para desenhar um quadrado, por exemplo, basta utiliza o comando “repita 4 [para frente 100 para direita 90]”.

Um dos principais recursos oferecido pelo SuperLogo é a criação de procedimentos. O procedimento é um conjunto de comandos ensinados ao SuperLogo que recebe um nome e se propõe a executar determinada ação. Por exemplo, pode-se ensinar um procedimento chamado quadrado. Sendo assim, sempre quando for necessário desenhar

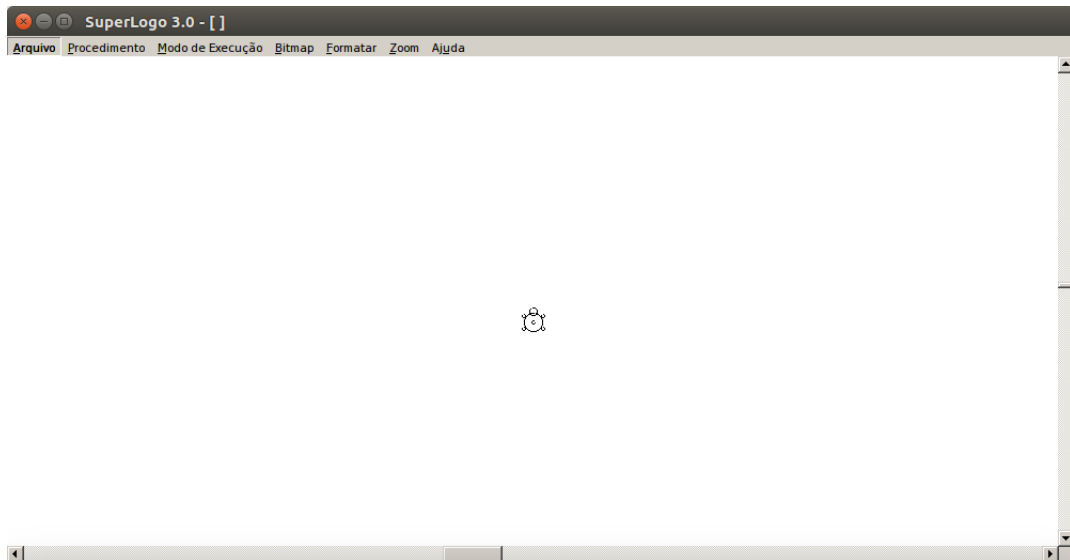


Figura 3.5: SuperLogo - Janela Gráfica



Figura 3.6: SuperLogo - Janela de Comandos

um quadrado, basta executar o este procedimento. Também é possível indicar parâmetros para um determinado procedimento. Dessa forma, indicando o parâmetro “tamanho” no procedimento “quadrado”, pode-se desenhar quadrados de vários tamanhos, de acordo com o parâmetro indicado.

Dentre as limitações do SuperLogo destaca-se a maneira na qual os seus conceitos de lógica são apresentados aos usuários. O ambiente deixa a desejar no aspecto lúdico porque a interface é pouco atraente. Geralmente, os alunos preferem ambientes que possibilitem uma interação lúdica. Como pontos positivos destacam-se a flexibilidade e a simplicidade da linguagem, além de apresentar uma enorme quantidade de comandos, possibilitando o desenvolvimento de vários objetos (Silva et al, 2014).

4 Considerações finais

A Computação é uma área de conhecimento que envolve grande parte das atividades humanas atualmente. O contato com a Computação permite o desenvolvimento de diversas capacidades úteis para qualquer atividade profissional, como capacidade de resolver problemas, de simplificar e interpretar, de liderar, trabalhar em equipe e de fluência no uso de tecnologias para quaisquer áreas do conhecimento. Dessa forma, o Pensamento Computacional torna-se necessário, assim como ler e escrever, para o pleno exercício da cidadania na sociedade atual.

No Brasil, a Sociedade Brasileira de Computação (SBC), através da sua Comissão de Especialistas de Ensino de Computação e Informática e do GT3 (Grupo de Trabalho de Licenciatura em Computação), tem incentivado as discussões sobre a importância de introduzir o Pensamento Computacional no ensino básico. Entretanto, em grande parte das escolas brasileiras os conceitos de Computação e Informática se confundem. O aluno aprende habilidades relacionadas à Informática, como navegar na internet, enviar e-mail, ou operar aplicativos de escritório. Todavia, o Pensamento Computacional consiste em saber utilizar o computador como um instrumento de aumento do poder cognitivo, para aumentar a produtividade, inventividade e criatividade.

Uma das metodologias mais aceitas para o ensino do Pensamento Computacional é a “Computação Desplugada”. Essa metodologia consiste em ensinar Computação sem a utilização de computadores através de atividades lúdicas que estimulam a criatividade dos alunos.

Outra metodologia adotada em várias instituições é o ensino de programação buscando desenvolver o raciocínio lógico dos alunos. Entretanto, aprender a programar é uma tarefa desafiadora para alunos iniciantes. A sala de aula é um ambiente complexo, onde muitos processos e ações estão acontecendo simultaneamente, e que é influenciado pelo estilo de aprendizagem dos alunos. Dessa forma, vários desafios são enfrentados no processo de ensino e aprendizagem da programação.

Iniciar estudantes em programação deve ser um processo que transfira emoções

positivas para o aprendiz, tornando-os capazes de construir a autossuficiência ideal para lidar com os diferentes problemas que surgirão quando estiverem desenvolvendo algoritmos. Este é um aspecto de extrema importância, principalmente, quando se trata de ensinar aos jovens as habilidades de programação. Assim, a presença de bons programadores não é suficiente para despertar o interesse de novatos em programação. É necessário entender como utilizar mecanismos que possam engajar os alunos para que o fluxo de aprendizagem seja mantido e eles possam enxergar o potencial das tecnologias a partir de uma nova perspectiva. E mais, que possam se enxergar capazes de resolver novos problemas a partir da fluência em tecnologias.

Dessa forma, é preciso realizar atividades que mantenham os estudantes engajados e motivados para que as dificuldades sejam superadas e eles continuem interessados em aprender. Sendo assim, a utilização apropriada de uma linguagem de programação pode contribuir para eliminar essas barreiras, mantendo o esforço do aluno concentrado na construção dos algoritmos.

Nesse contexto, os ambientes visuais de programação se tornam um grande facilitador no processo de ensino e aprendizagem da programação. Esses ambientes possuem uma interface amigável e de fácil manipulação, favorecendo o entendimento dos conceitos discutidos durante as aulas. Além disso, a criatividade dos estudantes é explorada durante as tarefas que envolvem o uso dos ambientes de programação em questão.

As ferramentas analisadas nesse trabalho, apresentam grande potencial para facilitar o processo de ensino do Pensamento Computacional, de forma lúdica e dinâmica. A utilização dessas ferramentas no âmbito escolar propõem mais que o ensino dos conceitos de Computação. Através delas, a criatividade e raciocínio lógico dos alunos são estimulados. Além disso, o aluno desenvolve a capacidade de resolução de problemas habituais, construindo assim o Pensamento Computacional.

A escolha de uma das ferramentas para o apoio ao ensino da programação deve levar em consideração o contexto onde ela será inserida. Nesse sentido, é preciso analisar a faixa etária dos alunos, o grau de conhecimento sobre computação, a possibilidade de promover o ensino da Computação de modo interdisciplinar entre outros fatores. O Scratch, por exemplo, apresenta um ambiente mais lúdico, sendo possível criar histórias onde o

aluno pode interagir com o programa que está sendo desenvolvido, definindo caminhos diferentes para a história de acordo com suas ações. Assim como o Scratch, o RoboMind também apresenta um ambiente lúdico, porém os algoritmos que podem ser desenvolvidos são mais limitados, não possibilitando a interação entre o programa e o mundo real. Já o App Inventor apresenta uma linguagem muito parecida com a linguagem do Scratch, no entanto o grande diferencial é a possibilidade de desenvolver aplicativos para o Sistema Operacional *Android*. Por fim, o SuperLogo apresenta um ambiente muito simples, entretanto permite a sua utilização para estimular os alunos durante aulas de geometria, promovendo assim, um processo de ensino e aprendizagem interdisciplinar.

Acredita-se que o ensino da Computação é necessário para todas as pessoas que vivem na sociedade atual. Através do uso de alguma das ferramentas analisadas é possível ensinar e aprender os conceitos de lógica para programação de forma lúdica, e, através da aprendizagem desses conceitos, os alunos conseguem desenvolver o Pensamento Computacional, favorecendo a preparação deles para a sociedade e o mercado de trabalho. Considera-se importante também a influência que o ensino da Computação pode exercer em alunos do ensino básico na escolha de suas carreiras, despertando o interesse em cursos de nível superior de Computação.

Não faz parte do escopo desse trabalho, apresentar todas as funcionalidades das ferramentas, ou comparar e determinar qual a melhor ferramenta, deixando assim uma lacuna para possíveis trabalhos futuros.

Bibliografia

- Andrade, D.; Carvalho, T.; Silveira, J.; Cavalheiro, S.; Foss, L.; Fleischmann, A. M.; Aguiar, M. ; Reiser, R. Proposta de atividades para o desenvolvimento do pensamento computacional no ensino fundamental. **Anais do Workshop de Informática na Escola**, v.1, n.1, p. 169, 2013.
- Aureliano, V. C. O.; Tedesco, P. C. d. A. R. **Avaliando o uso do scratch como abordagem alternativa para o processo de ensino-aprendizagem de programação**. In: CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, volume 32, p. 1–10, 2012.
- Barcelos, T. S.; Silveira, I. F. **Pensamento computacional e educação matemática: Relações para o ensino de computação na educação básica**. In: XX Workshop sobre Educação em Computação, 2012.
- Bell, T.; Witten, I. H.; Fellows, M.; Adams, R. ; McKenzie, J. **Ensinando Ciência da Computação sem o uso do computador**. Computer Science Unplugged ORG, 2011.
- Benitti, F. B. V.; Vahldick, A.; Urban, D. L.; Krueger, M. L. ; Halma, A. **Experimentação com robótica educativa no ensino médio: ambiente, atividades e resultados**. In: Anais do Workshop de Informática na Escola, volume 1, p. 1811–1820, 2009.
- Blikstein, P. **O pensamento computacional e a reinvenção do computador na educação**, 2008. Disponível em: http://www.blikstein.com/paulo/documents/online/ol_pensamento_computacional.html - Acessado em Abril de 2016.
- CSTA; ISTE ; NSF. **Computational thinking teacher resources**, 2011. Disponível em: http://www.csta.acm.org/Curriculum/sub/CurrFiles/472.11CTTeacherResources_2ed-SP-vF.pdf - Acessado em Abril de 2016.
- Carnegie Mellon. **Center for computational thinking**, 2013. Disponível em <http://www.cs.cmu.edu/CompThink/> - Acessado em Abril de 2016.
- França, R. S.; dos Santos Ferreira, V. A.; de Almeida, L. C. F. ; do Amaral, H. J. C. A disseminação do pensamento computacional na educação básica: lições aprendidas com experiências de licenciandos em computação. **Anais do Workshop de Informática na Escola, Brasília–DF**, 2014.
- França, R. S.; Tedesco, P. Explorando o pensamento computacional no ensino médio: do design à avaliação de jogos digitais. **Researchgate**, 2015.
- Gal-Ezer, J.; Beerli, C.; Harel, D. ; Yehudai, A. A high school program in computer science. **Computer**, v.10, p. 73–80, 1995.
- Gil, A. C. Como elaborar projetos de pesquisa. **São Paulo**, v.5, p. 61, 2002.

- Gomes, T. C.; de Melo, J. C. **App inventor for android: Uma nova possibilidade para o ensino de lógica de programação**. In: Anais dos Workshops do Congresso Brasileiro de Informática na Educação, volume 2, 2013.
- Google. **Center exploring computational thinking**, 2016. Disponível em: <http://www.google.com/edu/computational-thinking/> - Acessado em Abril de 2016.
- Isbell, C. L.; Stein, L. A.; Cutler, R.; Forbes, J.; Fraser, L.; Impagliazzo, J.; Proulx, V.; Russ, S.; Thomas, R. ; Xu, Y. (re) defining computing curricula by (re) defining computing. **ACM SIGCSE Bulletin**, v.41, n.4, p. 195–207, 2010.
- MEO. **Currículo para o ensino de ciência da computação nas escolas**, 2008. Disponível em: www.edu.gov.on.ca - Acessado em Maio de 2016.
- MIT. **App inventor**, 2016. Disponível em: www.appinventor.mit.edu - Acessado em Junho de 2016.
- MOTTA, M. S. **Contribuições do SuperLogo ao Ensino de Geometria do Sétimo Ano da Educação Básica**. 2008. Tese de Doutorado - Pontifícia Universidade Católica de Minas Gerais.
- Motta, M. S. Contribuições do superlogo ao ensino de geometria. **Colabor@-A Revista Digital da CVA-RICESU**, v.6, n.21, 2010.
- Nunes, D. J. Ciência da computação na educação básica. **Jornal da Ciência**, v.9, 2011.
- PUC-Rio. **Scalable game design brasil**, 2016. Disponível em: <http://www.serg.inf.puc-rio.br/wiki/index.php> - Acessado em Abril de 2016.
- Scaico, P. D.; Corlett, E. F.; Paiva, L. F.; Raposo, E. H. S. ; Alencar, Y. **Relato da utilização de uma metodologia de trabalho para o ensino de ciência da computação no ensino médio**. In: Anais do Workshop de Informática na Escola, volume 1, 2012a.
- Scaico, P. D.; Henrique, M. S.; Cunha, F. O. M. ; de Alencar, Y. M. Um relato de experiências de estagiários da licenciatura em computação com o ensino de computação para crianças. **RENOTE**, v.10, n.3, 2012b.
- Scaico, P. D.; de Lima, A. A.; Azevedo, S.; da Silva, J. B. B.; Raposo, E. H.; Alencar, Y.; Mendes, J. P.; Scaico, A. ; others. Ensino de programação no ensino médio: Uma abordagem orientada ao design com a linguagem scratch. **Revista Brasileira de Informática na Educação**, v.21, n.02, p. 92, 2013.
- Silva, E. G.; Barbosa, A. F.; Neto, S. R.; HO, R. ; Lopes, A. N. **Análise de ferramentas para o ensino de computação na educação básica**. In: XXXIV Congresso da Sociedade Brasileira de Computação, 2014.
- Sousa, R. V.; Barreto, L. P.; Andrade, A. ; Abdalla, D. Ensinando e aprendendo conceitos sobre ciência da computação sem o uso do computador: Computação unplugged!!! **Jornada de Atualização em Informática na Educação**, v.1, n.1, 2011.
- Wing, J. M. Computational thinking. **Communications of the ACM**, v.49, n.3, p. 33–35, 2006.