

UNIVERSIDADE ESTADUAL DE  
MATO GROSSO DO SUL

Computação, Licenciatura

Elison Fonseca Novaes

Análise de Ferramentas no Apoio ao Pensamento  
Computacional no Ensino Básico

UEMS  
Novembro/2018

Elison Fonseca Novaes

# Análise de Ferramentas no Apoio ao Pensamento Computacional no Ensino Básico

Orientador: Prof. Me. Jorge Marques Prates

Monografia apresentada a Universidade Estadual de Mato Grosso do Sul – UEMS, para o Trabalho de Conclusão de Curso, como parte dos requisitos para obtenção do título de Licenciando em Computação.

UEMS  
Novembro/2018

N815a Novaes, Elisa Fonseca  
Análise de ferramentas no apoio ao pensamento computacional  
no ensino básico/Elisa Fonseca Novaes. Nova Andradina, MS: UEMS, 2018  
57p. ;30cm.

Monografia (Graduação) - Computação, Licenciatura - Universidade  
Estadual de Mato Grosso do Sul, 2018

Orientador: Prof. Me. Jorge Marques Prates.

1. Análise de ferramentas 2. Ensino de programação 3. Pensamento  
I. Título.

CDD 23.ed. 005.713

Elison Fonseca Novaes

## Análise de Ferramentas no Apoio ao Pensamento Computacional no Ensino Básico

Monografia apresentada a Universidade Estadual de Mato Grosso do Sul – UEMS, para o Trabalho de Conclusão de Curso, como parte dos requisitos para obtenção do título de Licenciando em Computação.

Membros da banca:

---

Prof. Me. Jorge Marques Prates (Orientador)  
UEMS – Nova Andradina

---

Prof. Esp. André de Castro Garcia  
UEMS – Nova Andradina

---

Prof. Esp. Letícia de Godoy Enz  
UEMS – Nova Andradina

*Dedico este trabalho aos meus pais que não mediram esforços para me ajudar em tudo que precisei.*

# *Agradecimentos*

---

---

Agradeço a Deus que sempre esteve comigo, sempre soube de meus medos e mesmo assim me fortaleceu para que conseguisse superá-los.

Aos meus pais e familiares, o meu muito obrigado é pequeno diante da grandeza do que fizeram por mim. Tenho certeza que dei o primeiro passo para retribuir toda a dedicação. São a vocês, meus pais, do fundo do coração que agradeço, por ter me ensinado que vale acreditar em nossos sonhos e lutar por ele.

Aos professores agradeço todos os ensinamentos, sem vocês não teria conseguido, tudo vai fazer falta, suas broncas, suas puxadas de orelhas e até mesmo as risadas. Obrigado por tudo que me ensinaram.

*"O prazer no trabalho aperfeiçoa a obra."  
Aristóteles*

# *Resumo*

---

---

O avanço das tecnologias tem trazido diversas facilidades, principalmente no processo de ensino e aprendizagem de algumas disciplinas, sendo a disciplina de programação uma delas. Atualmente existem diversas ferramentas e novos métodos para o ensino desse conteúdo, com o objetivo de facilitar o aprendizado dos alunos. Com isso, podemos averiguar se os resultados que essas ferramentas podem trazer para o ensino, realmente auxiliam no aprendizado de programação e os benefícios que o pensamento computacional pode trazer para cada indivíduo. Neste trabalho foram levantadas algumas ferramentas e, dentre elas, foram selecionadas cinco para estudo, sendo elas: Scratch, App Inventor, Alice, Code.org e Blockly Games. Foi realizada uma análise da utilização das mesmas, além de serem levantados métodos de ensino que podem ser desenvolvidos nessas ferramentas. Com isso, foram demonstrados os benefícios e desvantagens que essas ferramentas podem trazer para os alunos, em paralelo com a importância do ensino de programação.

**Palavras-chave:** Ensino de programação, Pensamento computacional, Resolução de problemas, Análise de ferramentas.

# *Abstract*

---

---

The advancement of technologies has brought facilities, especially in the teaching and learning of some courses, such as a programming. Currently there are several tools and new methods for teaching this content, in order to facilitate students' learning. With this research, we can see if the results that these tools can bring to teaching really help in programming learning and the benefits that computational thinking can bring to everyone. In this work some tools were raised and among them, five were selected for this study: Scratch, App Inventor, Alice, Code.org and Blockly Games. An analysis was made of their use, in addition it was raised teaching methods that can be developed in these tools. Finally, we have demonstrated the benefits and disadvantages that these tools can bring to students, in parallel with the importance of teaching programming.

**Keywords:** Programming teaching, Computational thinking, Problem solving, Analysis of tools.

# *Lista de Figuras*

---

---

2.1	Interface inicial do Scratch 1.4. Extraído de UPF (2018) . . . . .	16
2.2	Bloco de comandos do Scratch. Extraído de IFSC. (2014) . . . . .	17
2.3	Interface inicial do Scratch 2.0. Extraído de SACHETTO (2014) . . . . .	18
2.4	Tipos de blocos de comandos do Scratch. Extraído de ARANTES et al. (2015) . . . . .	19
2.5	Interface inicial do App Inventor 2. Extraído de CORDEIRO (2017) . . . . .	20
2.6	<i>Bloco Editor</i> do App Inventor. Extraído de CORDEIRO (2017) . . . . .	21
2.7	Interface principal do Alice. Extraído de Duke (2013) . . . . .	22
2.8	Comandos do ambiente Alice. Extraído de PBWORKS (2012) . . . . .	23
2.9	Mundos do ambiente Alice. Elaborada pelo autor. . . . .	24
2.10	<i>Code Studio</i> do Code.org. Elaborada pelo autor. . . . .	24
2.11	Interface do Blockly Games. Elaborada pelo autor. . . . .	25

# *Lista de Tabelas*

---

---

3.1	Comparativo das ferramentas . . . . .	49
3.2	Vantagens e Desvantagens das Ferramentas . . . . .	50

# Conteúdo

---

---

<b>1</b>	<b>Introdução</b>	<b>11</b>
1.1	Contexto . . . . .	11
1.2	Motivação . . . . .	11
1.3	Objetivos . . . . .	12
1.4	Organização . . . . .	12
<b>2</b>	<b>Ferramentas no Apoio ao Pensamento Computacional</b>	<b>13</b>
2.1	Considerações Iniciais . . . . .	13
2.2	Pensamento Computacional . . . . .	14
2.3	Scratch . . . . .	15
2.4	App Inventor . . . . .	19
2.5	Alice . . . . .	21
2.6	Code.org . . . . .	22
2.7	Blockly Games . . . . .	25
<b>3</b>	<b>Análise das Ferramentas no Apoio ao Pensamento Computacional</b>	<b>27</b>
3.1	Introdução . . . . .	27
3.2	Scratch . . . . .	28
3.3	App Inventor . . . . .	33
3.4	Alice . . . . .	39
3.5	Code.org . . . . .	45
3.6	Blockly Games . . . . .	46
3.7	Comparação . . . . .	48
<b>4</b>	<b>Conclusão</b>	<b>51</b>
	<b>Referências Bibliográficas</b>	<b>55</b>
<b>A</b>	<b>Apêndice A</b>	<b>56</b>

---

# *Introdução*

---

## **1.1 Contexto**

O pensamento computacional pode trazer inúmeros benefícios para os alunos. Um dos seus principais focos é a solução de problemas, em que os alunos desenvolvem o raciocínio lógico ao realizar atividades de resolução desses problemas. A sua aplicação traz resultados não só para os alunos da área da Computação, mas também para os de outras áreas.

Atualmente há uma iniciativa da Sociedade Brasileira de Computação (SBC), para que o ensino de fundamentos da Computação seja inserido aos alunos do Ensino Básico, tentando fazer com que a Computação seja reconhecida com uma área do conhecimento. Assim, podendo demonstrar a importância do ensino de programação nas escolas ([WEISSHAHN et al., 2015](#)).

Além disso, a programação é um meio de ensinar os conceitos do pensamento computacional, podendo trazer diversos benefícios para os alunos, não só para quem é da área da Computação. Entretanto, o aprendizado de programação pode ser um tanto quanto complexo. Com o intuito de facilitar esse aprendizado, existem ferramentas de apoio que podem facilitar esse aprendizado.

## **1.2 Motivação**

Atualmente as pessoas estão totalmente ligadas às tecnologias, e isso pode ser aproveitado no âmbito escolar, visto que através delas os alunos podem ter acesso a diversos conteúdos de forma mais rápida do que tinham alguns anos atrás. Na atualidade existem diversas ferramentas que tem como intuito auxiliar no ensino de programação, facilitando o processo de ensino e aprendizagem dessa disciplina.

## 1.3 Objetivos

Esse monografia tem como objetivo investigar e analisar ferramentas no apoio ao ensino do pensamento computacional e programação aos alunos das escolas de ensino básico. Também é fornecido um comparativo entre as atuais ferramentas.

## 1.4 Organização

A organização desse trabalho segue do modo em que é descrito a seguir. No Capítulo 2 apresentam-se as funcionalidades das ferramentas por meio de uma descrição dos componentes que estão presentes nelas. O capítulo 3 apresenta uma análise da utilização e um comparativo entre essas ferramentas. Por fim, no Capítulo 4, apresentam-se as considerações finais desta monografia.

---

# *Ferramentas no Apoio ao Pensamento Computacional*

---

## **2.1 Considerações Iniciais**

Logo no início da inclusão dos computadores nas escolas, na década de 80, houve uma grande euforia para ensinar programação às crianças. Isso foi feito utilizando a linguagem Logo, que colocava o computador como um aliado na construção do conhecimento. Essa fase durou pouco, pois muitas das escolas pararam de utilizar o ensino de programação, passando então a utilizar o computador para realizar tarefas diferentes, como: criar textos e navegar na internet, o que ainda ocorre muito nos dias de hoje (ARANTES et al., 2015).

Atualmente a tecnologia evoluiu bastante, o que mudou também o seu tamanho, passando a ser portátil e sendo itens que fazem parte do nosso cotidiano, um exemplo disso são os *smartphones* que praticamente são utilizados pela maioria das pessoas (ARANTES et al., 2015).

Os jovens de hoje em dia são considerados “nativos digitais” devido ao fato de serem habituados com as tecnologias que existem atualmente. Utilizam-se dessas tecnologias para mandar mensagens, usar jogos online e navegar na internet. Apesar de ter uma grande interação com essas tecnologias, nem todos são capazes de criar seus próprios jogos ou animações, revelando que sabem “ler”, mas não sabem “escrever”. Ou seja, são capazes de utilizar, mas não de criar. Para que se possa passar a criar é necessário que ele aprenda a programar, essa habilidade traz benefício como expandir a capacidade de criação (ARANTES et al., 2015).

As ferramentas que serão descritas nas próximas seções têm o intuito de auxiliar o ensino de programação. Mais especificamente serão analisadas ferramentas que possibilitam o desenvolvimento de códigos de maneira lúdica, assim como a lógica dos alunos. Essas ferramentas foram obtidas através de pesquisas realizadas em estudos e artigos. Foram selecionadas cinco fer-

ramentas, que podem ser utilizadas pelos alunos do ensino básico, que utilizavam do conceito de montagem de bloco, além de ser ferramentas atuais. A listagem completa das ferramentas se encontra no Apêndice A.

## 2.2 Pensamento Computacional

O pensamento computacional pode ser entendido como um conjunto de técnicas que utilizam conceitos de Computação para auxiliar na solução de problemas, tanto do dia a dia quanto de problemas mais complexos. O pensamento computacional é considerado como uma habilidade fundamental para todos, não só para estudantes da computação, pois abrange várias áreas do conhecimento (WEISSHAHN et al., 2015).

Tendo isso em vista, é importante que as pessoas tenham contato com esse assunto, facilitando na resolução de problemas que passamos no dia a dia, pois o pensamento computacional envolve tanto formulação de problemas quanto a sua solução. “O pensamento computacional é o processo do pensamento envolvido na formulação de problemas e suas soluções para que as soluções sejam representadas em uma forma que possa ser efetivamente realizada por um agente de processamento de informações” (WING, 2010).

A ideia do pensamento computacional vem através das “grandes ideias” da Computação. A computação é uma atividade humana criativa, tendo em vista que para solucionar problemas computacionais temos que buscar as melhores maneiras para as suas soluções. Para isso, são necessárias informações e detalhes para se concentrar na compreensão e na solução de problemas, dando dados e informações que facilitam a criação de conhecimento (GROVER e PEA, 2013). Algoritmos são ferramentas para desenvolver e expressar soluções para os problemas computacionais, já que a programação é um processo criativo que produz artefatos computacionais. Isso permite a inovação de outras áreas do conhecimento humano, sendo um meio para explorar outros domínios, como Matemática ou Ciências (GROVER e PEA, 2013).

A programação é considerada uma das ferramentas fundamentais para apoiar o ensino do pensamento computacional, pois a criação de algoritmos é uma atividade que força o pensamento, pois encontra uma grande quantidade de problemas para ser solucionado. Sendo assim, deve ser pensado da sua concepção até a sua solução.

O pensamento computacional é baseado nas técnicas da computação, que busca pegar problemas mais complexos e dividir em problemas menos complexos, pegando esses problemas menores e comparando entre si, e identificado problemas que terão as soluções parecidas, focando em detalhes mais importantes, diminuindo assim a quantidade de informações sobre esse determinado problema (BRACKMANN et al., 2016).

Os algoritmos são considerados como regras ou etapas para resolver esse problema computacionais, que é compreendido pelos sistemas computacionais, que pode ser usado por todos os alunos mesmo que não seja da área da computação. Que é considerada para alguns países uma maneira de se desenvolver, pois pode gerar benefícios sociais, assim como também na economia. Já que formar novos profissionais mais capacitados e para a educação, traz à tona que o

mundo está ligado à tecnologia digital.

No Brasil o pensamento computacional ainda não é reconhecido como em outros países, mesmo com iniciativa da Sociedade Brasileira de Computação (SBC) de ser inserida na Base Nacional Comum Curricular (BNCC), alguma disciplina ou área focada no ensino do Pensamento Computacional. Mesmo não sendo acrescentada nem uma área, a tecnologia digital vem integrada com outras disciplinas, ou seja, ainda não é reconhecida como uma área do conhecimento, apesar da intenção da SBC de que a Computação seja reconhecida como uma área do conhecimento (WEISSHAHN et al., 2015).

Apesar de não ser reconhecida como uma área do conhecimento, há vários projetos de pesquisa sobre esse assunto. Tratam-se de trabalhos apresentados em *Workshops* de Educação, como por exemplo, o Congresso Nacional Brasileiro de Computação na Educação (CBIE) e o Congresso Anual da Sociedade Brasileira de Computação (CSBC) (WEISSHAHN et al., 2015).

Através da gamificação podem ser ensinados os conceitos de programação, que se trata do uso de jogos para o ensino. Nesses jogos os alunos têm tarefas para que sejam cumpridas, cada vez que eles cumprem a tarefa ganham ponto, e se não conseguirem perdem pontos. Assim, usam os mesmos conceitos dos jogos que não são educativos, envolvendo os alunos no aprendizado sobre programação ou nos conceitos do pensamento computacional (LEE et al., 2011).

O pensamento computacional pode ser realizado mesmo sem ter uma infraestrutura adequada, com salas de tecnologias ou sem acesso à internet, podendo ser realizado através da computação desplugada, que se trata do ensino dos conceitos do pensamento computacional sem o uso do computador. Em muitos países que usaram essas técnicas já mostraram um avanço significativo dos alunos mesmo com pouco contato com as técnicas do pensamento computacional (WEISSHAHN et al., 2015).

Várias dessas experiências computacionais introdutórias usam a progressão "use - modify - create" de três estágios para ajudar o aluno a passar do usuário para o modificador e, posteriormente, para o criador de artefatos computacionais (LEE et al., 2011). Com essas experiências os alunos não ficam presos a usar um determinado programa, podendo também modificar ou até mesmo criar o seu próprio programa, de acordo com o que ele necessita. A seguir são apresentadas algumas ferramentas que apoiam o desenvolvimento do pensamento computacional e no ensino de programação.

## 2.3 Scratch

O Scratch é um software livre que pode ser utilizado por crianças e jovens, visando auxiliar o aprendizado de programação, ou para outras disciplinas, sendo desenvolvido pela Media Lab do *Massachusetts Institute of Technology* (MIT). Tem como público alvo uma faixa etária entre 8 e 16, mas pode ser utilizado também por adultos (RIBEIRO et al., 2014).

O software utiliza o conceito de montagem de blocos de programação. Não necessita que o aluno tenha conhecimento de alguma linguagem de programação, o que pode facilitar o aprendizado sobre os conceitos básicos de programação, ficando mais fácil o entendimento de alguma

linguagem de programação posteriormente. Além disso, é encontrado em várias línguas, inclusive em português, o que pode facilitar a utilização do software, já que a interação fica mais fácil (RIBEIRO et al., 2014).

Essa ferramenta, criada para o ensino de programação e de outras disciplinas, incentiva o pensamento criativo, o raciocínio e o trabalho em grupo. Utiliza uma linguagem visual, ou seja, não é necessário ficar digitando linhas de códigos. É possível criar jogos e animações, podendo assim ser misturadas várias mídias como imagens, gráficos, sons e músicas (NETO, 2013).

Com o Scratch é possível aproximar o usuário cada vez mais desse ambiente, já que não necessita do aprendizado de uma determinada linguagem de programação. Possibilita também a criação de programa de maneira mais simples, e que permite a visualização do que foi desenvolvido, deixando assim mais fácil de ver possíveis erros que o código pode ter (NETO, 2013).

A interface principal do Scratch conta com um conjunto de blocos que podem ser arrastados, e criando os comandos que serão executados. À esquerda é possível ver um conjunto de comandos separados por categoria, sendo eles: movimento, aparência, som, caneta, controle, sensores, operadores e variáveis. Cada categoria é destacada por cores diferentes, para que seja possível diferenciar e saber à qual categoria cada comando pertence. No meio fica o espaço para arrastar os comandos que o ator irá executar. Na direita, na parte superior, há um espaço que é possível ver o resultado do programa executado, chamado de palco. Por fim, na parte inferior é possível ver os atores, como mostra a Figura 2.1. Dependendo da versão, a tela principal tem alterações, como o local onde está o palco, o conjunto de comandos, os atores e o local onde fica os comandos que o ator irá executar, como mostra a Figura 2.3 (ARANTES et al., 2015).

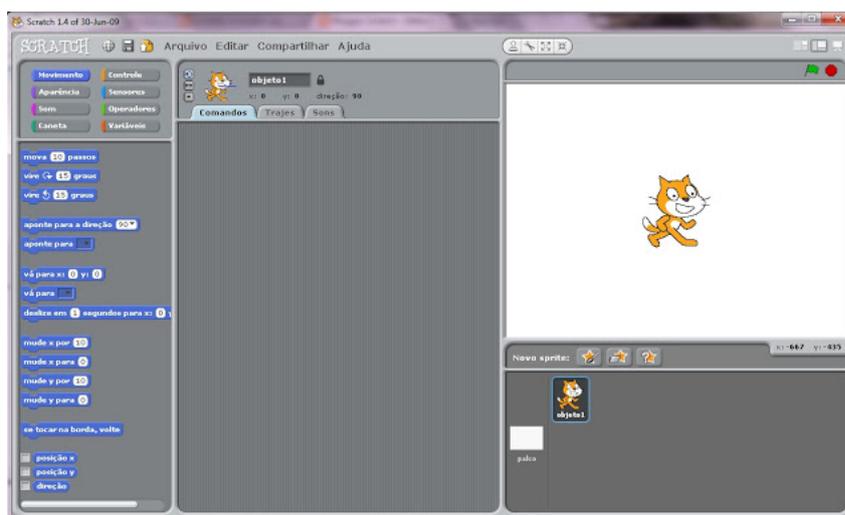


Figura 2.1: Interface inicial do Scratch 1.4. Extraído de UPF (2018)

É possível ver o resultado imediatamente, basta clicar no bloco de código que deseja executar e os dados se tornam visíveis, facilitando a localização do possível erro, pois os blocos de comando ficam destacados por uma borda branca brilhante. Se houver ocorrência de erro (como em uma divisão por zero, por exemplo) o bloco responsável pelo erro fica destacado em

vermelho. A interface é simples, com isso o usuário tem uma boa interação com a ferramenta. Há também painéis para os comandos, deixando os principais mais visíveis. Os comandos ficam divididas em oito categorias, o que evita uma poluição visual causada por uma lista de comandos muito longa. Em cada categoria os comandos mais úteis aparecem primeiro (ARANTES et al., 2015).

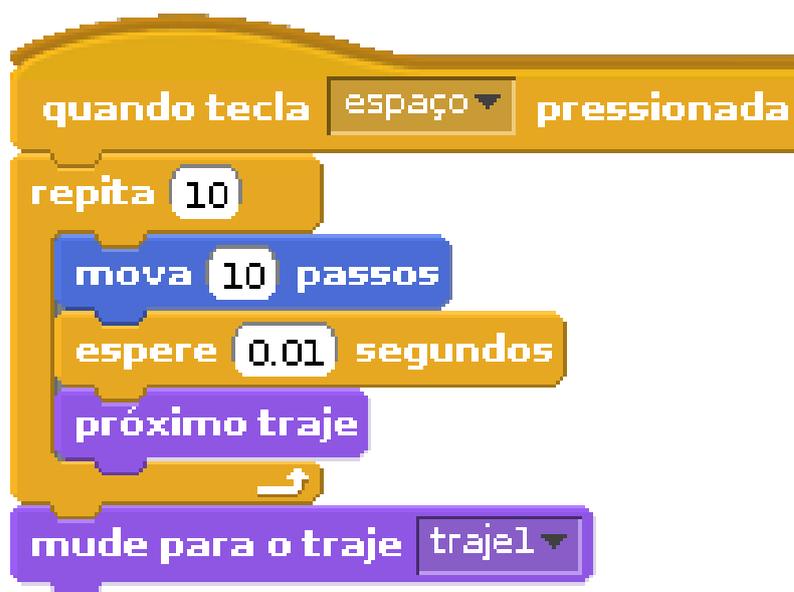


Figura 2.2: Bloco de comandos do Scratch. Extraído de IFSC. (2014)

O Scratch possui um palco, que é um plano de fundo estático onde o personagem executa os comandos que foram programados anteriormente. Podendo ter mais de um personagem por vez, as ações de cada um serão definidas por cada bloco de comando que será atribuído a cada personagem (BRESSAN e AMARAL, 2015).

Os blocos de comando são coloridos, assim como mostra a Figura 2.2, o que pode atrair a atenção das crianças. Esses comandos são montados como se fosse um quebra cabeça, basta arrastar e soltar onde deseja encaixar o comando, formando assim os conjuntos de comandos. Os blocos podem ser alterados a qualquer momento, o que facilita experimentar novas ideias de maneira simples. Podem ser testados vários blocos de comandos ao mesmo tempo. Ao arrastar e construir os blocos de comandos, os alunos aprendem com os erros, analisando e criando hipóteses para que possam corrigir o erro (BRESSAN e AMARAL, 2015).

O Scratch tem códigos prontos, basta o desenvolvedor arrastar o comando que deseja e montar a estrutura do programa que está sendo desenvolvido. Os comandos têm características específicas, que são responsáveis por executar funções específicas. Existem comandos para diferentes funções, como o comando para iniciar, para executar movimentos, além do comando que dá a permissão do programador inserir valores em variáveis. Os scripts do Scratch são as junções dos comandos, que também podem ser chamados de blocos. Os formatos dos blocos



Figura 2.3: Interface inicial do Scratch 2.0. Extraído de [SACHETTO \(2014\)](#)

mostram onde eles podem ser encaixados. Entretanto os comandos não podem ser encaixados em determinada situação, o que pode facilitar o desenvolvimento ([ARANTES et al., 2015](#)).

No Scratch há quatro tipos de blocos, como mostra a Figura 2.4. O primeiro são blocos de comando (Figura 2.4 (a)), esses blocos podem ser unidos para criar uma sequência de comandos. Possuem encaixes na parte superior e um correspondente na parte inferior. O segundo são blocos de função (Figura 2.4 (b)), esses blocos não seguem uma sequência como os demais blocos de comando, são comandos usados como argumentos para outros comandos e reunidos para construir expressões. Podem ter formatos diferentes, dependendo do tipo de valores que retornam: ovais para números e hexágonos para booleanos (operadores lógicos) ([ARANTES et al., 2015](#)).

O terceiro são os blocos de gatilhos (Figura 2.4 (c)), que são blocos de eventos tais como iniciar o programa, executar uma determinada ação ao clicar o mouse ou alguma tecla. E o quarto são os blocos de estruturas de controle (Figura 2.4 (d)), que são blocos de comando com uma ou mais sequências embutidas. O formato desses blocos, os tornam mais fáceis de ser utilizados. Na maioria das linguagens de programação textuais pode ocorrer o esquecimento dos delimitadores e por causa disso acabam ocorrendo erros. Já no Scratch esses blocos são indivisíveis o que evita esses tipos de erros ([ARANTES et al., 2015](#)).

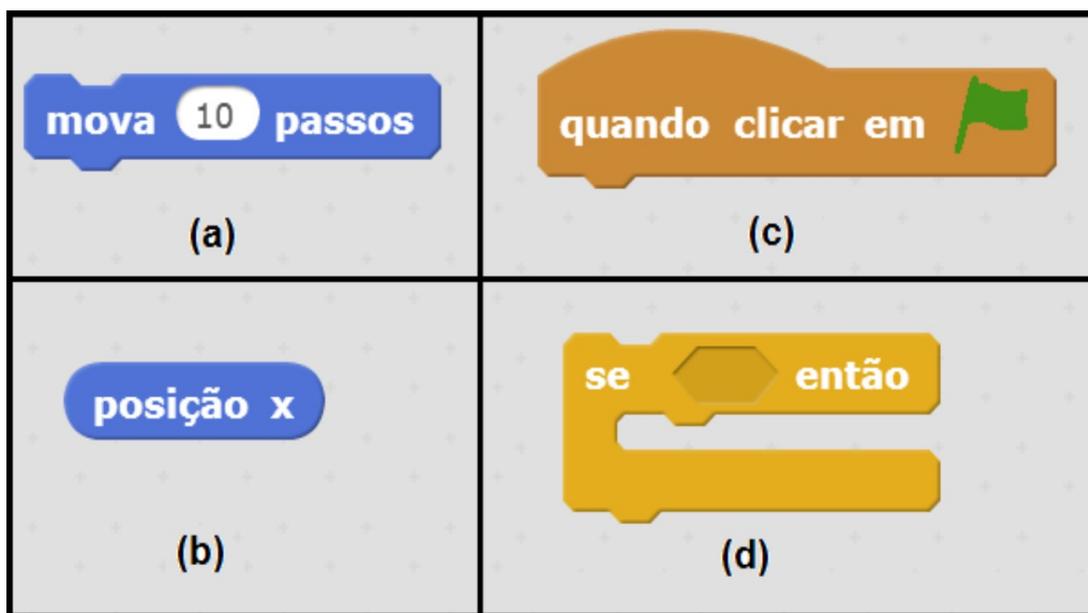


Figura 2.4: Tipos de blocos de comandos do Scratch. Extraído de [ARANTES et al. \(2015\)](#)

## 2.4 App Inventor

O App Inventor é um aplicativo via web que permite desenvolver aplicativos para Android, utilizando blocos de comando. Todos o desenvolvimento é visual ,já que você junta peças como se fosse um quebra cabeça, ou seja monta blocos que especificam como os componentes devem se comportar. Posteriormente ao término do desenvolvimento, é possível executar e testar o aplicativo em seu próprio celular.

O sistema conta com duas seções, o *App Inventor Designer* e o *Bloco Editor*. Na interface inicial encontramos a seção *App Inventor Designer*, que se trata da seção onde será desenvolvida a parte gráfica do seu aplicativo. Ou seja, a interface que é composta pela paleta, visualizador, componentes e propriedades.

Como podemos ver na Figura 2.5, a paleta fica no lado esquerdo com todos os componentes que podem ser utilizados. Mais ao centro temos o visualizador onde podemos organizar os componentes. Um pouco mais ao canto direito temos os componentes que fica em uma lista com tudo o que foi adicionado, e no canto direito fica a propriedade que são as definições de cada componente ([BARBOSA et al., 2015](#)).

A coluna chamada de Paleta (*Palette*) é onde estão disponíveis todos os componentes que podem ser utilizados. Essa coluna é dividida em seções para facilitar a localização dos componentes, que são basicamente botões, imagens e textos. É possível utilizar um componente clicando no que deseja escolher e arrastando para a coluna chamada de Visualizador. Na coluna Visualizador, é possível organizar cada um dos seus objetos conforme ele deve ser. Também, nessa coluna, há um simulador que apresenta uma versão aproximada do programa final.

Na coluna Components ficam armazenadas todos os componentes que foram adicionados. Mesmo que não sejam visíveis na tela, são listados de forma ordenada e acessível. Também é

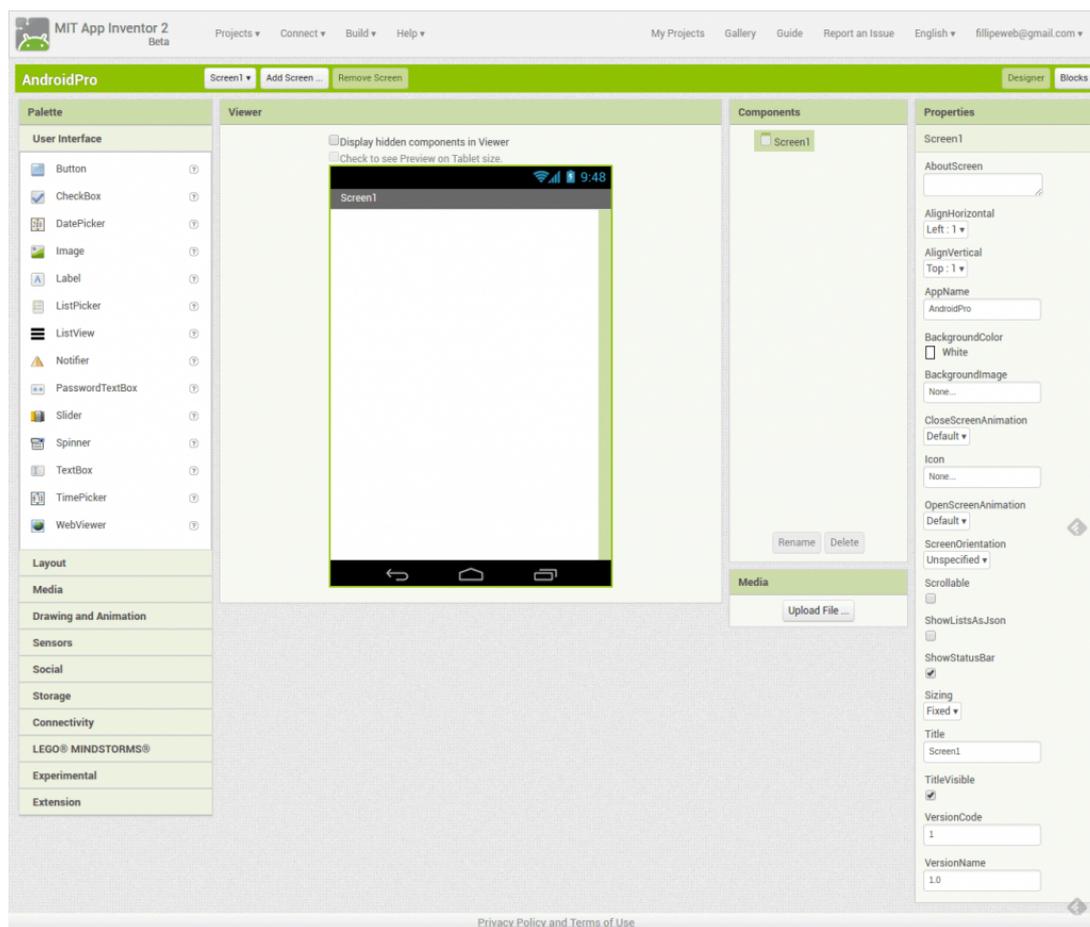


Figura 2.5: Interface inicial do App Inventor 2. Extraído de [CORDEIRO \(2017\)](#)

possível realizar alterações de nome, além de ser possível inserir arquivos de mídia se tiver a necessidade de utilizar no seu projeto.

A coluna Properties é a coluna mais importante do *App Inventor Designer*, já que é onde é possível alterar as definições. Podem ser alterados o tamanho dos textos dos botões, caixa de informações, tamanho das imagens, cores de fundo, largura e a altura do objeto. Essas mudanças são possíveis de ver instantaneamente, permitindo ter uma ótima ideia das mudanças que estão ocorrendo. Já na seção *Bloco Editor*, é onde se definem as ações que a ferramenta deve executar. São associadas a cada componente do aplicativo, que é composto pelo bloco e o visualizador.

Como podemos ver na Figura 2.6, os blocos são divididos em dois: os blocos do componente e os internos. Os blocos do componente correspondem aos componentes escolhidos para os aplicativos e os internos, que são os comandos, estão sempre disponíveis. O visualizador se trata da área de trabalho onde são programados os blocos do aplicativo ([BARBOSA et al., 2015](#)). O *Bloco Editor* tem uma interface simples e intuitiva, que se parece muito como montar um quebra cabeça. Isso facilita a montagem, já que os comandos foram construídos para parecer um quebra cabeça, apenas as funções compatíveis podem ser encaixadas.

Os Blocos de comandos internos são divididos em conjuntos, que são diferenciados por

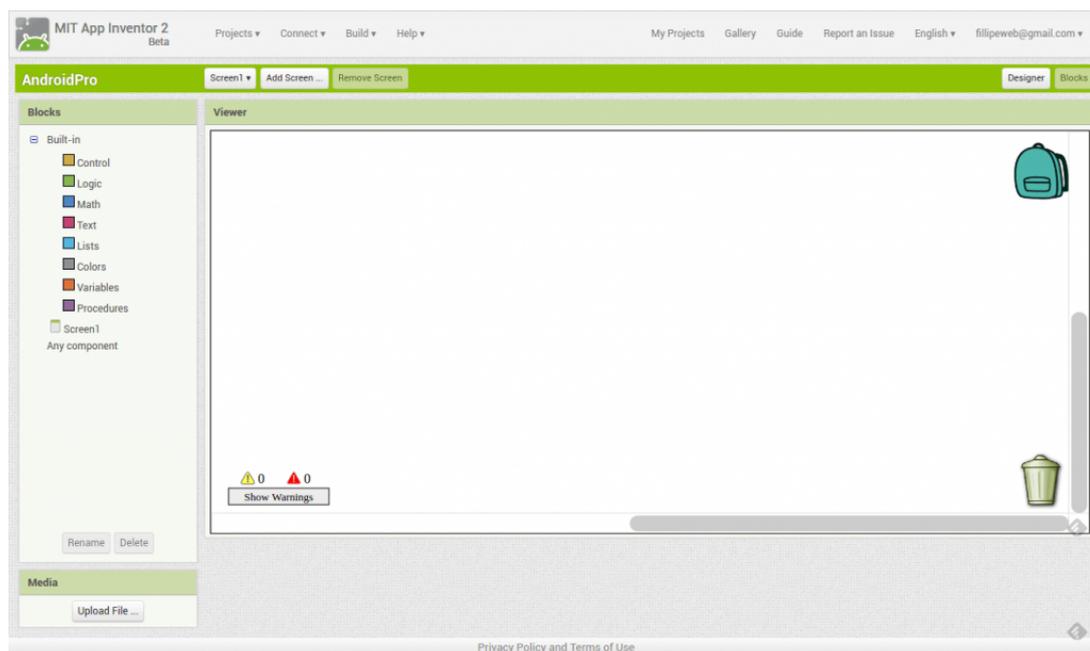


Figura 2.6: *Bloco Editor* do App Inventor. Extraído de [CORDEIRO \(2017\)](#)

cores. Os conjuntos são nomeados em: Controle, Lógica, Matemática, Texto, Lista, Cores, Variáveis e Procedimentos. Então, o conjunto de controle é composto por estrutura de repetição, estrutura de condicional, além de outros blocos de comando. O conjunto de bloco lógico é onde podemos encontrar os blocos que são responsáveis por realizar as ações lógicas do programa. O conjunto matemático contém os blocos que tem funções matemáticas, sendo elas fórmulas matemáticas que já estão prontas.

Nos blocos internos é fácil identificar quais as suas funcionalidades, já que na maioria dos casos os próprios nomes dos conjuntos já demonstram qual a sua funcionalidade utilizada. Os blocos dos componentes são específicos de cada item e que podem ser inseridos na tela inicial. Podem ser blocos de diferentes conjuntos de comandos internos, mas que, em alguns casos, não se repetem nas duas listas, o que evita lista muito extensas.

O App Inventor tem sido utilizado em várias instituições de ensino, sendo utilizado como uma plataforma de ensino de programação para iniciantes e para alunos do Ensino Médio e Fundamental. Atualmente, está sendo usado também para auxiliar nas formações de professores. Essa ferramenta vem mostrando que pode ser muito útil, demonstrando um potencial didático e pedagógico, principalmente para os iniciantes ([GOMES e MELO, 2013](#)).

## 2.5 Alice

Alice é uma ferramenta de desenvolvimento em 3D, que atualmente se encontra na versão 3. Conta com uma interface simples do estilo clique e arraste e é considerada excelente para quem está tendo a primeira experiência com a programação. Sua primeira versão foi desenvolvida na década de 90, que contou com a iniciativa de várias universidades ([GONDIM et al., 2009](#)).

O objetivo da ferramenta é ensinar a base das linguagens de programação. O seu ambiente



Figura 2.7: Interface principal do Alice. Extraído de [Duke \(2013\)](#)

conta com comandos do tipo *If/Else*, *While*, *For*, *Print*, *Up*, *Down*, além de métodos, parâmetros e funções. O aluno pode ver os resultados do que foi desenvolvido aplicando esses comandos nos objetos inseridos no mundo, que realiza a simulação do que foi desenvolvido. A visualização em tempo real auxilia o aluno a assimilar os conceitos de programação ([GONDIM et al., 2009](#)).

O ambiente Alice possui uma série de tutoriais que são iterativos e que possibilitam ao usuário conhecer o ambiente. Também há uma caixa de diálogo que passa as informações do que deve ser feito. No site do Alice podem ser encontrado vários vídeos, matérias e artigos, para que possa haver troca de informação. Entretanto essas matérias estão disponíveis essencialmente em inglês, como nem todos possuem o conhecimento sobre a língua, a utilização desses materiais é limitada. Mesmo que o ambiente seja de interface simples, isso acaba gerando algumas limitações que geram obstáculos na aprendizagem ([GONDIM et al., 2009](#)).

O programa dispõe de um ambiente chamado de mundo onde é possível ver as ações do objeto, que são as interações desenvolvidas pelo usuário. Podemos ver alguns exemplos na Figura 2.9. O Alice permite o aprendizado de maneira divertida, o que ajuda na compreensão de conceitos que são difíceis de se entender e contribui para o desenvolvimento do raciocínio lógico do usuário ([SILVA et al., 2014](#)).

## 2.6 Code.org

O Code.org é uma plataforma de ensino online de ensino de programação. Utiliza de jogos educacionais digitais que trabalham com diversos conceitos relacionado a lógica de programação como: Comandos básicos, princípios de computação gráfica, estruturas de condição, laços de repetição, rastreamento de algoritmos, pensamento computacional, funções e contadores ([MARTINS et al., 2016](#)).

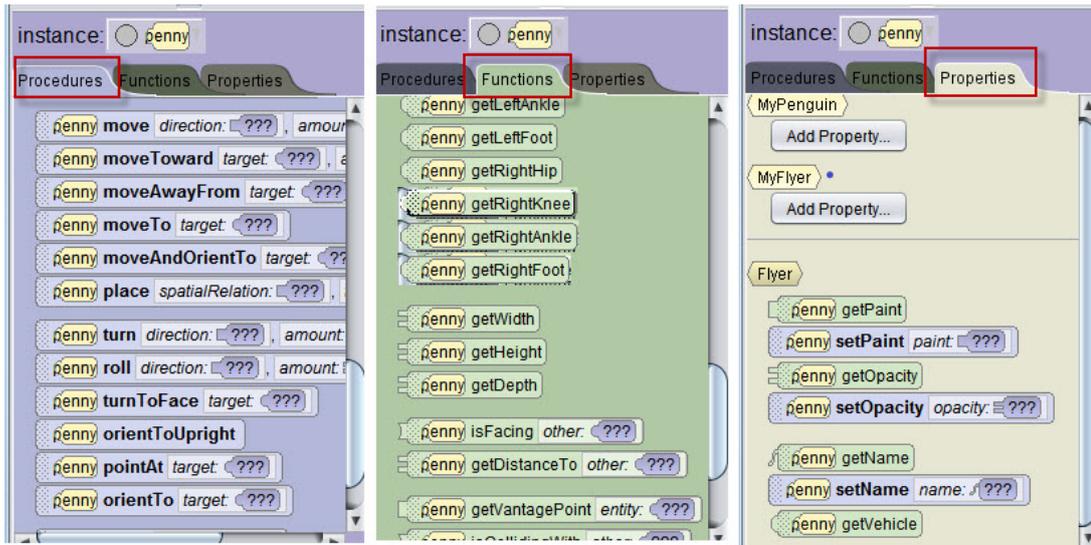


Figura 2.8: Comandos do ambiente Alice. Extraído de [PBWORKS \(2012\)](#)

Além de disponibilizar recursos para aprender e ensinar, a plataforma é gratuita. A plataforma disponibiliza alguns cursos que trabalha os fundamentos da Ciência da Computação para todas as idades, que podemos encontrar na área do aluno que é denominada de *Code Studio*. Mas a principal iniciativa é o evento “A Hora do Código”. O objetivo do evento consiste em promover o ensino de programação de maneira divertida, não somente para os alunos, mas também para os professores, dentro de um período de uma hora. O material que foi utilizado em cada evento fica disponível na plataforma, através de tutorias, o que possibilita a utilização do mesmo posteriormente. Cada tutorial aborda os conteúdos do evento através de jogos educacionais ([MARTINS et al., 2016](#)).

A plataforma possui diversos cursos, ilustrados na Figura 2.10, que são divididos por faixa etária, podendo ser utilizado desde crianças até jovens. No curso 1, indicado para crianças de 4 a 6 de idade, tem foco nos alunos que estão aprendendo a ler. Nele são trabalhados conceitos de lógica através de montagens de quebra-cabeça usando desenhos. Com isso, o aluno aprende conceitos básicos de lógica, como funciona a sequência lógica de comandos e noções de laços de repetição.

O curso 2 é focado para alunos que já sabem ler, a partir dos 6 anos de idade. Nesse curso, é inserida a programação com a utilização de blocos escritos, o que o difere do curso anterior. Trabalha com elementos de sequência lógica, laços de repetição e condicionais. Além disso, trabalha programação para diversos objetos que podem ser utilizados nas atividades.

Já o curso 3, é focado para alunos de 8 a 18 anos, sendo uma continuação do curso 2. Neste curso são ensinados os conceitos de função. O curso 4 é focado para alunos de 10 a 18 anos e que tenham realizado os cursos 2 e 3. Nesse curso os conceitos que foram aprendidos anteriormente, são trazidos novamente com um grau de complexibilidade maior. Sendo assim, são acrescentados novos conceitos sobre os conteúdos, que ainda não tinham sido apresentados para os alunos.

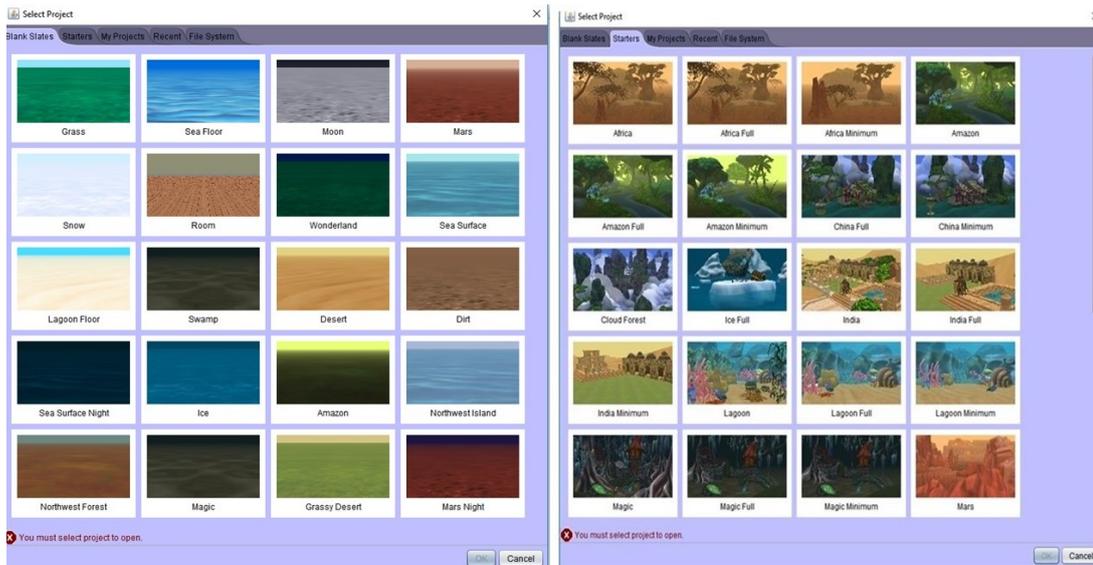


Figura 2.9: Mundos do ambiente Alice. Elaborada pelo autor.

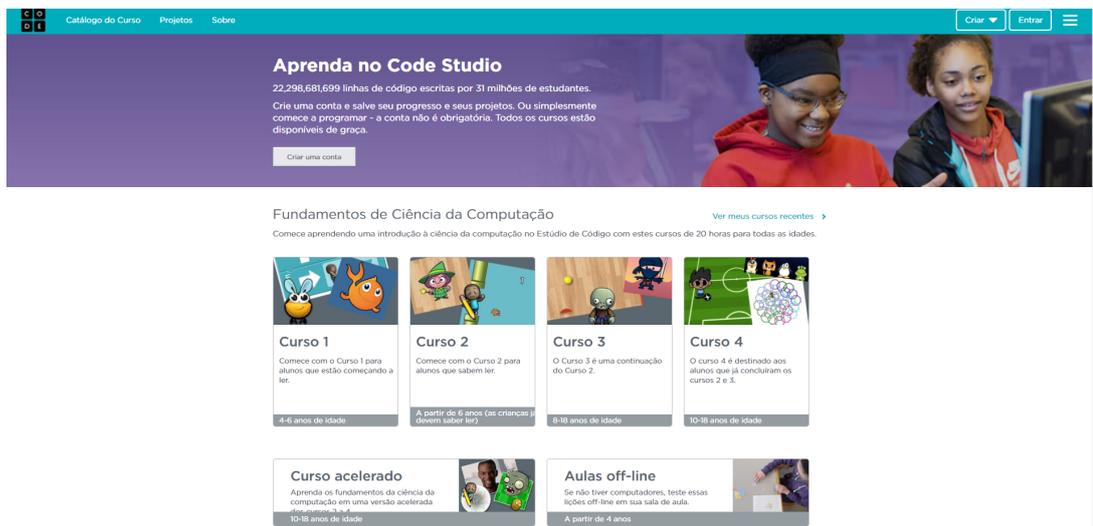


Figura 2.10: *Code Studio* do Code.org. Elaborada pelo autor.

Além dos quatro cursos, existem ainda dois cursos extras. O primeiro se trata de aulas off-line, caso não se tenham computadores que possam ser utilizados nesse curso. O segundo é um curso intensivo que utiliza os conteúdos dos cursos 3 e 4.

Na Hora do Código há diversos tutoriais, entre eles o Labirinto Clássico que aborda conceitos de lógica de programação, utilizando cenários do *Angry birds*, *Plants vs Zombies* e *A Era do Gelo*. O objetivo do jogo é direcionar o personagem principal de um ponto até o outro. A cada nível que vai passando a dificuldade aumenta, fazendo que o jogador aprenda conceitos mais complexos, como laços de repetição e estrutura condicionais (MARTINS et al., 2016).

Os conceitos são apresentados a cada usuário através de vídeos explicativos dos conceitos que serão necessários para a fase. Caso o aluno não deseje ver o vídeo, há a opção de ver as instruções dos novos comandos daquela fase por escrito, através de notas explicativas. O

jogo Labirinto utiliza a linguagem de programação visual, onde é possível só arrastar e soltar. São montados os códigos conectando cada bloco de comando, sem ter que se preocupar com a sintaxe. Ao executar o código com sucesso, aparecerá uma janela dizendo que o objetivo foi cumprido, onde é possível ver o código em uma linguagem de programação tradicional. Caso o objetivo não seja alcançado, o usuário pode corrigir o código para conseguir atingir o objetivo (MARTINS et al., 2016).

## 2.7 Blockly Games

O Blockly Games é uma plataforma que possui vários jogos educacionais. O principal objetivo dessa plataforma é o ensino de programação e lógica para crianças, que é constituído de diversos desafios para estimular o raciocínio e o pensamento lógico. O Blockly é intuitivo, o que facilita a utilização pelo usuário, permitindo a descoberta da ferramenta de forma independente. Além disso, não necessita de uma grande infraestrutura já que funciona em *tablets* e celulares, para isso basta ter um navegador instalado. Também é disponibilizada uma versão para ser utilizada off-line (JUNIOR, 2017).

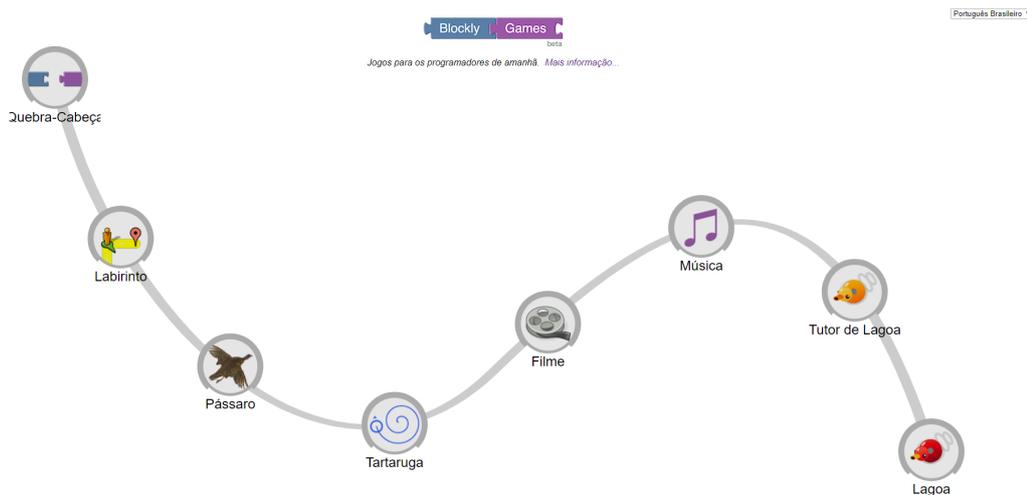


Figura 2.11: Interface do Blockly Games. Elaborada pelo autor.

Os jogos possuem diversos níveis que vão se tornando mais difíceis na medida em que o usuário vai subindo de nível. Cada fase pode ser resolvida através da montagem de blocos; em cada nível apresenta blocos novos, realizando funções diferentes. Caso o usuário não consiga resolver um determinado exercício, ele pode pular o nível e voltar posteriormente nele. Sendo assim, o usuário não se cansa do exercício e não desiste na primeira dificuldade (JUNIOR, 2017).

Dentre os vários jogos dessa ferramenta, o que mais se destaca é o jogo Tartaruga. O objetivo do jogo é introduzir as estruturas de repetição, utilizando representações de formas geométrica através do uso de blocos que representam as ações dos personagens. Essa ferramenta conta com um espaço para representar as ações do personagem, que fará o desenho da forma geométrica

através dos comandos que foram gerados pelo usuário. Há outro espaço que contém a área para desenvolver as instruções a serem executadas pelo personagem, além dos blocos que podem ser utilizados na fase em questão [JÚNIOR e BONIATI \(2015\)](#). Além do jogo tartaruga podemos encontrar outros jogos:

- *Puzzle*: introduz noções de como funciona a ferramenta.
- *Labirinto*: dá a introdução sobre as estruturas de repetição e condicional.
- *Bird*: faz com que o aluno aprofunda seus conhecimentos sobre condicionais.
- *Filme*: utiliza cálculos matemáticos para realizar animações, introduzindo assim equações matemáticas.
- *Música*: introduz os conceitos de funções.
- *Pond Tutor*: utiliza blocos ou textos; a cada nível que se passa, há a alteração entre esses dois modos.
- *Pond*: o usuário pode escolher qual modo vai utilizar, texto ou bloco. Com isso os alunos conseguem aprender tanto a lógica de programação, quanto uma linguagem específica, o que diferencia essa ferramenta das demais.

---

## *Análise das Ferramentas no Apoio ao Pensamento Computacional*

---

### **3.1 Introdução**

Atualmente estamos rodeados por sistemas computacionais e diversos modelos de plataformas, que acabaram se tornando indispensável na nossa rotina. Com os avanços da tecnologia, criaram-se diferentes recursos, que não nos trouxe somente uma maior agilidade e rapidez, mas também trouxe flexibilidade para o nosso dia a dia e para a sociedade. Além disso, são tecnologias que se tornaram acessíveis para praticamente todas as pessoas. Atualmente não estamos limitados somente a um espaço físico e a um tempo determinado, podemos acessar uma quantidade de informações muito maior do que a alguns anos atrás, com diferentes opiniões sobre um determinado assunto e novos modos de como fazer algo que desejamos (RIBEIRO et al., 2014).

Na área da educação não foi diferente. Com as novas tecnologias as formas de ensinar e aprender acabaram se modificando e se adequando a nova realidade. Já está se tornando comum a afirmação de que os professores não podem mais se omitir de transmitir a informação, já que se pode acessar através de pesquisas na rede. Nos dias atuais não podemos mais admitir essas práticas engessadas, de que somente o professor é o detentor do saber. O professor fica então responsável por acompanhar e durante a gestão do aprendizado do indivíduo, o professor acaba vivendo em uma eterna fase de aprendizagem. Assim como as ferramentas precisam nos oferecer práticas que sejam flexíveis (RIBEIRO et al., 2014).

Um dos desafios dessa nova prática pedagógica é a aprendizagem colaborativa, além da utilização do erro como um meio de aprendizagem e pensando no que pode ter proporcionado o seu erro. Para que o indivíduo não precisa ser somente um usuário das tecnologias, mas sim desenvolver artefatos computacionais (RIBEIRO et al., 2014).

Ter o conhecimento básico sobre computação e seus fundamentos está sendo considerado como fundamental nos dias de hoje. O desafio está em encontrar ferramentas que trabalhem os conceitos de programação que possam alcançar os objetivos e as expectativas dos participantes e, que ao mesmo tempo, não necessitam de um conhecimento técnico do usuário. As ferramentas que trabalham com a utilização do conceito de programação em bloco, que são semelhantes a peças de quebra-cabeça, facilitam na aprendizagem de programação, conseguindo então que o indivíduo não cometa erros de sintaxe (RIBEIRO et al., 2014). A seguir são apresentadas análises de algumas dessas ferramentas.

## 3.2 Scratch

No Scratch as instruções podem ser alteradas a qualquer momento, o que ajuda nos desenvolvimentos de novas ideias a realizar testes da mesma, aprendendo com os seus próprios erros. Além de poder serem executadas instruções de dois ou mais conjuntos de blocos ao mesmo tempo. Por meio dos erros do aluno, pode ser feita uma análise, assim como elaborar soluções para o problema até que consiga a solução desejada, alcançando o objetivo proposto (BRESSAN e AMARAL, 2015).

O Scratch tem como sua potencialidade desenvolver o pensamento criativo e o raciocínio lógico, utilizando as próprias criações que o aluno está desenvolvendo. Mesmo que os alunos trabalhem em projetos diferentes, pode ficar em conjunto para analisar e tentar solucionar o problema (BRESSAN e AMARAL, 2015).

De acordo com BRESSAN e AMARAL (2015) é interessante que os alunos sigam os seguintes passos: compreender o problema, elaborar um plano, executar o plano e verificar os resultados. A partir desses quatro passos, os alunos conseguem aprender e desenvolver a capacidade de identificar e solucionar problemas.

O pensamento criativo, como o próprio nome diz, se trata do método que utiliza a criatividade para solucionar os problemas de maneira eficaz e pode ser estimulado e desenvolvido. O ambiente em que o aluno vive pode influenciar diretamente no desenvolvimento dessa competência, se o estudante conviver em um ambiente que o estimula a estar conhecendo ou encontrando novas soluções para os problemas do seu dia a dia (BRESSAN e AMARAL, 2015).

A utilização das Tecnologias de Informação e Comunicação (TIC's) com fins educacionais interfere em aspectos do saber pedagógico: o saber ensinar e o saber avaliar. Ao participar do processo de construção e não somente de uma comparação de pré-teste e pós-teste, mas sim de um todo que acontece durante o processo e que engrandece mais a coleta e a análise de dados, o que possibilita detectar os problemas (BRESSAN e AMARAL, 2015).

De acordo com BRESSAN e AMARAL (2015), o Scratch mostrou que é capaz de auxiliar no desenvolvimento da capacidade avaliativas, já que o aluno podia observar os blocos de comandos utilizados que estavam sendo desenvolvidos por eles. Isso permitia realizar uma reflexão e realizar alterações de acordo com a detecção de erros, através da representação dos processos mentais como calculo mental, facilitando a estruturação, organização do pensamento

de maneira criativa.

BRESSAN e AMARAL (2015) ainda mostraram que além de ajudar no desenvolvimento do pensamento criativo, também houve a melhora da capacidade de solução de problemas, formulação, reflexão e organização de estratégias para que consiga realizar o objetivo e quando necessário reformulação e reorganização.

Os estudantes foram capazes de aprender alguns conceitos da Computação, demonstrando aprendizagem de alguns conceitos como paralelismo, sequência, evento, repetição, condicionais, operadores e dados. Alguns conceitos acabaram apresentando uma aprendizagem melhor sendo elas: sequência e eventos, já o paralelismo foi o que menos foi compreendido, sendo possível ter alcançado os objetivos traçados inicialmente, que foi explorar conceitos de computação além de auxiliar na divulgação do pensamento computacional na educação básica.

De acordo com NETO (2013), foi observado que os alunos da disciplina de lógica de programação tinham muitas dificuldades de aprender e aplicar os conceitos de programação, como na interpretação dos problemas que eram propostos. Alguns consideravam o conhecimento de lógica matemática insuficiente, já que estavam acostumados a decorar os conteúdos, ou seja, não conseguiam desenvolver o raciocínio lógico que era necessário para a disciplina. Muitos ainda não conseguia apresentar soluções lógicas computacionais para os problemas que estavam sendo propostos.

NETO (2013) ainda mostrou que haviam casos em que alguns alunos não conseguiam compreender os termos e conceitos. O que acabou gerando alunos desmotivados para o restante do curso. Alguns alunos ao terminar a disciplina não tinham conseguido aprender o conteúdo, alguns se achavam incapazes de resolver problemas computacionais.

Para amenizar ou até mesmo solucionar essas dificuldades, foi utilizada a ferramenta Scratch nas primeiras aulas da disciplina. A ferramenta foi usada para introduzir os conceitos de programação, além de incentivar os alunos a continuarem praticando posteriormente a cada aula, através de problemas que era levado para casa.

Na primeira aula, foram apresentados conceitos de lógica de programação e algoritmos como solução de situações a serem seguidos passo a passo. Também foi apresentada a ferramenta de maneira geral para os alunos. Na segunda aula foram apresentados os conceitos iniciais de programação utilizando o Scratch, sendo eles comandos de entrada e saída, operadores, variáveis e atribuição de dados.

A terceira aula foi dedicada à resolução de problemas, contando com o auxílio da ferramenta. Foram apresentadas algumas funcionalidades da ferramenta que ainda não tinham sido apresentado para os alunos. Na quarta aula foram ensinadas para os alunos as estruturas de decisão que estão presentes na ferramenta.

Para exemplificar o uso foi utilizado o problema de cálculo da média, que demonstrava a situação do aluno se tinha sido aprovado ou reprovado. O usuário digita os valores das notas e realiza o cálculo, de acordo com esse resultado a imagem de fundo acaba sendo alterada. Na última aula, foram mostradas as estruturas de repetição que a ferramenta possui.

Logo após o autor propôs um desafio para os alunos: produzirem um jogo utilizando todos os conhecimentos adquiridos. Com isso os alunos mostraram motivação e desejo de produzir o melhor programa possível. No decorrer da disciplina os conceitos que eram apresentados, eram feitas associações com o que já tinha sido visto anteriormente, com o intuito de deixar a compreensão mais fácil. Assim como na hora de aprender alguma linguagem de programação, tendo só que se adaptar à novas formas e particularidades.

De acordo com NETO (2013) alguns alunos preferiam utilizar linhas de códigos do que a montagem de blocos. Assim, conseguiam entender melhor o conteúdo, que se parece mais com o que é programar, no entendimento deles. Já com outros estudantes pode se identificar que tiveram um maior desempenho na disciplina no primeiro momento, onde estava sendo utilizado o Scratch, do que no restante da disciplina. Quando foram questionados a respeito dessa queda de desempenho, eles alegaram que a ferramenta era mais fácil de compreender e divertida na hora de criar os programas.

Além disso foi destacado que a ferramenta contar com blocos coloridos que são encaixados, sons, imagens, que possibilita a criação de jogos, animações e história, são pontos destacados pelos alunos. Com isso, acabou gerando um clima mais divertido e descontraído durante as aulas, além de ter provocado uma frustração nos alunos que já avinham cursado a disciplina e provocando entusiasmos nos que ainda teriam a disciplina.

Como tinha sido apontado anteriormente por NETO (2013), muitos alunos tinham dificuldades no aprendizado dos conceitos iniciais de programação, o que acabou sendo diferente depois da nova abordagem, contando com o auxílio do Scratch. De acordo com NETO (2013) muitos dos alunos conseguiram compreender os conceitos iniciais de programação, além de entender a forma de “raciocínio” do computador, outros que conseguiram criar programas com mais facilidade, muitos se sentiram mais estimulados em aprender o conteúdo da disciplina, além de se considerarem capazes de interpretar melhor os problemas que eram propostos em aula.

De acordo com NETO (2013), ao utilizar o Scratch para auxiliar no ensinamento, alguns alunos ficaram mais interessados em seguir na área de Computação, se sentiram mais motivados para o restante do curso e com outras ferramentas e metodologias que seriam utilizadas em outras disciplinas do curso. Alguns alunos se sentiram com as expectativas correspondidas, além de se sentirem capazes de resolver e interpretar a maioria dos problemas que estavam sendo propostos nas aulas. É diferente da turma anterior, em que nenhum aluno respondeu que sentia não ter a capacidade de resolver problemas computacionais. Com isso podemos observar que ocorreu uma melhora de desempenho dos alunos na disciplina.

Entretanto, nem tudo foi positivo. De acordo com NETO (2013), a ferramenta apresenta algumas desvantagens que foram citadas pelos alunos. Como na área de imagens, em que foram consideradas a qualidade de algumas imagens de pouca qualidade. Além disso, as opções e quantidade são limitadas, o editor de imagens possui poucos recursos para tratamento das imagens, os recursos que são disponíveis são bem restritos e difíceis de serem utilizados.

De acordo com NETO (2013), o recurso de áudio também foi apontado como uma desvan-

tagem já que os alunos demonstraram dificuldade para utilizar esse recurso. Alguns arquivos não podiam ser executados, além de não possuir um editor de áudio. Com a utilização de muitos recursos acabam gerando um baixo desempenho, o que deixa o programa pesado e se a capacidade da maquina utilizada for baixo a execução dos comandos ficam lentos e podem ocasionar travamentos. Outro ponto que foi destacado como negativo, é que dependendo do que o usuário precisa realizar, acaba tendo a necessidade de utilizar muitos blocos de comando, o que pode dificultar na criação do programa e o tornando trabalhoso.

NETO (2013) ainda destacou que alguns dos alunos estavam tendo dificuldades em criar programas no Scratch, como em saber qual dos comandos seriam utilizados para desenvolver algo, ou solucionar algum problema. Em alguns casos o aluno confundia qual bloco de comando que deveria ser utilizado, principalmente quando tinham que ler e armazenar alguma informação do usuário. Também houve dificuldades na utilização de alguns blocos de comando, com as estruturas de repetição, de decisão e os operadores.

Outro ponto que se destacou foi a dificuldade de colocar os objetos em cena, já que cada um deles possuem o seu próprio conjunto de bloco e tem que ser sincronizar as suas ações, como os objetos que estão no palco. Outro ponto que podemos destacar é que não se pode utilizar para todos os tópicos de programação, um dos fatores é a falta de algumas estruturas como vetor, matriz e registro, já que ter o conhecimento sobre essas estruturas é necessária para o desenvolvimento de programas mais completos.

Através desses resultados NETO (2013) conclui que a utilização dessa ferramenta pode gerar motivação nos alunos, assim como também se mostrou eficiente na participação na busca pela solução de problemas. Com isso o Scratch se demonstrou ser útil para auxiliar nessa nova abordagem, trazendo resultados favoráveis. Então pode-se ver que o ambiente interativo propicia aos alunos que a programação se torne atraente, fácil e intuitiva, já que o mesmo pode desenvolver o seu programa utilizado de elementos como sons e imagens pode ser ou não produzidas por ele.

NETO (2013) notou também que a ferramenta não pode ser utilizada sozinha, sem que seja desenvolvido um planejamento das aulas que sejam interessantes e motivadoras para que possa gerar interesse no aluno. Como a ferramenta tem a ausência de algumas estruturas, ela não deve ser utilizada isolada, já que essas estruturas são importantes para o desenvolvimento da programação dos alunos. A ferramenta demonstrou que pode facilitar o entendimento das estruturas e comando, principalmente no ensino de programação para alunos que estão iniciando, amenizando problemas que teriam na compreensão da lógica computacional, contribuindo com o seu desenvolvimento.

A construção do projeto tem que ser contextualizado de acordo com o que o aluno considera importante, com isso aumenta a chance do conceito que está sendo trabalhando possa ser aprendido da melhor maneira possível. Mas para que isso possa ser possível, o aluno tem que participar da escolha do processo, contando com o auxílio do professor para que o processo possa ser desafiador e que possa ser concluída (SÁPIRAS et al., 2015).

É importante que o aluno entenda o significado de cada conceito e não como meros símbolos e formalismos sem que compreendam qual a sua utilidade, por meio da utilização e construção dos conceitos que levam a descobertas dos novos conceitos. A integração das atividades com as relações culturais e sociais, com a computação e a tecnologia são excelentes para demonstrar essas possibilidades, o professor fica responsável por propiciar essa utilidade de forma educacional (SÁPIRAS et al., 2015).

Atualmente a escola precisa auxiliar o aluno a lidar e compreender com as novas mídias digitais. Os jovens estão conseguindo construir habilidades e competências através da interação com as mídias, apesar de não estarem dando uma grande importância no âmbito educacional. Algumas habilidades são construídas através da colaboração (SÁPIRAS et al., 2015).

O professor pode ensinar como utilizar as mídias digitais para que o aluno possa tirar o que a de melhor nela, já que nem todo o conteúdo que estão disponíveis na rede estão corretos, sendo assim, o aluno não necessitaria da presença do professor a todo momento para tirar as suas dúvidas.

Através das simulações os alunos podem se desenvolver mentalmente, podendo lidar com uma grande quantidade de informações. Além de poder criar hipóteses e testes para que possam compreender informações mais complexas. Sendo assim, podendo construir o seu conhecimento através dessas informações (SÁPIRAS et al., 2015).

Já que a simulação utiliza do meio de tentativa de erros ou acertos, com isso podendo redefinir de acordo com os resultados das simulações. A construção do conhecimento está além do certo ou errado e a busca das teorias que quando realizado uma análise a partir de um olhar que tenha como base o conhecimento formal (SÁPIRAS et al., 2015).

Atualmente muitos jogos possibilitam aos alunos não se limitar somente a observar, mas sim a experimentar as possibilidades. A simulação disponibiliza informações que possibilita tomar a melhor decisão e com isso entender o objetivo que se deseja alcançar, com isso possibilita o planejamento de ações futuras, tornando assim mais simples para realizar o projeto. Com isso, os alunos se tornam mais flexíveis na personalização de modelos e na manipulação dos dados, o que pode ser mais fácil contando com a utilização da tecnologia (SÁPIRAS et al., 2015).

SÁPIRAS et al. (2015) utilizaram na primeira etapa a construção Scratch Cards, que são cartões com atividades que já tinham sido definidas anteriormente para que os alunos possam desenvolver uma sequência de comandos. Na segunda etapa, os alunos tiveram que desenvolver um jogo ou animação que seria de seu interesse, contando com auxílio do Scratch. Os alunos foram divididos em cinco grupos de três pessoas, e em grupo decidiram o que iriam produzir. Foi definido que um dos jogos que se chamaria *Tom&Gerry*, que consiste em fazer com que o rato escape do gato. É jogado por duas pessoas sendo que cada uma delas performa o papel de um dos personagens.

SÁPIRAS et al. (2015) puderam observar que os alunos tiveram dificuldade de posicionamento inicial do personagem, em saber qual comando seria o correto. Com isso permitiu que o aluno utilizasse do meio de simulações, utilizando o comando correto para tal necessidade. A

simulação não visa mostrar o produto, mas sim como será o processo de construção através da tentativa e erro. Através das mudanças dos comandos que estão sendo utilizados, os processos de construção de seu código podem gerar erros e acertos. Assim, fazendo com que os estudantes busquem um curso natural até a resolução do problema, através da compreensão da situação até o modo de como se realiza o processo.

De acordo com [SÁPIRAS et al. \(2015\)](#) puderam observar ainda que o posicionamento do personagem está relacionado a conceito matemático, já que para isso tiveram a necessidade de utilizar um comando que pode usar o posicionamento no plano cartesiano. Mesmo que os alunos já tenham o conhecimento sobre o assunto, puderam então melhorar o seu conhecimento. A associação desse conceito com os aspectos do jogo acabou sendo reconhecida sem que tenha sido apresentado para eles.

Pode ser constatado que a matemática foi utilizada como um meio para que o processo pudesse ser inicializado e o objetivo se conclua. O que foi de interesse não era o produto final e sim o processo de simulação que foram utilizados pelos alunos. As simulações só são possíveis pelo conceito computacional, mesmo que embora não seja especificamente matemático, mas sim lógica computacional, o que pode auxiliar de modo decisivo nas construções matemáticas utilizadas no Scratch ([SÁPIRAS et al., 2015](#)).

Então, [SÁPIRAS et al. \(2015\)](#) cluíram que no decorrer das aulas apresentaram mudanças no modo em que as simulações foram encaminhadas. Os conceitos matemáticos que foram utilizados como meio para que as simulações fossem possíveis. Na parte final o processo de simulação que permitiam experimentar um conjunto de situações para que pudessem tornar o jogo melhor, a capacidade de simulações pode contribuir com a capacidade de interpretar e utilizar as mídias digitais.

## 3.3 App Inventor

O App Inventor inicialmente foi utilizado no Estados Unidos em projetos pilotos, que foram patrocinados pela Google. Esses projetos visaram explorar os potenciais pedagógicos da ferramenta como uma plataforma de ensino, sendo utilizados em cursos de introdução a programação no ensino médio e superior. Atualmente está sendo utilizado desde o ensino fundamental, médio e superior, sendo até mesmo usado na formação de professores. Através disso vem demonstrando o seu potencial pedagógico, principalmente em cursos para iniciantes em programação ([GOMES e MELO, 2013](#)).

Com o auxílio dessa ferramenta é possível que os alunos consigam desenvolver aplicativos para Android, mesmo sem ter conhecimento nenhum sobre programação. Esses programas podem instalados e executados em seus dispositivos Android, o que acaba deixando os alunos mais motivados por estarem desenvolvendo aplicativos que poderão ser utilizados no término da programação ([GOMES e MELO, 2013](#)).

Inicialmente [GOMES e MELO \(2013\)](#) realizaram uma experiência de lógica de programação contando com o auxílio do App Inventor. Os alunos podem utilizar a ferramenta não só

para se divertir e jogar para explorar os conhecimentos de computação, processamento da informação e raciocínio lógico. Mas puderam desenvolver aplicativos para Android contando com auxílio dessa ferramenta. Os aplicativos que foram desenvolvidos exploravam conceitos iniciais de programação (variáveis, estruturas de controle, estruturas de decisão). Os resultados mostraram que os alunos tiveram um aprendizado significativo, além de muito alunos demonstrarem que desejaram seguir na área da Computação.

Foram realizadas oficinas nas quais os conteúdos eram ministrados com foco na prática. Para isso, o conteúdo foi dividido em quatro partes principais, sendo elas: introdução ao desenvolvimento de aplicativos para dispositivos móveis, linguagem de programação visual App Inventor, emulador e teste, publicando os aplicativos na web. O objetivo era explorar os conhecimentos iniciais de computação móvel através do desenvolvimento de aplicativos educacionais para dispositivos Android, além de explorar os benefícios e técnicas de programação para o aprendizado de lógica de programação, assim como promover formas divertidas de aprendizagem através da ferramenta.

Em cada oficina era feita uma pequena introdução ao desenvolvimento de aplicativos para dispositivos móveis. Posteriormente, era iniciada a parte prática que contou com o desenvolvimento de aplicativos que buscavam conter os principais conceitos de lógica de programação. O objetivo era, que no decorrer do desenvolvimento dos aplicativos, fossem criados de maneiras divertidas e que o participante adquiriria o conhecimento de acordo com o avanço do curso.

Na oficina os alunos desenvolveram quatro aplicativos. No *HelloDroid*, o aplicativo era formado por uma caixa de texto que o usuário poderia digitar uma mensagem e pressionar a tecla de confirmação, seria exibida na tela a mensagem que foi digitada. Esse aplicativo tinha como objetivo de demonstrar os principais recursos da ferramenta.

O *MiniPaint* se tratava de um aplicativo de criação e edição de imagens simples, que tinham somente recursos básicos, que permitia ao usuário fazer traços e pontos de cores variadas, importar imagem da galeria, salvar o desenho que foi criado. Com o desenvolvimento desse aplicativo os alunos puderam compreender conceitos de lógica de programação (GOMES e MELO, 2013).

O aplicativo *MoleMash* simula uma toupeira que fica percorrendo um conjunto de pontos aleatórios na tela, a cada vez que o usuário acerta a toupeira ele ganha ponto. Esse aplicativo tem como objetivo de reforçar os conhecimentos que foram aprendidos anteriormente, e iniciar o aprendizado sobre estruturas de controle, operadores aritméticos e relacionais (GOMES e MELO, 2013).

O último aplicativo que foi desenvolvido é o *QuizMe*, um jogo de perguntas e respostas, que utiliza pontuação e tempo. Através do desenvolvimento desse aplicativo o aluno aprofundava mais os seus conhecimentos sobre as estruturas de controle e a utilização de operadores aritméticos relacionados a lógicas (GOMES e MELO, 2013).

Através dos blocos pré-definidos permitia que os participantes tivessem maior liberdade para desenvolver os códigos. De acordo com os conteúdos eram ensinados, os aplicativos eram desenvolvidos e os estudantes tinham total liberdade de alterar o código para que pudesse ser

testados novos blocos de comando. As atividades de desenvolvimento ocorriam de acordo como a evolução da compreensão sobre a lógica de resolução de problemas, explorando melhor a ferramenta (GOMES e MELO, 2013).

Foi observado que no decorrer das oficinas que os alunos se sentiam motivados por poderem desenvolver aplicativos que pudessem ser utilizados nos dispositivos. Muitos deles já testavam diretamente em seus dispositivos Android conectando o aparelho ao computador.

GOMES e MELO (2013) concluíram que ao desenvolver aplicativos que possam ser utilizados em seus dispositivos Android e que contava com funcionalidades específicas de dispositivos móveis, como a mobilidade e o uso do sistema de geolocalização, que são conexões reais do aluno com os conceitos que foram aprendidos. Os conceitos em a sua importância para os educandos, pois possibilita o desenvolvimento de aplicativos reais que os deixaram mais motivados.

Com a facilidade de descoberta de erros no desenvolvimento dos aplicativos, era possível que os alunos aprendessem através da construção de seus próprios conhecimento. Já que o aluno é um participante ativo na construção do seu conhecimento, fica por sua conta conduzir o aprendizado. Sendo assim, ao invés de ser somente passado o conhecimento para o aluno e levá-lo a fornecer uma resposta, o computador recebe as instruções e as operações que devem ser executadas para que possa chegar nas respostas desejadas pelo aluno.

Os estudantes além de desenvolver aplicativos com um conteúdo relevante, ainda têm uma ideia do que está sendo desenvolvido. Sabendo assim se as ações que estão sendo executadas pelos blocos estão corretas, podendo assim comparar os resultados que foram obtidos com o resultados que era esperados. Desse modo, podia se ter uma melhor compreensão de assuntos que seriam complexos e permitindo ainda ao aluno a entender o porquê do código que foi desenvolvido não gerou o resultado que estava sendo esperado.

Por fim, GOMES e MELO (2013) concluíram que de acordo com o que foi apresentado a ferramenta demonstrou o seu potencial didático-pedagógico, se demonstrando assim ser uma ferramenta muito útil para auxiliar no ensino de lógica de programação e na transmissão dos conceitos do pensamento computacional, além de propiciar a inclusão das tecnologias móveis no meio educacional. Através da interface gráfica simples, é permitido ao usuário focar em conceitos básicos de programação e algoritmos, sem ter que se preocupar com a sintaxe, o que resulta em uma melhor compreensão sobre o conteúdo que está sendo abordado.

BARBOSA et al. (2015) utilizaram o App Inventor com foco nos aplicativos educacionais voltados a princípios matemáticos. Com isso, tinha o objetivo de ampliar o conhecimento de ferramentas e as possibilidades do App Inventor. Contando com auxílio dessa ferramenta, foi desenvolvida uma calculadora, já que é um recurso bem conhecido, além de, possuir diversas funcionalidades matemáticas. Ainda foi ressaltado que o objetivo não era desenvolver uma aplicativo diferente do que já existia, mas sim analisar o potencial da ferramenta para o desenvolvimento de aplicativos para a Matemática.

Inicialmente foi desenvolvida uma calculadora com funções básicas (soma, subtração, multiplicação e divisão), posteriormente foram inseridas novas funcionalidades que pertencem a

uma calculadora científica. Para isso, contou como auxílio de vídeo aulas e informações que estavam disponíveis em fórum oficiais do App Inventor, o que contribuiu para algumas ações de desenvolvimento do aplicativo.

[BARBOSA et al. \(2015\)](#) puderam identificar que a ferramenta contém blocos matemáticos de várias operações que já estão prontos para serem utilizados. O que facilita para o usuário na familiarização com o App Inventor, pois não tem a necessidade de serem programados. A primeira versão da calculadora continha apenas as principais operações matemáticas, e continha uma caixa de texto, para a inserção dos valores, esses dados entrava via teclado. O resultado da operação que tinha sido realizada aparecia em outra caixa de texto. Os testes para verificar se as operações estavam corretas eram realizadas através do emulador, que é um recurso presente na ferramenta para a realização de testes. Com isso o desenvolvedor poderia ter uma ideia em tempo real do que está sendo desenvolvido.

Durante uma reunião do grupo de pesquisa, foi decidido que seriam realizadas algumas melhorias, como os campos para digitar os dados para que aceitem somente números; ocultar o teclado ao aparecer o resultado; alteração na interface do aplicativo; trocar os nomes da operação pelo seus símbolos e, por fim, uma função para limpar todos os campos em um único botão.

[BARBOSA et al. \(2015\)](#) analisaram a segunda versão e optou por encerrar os testes que foram feitos com entrada de dados através do teclado do dispositivo, desse modo preferiu trabalhar com o teclado numérico inserido na interface do aplicativo. Então para a terceira versão foi decidido realizar essa mudança de inserir o teclado numérico como o dos operadores, que deixou mais parecido com a calculadora convencional. Além dessa alteração foram inseridas novas operações como: raiz quadrada, porcentagem, apagar o último número inserido e o de limpar todos os dados. Só o botão de limpar foi programado nesta versão, já os demais foram inseridos na interface, mas só serão programados nas próximas versões.

Foi realizada uma nova análise na versão última versão, e através de reunião foi decidido novas alterações, a calculadora contará com botões para calcular raiz quadrada e potencialização, como também botões da calculadora científica. Os botões que são típicos da calculadora científica foram colocados separados. A quarta versão contou ainda com alteração da interface, diminuição dos botões e alteração da cor de fundo.

Através da criação dessa calculadora [BARBOSA et al. \(2015\)](#) com ocorre em outras linguagens de programação, que contam com bibliotecas para a realização de cálculos matemáticos (Java, C e Python). Essa ferramenta contém blocos matemáticos o que facilita a realização do desenvolvimento de aplicativos que contam com esses cálculos. Esses blocos já possuem algumas opções para realizar os cálculos pré-programados, sendo eles as quatro operações matemáticas básicas, além de raiz quadrada, potência e logaritmo natural. A ferramenta conta ainda com funções trigonométricas e seus arcos, além da conversão de graus para radianos.

Como isso [BARBOSA et al. \(2015\)](#) concluíram, a facilidade que a ferramenta possui para a realização de teste, favorece ao processo de criação do aplicativo. Outro ponto que se desta-

cou positivamente foi a de que a chance de cometer erros são diminuídas, já a ferramenta não permite conectar os blocos que não são compatíveis, o que diminui a possibilidade de erros. O ponto que se destacou negativamente foi o desenvolvimento da interface do aplicativo que está sendo desenvolvido se mostrou inflexível, já que permitia pouco controle sobre onde os componentes são colocados na tela de desenvolvimento.

[BARBOSA et al. \(2015\)](#) ainda desenvolveram outros aplicativos voltado para a matemática, esse novo aplicativo foi intitulado de Matrizes e Determinantes. O objetivo desse aplicativo era realizar operações com matrizes e ao cálculo de determinantes, com isso deverá conter na interface principal botões para escolher a ação que serão realizados, sendo elas: adição, subtração, multiplicação, transposta, inversa e determinantes.

[BARBOSA et al. \(2015\)](#) puderam identificar limitações do App Inventor. Por exemplo, não podem ser utilizadas variáveis globais, somente variáveis locais que funcionam somente na tela que foram inseridas. O mesmo ocorria com o banco de dados que era restrito somente a uma única tela. Para que fosse possível a utilização de dados que pertence ao um banco de dados, é necessário que seja inserido o banco de dados em outras telas. Apesar das limitações, o App Inventor é uma das ferramentas mais completas para o desenvolvimento de aplicativo para Android visando o usuário final.

[BARBOSA et al. \(2015\)](#) concluíram que a ferramenta é um recurso que se tem mostrado positivo para o desenvolvimento de aplicativos, além de facilitar com a disponibilização de alguns recursos e a sua flexibilidade nos momentos que foram necessários realizar algumas adaptações. Em relação à calculadora, essas características foram relevantes, pois facilitaram o desenvolvimento e ainda oportunizou a realização de ajustes necessários, melhorando assim a usabilidade do aplicativo.

Além de propiciar o trabalho em grupo através de discussões com possíveis sugestões e opiniões. Já sobre o aplicativo de matriz e determinantes não foram realizadas muitas considerações, pois o desenvolvimento ainda não tinha sido concluído. Entretanto, foi observado que a ferramenta não oferece biblioteca para facilitar a realização de cálculos com matrizes, o que é comum em outra linguagem de programação, mas se mostrou bem flexível.

[DUDA e SILVA \(2015\)](#) tinham o objetivo de desenvolver e aprimorar o raciocínio e pensamento algébrico, com atividades que são relacionadas à resolução algébrica de equações e a modelo de resolução de problemas, que visa relacionar a matemática escola com o cotidiano. Inicialmente os alunos recebiam instruções sobre o App Inventor, para capacitar dos mesmos, que tinha como objetivo desenvolver as habilidades matemática, focando principalmente as que são relacionadas ao uso de linguagem algébrica. Foi utilizado o princípio de que o aluno também é responsável pela construção do seu conhecimento e o professor é um mediador que auxilia nesta construção de conhecimento.

A primeira atividade que foi desenvolvida foi o desenvolvimento de um aplicativo para a resolução de regra de três. Os alunos foram incentivados a indicar algebricamente. A regra de três foi escolhida por não necessitar de um alto nível de conhecimento, pois tem um grau

de complexidade baixa, focando principalmente nos conceitos iniciais do App Inventor. O objetivo foi simplificar ao máximo para que os alunos tivessem um contato agradável com a ferramenta de designer de aplicativos, em simultâneo com a reflexão sobre a utilização da linguagem algébrica para a representação a solução de problemas.

Para a representação de uma regra de três na interface de um aplicativo foi utilizado os componentes que prendessem ao App inventor. Para isso as funcionalidades de cada componente foram apresentadas aos alunos e posteriormente cada um ficou responsável por desenvolver uma interface de um aplicativo. Depois disso, foram apresentados para os mesmos a forma com que cada botão assumia um conjunto de blocos.

Posteriormente o desenvolvimento do aplicativo de regras de três, foram elaboradas atividades visando situações problemáticas, sobre a utilização de linguagem algébrica. Dando autonomia e espírito colaborativo dos alunos para as voltadas para a implementação dos recursos e funções nos aplicativos. Para o desenvolvimento, Duda dividiu o processo em etapas que eram ligadas, de modo que pudesse ocorrer a aprendizagem do conteúdo que estava sendo abordado, sendo assim, os alunos poderiam elaborar mais de um aplicativo.

A primeira etapa, intitulada como Pesquisa, se trata da apresentação dos temas e que são realizadas pesquisas, conseguindo assim pontos chaves sobre o objeto de estudo. Sendo indicado pelo professor o foco e o objetivo da pesquisa. A segunda etapa foi intitulada como Arquitetura, se trata da passagem dos temas que foram pesquisados, são aplicados na estruturação do aplicativo. A terceira etapa denominada como Revisão, se trata da etapa onde o professor verifica possíveis falhas ou erros na execução dos cálculos, onde e como o aplicativo pode ser corrigido ou adaptado. Onde é possível constatar se todos os pontos elencados foram utilizados no desenvolvimento dos aplicativos, se só foram listados, mas não se teve a preocupação se foram compreendidos. Na quarta e última etapa, nomeada de Consolidação, se trata da etapa na qual o aplicativo são validados com relações à sua estruturação e funcionalidade.

DUDA e SILVA (2015) destacaram que os aplicativos criados pelos alunos passaram das expectativas no que diz respeito à sua estruturação lógica e organização. Com isso, foram abordados conhecimentos mais básicos, mas na medida que o projeto fosse desenvolvido a equipe foi descobrindo novas maneiras de como associar e ativar comandos, o que aumentou a qualidade dos aplicativos.

A utilização de recursos computacionais cria um ambiente que modifica continuamente a aprendizagem, no qual a capacidade de criação dos alunos é desenvolvida aos poucos. Com as características da ferramenta os alunos passam a ser mais participativos no seu desenvolvimento. Isso foi notado através dos aplicativos que foram desenvolvidos pelos alunos, pois os resultados que foram apresentados passaram das expectativas que eram esperadas.

Com isso, DUDA e SILVA (2015) concluíram que mesmo com a limitação a respeito de cálculos acerca de recorrências e números complexos, mas mesmo assim a ferramenta se demonstrou ser uma alternativa viável para a utilização do uso da linguagem algébrica. Além disso, há potencial suficiente para a estruturação de atividades variadas, como a exploração de concei-

tos matemáticos até a modelagem de situações problemáticas.

DUDA e SILVA (2015) ainda concluíram que é possível o reaproveitamento do roteiro de estudo e as demais atividades que foram decorrentes do processo de elaboração dos aplicativos que visaram auxiliar no processo de avaliação em matemática, já que os trabalhos que foram desenvolvidos pelos alunos apresentam uma riqueza de detalhes. Com os resultados pretende dar continuidade no projeto, visando ampliar mais o número de temas abordado, e mantendo o blog sempre atualizados. Sendo assim, dar continuidade a formação de professores, através de oficinas e minicursos a respeito do aplicativo em atividades educacionais.

Com isso a ferramenta visa contribuir significativamente para a melhoria do modo de ensino e aprendizagem, incentivando a utilização dos recursos computacionais nas atividades de sala de aula, além de compartilhar diferentes meios de estratégias na promoção da aprendizagem matemática.

## 3.4 Alice

A aprendizagem de conceitos computacionais pode nos trazer várias dificuldades em relação ao aprendizado de programação, que pode ir desde a compreensão dos problemas até a resolução algorítmica dos mesmos. A ferramenta pode auxiliar no ensino desses conceitos para os alunos, já que facilita a aprendizagem de forma divertida e permite que participe diretamente em relação ao seu aprendizado, além de se poder trabalhar em grupos (SILVA et al., 2014).

SILVA et al. (2014) acreditam que o ensino de computação nas escolas, pode trazer bons resultados. Além disso, pode colaborar no ensino de conceitos básicos de computação. Contando com o auxílio da ferramenta, podem ser aprendidas noções de lógicas de programação, além de permitir a programar de forma divertida e desenvolver as competências do pensamento computacional dos alunos, preparando os alunos para que possam viver na sociedade da informação que vivemos atualmente.

SILVA et al. (2014) destacaram que a Alice apresentou algumas limitações, como a dificuldade de interação o que pode deixar a compreensão dos conteúdos mais difícil, além de conter diversos bugs, travamento, problemas de salvamento e na inserção de sons. Mas também foram destacados alguns pontos positivos, pois propicia a aprendizagem de programação de forma divertida, com isso pode ajudar na compreensão de conceitos complexos, além de poder trabalhar conceitos de orientação a objetos (como objeto, classe e métodos) e com isso contribuir significativamente para o desenvolvimento do raciocínio lógico.

GONDIM et al. (2009) utilizaram um método que era baseado no aprendizado através da solução de problemas. Nesse método o aluno quando encontrava um problema, ele deveria procurar a solução para o mesmo, sem ter recebido exemplo de soluções para o problema que foi encontrado. O papel do professor se trata de dar instruções, citando fontes onde o aluno pode encontrar possíveis soluções do problema, acompanhar o que está sendo desenvolvido pelo aluno. Caso fuja do objetivo, o professor pode intervir e o redirecionar de volta para o caminho, mas o aluno é o único responsável por realizar a pesquisa, desenvolver e apresentar

uma solução. Pode se empregar o trabalho colaborativo, com isso os alunos podem ser divididos em grupos.

O laboratório que foi utilizado para as aulas das disciplinas contava com quatro mesas com capacidade para seis alunos, além de ter sido disponibilizado um computador que já possuía a Alice instalado. Os alunos foram divididos em duas turmas de vinte alunos, cada turma conta com diferentes professores e horários diferentes. A equipe era formada pelos professores, um monitor e dois estagiários. Os estagiários auxiliavam os professores na sala de aula, enquanto o monitor cumpria uma determinada carga horária no laboratório fora do horário das aulas, para que pudesse ajudar os alunos que estão com dificuldade de compreensão do conteúdo.

Para isso [GONDIM et al. \(2009\)](#) dividiram a disciplina em duas partes, a primeira contava com auxílio do ambiente Alice e a segunda utilizando o ambiente Java. Na primeira parte foram passados os conceitos iniciais sobre programação, exceto vetores e matrizes, já que a ferramenta ainda não possui um bom suporte sobre essas estruturas de dados, existe erros nessas estruturas.

Para que fosse possível realizar a apresentação da ferramenta foi contado com o auxílio de tutorias que estão disponíveis no programa, também foram desenvolvidos exemplos em sala, que foram acompanhados pelos alunos. Com o método que foram utilizados os conceitos eram passados aos poucos através de novos problemas, começando com os mais simples e cada vez ficava mais difíceis. Já que o primeiro problema bastava escolher um personagem da galeria e animá-lo da forma que desejar.

O desafio final era simular uma lanchonete onde o personagem deveria atender cinco mesas com vários clientes. Além de ter que realizar um controle sobre a quantidade de bebida que o freezer possui, com isso cada bebida era retirada, se estiver acabado a mesma no freezer deveria ser repostos. Então os alunos foram avaliados através da criatividade e de como foi a utilização dos comandos.

Com intuito de avaliar a ferramenta Alice, foram realizadas avaliações com os alunos, que contavam com prova escrita, correção dos problemas e acompanhamento dos alunos. A primeira avaliação foi realizada depois de três semanas de utilização da ferramenta, que tinha como objetivo avaliar o nível de familiaridade dos alunos com a ferramenta e quais os conhecimentos que os aluno já tinha adquiridos sobre programação com o auxílio do ambiente. Essa primeira avaliação contava com quatorze questões, sendo elas, seis sobre o nível de familiaridade, sete sobre os conceitos de programação que eram relacionados à programação clássica e uma questão subjetiva para saber a opinião dos alunos sobre o programa Alice.

No final do curso, foi realizado outro questionário com os alunos, para saber dos mesmos quais foram a contribuição da ferramenta para a aprendizagem de conceitos básicos. Então essa avaliação foi elaborada da seguinte maneira: oito questões objetivas, que contavam com um campo para o aluno expressar sua opinião e ainda contava com uma questão subjetiva para saber se o aluno utilizaria essa ferramenta como meio para auxiliar o ensino e o que ele faria de diferente na disciplina. Através disso foi dada total liberdade para os alunos criticarem a ferramenta, os métodos que foram utilizados, os recursos e também os professores.

As avaliações serviram para saber o nível de abstração que os alunos tiveram, além de servir para analisar a transição da ferramenta para uma linguagem tradicional, ou seja sem utilizar a montagem de blocos e sim a criação de código através de texto em uma linguagem de programação. Essa transição mostrou ser tranquila, já que com a Alice os alunos conseguiram desenvolver um bom conhecimento em resoluções de problemas e estavam motivados. Foi destacado ainda que o método utilizado contando com auxílio da ferramenta, foi muito importante para que os alunos tivessem sucesso na disciplina. Com isso, conseguiram como resultados uma baixa quantidade de desistência e reprovação.

No primeiro questionário, aplicado depois de um certo período de adaptação com a ferramenta, continha treze questões de múltipla escolha com o objetivo de avaliar a ferramenta. Foram considerados alguns pontos como suas funções, visual, facilidade e também o nível de conhecimento adquirido pelos alunos sobre os conceitos de programação. De forma geral, os alunos tiveram um bom aprendizado com a ferramenta, além de terem gostado bastante de utilizar a mesma.

Entretanto, parte dos alunos não gostaram de ferramenta, pois a mesma conta com alguns problemas como: bugs e lentidão, alguns erros na definição dos vetores, falhas em algumas funções e além de acontecer alguns travamentos. Mas mesmo com esses problemas foi destacado que os alunos mostraram, de modo geral, um bom aprendizado dos conceitos que foram trabalhados no ambiente Alice.

Ainda foi destacado que na última questão, a única descritiva, os alunos destacaram que a ferramenta é muito lenta e possui travamentos constantes. Além de ser difícil encontrar alguns recursos e comandos, o que pode tornar o desenvolvimento um pouco mais demorado. Foi destacado também pelos alunos que a movimentação dos personagens não era suave e do lado positivo foi destacado a semelhança dos comandos da ferramenta com a linguagem de programação tradicional, o que facilita a compreensão da lógica dos problemas.

O segundo questionário que foi aplicado no final do curso, contava com oito questões de múltipla escolha. Através do mesmo foi possível verificar que o nível de desenvolvimento dos alunos foi satisfatório, em relação aos conceitos de lógica algorítmica e da linguagem de programação. Com isso foi possível notar que mesmo a disciplina não tratando tópicos de orientação a objetos, os alunos mostraram ter compreendido sobre o que se tratava o assunto.

A nona questão pedia a opinião dos alunos sobre o método que utilizou a Alice como ferramenta de ensino. A maioria dos alunos disseram que utilizariam sim, mas todos ressaltaram que utilizaria esse método para iniciantes em programação. Com isso [GONDIM et al. \(2009\)](#) puderam concluir que nos métodos tradicionais, na maioria das vezes, os alunos podem não se interessar, ou seja, não ficam motivados.

Isso ocorre pois os alunos tem que se preocupar com questões de codificação e compilação, que são utilizadas nas linguagens de programação tradicionais. Para que os alunos não sofram com esses problemas foram desenvolvidos novos métodos e ferramentas que os acompanham. Com isso, as ferramentas poderiam tornar o ensino de programação mais intuitivo e atraente

para os alunos que estão iniciando seus estudos na área.

GONDIM et al. (2009) destacaram que os resultados foram satisfatórios. Mesmo não contando com um aumento significativo nas médias de notas, houve um baixo nível de reprovações e contou ainda com somente uma desistência. Com os questionários que foram aplicados, foi possível ver que os alunos conseguiram compreender bem sobre os conceitos que foram abordados. Ainda houve casos que foram além do conteúdo proposto. Com isso, os alunos acharam que o aprendizado se tornou mais fácil.

GONDIM et al. (2009) destacaram que a ferramenta deixou a ensino e aprendizagem mais intuitivo, o que acabou gerando nos alunos entusiasmo e motivação. Mas esse sucesso só foi possível por causa da metodologia de ensino que foi utilizada para a exploração da ferramenta. O método auxiliou muito no desenvolvimento dos alunos, o que se pode ver através do desenvolvimento e do ânimo do mesmo.

Ainda foi destacado que deve ser levado em consideração o perfil do aluno, para que a metodologia utilizada seja adequada, juntamente com a ferramenta que será utilizada para o apoio. Os alunos que estão ingressando no Ensino Superior no Brasil têm o mesmo perfil dos alunos do Ensino Médio, que são acostumados a receber as informações que são necessárias para resolvam um problema. Com isso, quando se tem o surgimento de um novo problema, os alunos não conseguem resolver de forma diferente na qual eles estão acostumados.

Através disso GONDIM et al. (2009), destacaram que o método que foi utilizado não foi muito aceito no início, sendo visto de forma negativa pelos alunos, já que se vinham obrigados buscar os conhecimentos pelos seus próprios meios. Alguns alunos acabaram transferindo esse descontentamento para o Alice. Através desse método os alunos se tornaram mais ativos e independente, além de trabalharem de forma colaborativa.

VALASKI e PARAISO (2012) utilizaram o Alice na disciplina de *Oficina de Programação* do curso de Sistemas de Informação, sendo contemplada para alunos do primeiro semestre do curso, que tem como objetivo reforças os conhecimentos sobre programação básica. Sendo assim, foram trabalhados conceitos como: variáveis, estruturas sequenciais, estruturas de seleção, estrutura de repetição, objeto, classe, métodos e eventos. Estes mesmos conceitos também eram abordados ao mesmo tempo na disciplina de Programação I.

De acordo com VALASKI e PARAISO (2012) os alunos que estavam matriculado na Oficina de Programação eram responsáveis por desenvolver jogos digitais. Esses jogos eram simples e que eram baseados em uma estrutura em Java, que eram previamente desenvolvido pelo o professor da disciplina. Já na disciplina de programação, os alunos começaram aprendendo os conceitos que eram desenvolvidos em Portugol e depois de passavam a criar programas em Java. Esses mesmos conceitos eram passados para os alunos na disciplina Oficina de Programação através do desenvolvimento de animações, contando com o auxílio do programa Alice. No primeiro semestre não foram trabalhando os conhecimentos da orientação a objeto, sendo trabalhados somente posteriormente.

VALASKI e PARAISO (2012) destacaram que para que fosse possível criar uma animação

no Alice é necessário inserir os objetos. É possível ver todos os objetos que estão inserido no cenário que estão organizados em uma *TreeView*, além disso um objeto pode ser composto por outros objetos. E cada uma desses objetos podem receber propriedade, métodos e funções próprias. As animações são programadas através de blocos de comandos, ou seja, basta arrastar e soltar qualquer função, método, função e as instruções, isso também ocorre com os objetos.

Para que fosse possível avaliar VALASKI e PARAISO (2012) elaboraram um instrumento de coleta de dados, que foi realizado com o intuito de avaliar a aprendizagem dos alunos sobre os conceitos básicos de programação através da utilização do Alice, sendo realizada no final da disciplina. Para que essa coleta de dados fosse possível, foi utilizado o Google Docs, que contava com sete questões.

A primeira questão tinha o objetivo de saber quais dos conhecimentos que tinham sido apresentados na disciplina, os alunos já possuíam o conhecimento antes do início da mesma. Através disso, observou que mais da metade dos alunos acreditava que não conhecia mais do que os conteúdos que foram ensinados antes do início da disciplina. Já na segunda questão pode-se destacar que posterior a disciplina, a maioria dos alunos julga ter conhecimento de pelo menos 50% dos conteúdos trabalhados na disciplina. Com isso, pode-se observar que os alunos conseguiram adquirir novos conhecimentos. Entretanto, somente com esse resultado não é possível saber se a ferramenta influenciou no aprendizado.

VALASKI e PARAISO (2012) puderam notar na questão três, que mais da metade dos alunos considerou que o Alice não ajudou no aprendizado de conceitos básicos de programação. Através da questão, quatro dos alunos foram indagados se recomendariam o uso da ferramenta para outros alunos. A maioria dos alunos não aconselha o uso do Alice para o ensino de programação, o uso é indicado somente para os alunos que declaram não ter nenhum conhecimento sobre programação. O uso da ferramenta também pode facilitar a compreensão dos conteúdos para os alunos que estão tendo dificuldade. Foi destacado que para os alunos que já possuem algum conhecimento ou não tem dificuldade no aprendizado dos conteúdos, a ferramenta pode os desmotivar.

Através da questões cinco e seis VALASKI e PARAISO (2012) tinham como objetivo identificar os pontos positivos e negativos da ferramenta. Dentre do pontos positivos podem-se destacar que a possibilidade de trabalhar conceitos básicos de programação de maneira mais lúcida e descontraída, além de permitir novos métodos de ensino evitando os métodos engessados dos métodos tradicionais. Ainda a ferramenta possibilita o entendimento de conceitos que podem ser abstratos no método tradicional.

De acordo com VALASKI e PARAISO (2012), outro ponto positivo foi abordagem à orientação a objetos, já que todos os elementos têm propriedade e métodos, sendo organizados como objetos. Com isso os alunos aprendem conceitos relacionados com a orientação a objeto de maneira natural. Além de auxiliar a desenvolver o raciocínio lógico. Ainda foi reforçada a ideia de que a Alice é mais indicado para um grupo específico de alunos, que são os que não tiveram contato algum com programação.

VALASKI e PARAISO (2012) afirmaram que os alunos apontaram como pontos negativos, como a dificuldade de interação com a interface. Também foi apontado que houve dificuldade na criação das animações utilizando os conceitos que foram ensinados. Além de apresentar vários bugs durante o desenvolvimento das animações, sendo eles: travamento, erro ao recuperar uma animação salva, execução da animação com problema. Outro fato que gerou dificuldade foi da ferramenta ser em inglês, dificultando a compreensão de algumas funcionalidades.

Os alunos ainda citaram como ponto negativo a diferença da ferramenta com as ferramentas de desenvolvimento comerciais. Mas esse último ponto negativo pode ser levado em conta questões econômicas e culturais. Além disso, alguns alunos já trabalharam ou desejam entrar rapidamente para o mercado de trabalho. Como os alunos desejam aprimorar os seus currículos e ter boas oportunidades de se inserir no mercado de trabalho, a utilização de uma ferramenta que possa ser utilizada no mercado de trabalho e não somente para fins educacionais podem os deixar mais motivados.

Através da questão sete VALASKI e PARAISO (2012) desejavam saber quais seriam os outros métodos que poderiam ser utilizados para auxiliar na aprendizagem dos conceitos básicos de programação. Foram descritos pelos alunos que a utilização de programa que tenha sido construído para entendimento do código fonte e conhecimento de como é possível melhorar a estrutura do programa. Os alunos ainda acreditam que é melhor aprender através do método tradicional, através de muitos exercícios em Portugol e em uma linguagem de programação específica.

Ainda foi feita uma análise dos impactos dessa atividade para reforçar os conceitos que eram ensinados na disciplina de Programação I. Essa disciplina é a que contém o maior nível de reprovação do primeiro semestre. Sendo que nas ultimas quatro turmas dos anos anteriores de 2012, tem uma média de aprovação de 53%. Se comparamos com o ano que foi utilizado o Alice para auxiliar no ensino teve uma taxa de aprovação de 55%, através desse dado não é possível concluir que ocorreu uma melhora significativa.

VALASKI e PARAISO (2012) concluíram que se tem uma alta quantidade de reprovação e desistência dos alunos do primeiro semestre dos cursos na área da Computação. Pois os alunos têm muitas dificuldades nas matérias de programação. Para que se possa melhorar a aprendizagem dos alunos estão sendo desenvolvidos diversos software com o intuito de auxiliar no ensinar dos alunos. Em relação ao Alice menos da metade dos alunos apontaram que a ferramenta auxilia no aprendizado, isso pode ter relação como o fato da maioria dos alunos já possuem conhecimento prévio. Então esses alunos acham melhor utilizar uma linguagem específica, que é considerado o melhor modo de desenvolver os conhecimentos sobre programação.

Pode-se concluir que a ferramenta Alice é mais indicada para o aluno que não possui nenhum conhecimento sobre programação, assim como alunos que estão encontrando dificuldades na compreensão dos conceitos de programação. Ainda pode-se observar que os alunos desejam entrar rapidamente para o mercado de trabalho, mesmo enquanto ainda são alunos de graduação. Nesse viés pode-se concluir que o Alice não contribuiu significativamente na aprovação da

disciplina. Entretanto, houve uma redução na evasão dos alunos da disciplina.

## 3.5 Code.org

[DANTAS e COSTA \(2013\)](#) apontaram que o ensino de programação pode ser útil no contexto escolar e trazer benefício para a vida dos alunos, pois contribui para desenvolvimento da criatividade do indivíduo. Indicou que o aluno se torna mais importante quando assume que o papel de criador, sendo assim, aumenta as suas habilidades gradativamente.

De acordo com [DANTAS e COSTA \(2013\)](#) o projeto Code.org vê através da programação um meio de desenvolver a aprendizagem, além de poder oportunizar o crescimento profissional. Desse modo, o aluno pode melhorar a forma como se comporta na decisão para solucionar o problema, podendo ser do trabalho ou da escola.

O Code.org tem como objetivo disseminar o ensino de programação nas escolas, gerando a oportunidade para que os alunos possam aprender sobre programação. Com isso, ainda defende que a informática deve fazer parte da grade curricular, juntamente com outras ciências que já fazem parte da mesma.

Através do estudo que foi realizado, [DANTAS e COSTA \(2013\)](#) puderam observar que a plataforma utiliza do método de que o aluno tem que construir o seu conhecimento. Quando o aluno é responsável por construir algo, ele pode desenvolver um conhecimento mais sólido, pois consegue ver os resultados dos seus trabalhos imediatamente.

A plataforma cumpre muito bem o seu papel de ensinar a programação, não só para pessoas que são da área, mas para todos que desejam aprender a programar. Com isso, pessoas de qualquer área conseguem aprender sobre o assunto, além de desenvolver o raciocínio lógico, resoluções de problemas e conceitos matemáticos, ou seja, os indivíduos não aprendem apenas os conceitos de programação, mas é possível que adquirir conhecimento sobre outras áreas ([DANTAS e COSTA, 2013](#)).

Através disso [DANTAS e COSTA \(2013\)](#) concluíram que o Code.org é uma excelente iniciativa, que pode alterar a visão sobre a Computação, que através dela pode auxiliar no aprendizado dos alunos. Pode trazer benefícios para os alunos, já que o mesmo pode desenvolver o seu conhecimento e adquirir novas habilidades.

O Code.org utiliza de vídeos para ensinar os conceitos de programação, na maioria dos casos de curta duração. Traz somente conhecimentos que são necessários para cumprir os objetivos das próximas fases. De acordo com [ZANATTA \(2015\)](#), através dos vídeos podemos adquirir conhecimento sem que conseguimos perceber, ou seja, através das imagens, áudios, alguma cena ou até mesmo uma parte da videoaula fica gravado no nosso subconsciente. No momento que encontramos alguma situação podemos nos recordar desses conhecimentos.

Segundo [ZANATTA \(2015\)](#), um vídeo de qualidade pode aguçar a curiosidade dos alunos sobre o assunto que foi introduzido através do mesmo. Motivando o aluno a buscar mais informações sobre o assunto, além de criar o desejo de realizar pesquisas e aprofundar os conhecimentos sobre o tema do vídeo. [ZANATTA \(2015\)](#) ainda destaca que os vídeos também

podem ser utilizados para reforçar conteúdos que já foram passados para os alunos, já que é um meio didático e pedagógico de transmitir o conhecimento, se mostra eficaz na transmissão dos ensinamentos.

De acordo com [GERALDES \(2014\)](#) o ensino de programação para crianças e jovens não é aceito por todos, existem especialistas da área de Tecnologia da Informação e educadores que são contra essa ideia. Alegam que os alunos necessitam de um raciocínio lógico perfeito, assim como também serem capaz de resolver problemas de alta complexidade, o que não ocorre com todos. Através disso, o ensino de programação nas escolas pode nos trazer péssimos resultados, já que os desenvolvedores que foram formados na escola não seriam de qualidade.

Alguns especialistas e educadores apresentam argumentos contra o ensino de programação no âmbito escolar. De acordo com eles, a inserção da computação muito cedo na vida das crianças afeta a infância e a juventude, o que pode causar desastres mentais nos alunos que não podem ser vistos. E muito educadores acreditam e defendem essa teoria, de que o ensino de programação prejudica a infância, perdendo assim a oportunidade de interagir com as outras crianças através das brincadeiras que fazem parte dessa fase ([GERALDES, 2014](#)).

Durante o estudo da programação é possível aprender mais do que somente programar. O que reforça a importância de ter os conhecimentos sobre os métodos que são utilizados para a programação de computadores. Deve-se destacar que o ensino desses conceitos nas escolas não deve atrapalhar o desenvolvimento social e o processo de aquisição de conhecimento do aluno.

Durante os anos escolares ocorre não somente o ensino dos alunos, mas também são construídas as relações sociais entre eles. Com isso, não se deve esquecer das brincadeiras e jogos que fazem parte da infância da criança, mesmo que atualmente esses jogos e brincadeiras tenham sido influenciadas pelas tecnologias ([GERALDES, 2014](#)).

Através disso, [GERALDES \(2014\)](#) termina a sua conclusão falando que as crianças não devem ser privadas de interações com outras crianças e com adultos, já que essa relação é importante para o desenvolvimento social do indivíduo, respeitando o desenvolvimento das crianças. Sendo assim, pode trazer vários benefícios sem trazer consequências para os alunos, não só para a programação, mas também no processo de aprendizado de outras matérias.

## 3.6 Blockly Games

O Blockly Games utiliza em seus jogos blocos de comando simples, que são responsáveis por fazer que o personagem realize alguma ação. Os primeiros níveis dos jogos são simples, mas a cada nível que passa se tornam mais difíceis, com a inserção de novos conceitos e estruturas. Muitos desses comandos podem deixar a programação bem mais simples ([JUNIOR, 2017](#)).

De acordo com [JUNIOR \(2017\)](#) os blocos de comando do Blockly Games representam comandos de programação em uma linguagem tradicional. Então com os comandos em blocos os alunos podem entender os conceitos de programação facilmente. Além disso, a ferramenta mostra aos alunos como ficam os códigos na linguagem de JavaScript, posteriormente ao término do nível.

É importante que os alunos tenham capacidade de resolver problemas e tomar decisões. Essas capacidades podem ser ensinadas na escola de ensino básico. Além disso, a lógica de programação inclui esses conceitos, que podem auxiliar os alunos a decidir qual área profissional vai seguir (JUNIOR, 2017).

JUNIOR (2017) acredita que pode-se ter uma disciplina que trabalhe os conceitos de lógicas de programação, aprendendo assim fundamentos suficientes para os alunos. Assim os alunos podem se desenvolver em várias áreas, não só as que estão ligadas à Computação. Através dos conceitos da Computação, os alunos podem compreender um determinado problema e conseguir uma maneira que o mesmo possa ser resolvido.

O Blockly Games auxilia no desenvolvimento do raciocínio lógico dos alunos, além de deixar os alunos motivados em aprender. Permite ainda que os alunos se desenvolvam de forma divertida, podendo assim os estimular a continuar aprendendo, mostrando a importância de que os alunos possam aprender de forma lúdica.

De acordo com GOMES et al. (2018) a programação muitas vezes pode não ser de fácil entendimento para os alunos que estão começando a aprender. Através disso podemos ver a importância de se utilizar meios para que possa se tomar mais fáceis o aprendizado desses conceitos, além de facilitar a comunicação sendo assim se torna útil para o professor e para o aluno.

GOMES et al. (2018) ainda nos falam que somente utilizar jogos para os ensinamentos nem sempre quer dizer que o aluno está aprendendo. Devemos escolher uma ferramenta que possa trazer resultados. Para que isso possa ser possível, é importante levar em conta para quais alunos será aplicada a ferramenta, além do nível escolar.

Com isso, GOMES et al. (2018) ainda destacam que a escolha da ferramenta que sejam divertidas pode ser bem difícil. Já que se tem que ser levado em conta os aspectos educacionais e deve se ter cuidado, além de se planejar como a ferramenta será utilizado nas aulas.

O Blockly Games é utilizado para ensinar programação para crianças de maneira divertida e de fácil utilização. De acordo com LUCRECIO (2016) a ferramenta demonstrou ser uma excelente opção, pois não utiliza de pseudocódigo. Além disso, os códigos em blocos se transformam em código funcional na linguagem Javascript, o que a difere das demais ferramentas.

Atualmente podemos ver que os alunos estão poucos motivados com o método que está sendo utilizado pelas escolas, o método tradicional. Essa desmotivação pode ser vista no Ensino Médio e Fundamental. Pode-se notar que os alunos não estão se interessando por esses métodos de ensino, se interessando mais por jogos eletrônicos e atividade de lazer (TRAVNIK e GIGLIO, 2017).

O algoritmo é um meio utilizado para resolver problemas reais. Além de melhorar o raciocínio, pois possibilita o processo de busca de meios para o solucionar, elaborando os passos para solucionar a tarefa. Com isso, o uso de ferramentas pode auxiliar no aprendizado dos conceitos de algoritmo de maneira eficaz (TRAVNIK e GIGLIO, 2017).

Atualmente podemos utilizar da técnica de gamificação, que permite aos alunos realizar

atividades do cotidiano com jogos, com o intuito de aprender sobre algum assunto. Os jogos são atividades livres, que permite que aluno aprenda de forma livre sem seguir um roteiro específico. É importante destacar que se tenha uma meta definida, ou seja, o que se deseja alcançar com essa atividade. Além de se determinar regras, que pode ser o caminho que deve ser seguido, ou as atividades que devem ser executadas, naquele nível. Uma característica da gamificação é o que os jogadores ganhem pontuações, prêmios ou outros tipos de incentivos (TRAVNIK e GIGLIO, 2017).

TRAVNIK e GIGLIO (2017) ainda ressaltam que a gamificação não se trata especificamente de criar jogos, mas a utilização dos elementos e a utilização da capacidade de fazer com que os usuários prendam a atenção no jogo. Com o Blockly Games é possível utilizar das características da gamificação, já que a ferramenta utiliza de jogos para ensinar a programação. Para que se possa motivar os alunos pode ser utilizado de premiações para os mesmos consigam cumprir o objetivo de cada nível. Além disso, os alunos vão aprendendo a linguagem JavaScript de maneira divertida e se sentem motivados a aprender.

## 3.7 Comparação

Como podemos ver na Tabela 3.1 algumas ferramentas possuem mais de uma linguagem, o Scratch, App Inventor, Code.org e Blockly Games possuem mais linguagens do que as citadas, pois são bem numerosas. Além de podemos ver para qual público essas ferramentas podem ser aplicadas, com isso pode se saber se ela se encaixa no público que se deseja. Com também podemos saber se elas utilizam somente linguagem visual, ou se gera uma linguagem de programação tradicional, ou seja, se mostra os códigos que foi desenvolvido na linguagem visual em linguagem de programação tradicional, que no caso foram todas em JavaScript.

Todas as ferramentas são gratuitas, além de possuir uma interface de fácil utilização, no caso do Alice isso se torna um pouco mais difícil pois se encontra somente em inglês, nesse caso é só a dificuldade de compreensão do idioma. Conseguimos identificar o contexto em que as ferramentas trabalham, ou seja, se com elas é possível criar jogos, histórias, animações e aplicativos. Além disso, podemos ver os conceitos que podem ser ensinados com as ferramentas.

Através da Tabela 3.2 podemos ver quais as vantagens e desvantagens que cada ferramenta apresenta para o ensino de programação, sendo possível identificar quais ferramentas nos trarão benefícios e quais podem trazer dificuldades para o ensino.

Tabela 3.1: Comparativo das ferramentas

<b>Ferramentas</b>	<b>Scratch</b>	<b>App Inventor</b>	<b>Alice</b>	<b>Code.org</b>	<b>Blockly Games</b>
<b>Gratuita</b>	Sim	Sim	Sim	Sim	Sim
<b>Linguagem</b>	Inglês, English, Português, Português Brasil, Espanhol, Francês, Italiano, Língua Indonésia, Idioma Malaio, Latina, Holandês, Polonês.	Inglês, Espanhol, Português, Português Brasil, Italiano, Francês. Holandês, Sueco.	Inglês	Inglês, Espanhol, Português, Português Brasil Italiano, Italiano, Frances, Alemão, Dinamarquês, Holandês, Finlandês, Bósnio, Irlandês.	Inglês, Espanhol, Português, Português Brasil Francês, Italiano, Alemão, Polonês, Húngaro, Holandês, Língua indonésia, Idioma Malaio.
<b>Público alvo</b>	8 a 16	12 a 18	16+	4 a 18	10 a 18
<b>Linguagem de Programação</b>	Visual	Visual	Visual	Visual e JavaScript	Visual e JavaScript
<b>Interface</b>	Simples	Simples	Simples	Simples	Simples
<b>Utiliza de mídias</b>	Imagens, sons, gráficos e músicas	Imagens, sons e vídeos	Não	Vídeos	Não
<b>Contexto</b>	Jogos e Historias	Aplicativo	Historia e Animação	Vídeos e Jogos	Jogos e Animações
<b>Conceito</b>	Funções, Estruturas de controle, Estruturas de repetição e Estrutura condicionais.	Estruturas de repetição, Estruturas condicionais, Lista e Banco de dados.	Estruturas condicionais, Estrutura de repetição, Métodos, Parâmetros e Funções.	Conceitos de lógica, Laços de repetição, Estrutura condicionais e Funções.	Estruturas de repetição, Estruturas condicionais e Funções.

Tabela 3.2: Vantagens e Desvantagens das Ferramentas

<b>Ferramentas</b>	<b>Vantagens</b>	<b>Desvantagens</b>
<b>Scratch</b>	<p>Desenvolve o raciocínio lógico.  Desenvolve o pensamento criativo.  Desenvolve o trabalho em grupo.  Ensina programação de maneira divertida.  Facilidade na criação dos jogos ou histórias.</p>	<p>O editor de imagem possui poucos recursos.  Qualidade das imagens de baixa.  Não possui um editor de áudio, além de ser difícil de ser utilizado alguns arquivos não executa.  A utilização de muitos recursos provoca um baixo desempenho.</p>
<b>App Inventor</b>	<p>Desenvolver o raciocínio lógico.  Desenvolve o pensamento criativo.  Desenvolve o trabalho em grupo.  Possui blocos já programados de funções matemáticas.  Facilidade para realizar teste.</p>	<p>Variáveis globais na verdade são locais.  O Banco de dados é específico de uma tela.</p>
<b>Alice</b>	<p>Desenvolve o raciocínio lógico.  Desenvolve o trabalho em grupo.  Desenvolve o pensamento criativo.  Ensina de programação de maneira divertida.  Orientação a objetos.</p>	<p>Somente em Inglês.  Difícil interação.  Travamentos.  Lenta.  Problemas de salvamento.</p>
<b>Code.org</b>	<p>Desenvolve o pensamento criativo.  Ensino de programação de maneira divertida.  Desenvolve o raciocínio lógico.  Transforma o código em blocos para JavaScript.</p>	<p>O curso offline está disponível somente em inglês.</p>
<b>Blockly Games</b>	<p>Desenvolve o pensamento criativo.  Desenvolve o raciocínio lógico.  Ensino de programação de maneira criativa.  Pode ser utilizado em diferentes plataformas (computadores, celulares e etc.).  Transforma o código em blocos para JavaScript.  Pode ser desenvolvido código em JavaScript.  Pode ser utilizado Offline.</p>	<p>Não possui explicações sobre o conteúdo na ferramenta.</p>

---

## *Conclusão*

---

O mundo está ligado com a tecnologia e os alunos lidam com ela a todo momento. Também é importante que os alunos aprendam a lidar com a grande quantidade de informação que se tem disponível na rede, desenvolvendo a capacidade de analisar as informações, buscando a informação correta.

O professor pode utilizar o sistema de erros e acertos para que os alunos consigam aprender o conteúdo que está sendo ensinado. Mas para que isso ocorra, ele precisa conscientizar o aluno de que o erro não é um empecilho no seu aprendizado. Assim, o professor deve buscar meios para que os alunos não se desmotivem ao cometer erros e os incentivem a continuar tentando.

A escolha da ferramenta deve ser uma etapa cuidadosa do processo de ensino, pois em vez de ajudar pode causar o efeito contrário. Deve ser feito o planejamento das aulas, o levantamento de quais alunos serão o público alvo e realizar uma análise da ferramenta. Este trabalho de conclusão de curso apresenta uma descrição e análise dessas ferramentas.

O Scratch demonstrou ser uma ferramenta que auxilia no ensino de programação. Através dela, os alunos podem aprender conceitos da programação de computadores e do pensamento computacional. Além disso, é possível desenvolver o raciocínio lógico, o pensamento criativo e a capacidade de resolução de problemas.

O *App Inventor* corresponde muito bem com o que é esperado, pois trouxe diversos benefícios para os alunos, sendo eles não só específicos da área da Computação. Outra área impactada foi a da Matemática, já que os alunos podem desenvolver aplicativos que resolvem problemas matemáticos. Por meio dessa ferramenta, os alunos são capazes de desenvolver habilidades como a resolução de problema, do processo criativo e na criação de simulações.

A Alice é uma ferramenta indicada no auxílio do aprendizado dos alunos que não tiveram contato anteriormente com a programação. Entretanto, o que a diferencia das outras de maneira negativa é que a ferramenta apresenta momentos instáveis.

---

A plataforma Code.org é um excelente meio para o ensino de programação. Pode ser utilizada por alunos de diversas idades, já que o conteúdo da ferramenta é de fácil compreensão. Apresenta cursos offline, o que facilita o seu uso em alguns casos. Todavia, sua interface só está disponível em inglês, com isso o professor e os alunos podem ter dificuldades de utilizar esse recurso. Outra vantagem apresentada é que os alunos já podem estar sendo inseridos em uma linguagem de programação, pois a ferramenta transforma o código que foi desenvolvido em blocos em JavaScript.

O Blockly Games é uma ferramenta muito interessante para o ensino de programação, além de cumprir muito bem o seu papel de auxiliar no ensino. Além de mostrar os códigos desenvolvidos em blocos, a ferramenta os transforma para uma linguagem de programação, sendo ela JavaScript. O aluno ainda pode começar a desenvolver o seu código em JavaScript através da ferramenta, podendo aprender uma linguagem de programação específica.

Neste trabalho podemos constatar que as ferramentas podem auxiliar no ensino de programação e pensamento computacional. Mas para que isso possa acontecer, deve ser levado em consideração quais alunos utilizarão a ferramenta, além de realizar o planejamento das atividades. Essas ferramentas não devem ser utilizadas sozinhas, deve haver a combinação de conceitos teóricos juntamente com aplicação da ferramenta.

## Referências Bibliográficas

---

ARANTES, F. L.; DA S. FERREIRA, J. M. L.; RIBEIRO, P. E. J. Scratch - um primeiro olhar. In: *Texto Livre: Linguagem e Tecnologia*, 2015, p. 1–16.

BARBOSA, E. S.; BATISTA, S. C. F.; BARCELOS, G. T. App inventor: Análise de potencialidades para o desenvolvimento de aplicativos para matemática. In: *Congresso Integrado da Tecnologia da Informação*, 2015, p. 1–12.

BRACKMANN, C.; BARONE, D.; CASALI, A.; MUÑZ-HERNANDEZ, R. B. S. Computational thinking: Panorama of the americasl. In: *IEEE*, 2016, p. 1–6.

BRESSAN, M. L. Q.; AMARAL, M. A. Avaliando a contribuição do scratch para a aprendizagem pela solução de problemas e o desenvolvimento do pensamento criativo. In: *Revista Intersaberes*, 2015, p. 509–526.

CORDEIRO, F. App inventor: Guia de criação de apps. 2017.

Disponível em <https://www.androidpro.com.br/blog/desenvolvimento-android/app-inventor/>

DANTAS, R. F.; COSTA, F. E. A. 5º simpósio hipertexto e tecnologia na educação, 1º colóquio internacional de educação com tecnologia. In: *CODE: O ensino de linguagens de programação educativas como ferramentas de ensino/aprendizagem*, 2013, p. 1–14.

DUDA, R.; SILVA, S. D. R. Desenvolvimento de aplicativos para android com uso do app inventor: Uso de novas tecnologias no processo de ensino aprendizagem em matemática. 2015, p. 310–323.

DUKE, U. Tutoriais alice versão 3 verão 2013-2017. Acessado em 20 jun. 2018, 2013.

Disponível em <https://www2.cs.duke.edu/csed/web/alice09/tutorialsAlice3.php>

GERALDES, W. B. Programar é com para as crianças? uma visão crítica sobre o ensino de programação nas escolas. In: *Texto Livre Linguagem e Tecnologia*, 2014, p. 105–117.

GOMES, H. A. C.; MELO, J. M.; FARRAPO, H. P.; BONATES, M. F.; COUTINHO, E. F. Revista sistemas e mídias digitais (rsmd). In: *Descrição e Comparação de Jogos Digitais para Auxiliar no Ensino de Programação*, 2018, p. 1–14.

GOMES, T. C. S.; MELO, J. C. B. App inventor for android: Uma nova possibilidade para o ensino de lógica de programação. In: *II Congresso Brasileiro de Informática na Educação (CBIE 2013), Workshops (WCBIE 2013)*, 2013, p. 620–629.

GONDIM, H. W. A. S.; AMBRÓSIO, A. P.; COSTA, F. M. Uma experiência no ensino de algoritmos utilizando ambientes visuais de programação 3d. In: *XVII Workshop sobre Educação em Informática*, 2009, p. 1–10.

GROVER, S.; PEA, R. Computational thinking in k-12: A review of the state of the field. In: *Educational Researcher*, 2013, p. 37–43.

IFSC. Projeto forma-engenharia. Online; accessed 13 jun. 2018, 2014.

Disponível em <http://programacaoscratchupf.blogspot.com/2012/05/ola-pessoal-bem-vindos-esse-blog.html>

JUNIOR, E. J. O. PLATAFORMA DE SUPORTE ÀS FERRAMENTAS DE APRENDIZAGEM PARA O ENSINO DE LÓGICA DE PROGRAMAÇÃO NA EDUCAÇÃO BÁSICA. Monografia (Pós-Graduação em Educação), UFVJM (UNIVERSIDADE FEDERAL DOS VALES DO JEQUITINHONHA E MUCURI), Diamantina, Brazil, 2017.

JÚNIOR, R. P. M.; BONIATI, B. B. Anais do eati - encontro anual de tecnologia da informação e semana acadêmica de tecnologia da informação. In: *LogicBlocks: Uma Ferramenta para o Ensino de Lógica de Programação*, 2015, p. 63–70.

LEE, I.; MARTIN, F.; DENNER, J.; COULTER, B.; ALLAN, W.; ERICKSON, J.; MALYN-SNITH, J.; WERNER, L. Computational thinking for youth in practice. In: *ACM Inroads*, 2011, p. 32–37.

LUCRECIO, A. I. Comparação e Aplicação de Diferentes Ferramentas para Ensino de Programação para Crianças. Monografia (Trabalho de Conclusão de Curso), UFSC (UNIVERSIDADE FEDERAL DE SANTA CATARINA), ARARANGUÁ, Brazil, 2016.

MARTINS, R. S.; REIS, R. J. A.; MARQUES, A. B. Inserção da programação no ensino fundamental: Uma análise do jogo labirinto clássico da code.org através de um modelo de avaliação de jogos educacionais. In: *V Congresso Brasileiro de Informática na Educação (CBIE 2016), XXII Workshop de Informática na Escola (WIE 2016)*, 2016, p. 121–130.

NETO, V. S. M. A utilização da ferramenta scratch como auxílio na aprendizagem de lógica de programação. In: *II Congresso Brasileiro de Informática na Educação (CBIE 2013)*, 2013, p. 1–10.

PBWORKS Métodos incorporados. Online; accessed 20 jun. 2018, 2012.  
Disponível em <http://alice3.pbworks.com/w/page/57585609/Built-in%20Methods>

RIBEIRO, A. S. M.; DE B. V. RODRIGUES, F.; PERREIRA, V. M. D. S. Conhecendo o scratch e suas potencialidades pedagógicas. In: *I Seminário Internacional de Inclusão Escolar: práticas em dialogo*, 2014, p. 1–6.

SACHETTO, A. Oficina de programação - htpc 14/10. Online; accessed 20 jun. 2018, 2014.

Disponível em <https://flaminioangel.wordpress.com/2014/10/14/oficina-de-programacao-htpc-1410/>

SILVA, G. E.; BARBOSA, A. F.; NETO, S. R. S.; LOPES, R. H. O.; RODRIGUES, A. N. Análise de ferramentas para o ensino de computação na educação básica. In: *XXXIV Congresso da Sociedade Brasileira de Computação - CSBC*, 2014, p. 1495–1504.

SÁPIRAS, F. S.; VECCHIA, R. D.; MALTEMPI, M. V. Utilização do scratch em sala de aula. 2015, p. 1–16.

TRAVNIK, G. R. R. P.; GIGLIO, G. P. M. In: *UTILIZAÇÃO DE GAMEFICAÇÃO NA IMPLEMENTAÇÃO DE UMA FERRAMENTA DE APOIO AO APRENDIZADO DE ALGORITMOS*, 2017, p. 1–17.

UPF O que é o scratch? Online; accessed 13 jun. 2018, 2018.  
Disponível em <http://programacaoscratchupf.blogspot.com/2012/05/ola-pessoal-bem-vindos-esse-blog.html>

VALASKI, J.; PARAISO, E. C. Limitações da utilização do alice no ensino de programação para alunos de graduação. In: *23º Simpósio Brasileiro de Informática na Educação*, 2012, p. 1–10.

WEISSHAHN, Y.; DUARTE, A. I. R.; CAVALHEIRO, S.; FOSS, L.; AGUIAR, M.; BOIS, A. D.; FLEISCHMANN, A. M. P.; REISER, R. H.; DE B. PIANA, C. F. Pensamento computacional no brasil: O estado da arte. 2015, p. 1–6.

WING, M. J. Computational thinking: What and why? 2010, p. 1–6.

ZANATTA, A. C. Programação de Computadores para Crianças Metodologia do CODE CLUB Brasil. Monografia (Bacharel em Tecnologias da Informação e Comunicação.), UFSC (Univercidade Federal de Santa Catarina), Araraguá, Brazil, 2015.

## *Apêndice A*

---

---

### Lista de ferramentas 1 - Alice

- 2 - AMBAP
- 3 - App Inventor
- 4 - ASA
- 5 - ATMUF
- 6 - AWTM
- 7 - Blockly
- 8 - BlueJ
- 9 - Cargo-Bot
- 10 - CC50
- 11 - Code.org
- 12 - Codecademy
- 13 - CodeCombat
- 14 - Codeschool
- 15 - G-Portugol
- 16 - Greenfoot
- 17 - Hackety Hack
- 18 - Hopscotch
- 19 - Hour of Code
- 20 - Javatool
- 21 - Jeliot
- 22 - Kids Ruby
- 23 - Kodable
- 24 - Kodu

- 25 - Lego Mindstorms
- 26 - Lissa Explains it all
- 27 - Minecraft
- 28 - Moocs
- 29 - Portugal 2.1
- 30 - Processing
- 31 - Programaê
- 32 - Robocode
- 33 - Robomind
- 34 - Robotopia
- 35 - Scratch
- 36 - Sparki
- 37 - StarLogo TNG
- 38 - Try Ruby
- 39 - Tuts+
- 40 - Tynker
- 41 - Visualg
- 42 - Webportugol