

Avaliação das Ferramentas Utilizadas em um Curso de Programação para Crianças: Percepções dos Alunos e seus Instrutores

Caio de Oliveira Cafiero, João Victor Moreira, Janaina Gomide

¹Universidade Federal do Rio de Janeiro (UFRJ) - Campus Macaé
Macaé - RJ - Brasil

{caioocafiero, jvmoreira97}@gmail.com, janainagomide@macae.ufrj.br

Abstract. *The aim of this article is to evaluate tools focused on computer coding for children, under the perspective of instructors and students. Five different tools, which approach written and block coding, were used by forty children. When it comes to the educational platforms, they were evaluated under three aspects: tool inspection to verify its adequacy to the good practices of educational softwares, instructors testimony about their teaching experience and children report on use of the tool. It was observed that while some platforms had excellent acceptance, some generated differences on the monitors' and students' evaluation.*

Resumo. *Este artigo tem como objetivo avaliar ferramentas voltadas ao ensino de programação para crianças sob a perspectiva de instrutores e alunos. Cinco ferramentas diferentes, que abordam linguagem escrita e em blocos, foram utilizadas por quarenta crianças. A avaliação das plataformas educacionais foi composta por três aspectos: inspeção das ferramentas para verificar sua adequação às boas práticas de softwares educacionais, depoimento dos instrutores sobre a experiência de ensino e relato das crianças quanto ao uso de cada ferramenta. Observou-se que enquanto algumas plataformas tiveram excelente aceitação, outras geraram divergências na avaliação.*

1. Introdução

Diferentes linguagens de programação como Java, C++ e Python são utilizadas nos cursos de graduação e nas empresas para o desenvolvimento de softwares e aplicativos. Tais linguagens exigem do usuário conhecimentos que abrangem não só conceitos de computação, mas também cuidados com a construção do código, podendo causar dificuldade aos iniciantes em programação.

O pensamento computacional, desenvolvido ao longo da aprendizagem da programação, é uma habilidade fundamental para todos, não apenas para cientistas da computação, [Wing 2006]. Na Inglaterra, por exemplo, as disciplinas de programação já pertencem ao currículo escolar e abordam vários conceitos como lógica, pensamento computacional e algoritmos, [Paula et al. 2014]. No Brasil, há uma proposta de currículo para inclusão dessa disciplina na educação básica desenvolvida pela Sociedade Brasileira de Computação, [Raabe et al. 2017].

Um dos desafios do ensino de programação para crianças e jovens é a escolha adequada das ferramentas utilizadas no ensino para que os alunos se sintam confortáveis

com as tarefas e despertem o desejo pelo aprendizado, [Aono et al. 2017]. É interessante, durante as aulas, ressaltar aos alunos a possibilidade de programar algo concreto como foi proposto em [Santos et al. 2017]. Nesse contexto, a programação em blocos é a abordagem utilizada por diversas plataformas destinadas ao ensino de programação para esse público. Os blocos são como peças de quebra-cabeça que se conectam formando o código do programa. Para cada tipo de dado e estrutura de controle há blocos com formatos diferentes que são especialmente moldados para serem encaixados somente quando permitido, prevenindo erros de sintaxe, [Bau et al. 2017]. Exemplos de plataformas que usam essa abordagem são *Scratch*¹, *Monster Coding*² e *Hour of Code*³.

Existem também ferramentas que permitem aos usuários alternarem livremente entre a programação escrita e blocos, o que possibilita uma transição gradual de uma forma de programação para a outra, como a ferramenta *Pencil Code*⁴. A lógica já assimilada nos blocos pode ser transposta de forma que os usuários compreendam mais facilmente a sintaxe utilizada e a eficiência do texto. Entretanto, tal fator exige equilíbrio por parte da ferramenta para que os erros de construção textual sejam tratados de forma eficiente na transposição de uma abordagem para outra [Bau et al. 2017].

Há também plataformas que não utilizam blocos e são voltadas para crianças, por exemplo o *Code Monkey*⁵ que trabalha exclusivamente com linguagem escrita. Nessa ferramenta para saber quais comandos existem, há botões que quando apertados introduzem o código escrito na tela.

O objetivo dessa pesquisa é analisar tais plataformas sob a perspectiva de instrutores e crianças. A análise envolve três aspectos: (1) Inspeção das ferramentas para verificar sua adequação às boas práticas de softwares educacionais; (2) Depoimento dos instrutores sobre a experiência de ensino; (3) Relato das crianças quanto ao uso de cada ferramenta, suas preferências e dificuldades.

Alguns trabalhos como em [Kim and Ko 2017] e [Gouws et al. 2013] analisaram diversos aspectos e dimensões de várias plataformas *online* de ensino de programação, fóruns e plataformas de cursos à distância. [da Silva Junior and França 2017] desenvolveram uma análise comparativa entre ferramentas de ensino de programação voltadas ao público geral. Nesse artigo são analisadas ferramentas de programação para crianças envolvendo o relato de alunos e seus instrutores.

2. Ferramentas Utilizadas

A seguir, são apresentadas as cinco ferramentas avaliadas e suas particularidades. *Hour of Code* é um projeto desenvolvido pelo *Code.org* que consiste na realização de atividades com duração de uma hora cujo objetivo é mostrar que qualquer pessoa é capaz de aprender programação⁶. A plataforma contém diversos tutoriais que utilizam linguagem em blocos e explicam conceitos básicos de programação com vídeos explicativos entre uma tarefa e outra. Os temas das atividades envolvem assuntos conhecidos como: Mi-

¹Scratch - link: <https://scratch.mit.edu/>

²Monster Coding - link: <http://monstercoding.com/>

³Hour of Code - link: <https://br.code.org/learn>

⁴Pencil Code - link: <https://pencilcode.net/>

⁵Code Monkey - link: <https://www.playcodemonkey.com/blog/hour-of-code/coding-adventure/>

⁶<https://hourofcode.com/pt#faq>

necraft, Angry Birds, Star Wars, Moana e Frozen. A identificação do usuário com esses temas despertam seu interesse e é um dos fatores positivos para o sucesso da ferramenta, [Godinho et al. 2017].

Scratch é uma plataforma que utiliza a linguagem em blocos onde é possível criar histórias, animações e jogos. A ferramenta possui uma biblioteca com diversos personagens, sons e imagens a serem usados livremente pelo usuário. Ao usar o Scratch, não há um tutorial autoguiado e a criança é estimulada a desenvolver suas próprias ideias e a usar a criatividade para resolver um mesmo problema proposto de formas diferentes, [Vorderman et al. 2014] e [de Oliveira et al. 2016].

Pencil Code permite que os alunos alternem livremente entre a programação escrita e blocos. Essa abordagem possibilita que os alunos façam a transição gradual de blocos para texto à medida que se familiarizam com a sintaxe. Ele tem sido usado para permitir que programadores iniciantes de todas as idades criem programas em JavaScript e CoffeeScript, [Bau et al. 2015].

A ferramenta *Monster Coding* utiliza a programação em blocos e contém um tutorial autoguiado no qual a criança cria um personagem, um monstro, e deve dar comandos para ele realizar. Ao longo do tutorial aparecem questões a serem resolvidas pelo usuário, testando seu conhecimento.

Code Monkey é uma ferramenta que utiliza exclusivamente a linguagem escrita, fator interessante de ser analisado em contraponto com a abordagem utilizada pelas demais ferramentas. O usuário pode, também, utilizar botões que adicionam texto à caixa de programação, não precisando necessariamente digitar todo o código utilizando. Ao longo do tutorial autoguiado, a criança tem como principal objetivo utilizar comandos para fazer com que o macaco pegue todas as bananas na tela.

3. Avaliação das Ferramentas

A avaliação das ferramentas foi realizada sob perspectiva de nove instrutores e quarenta crianças que participaram de um curso de doze horas de duração durante um período de três semanas. Os alunos foram divididos em cinco turmas e em todas as turmas havia presença de no mínimo dois monitores responsáveis. Os instrutores são graduandos de cursos de engenharia da UFRJ - Campus Macaé que participam do projeto de extensão "Aprenda a Programar Jogando"⁷ com foco de ensino de programação para crianças. As aulas do curso abordaram os conceitos básicos de programação e cada aula envolvia a explicação de um conceito e a execução de uma das ferramentas.

Foram considerados três aspectos: (1) a inspeção das ferramentas para verificar sua adequação às boas práticas de softwares educacionais; (2) o depoimento dos instrutores sobre a experiência de ensino; e (3) o relato das crianças quanto ao uso de cada ferramenta. A seguir, cada aspecto é apresentado em detalhe.

3.1. Boas Práticas para Software Educacional

Para analisar as boas práticas para software educacional de programação foi feita uma adaptação das "Dez Regras de Ouro" propostas em [Jha and Duffy 2002], que analisam

⁷Aprenda a Programar Jogando - <https://www.aprendaprogramar.macaue.ufrj.br/>

a efetividade de ensino de outros tipos de softwares educacionais. As regras adaptadas estão listadas na Tabela 1.

Tabela 1. As ‘Dez Regras de Ouro’ para os softwares de ensino de programação.

Regra 1	A ferramenta relaciona de forma relevante o que é ensinado com o que é efetivamente utilizado na programação computacional.
Regra 2	A ferramenta utiliza métodos de ensino com embasamento científico.
Regra 3	A ferramenta faz uso de recursos de hipermídia e hipertexto, com sons, animações e imagens que se relacionam de maneira interativa.
Regra 4	A interface da ferramenta é instigadora, interessante e agradável para o usuário.
Regra 5	O uso de multimídia da ferramenta é apropriado, com textos, gráficos, imagens ou vídeos que auxiliam na orientação e direcionamento dos usuários.
Regra 6	Os desafios propostos simulam efetivamente problemas reais presentes na prática de programação.
Regra 7	A ferramenta estimula o usuário a analisar o problema criticamente, para que ele possa então aplicar os conceitos de programação necessários e chegar à solução.
Regra 8	A ferramenta é amigável ao usuário e de fácil navegação.
Regra 9	Dados os benefícios oferecidos pela ferramenta, ela é definida como apropriada para uso no que tange ensino de programação.
Regra 10	A ferramenta pode ser definida como de fácil manutenção, apresentando abertura para novas atualizações.

Os nove instrutores foram apresentados às regras e solicitados a responder se concordam que a regra foi atendida pela ferramenta (C), se discordam sobre o cumprimento da regra (D), ou se são indiferentes (I). A avaliação foi feita ao final do curso, depois que todas as ferramentas já haviam sido utilizadas pelos alunos e o resultado com as respostas pode ser visualizado na Tabela 2.

A partir da observação da Tabela 2 é possível notar que as ferramentas *Code Monkey*, *Hour of Code* e *Scratch* foram as que mais atenderam às regras, enquanto *Monster Coding* e *Pencil Code* despertaram divergências nas respostas dos monitores, com o número de discordâncias e indiferenças maior que o de concordâncias.

A regra um, sobre a ferramenta relacionar de forma relevante o que é ensinado com a programação computacional, foi cumprida por todas as ferramentas de acordo com a maioria dos monitores. As plataformas foram, quase sempre, consideradas como indiferentes às segunda e décimas regras que falam sobre a ferramenta utilizar métodos com embasamento científico e ser de fácil manutenção, respectivamente. As regras três, quatro, cinco, sete, oito e nove, são atendidas pelas ferramentas *Hour of Code*, *Scratch* e *Code Monkey*. Já para as ferramentas *Pencil Code* e *Monster Coding*, os monitores não concordaram e alguns acreditam que elas não atendem às essas regras.

Sobre a regra seis, foi possível observar que as respostas variam entre os monitores principalmente para a ferramenta *Scratch*. Acredita-se que essa variação ocorra por depender da atividade realizada nessa plataforma, que pode abordar ou não problemas

Tabela 2. Resposta dos nove instrutores quanto às “Dez Regras de Ouro para Softwares Educacionais”: concordam (C), discordam (D) ou indiferentes (I).

	Hour of C.			Scratch			Pencil C.			Monster C.			C. Monkey		
	C	I	D	C	I	D	C	I	D	C	I	D	C	I	D
Regra 1	9	0	0	9	0	0	8	1	0	6	2	1	9	0	0
Regra 2	4	5	0	3	5	1	2	6	1	2	7	0	5	4	0
Regra 3	8	1	0	9	0	0	3	4	2	7	2	0	9	0	0
Regra 4	9	0	0	8	1	0	6	2	1	3	2	4	9	0	0
Regra 5	8	1	0	6	1	2	2	3	4	2	2	5	9	0	0
Regra 6	8	0	1	4	4	1	5	2	2	5	2	2	7	1	1
Regra 7	8	1	0	8	0	1	6	1	2	3	2	4	7	2	0
Regra 8	9	0	0	8	1	0	2	3	4	1	2	6	9	0	0
Regra 9	9	0	0	9	0	0	8	1	0	3	2	4	9	0	0
Regra 10	3	6	0	1	8	0	1	8	0	1	6	2	3	6	0
Total	75	14	1	65	20	5	43	31	16	33	29	28	76	13	1

reais da programação.

A ferramenta *Monster Coding* é a com a pior avaliação pelos monitores. As regras cinco e oito, foram consideradas pela maioria como não cumpridas.

3.2. Relato dos Instrutores

Os instrutores responderam a um questionário com perguntas sobre as plataformas e também escreveram um relato com sua opinião sobre cada uma delas. Conforme ressaltado em [Petri and Gresse von Wangenheim 2017], questionários são comumente utilizados para esse tipo de análise. O questionário continha seis perguntas, que avaliavam a didática, variedade de conteúdo, facilidade de uso e capacidade de captar interesse do usuário. As respostas são apresentadas a seguir.

P1 - "Na sua opinião, em qual ferramenta os conceitos abordados durante as aulas foram melhor explicados? Por que?"

Nela, assim como na primeira pergunta, todos os monitores relataram que o *Scratch* abordava melhor os conceitos em relação às outras plataformas. Foram destacadas a interface da ferramenta e aceitação pelos alunos, como na resposta: *"Scratch, pois a ferramenta apresenta uma interface simples e intuitiva. A maneira com que os blocos são divididos em sessões (movimento, controle, aparência etc) facilita a aplicação dos conceitos abordados em aula, especialmente porque essas têm nomes semelhantes aos conceitos. Pode-se acrescentar, ainda, o fato da boa recepção por parte dos alunos com a plataforma incentivá-los a aprender."*

P2 - "Na sua opinião, em qual ferramenta os conceitos abordados durante as aulas foram pior aplicados? Por que?"

Nessa questão houve uma discordância por parte dos monitores, seis deles acreditam que o *Monster Coding* é a plataforma que pior aborda os conceitos apresentados, enquanto três dizem ser o *Code Monkey*. O principal argumento daqueles que responderam *Monster Coding* foi relacionado à tradução ruim, como reforçado na seguinte resposta: *"Monster Coding. Não fosse pela tradução mal feita, certamente não haveria o que citar"*

aqui.". Já os que optaram pelo *Code Monkey*, destacaram é o tutorial é muito extenso e não era possível abordar todos os conceitos de programação durante o tempo da aula.

P3 - "Em relação à didática de ensino, qual plataforma cumpre melhor a função?"

Sete monitores escolheram a plataforma *Scratch* e dois monitores optaram pela *Hour of Code*.

P4 - "Em relação à variedade de conteúdo disponível, qual plataforma cumpre melhor a função?"

Assim como na pergunta anterior, sete monitores participantes optaram pelo *Scratch*, enquanto que os outros dois optaram pela *Hour of Code*.

P5 - "Em relação à facilidade de uso (interface intuitiva, fácil compreensão, etc), qual plataforma cumpre melhor a função?"

As respostas foram variadas mostrando que os instrutores do curso ficaram com opiniões mais divididas. Quatro deles optaram pela *Hour of Code*, três pelo *Scratch* e os dois restantes pelo *Code Monkey*.

P6 - Em relação ao interesse gerado nos alunos, qual plataforma cumpre melhor a função?"

A resposta foi unânime e os nove monitores escolherem o *Scratch* como sendo o responsável por gerar o maior interesse nos alunos durante as aulas.

Os monitores escreveram também um relato com pontos positivos e negativos de cada ferramenta e os comentários mais frequentes estão resumidos na Tabela 3.

3.3. Relato das Crianças

Quarenta crianças com idade entre seis e doze anos participaram do curso de programação. Ao final do curso foi aplicado um questionário, que avaliaria as ferramentas sob a perspectiva dos alunos, sendo similar ao utilizado em [Duarte et al. 2017]. Para tal, foram analisados aspectos que levavam em consideração o divertimento e capacidade de captação de interesse do aluno, sendo esse fator importante para motivar o aprendizado [Gomes et al. 2017].

O questionário iniciava com a pergunta "*Você gostou de usar a ferramenta?"* e as opções de respostas eram: "*Gostei muito*", "*Gostei*" e "*Não gostei*". Essa pergunta foi feita para todas as cinco ferramentas e resposta se encontra na Figura 1. Pôde-se perceber que a maioria dos alunos gostou muito de todas as ferramentas. O índice de insatisfação é praticamente nulo para *Scratch* e *Pencil Code*. Algumas crianças não gostaram das ferramentas *Monster Coding* e *Code Monkey*.

Em seguida alunos foram questionados sobre o nível de dificuldade de cada uma das ferramentas, "*Você achou o fácil, médio ou difícil?"*. As crianças podiam escolher entre as opções: "*Fácil*", "*Médio*" e "*Difícil*". As respostas estão sumarizadas na Figura 2. A maioria dos alunos considerou a plataforma *Code Monkey* como "Médio" ou "Difícil", possivelmente por ser uma ferramenta que utiliza exclusivamente a linguagem escrita. As plataformas que a maioria marcou como "Fácil" foi *Scratch* e *Pencil Code*.

Por fim, perguntou-se qual ferramenta que eles mais gostaram de utilizar no curso. Com isso pôde-se observar que a ferramenta com mais votos foi o *Scratch*, com 32,5%

Tabela 3. Pontos positivos e negativos de cada ferramenta segundo monitores.

Ferramenta	Pontos positivos	Pontos negativos
<i>Hour of C.</i>	<ul style="list-style-type: none"> • O tutorial é autoguiado, dando ao aluno uma certa autonomia para concluir a atividade no seu próprio tempo; • Variedade grande de opções de tutoriais com temas conhecidos pelas crianças, como por exemplo Moana e Minecraft. 	<ul style="list-style-type: none"> • Os vídeos possuem somente áudio em inglês e as legendas são rápidas, o que dificulta o entendimento das crianças; • Muitos conceitos de programação são ensinados de uma só vez e alguns alunos não conseguem entender tudo que estão usando.
<i>Scratch</i>	<ul style="list-style-type: none"> • Os alunos têm liberdade de criar seus próprios jogos sem limitar-se a um tutorial específico; • Os alunos têm acesso a outros projetos já desenvolvidos e também podem compartilhar os seus. 	<ul style="list-style-type: none"> • A enorme quantidade de funcionalidades disponíveis, gera a dispersão de alguns alunos que queriam explorar outras possibilidades fora do escopo da aula.
<i>Pencil C.</i>	<ul style="list-style-type: none"> • Os alunos se divertem ao utilizar a ferramenta, mostrando interesse em variar os diversos comandos de repetição, para formar diferentes desenhos; • Aborda de forma efetiva todos os conceitos de programação. 	<ul style="list-style-type: none"> • A ferramenta não possui tradução dos blocos de comando para o português; • Os comandos são abreviados, gerando confusão na compreensão.
<i>Monster C.</i>	<ul style="list-style-type: none"> • Ao longo do tutorial algumas questões conceituais são apresentadas para verificar o aprendizado do aluno; • A opção de customização do personagem gera identificação e interesse maior da criança com o jogo. 	<ul style="list-style-type: none"> • A tradução da ferramenta à língua portuguesa é falha, comprometendo a compreensão em certos momentos; • A interface não é intuitiva, os alunos não compreenderam de início aonde estavam os comandos e como executá-los; • Tutorial não possui um grau de dificuldade gradativo.
<i>C. Monkey</i>	<ul style="list-style-type: none"> • Faz uma boa introdução à linguagem escrita Coffeescript, mostrando conceitos básicos; • Tutoriais autoexplicativos, facilitando o entendimento do aluno aos conceitos abordados em cada fase. 	<ul style="list-style-type: none"> • Como a linguagem Coffeescript é desenvolvida em inglês, não há tradução do código, o que dificultou inicialmente a compreensão dos alunos.

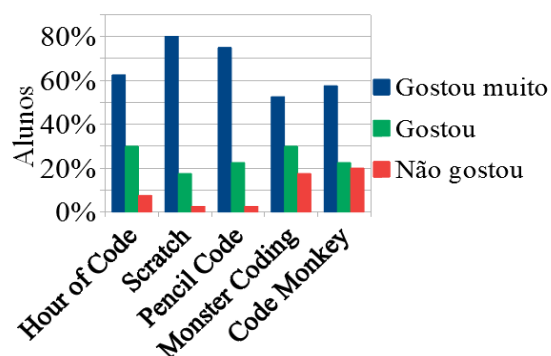


Figura 1. Satisfação dos alunos em relação às ferramentas.

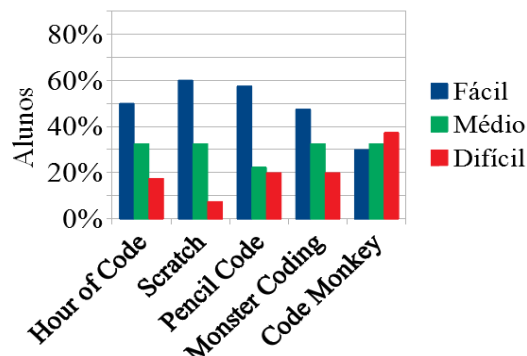


Figura 2. Dificuldade das ferramentas segundo os alunos.

de aprovação, logo seguida o *Pencil Code*, com 25,0% dos votos e *Code Monkey*, com 22,5%. As ferramentas com menor preferência nesse quesito foram *Monster Coding* e *Hour of Code*, com 15,0% e 5,0%, respectivamente.

4. Conclusão

Esse artigo apresentou uma avaliação das ferramentas voltadas ao ensino de programação para crianças. Foram consideradas cinco ferramentas Web gratuitas: *Hour of Code*, *Scratch*, *Pencil Code*, *Monster Coding* e *Code Monkey*. Os objetivos foram inspecionar as ferramentas para observar o cumprimento de regras de boas práticas de programação, relatar a visão das crianças quanto ao uso das ferramentas ressaltando as dificuldades e preferências e analisar o ponto de vista dos instrutores quanto à experiência em sala de aula citando os pontos positivos e negativos de cada ferramenta.

A ferramenta *Hour of Code* atendeu à praticamente todas as regras para um software educacional e foi considerada como a que possui a melhor interface pela maioria dos monitores. Seu principal ponto positivo foi a grande variedade de tutoriais autoguiados com temas conhecidos pelas crianças, o que gera um grande entusiasmo por parte dos alunos. Além disso, por ser autoguiado, o aluno tem autonomia de seguir as atividades no seu ritmo. A desvantagem encontrada foi quanto à tradução dos vídeos explicativos que são legendados e as crianças não conseguem acompanhar.

O *Scratch* foi escolhido como a melhor ferramenta e a que mais permite explorar os conceitos de programação de computadores. Além disso, também atendeu de forma satisfatória à maioria das regras utilizadas para avaliação de softwares educacionais. Destacou-se a liberdade de criação de atividades devido à grande variedade de conteúdo, o que incentiva a criatividade dos alunos. Outro ponto positivo ressaltado foi a possibilidade de compartilhar os projetos desenvolvidos. Aproximadamente 97% das crianças gostaram ou gostaram muito de usar a ferramenta e apenas 5% acharam difícil. Além disso, 32% dos alunos tiveram preferência em utilizar a ferramenta *Scratch*. Vale ressaltar que essa ferramenta não foi citada por nenhum aluno quando questionados qual ferramenta menos gostou.

Com o *Pencil Code* foi possível constatar que a plataforma não explora o uso de multimídia apropriadamente e não é tão amigável ao usuário, segundo a avaliação das regras para softwares educacionais. Nessa ferramenta é possível explorar todos os conceitos

de programação. Os problemas encontrados são quanto à falta de tradução dos blocos de comando para o português e a abreviação dos comandos que geram mal entendido. As crianças relataram grande satisfação em trabalhar com essa ferramenta e usar comandos de repetição para desenhar diferentes formas geométricas. Ela foi escolhida como a melhor ferramenta por 25% dos alunos e a maioria classificou a dificuldade da ferramenta como fácil ou média.

A única ferramenta que utiliza exclusivamente a linguagem de programação escrita é a *Code Monkey*. Essa plataforma foi a que obteve a melhor avaliação das regras para softwares educacionais. Os instrutores listaram como pontos positivos o fato dela ser um tutorial autoexplicativo e autoguiado, além de ser adequada para introduzir a programação escrita para crianças. Entretanto, como a maioria das crianças não sabia falar inglês, foi necessária a tradução dos comandos. Os alunos ficaram divididos ao classificar o nível de dificuldade e aproximadamente a mesma quantidade de alunos classificou como fácil, médio e difícil, apesar da maioria dos alunos terem gostado ou gostado muito da ferramenta.

A ferramenta *Monster Coding* foi a que também menos cumpriu as regras para softwares educacionais. Os pontos positivos dessa ferramenta são a apresentação dos conceitos de programação ao longo do tutorial e a avaliação desses conceitos, que é feita por meio de perguntas para verificar o aprendizado do aluno. Um grande problema dessa ferramenta é a tradução falha para a língua portuguesa, visto que em vários momentos há frases que são difíceis de compreender ou palavras em inglês no meio dos textos ou comandos. Além disso, a interface não é intuitiva e os alunos possuem dificuldade de encontrar os comandos. É importante ressaltar que ela foi a ferramenta mais rejeitada pelos alunos.

Avaliar sob a perspectiva dos alunos e monitores possibilitou comparar as experiências que ambos tiveram ao utilizar as ferramentas. *Scratch* e *Hour of Code* obtiveram as melhores avaliações no consenso da maioria dos monitores e alunos. A plataforma *Monster Coding* foi considerada como a menos aceita em todas as análises, tornando-se a ferramenta menos recomendada.

Referências

- Aono, A. H., Rody, H. V. S., Musa¹, D. L., Pereira¹, V. A., and Almeida, J. (2017). A utilização do scratch como ferramenta no ensino de pensamento computacional para crianças. In *XXXVII Congresso da Sociedade Brasileira de Computação - 25º Workshop sobre Educação em Computação (WEI)*, pages 2169–2178.
- Bau, D., Bau, D. A., Dawson, M., and Pickens, C. S. (2015). Pencil code: Block code for a text world. In *Proceedings of the 14th International Conference on Interaction Design and Children, IDC '15*, pages 445–448, New York, NY, USA. ACM.
- Bau, D., Gray, J., Kelleher, C., Sheldon, J., and Turbak, F. A. (2017). Learnable programming: blocks and beyond. *Communications of the ACM*, 60(6):72–80.
- da Silva Junior, S. M. and França, S. V. A. (2017). Programação para todos: Análise comparativa de ferramentas utilizadas no ensino de programação. In *XXXVII Congresso da Sociedade Brasileira de Computação - 25º Workshop sobre Educação em Computação (WEI)*, pages 2199–2200.

- de Oliveira, M. V., Rodrigues, L. C., and de Queiroga, A. P. G. (2016). Material didático lúdico: uso da ferramenta scratch para auxílio no aprendizado de lógica da programação. In *V Congresso Brasileiro de Informática na Educação (CBIE 2016) - Anais do XXII Workshop de Informática na Escola (WIE 2016)*, pages 359–368.
- Duarte, K. T. N., Silveira, T. R. S., and Borges, M. A. F. (2017). Abordagem para o ensino da lógica de programação em escolas do ensino fundamental ii através da ferramenta scratch 2.0. In *VI Congresso Brasileiro de Informática na Educação (CBIE 2017) - Anais do XXIII Workshop de Informática na Escola (WIE 2017)*, pages 175–184.
- Godinho, J., Torres, K., Batista, G., Andrade, E., and Gomide, J. (2017). Projeto aprenda a programar jogando: Divulgando a programação de computadores para crianças e jovens. In *XXXVII Congresso da Sociedade Brasileira de Computação - 25º Workshop sobre Educação em Computação (WEI)*, pages 2140–2149.
- Gomes, T. C. S., de Castro, F. C., and Alencar, A. (2017). Evaluating the effectiveness of educational games: a digital game-based approach to teach programming concepts for kindergarteners. In *V Congresso Brasileiro de Informática na Educação (CBIE 2016) - Anais do XXVIII Simpósio Brasileiro de Informática na Educação (SBIE 2017)*, pages 574–584.
- Gouws, L. A., Bradshaw, K., and Wentworth, P. (2013). Computational thinking in educational activities: An evaluation of the educational game light-bot. In *Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education, ITiCSE '13*, pages 10–15, New York, NY, USA. ACM.
- Jha, V. and Duffy, S. (2002). Ten golden rules for designing software in medical education: results from a formative evaluation of dialog. *Medical Teacher*, 24(4):417–421.
- Kim, A. S. and Ko, A. J. (2017). A pedagogical analysis of online coding tutorials. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education, SIGCSE '17*, pages 321–326, New York, NY, USA. ACM.
- Paula, B. H., Valente, J. A., and Burn, A. (2014). O uso de jogos digitais para o desenvolvimento do currículo para a educação computacional na Ingraterra. In *Currículo sem Fronteiras*, volume 14, pages 46–71.
- Petri, G. and Gresse von Wangenheim, C. (2017). How games for computing education are evaluated? a systematic literature review. *Comput. Educ.*, 107(C):68–90.
- Raabe, A. L. A. et al. (2017). Referenciais de formação em computação: Educação básica. In *Sociedade Brasileira de Computação*.
- Santos, C. P., da Silva, D. R., da Silveira, M. G. F., and Ferreira, G. (2017). Desafio de programação para meninas do ensino médio: Um relato de experiência. In *VI Congresso Brasileiro de Informática na Educação (CBIE 2017) - Anais do XXIII Workshop de Informática na Escola (WIE 2017)*, pages 144–137.
- Vorderman, C., Woodcock, J., McManus, S., Steele, C., Quigley, C., and McCafferty, D. (2014). Help your kids with computer coding. DK Publishing.
- Wing, J. M. (2006). Computational thinking. In *Communications of the ACM*, volume 49, pages 33–35.