Um *benchmark* de ferramentas de programação em blocos que podem ser utilizadas nas salas de aula do Ensino Médio

Ana Paula Juliana Perin[1], Deivid Eive Silva[1] e Natasha Malveira C. Valentim [1]

[1]Universidade Federal do Paraná (UFPR)

[1]Curitiba, PR, Brasil

[1]Departamento de Informática

apjperin@inf.ufpr.br, desilva@inf.ufpr.br, natasha@inf.ufpr.br

Abstract. *Due to the complexity of teaching programming to novice students, block programming has come up with a more attractive, intuitive, and visual approach. This paper presents a benchmark that aimed to identify and compare characteristics of block programming tools used in high school through a manual search. In this benchmark the characteristics analyzed were: disciplines in which they can be used, platforms and operating systems they work on, and emerging technologies that can be combined with block programming, among others. Finally, 58 tools were analyzed and can help high school teachers in choosing the most appropriate tool for their context of use.*

Resumo. *Devido a complexidade de ensinar programação para estudantes novatos, a programação em blocos surgiu com uma abordagem mais atrativa, intuitiva e visual. Este artigo apresenta um benchmark que objetivou identificar e comparar características de ferramentas de programação em blocos utilizadas no Ensino Médio por meio de uma busca manual. Neste benchmark as características analisadas foram: disciplinas em que podem ser utilizadas, plataformas e sistemas operacionais que elas funcionam e tecnologias emergentes que podem ser aliadas à programação em blocos, entre outras. Por fim, 58 ferramentas foram analisadas e podem auxiliar professores do Ensino Médio na escolha da ferramenta mais adequada para o seu contexto de uso.*

1. Introdução

A programação textual pode ser vista como um problema para estudantes que estão tendo um primeiro contato com a programação (Souza e França, 2013) . O uso de linguagens de programação textual podem tornar o processo de aprendizagem mais difícil devido a complexidade da sintaxe dessas linguagens (Burnett e McIntyre, 1995) . Com a intenção de amenizar estas dificuldades, pesquisadores e profissionais projetaram novas maneiras de ensino de programação para facilitar a aprendizagem, os ambientes de Linguagem de Programação Visual (*Visual Programming Language*, VPL), comumente conhecido como ambientes de programação em blocos. Os ambientes de programação em blocos são formados por características e cores específicas que indicam sua função (comandos e valores). Essa união de características e cores permitem formar a estrutura dos programas. Os comandos e valores são blocos de encaixe, permitindo que a ação de programar seja menos complexa. Isso possibilita a aproximação de outros públicos a terem a experiência

DOI: 10.5753/sbie.2021.217765

com a programação, como é o caso de professores da educação básica (Souza Rios et al. 2019) .

Os ambientes de programação em blocos podem tornar a programação, naturalmente relacionada aos conceitos da Matemática e da Lógica, mais atrativa por meio de uma experiência mais intuitiva e visual. Outro conceito importante que pode ser relacionado à programação em blocos é o pensamento computacional, caracterizado por um conjunto de habilidades para resolver problemas, projetar sistemas e entender o comportamento humano, sendo fundamentado por conceitos da Ciência da Computação (Wing, 2006). O pensamento computacional é uma das habilidades do Século XXI, destacando sua necessidade de ser trabalhado no ensino básico, pois auxiliam os estudantes a entenderem conceitos computacionais e desenvolver outras competências, como autonomia e raciocínio lógico (Yett et al. 2020; Vinayakumar et al. 2018; Guggemos et al. 2019 e Papadakis e Orfanakis, 2018). Nesse sentido, o ensino do pensamento computacional pode ser inserido no Ensino Médio por meio da introdução de linguagens de programação em blocos (Vinayakumar et al. 2018). Desse modo, a programação em blocos pode auxiliar também no aprimoramento do raciocínio lógico, pensamento crítico, criatividade, autonomia, trabalho em equipe e resiliência nos estudantes. Além disso, a programação em blocos aliada a Robótica Educacional ajuda os alunos do Ensino Médio a aprender conceitos computacionais, mantendo-os engajados (Yett et al. 2020).

Diante deste contexto, este artigo apresenta um *benchmark* realizado com o objetivo de identificar e comparar características de ferramentas de programação em blocos que podem ser utilizadas por professores do Ensino Médio. A motivação para este *benchmark* surgiu com a intenção de auxiliar os professores na identificação de ferramentas de programação em blocos que são mais adequadas para o seu contexto de uso. No geral, foram identificadas 58 ferramentas. As características de cada ferramenta analisada foram: Plataformas e Sistema Operacional em que as ferramentas de programação em blocos funcionam; se a ferramenta fornece material de apoio ao professor e/ou ao aluno e qual o tipo de material disponível; idioma em que a ferramenta pode ser utilizada; se é necessário realizar cadastro para acesso e uso da ferramenta; disciplinas em que as ferramentas podem ser trabalhadas; e tecnologias emergentes que podem ser usadas em conjunto com as ferramentas de programação em blocos.

Vale ressaltar que o *benchmark* foi direcionado ao Ensino Médio, pois foi percebido que este nível de ensino tem apresentado baixos rendimentos escolares nas principais avaliações da Educação Básica, como Programa Internacional de Avaliação de Estudantes (PISA) e Sistema de Avaliação da Educação Básica (SAEB). O Ensino Médio encontra-se estagnado desde 2009 (SAEB, 2017), e mesmo existindo uma pequena evolução no desempenho dos estudantes, esse resultado ainda precisa ser melhorado necessitando de iniciativas neste contexto (PISA, 2015; PISA, 2018). Além disso, Scaico et al. (2013) reconhecem a importância de ensinar programação no Ensino Médio e mencionam que este ensino precisa ser acompanhado de uma metodologia que possibilite engajar e motivar os alunos, de forma que as dificuldades encontradas possam ser superadas e os estudantes se mantenham interessados em aprender. Uma forma de manter os estudantes engajados pode ser por meio da programação em blocos relacionado a conteúdos das disciplinas ou aliados à tecnologias emergentes.

Por fim, o *benchmark* é um método utilizado para medir, comparar, definir me-

lhores práticas, implementação e melhoria de um software ou produto (Anand e Kodali 2008). Esse artigo apresenta a ideia de *benchmark* que consiste na avaliação dos ambientes computacionais para o contexto da Informática na Educação, conforme descrito e utilizada por Kist et al. (2002). Nesta pesquisa, o *benchmark* foi utilizado como uma metodologia de avaliação quantitativa que possibilita verificar as características das ferramentas de programação em blocos.

Este artigo está organizado da seguinte forma: A Seção 2 apresenta os trabalhos relacionados. A Seção 3 apresenta a metodologia utilizada para a realização do *benchmark*. A Seção 4 apresenta os resultados e discussões do *benchmark*. Por fim, a Seção 5 apresenta as considerações finais e trabalhos futuros.

2. Trabalhos Relacionados

Mesmo com a relevância dos estudos relacionados às ferramentas de programação em blocos, alguns destes são de avaliação comparativa para verificar a adequação das ferramentas em um determinado contexto a partir de critérios pré-definidos para os processos de ensino e aprendizagem. Um exemplo de estudo comparativo, no qual as ferramentas de programação *Scratch* e *Alice* foram comparadas sobre seus efeitos na prática de ensino de programação, em relação ao envolvimento dos alunos, ao pensamento reflexivo, à resolução de problemas e ao desenvolvimento de competências e habilidades, foi realizado por Durak (2020). Neste estudo, Durak (2020) percebeu que a ferramenta de programação *Scratch* se sobressaiu de forma mais positiva do que *Alice* no que tange o engajamento e o pensamento reflexivo dos alunos para a resolução de problemas. Todavia, a ferramenta *Alice* afeta positivamente no desenvolvimento de habilidades relacionadas ao pesamento computacional. Como pode ser percebido neste exemplo, houve uma comparação de duas ferramentas sob a ótica do processo de ensino e aprendizagem.

Outro exemplo de estudo comparativo relacionado às ferramentas de programação em blocos foi realizada por (Begosso et al. 2020) . As ferramentas analisadas foram: *Alice*, *MIT App Inventor*, *Blockly Games*, *Code.org*, *Gameblox*, *Pencil Code*, *MakeCode* e *Scratch*, sob um olhar mais técnico e com o objetivo de identificar e classificar ambientes de programação em blocos como práticas pedagógicas para o ensino inicial de lógica e programação de computadores. As características analisadas foram: sintaxe básica, alteração de bloco para *script*, variáveis e tipos de dados, expressões, atribuições, entrada e saída de dados simples, estruturas e condicionais, passagem de funções e parâmetros, e conceito de recursão. Os resultados mostraram que *Alice* possibilita o ensino de programação orientada a objetos e programação em um ambiente gráfico virtual 3D, porém, não possibilita transformar o programa de bloco em código de programação de *script*; O *MIT App Inventor* auxilia na construção de aplicativos móveis para a plataforma Android mas também não possibilita gerar código de programação de *script* a partir dos blocos criados pelo usuário; *BlocklyGames* utiliza blocos para representar os conceitos de código de programação, porém não possui recursos que permitam trabalhar com conceitos de recursão; *Code.org* fornece recursos para aprender e ensinar conceitos da Ciência da Computação por meio de jogos educacionais digitais, porém não permite a geração de roteiros de programação e não possui a funcionalidade de representação de expressões lógicas como “e” e “ou’; Com o *PencilCode* é possível criar rotinas com blocos de ações, base da lógica de programação, porém não tem a funcionalidade de representar um tipo de dados booleano; O *MakeCode* cobre todos os conceitos básicos

de programação que ajudam a desenvolver a criatividade, o raciocínio computacional e as habilidades de colaboração, por meio das opções de blocos disponíveis; O *Scratch*, é um ambiente de programação visual e multimídia que ensina como produzir animação, com ou sem som, e auxilia no aprendizado de boas estruturas de programação, entretanto, não permite a geração de *scripts*, nem trabalha com os conceitos de recursão, funções e passagem de parâmetros; e, o *Gameblox* foi projetado para design de jogos divertidos incorporando algumas tarefas de programação mais complexas, como conteúdo de física (colisão e gravidade) dentro dos blocos. Foi constatado que o *Gameblox* é uma ferramenta completa, porém os recursos de atribuição, funções e parâmetros, e conceitos de recursão são utilizados apenas quando há alternância de programação em bloco para o *script* gerado pelo ambiente. De modo geral, os autores perceberam um predomínio de ferramentas de programação em blocos que favorecem o desenvolvimento de jogos. Além disso, nenhum dos ambientes explorados exige conhecimento prévio de programação, pois são considerados ambientes de ensino para alunos iniciantes.

Não foi identificado na literatura um estudo comparativo de ferramentas de programação em blocos levando em consideração características como o material de apoio necessário para o uso destas ferramentas, tecnologias emergentes que podem ser trabalhadas em conjunto com estas ferramentas, dentre outras. Portanto, um *benchmark* foi realizado a fim de identificar o máximo de ferramentas de programação em blocos que podem ser utilizadas nas salas de aula do Ensino Médio e compará-las sob o ponto de vista de algumas de suas características listadas abaixo. Este *benchmark* não levou em consideração especificações de versão das plataformas que as ferramentas funcionam. Além disso, diferencia-se dos estudos acima por apresentar características que podem auxiliar professores do Ensino Médio na escolha de ferramentas de programação em blocos.

3. Metodologia

O *benchmark* foi executado em três etapas: (1) Planejamento; (2) Execução; e (3) Análise. Na etapa de Planejamento (1), foram definidas quais estratégias seriam adotadas para atingir o objetivo do *benchmark* e quais características seriam identificadas nas ferramentas. As estratégias adotadas foram: i) Definição das palavras-chave para serem utilizadas na busca de ferramentas de programação em blocos na *Web*; e ii) Realização de uma revisão da literatura em busca de artigos que abordem o uso de ferramentas de programação em blocos no Ensino Médio. As características analisadas foram: a) Plataformas em que as ferramentas de programação em blocos funcionam; b) Sistema Operacional; c) Material de apoio ao professor e/ou ao aluno e qual o tipo de material disponível; d) Idioma em que a ferramenta pode ser utilizada; e) Se é necessário realizar cadastro para acesso e uso da ferramenta; f) Disciplinas em que as ferramentas podem ser trabalhadas; e g) Tecnologias Emergentes que podem ser usadas em conjunto com as ferramentas de programação em blocos. Estas características foram escolhidas porque observou-se uma ausência desse tipo de análise na literatura. Além disso, essas informações são pertinentes para uma melhor escolha das ferramentas por professores, pois caracteriza o contexto de uso em que podem ser utilizadas e se estão de acordo com a realidade de cada escola em relação à sua infraestrutura. As etapas do planejamento foram revisadas por pares.

A estratégia para a (2) Execução do *benchmark* foi a busca manual das ferramentas na *Web* por meio das palavras-chave em inglês como *"block programming"*, *"block-based programming*", *"block-based coding"*, *"block interface"*, *"block-based tool*,

"block-based platform", *"block-based language"*, *"block-based approach"*, *"block-based methodology"*, *"block-based process"* e *"visual block programming"*, e os mesmos termos em português. Em paralelo, por meio de uma revisão da literatura, buscou-se identificar artigos que faziam menção a programação em blocos no Ensino Médio. Por fim, na etapa de (3) Análise, as ferramentas identificadas foram analisadas individualmente em relação às características listadas e revisada pelos demais pesquisadores.

4. Resultados e Discussões

No *benchmark* foram encontradas ao todo 58 ferramentas de programação em blocos. A relação das ferramentas identificadas pode ser encontrada na Tabela 1. Além disso, por questões de espaço, um relatório técnico foi criado contendo todas as tabelas da análise deste *benchmark* e disponibilizado em um link [1] . Os *links* de acesso dessas ferramentas podem ser encontradas na Tabela I deste relatório. Pode-se perceber que nas análises descritas nesta seção a somatória das porcentagens não resulta em 100%, pois as ferramentas podem ser classificadas em mais de uma das opções de características investigadas.

Tabela 1. Ferramentas de programação em blocos identificadas

AgentsCubes	Alice	MIT AppInventor	Art:bit	Beetle Blocks	BlocklyDuino
Blockuino	Blocky	BloclyGames	BrickLayer	ChoiCo	Code Kano
Code.org	CodeMao Turtle	Csnap	DroneBlocks	DuinoBlock	Engage
Gameblox	Hackeduca Conecta	Judge	Kitten	KittenBlock	Kodetu
Kodular	Lego	Lightbot	LookingGlass	MBlock	Micro:bit
Mind+	MiniBloq's	MusicBlocks	ModKit	NetsBlox	OzoBlocky
PencilCode	Pilas Bloques	Poredu	Progkids	Rita	Roblockly
Scratch	S4A	Sketchware	Snap!	Snap4Arduíno	StartLogo
Tello Edu	Thunkable	Tickle	Tinkecard	Tynker	Ubbu
Vittascience	Stencyl	logicBlocks	Catrobat		

4.1. Plataformas e Sistemas Operacionais

Em relação às plataformas em que as ferramentas de programação em blocos funcionam (Tabela A do relatório técnico), 77,59% (N = 45) das ferramentas funcionam em plataformas *Web*, 31,03% (N = 18) funcionam em plataforma Desktop, e 20,69% (N = 12) funcionam em plataforma Mobile. Pode-se observar que algumas ferramentas de programação em blocos funcionam em duas ou mais das plataformas investigadas. Por exemplo, as ferramentas *MBlock* e *OzoBlocky* estão disponíveis nas plataformas *Web*, mobile e desktop.

No que diz respeito aos sistemas operacionais onde as ferramentas de programação em blocos podem ser instaladas (Tabela B do relatório técnico), 29,31% (N = 17) são Windows; 24,14% (N = 14) para macOS; 20,69% (N = 12) para Android; 18,97% (N = 11) para iOS; e 15,52% (N = 9) para distribuições baseadas em Gnu/Linux. Além disso, 6,90% (N = 4) são para sistemas ChromeOs. Por fim, 55,17% (N = 32)

[1] https://figshare.com/s/61062bffd85783125510

das ferramentas identificadas funcionam somente em plataformas *Web*. Desse modo, foi possível identificar que algumas ferramentas de programação em blocos funcionam em duas ou mais das plataformas investigadas, por exemplo, a ferramenta *OzoBlocky* que tem suas versões disponíveis em sistema operacional Windows, macOS, Android, iOS e CrhomeOS. Outro exemplo de ferramenta que funciona em mais de um sistema operacional é o *Scratch*, com suas versões disponíveis para distribuições Gnu/Linux, Windows, macOS, Android, iOS e ChromeOs.

4.2. Tipos de materiais de apoio ao professor e ao aluno

Em relação aos tipos de materiais de apoio aos processos de ensino e aprendizagem de programação em blocos identificados no *benchmark*, as categorias foram investigados tendo como alvo o professor e o aluno, respectivamente. Os materiais de apoio e o público-alvo foram categorizados a partir das descrições das páginas *Web* das ferramentas. Quando o site da ferramenta não descreveu o público-alvo, foi considerado que o material de apoio poderia ser utilizado pelo professor e aluno. Essa categorização foi revisada por pares.

No que diz respeito aos tipos de materiais de apoio ao professor (Tabela C do relatório técnico), 27,59% (N = 16) dos materiais disponibilizados são tutoriais; 17,24% (N = 10) são do tipo de curso, 13,79% (N = 8) são guias, que inclui materiais que visam auxiliar e apoiar o professor no processo de ensino de programação em blocos com os alunos; em 12,07% (N = 7) das ferramentas são fóruns; 10,34% (N = 6) das ferramentas foram identificados materiais multimídias; 10,34% (N = 6) disponibilizam materiais no formato PDF, ou seja, material de apoio que pode ser impresso e utilizado pelo professor ou pelo aluno nos processos de ensino e aprendizagem, contendo informações como exercícios e instruções. Por último, em 48,28% (N = 28) das ferramentas não mencionam materiais de apoio e foram categorizados como “não encontrado”.

Referente aos materiais de apoio aos alunos (Tabela D do relatório técnico), 20,69% (N = 12) dos materiais disponibilizados são tutoriais; 12,07% (N = 7) das ferramentas disponibilizam materiais em forma de curso; 10,34% (N = 6) são materiais multimídias; 8,62% (N = 5) são fóruns; 6,90% (N = 4) são guias; e em 6,90% (N = 4) das ferramentas disponibilizam materiais em formato PDF. Por fim, 58,62% (N = 34) das ferramentas não mencionam materiais de apoio e foram categorizadas como “não encontrado”.

Esses resultados demonstram uma variedade de materiais de apoio ao professor e aluno, e ao mesmo tempo, um número considerável de ferramentas que não disponibilizam nenhum tipo de material de apoio. Além disso, pode-se perceber que há mais materiais disponíveis para o professor que podem apoiar seu auto aprendizado em programação em blocos, ou que o apoie em sala de aula no processo de ensino.

4.3. Idioma e cadastro de acesso

Em relação ao idioma em que as ferramentas podem ser utilizadas (Tabela E do relatório técnico), foi identificado que 89,66% (N = 52) podem ser utilizadas no idioma Inglês, 36,21% (N = 21) no Português, e 36,21% (N = 21) no Espanhol, 48,28% (N = 28) das ferramentas podem ser encontradas em outros idiomas, além dos já mencionados.

No que diz respeito a necessidade de realizar cadastro para utilizar a ferramenta (Tabela F do relatório técnico), foi identificado que: 74,14% (N = 43) não necessitam realizar cadastro para uso, e 25,86% (N=15) necessitam realizar cadastro.

4.4. Disciplinas e Tecnologias Emergentes

Em relação às disciplinas indicadas e mencionadas nas páginas *Web* das ferramentas de programação em blocos (Tabela G do relatório técnico), foi percebido que: 22,41% (N = 13) mencionam que a ferramenta pode ser usada na disciplina de Matemática; 12,07% (N = 7) em Artes; 5,17% (N = 3) em Física; 3,45% (N = 2) em Biologia; 3,45% (N = 2) em História; 1,72% (N = 1) em Geografia; e 1,72 % (N = 1) em Química. Por fim, 68,97% (N = 40) das ferramentas não mencionam relação com qualquer disciplina do currículo da educação básica em sua página *Web*.

As tecnologias emergentes foram identificadas considerando a descrição da página *Web* das ferramentas de programação em blocos, descrição nas próprias ferramentas, e com base na percepção dos pesquisadores. Essa percepção se deu pois os pesquisadores notaram que a ferramenta também poderia ser utilizada com uma outra tecnologia emergente, por exemplo, o *Scratch* que menciona IoT (*Internet of Things*, Internet das Coisas) e Robótica, mas foi identificado pelos pesquisadores que esta ferramenta pode ser trabalhada também com Jogos Digitais 2D e 3D. Foram identificadas 10 tecnologias emergentes, sendo: Robótica; IoT; Realidade Aumentada; Realidade Virtual; Jogos Digitais 3D; Jogos Digitais 2D; Simulação 3D; Modelagem 3D; IA (Inteligência Artificial); e Vant's (Veículo Aéreo Não Tripulado).

Em relação as tecnologias emergentes relacionadas às ferramentas de programação em blocos (Tabela H do relatório técnico), foi percebido que 32,76% (N = 19) das ferramentas estão associadas a IoT; 31,03% (N = 18) à Robótica; 27,59% (N = 16) à Jogos Digitais 2D; 18,97% (N = 11) à Jogos Digitais 3D; 12,07% (N = 7) à Simulação 3D; 10,34% (N = 6) à IA; 6,90% (N = 4) à Modelagem 3D; 5,17% (N = 3) à Vant's; 3,45% (N = 2) à Realidade Aumentada; e, 3,45% (N = 2) à Realidade Virtual. Por fim, foram identificados que em 18,97% (N = 11) das ferramentas de programação em blocos não é mencionado relação com nenhuma tecnologia emergente.

Durante a revisão da literatura, foram encontrados alguns estudos que mencionavam disciplinas e tecnologias emergentes em que as ferramentas de programação em blocos foram utilizadas. Desse modo, foi possível investigar em quais disciplinas e/ou com quais tecnologias emergentes as ferramentas de programação em blocos estão sendo utilizadas e com quais disciplinas e tecnologias emergentes elas são indicadas em suas páginas *Web*. Das ferramentas identificadas nestes estudos, quatro ferramentas mencionam relação direta com pelo menos uma disciplina e/ou tecnologia emergente e que foram trabalhadas com alunos do Ensino Médio, são elas: *Blocky*, *Micro:bit*, *NetsBlox*, e *Scratch*. Além disso, foi identificado um estudo com a ferramenta *Snap!* utilizada para a formação de professores do Ensino Médio, mesmo que sua página *Web* não mencione relação com disciplinas desse nível de ensino.

A ferramenta *Blocky*[2] menciona em sua página *Web* a relação com a disciplina de Matemática. Não identificou-se na página *Web* desta ferramenta a informação sobre o uso

[2]https://developers.google.com/blockly

conjunto dela com uma tecnologia emergente. Um exemplo de estudo utilizando *Blocky* e EV3 LEGO Mindstorms relacionado às disciplinas de Matemática e Física por meio da Robótica foi realizado por Orlando et al. (2020). Nesse estudo os alunos testaram o robô programado e refletiram sobre os conceitos de Física.

A ferramenta *Micro:bit*[3] traz uma relação com as disciplinas de Artes, Física e Biologia e com as tecnologias emergentes de Robótica e IoT em sua página *Web*. Um exemplo de estudo com *Micro:bit* utilizada na disciplina de Ciências com IoT foi realizado por Mørch et al. (2019). Nesse estudo, foram aplicados métodos de análise qualitativa para ensinar e engajar os alunos na disciplina de Ciências. Os alunos demonstraram interesse na programação em blocos e comentaram que seria bom relacionar a disciplina com atividades extracurriculares. Os alunos perceberam uma relação interdisciplinar entre a programação em blocos e conteúdos do currículo comum da disciplina e comentaram que poderiam aprender sobre velocidade e aceleração e outros assuntos semelhantes, como calcular valores e outras fórmulas. Além disso, os autores encontraram evidências de que a programação em blocos pode auxiliar no aprendizado e que os alunos preferem aprender por meio da contextualização dos conteúdos com situações da vida real.

A ferramenta *NetsBlox*[4] pode ser utilizada com as disciplinas de Matemática e Geografia e utilizada em conjunto com a Robótica, de acordo com a sua página *Web*. Um exemplo de estudo utilizando *NetsBlox* relacionada à disciplina de Física e a tecnologia emergente Computação em Nuvem foi realizado por Mørch et al. (2019). O currículo trabalhado foi dividido em quatro módulos: (1) transporte terrestre em 1D envolvendo aceleração e desaceleração constantes, (2) movimento de velocidade constante em 2D para transporte através de um rio, e (3) movimento acelerado em 2D (com a gravidade como um fator) para entrega de pacote em uma área remota usando um drone voador; e (4) Movimento 1D e 2D com forças (incluindo atrito estático e dinâmico). Foi possível perceber que os alunos desenvolveram melhor compreensão dos conceitos e práticas de Física e Pensamento Computacional do que os alunos que aprenderam através de um currículo tradicional. Isso implica dizer que atividades envolvendo programação em blocos em conjunto com uma disciplina podem ser mais atrativas nos processos de ensino e aprendizagem, além de proporcionar um ambiente mais lúdico.

A ferramenta *Scratch*[5] menciona em sua página *Web* que pode ser utilizada nas disciplinas de Artes, História e Matemática, e em conjunto com tecnologias emergentes como Robótica, IoT, Realidade Aumentada, Jogos Digitais 3D, Jogos Digitais 2D, IA e Vant's. Um exemplo de estudo utilizando *Scratch* e *MIT App Inventor* [6] relacionada às disciplinas de Matemática e Ciências, e aliada à IoT foi realizado por Fronza et al. (2019). O objetivo das atividades de programação em blocos foi desenvolver habilidades de resolução de problemas no aluno, a partir de estratégias do pensamento computacional com atividades relacionadas à Engenharia de Software, onde os alunos precisaram desenvolver um software para dispositivos móveis em ciclo de vida (Análise, Implementação e Teste). Assim, foram ensinados conceitos fundamentais de algoritmos e programação e, conforme o desenvolvimento dos alunos, as atividades se tornaram mais complexas e

[3]https://microbit.org/
[4]https://netsblox.org/
[5]https://scratch.mit.edu/
[6]http://ai2.appinventor.mit.edu

são relacionadas às disciplinas de Matemática e Ciência. Além disso, houve o desenvolvimento de atividades de *software* e de *hardware*, simulando um semáforo utilizando *Scratch*, integrando como destino o *hardware Raspberry Pi*.

Por fim, o *Snap!*[7] em sua página *Web* não menciona as disciplinas do Ensino Médio que podem ser utilizadas, porém, mencionam que a ferramenta pode ser utilizada aliada às tecnologias emergentes, tais como Jogos Digitais 2D, Simulação 3D e Modelagem 3D. Um exemplo de estudo realizado com professores do Ensino Médio em relação a formação utilizando o *Snap!* foi realizado por ocius et al. (2020). Participaram deste estudo 116 professores das disciplinas de Humanas, e Ciências e Matemática. A formação dos professores iniciou com a apresentação de pensamento computacional (PRADA – *Pattern Recognition, Abstraction, Decomposition, and Algorithms*), seguido por sessões de infusão de código, no qual foi usada a estrutura de aprendizagem *Use-Modify-Create*, permitindo o uso, a modificação e a criação de novos códigos durante o processo de aprendizagem. Os professores também realizaram uma atividade colaborativa para mapear e descrever os padrões dos elementos PRADA, criar um plano de aula para sua disciplina e sugerir atividades que poderiam ser implementadas em sala de aula. Por fim, os professores criaram materiais pedagógicos complementares, como slides e apostilas para apresentar o que aprenderam aos demais participantes, convidados e gestores de escolas.

Não foram identificados estudos que mencionam todas as 58 ferramentas identificadas neste *Benchmark*. Com a análise descrita neste artigo foi possível perceber que as disciplinas que as ferramentas de programação em blocos foram mais utilizadas é a disciplina de Matemática, seguido de Ciências. Em relação as tecnologias emergentes, a maioria dos estudos utilizam a Robótica e IoT.

De modo geral, pode ser observado que as ferramentas de programação em blocos estão sendo utilizadas em disciplinas e com tecnologias emergentes que vão além das quais são sugeridas ou exemplificadas em sua página *Web*. Acredita-se que esse resultado pode estar relacionado ao fato do responsável por ensinar programação em blocos fazer uma associação e adaptação do uso da ferramenta à sua disciplina e ao seu contexto. Além disso, a maioria das ferramentas representadas na Tabela G do relatório técnico não mencionam relação com nenhuma disciplina. Isso demonstra que as ferramentas de programação em blocos podem ser usadas nas mais diversas disciplinas, independente da área do conhecimento, e baseadas nos objetivos a serem alcançados pelo professor.

Esse trabalho pode apoiar os professores do Ensino Médio no que diz repeito às possibilidades de ferramentas que podem ser utilizados por esses docentes, de acordo com seu contexto de uso. Além disso, contribui para a comunidade científica de Educação em Computação e Informática na Educação, pois não foi identificado na literatura um estudo comparativo de ferramentas de programação em blocos levando em consideração características como o material de apoio necessário para o uso destas ferramentas, tecnologias emergentes que podem ser trabalhadas em conjunto com estas ferramentas, dentre outras características, além de um levantamento de um maior número possível de ferramentas.

5. Considerações Finais e Trabalhos Futuros

O *benchmark* contribuiu para identificar quais sistemas operacionais e plataformas as ferramentas de programação em blocos funcionam; contribuiu na identificação de quais ma-

[7] https://snap.berkeley.edu/

teriais existem e estão disponíveis tanto para o professor quanto para o aluno. Contribuiu também para identificar o idioma que a ferramenta de programação em blocos pode ser utilizada, e se é necessário realizar cadastro para utilizá-la. Contribuiu na identificação de quais disciplinas e tecnologias emergentes as ferramentas de programação em blocos são indicadas em suas páginas *Web*. Adicionalmente, foi possível analisar quais disciplinas e tecnologias emergentes essas ferramentas estão sendo de fato utilizadas, fazendo uma relação com alguns estudos encontrados.

Esse estudo pode contribuir na escolha de ferramentas de programação em blocos por professores do Ensino Médio de acordo com seu contexto de uso, seja contendo material de apoio, pelo idioma – que pode facilitar o processo de ensino e aprendizagem quando utilizada uma ferramenta no idioma Português, língua nativa do Brasil –, além do Inglês e Espanhol, quando o objetivo do professor for trabalhar o ensino de língua estrangeira, trabalhando assim, de forma interdisciplinar, conceitos de programação com o idioma escolhido. Pode contribuir para professores de escolas públicas e privadas na escolha de ferramentas conforme o sistema operacional utilizado no laboratório de informática da instituição. Pode contribuir no incentivo do uso dessas ferramentas na sala de aula do Ensino Médio de forma interdisciplinar, e em outras disciplinas além das exatas, na qual são comumente trabalhadas.

Além disso, este estudo pode contribuir com a comunidade científica de Educação em Computação e Informática na Educação e docentes de disciplinas introdutórias de programação de universidades públicas e privadas no que tange a ter acesso à características específicas de ferramentas de programação de blocos que podem apoiar pesquisas e conteúdo de aulas de graduação neste contexto. Por exemplo, um pesquisador que esteja trabalhando com o tema pensamento computacional para o Ensino Médio pode se beneficiar deste *benchmark* ao identificar ferramentas que podem ser utilizadas e que melhor se adéquem ao foco da sua pesquisa. Para apoiar estes docentes e pesquisadores, foi disponibilizada na Tabela I do relatório técnico a relação das ferramentas de programação em blocos identificadas no *benchmark* e seus *links* de acesso.

Após a análise destas ferramentas, percebeu-se que o professor pode ficar confuso na escolha de qual ferramenta melhor se adéqua ao seu contexto e necessidades. Desse modo, notou-se a necessidade de, como trabalho futuro, desenvolver um assistente virtual que apoie o professor na escolha de ferramentas de programação em blocos no contexto do Ensino Médio. Acredita-se que um assistente virtual que apoie o professor neste propósito, pode auxiliá-lo em seu processo de ensino, levando em consideração algumas características destas ferramentas. Além disso, pretende-se realizar um estudo com professores do Ensino Médio, apresentando as ferramentas e suas características, a fim de investigar se utilizariam uma determinada ferramenta em sua sua disciplina e que não está caracterizado na mesma. Por fim, pode ser interessante estender este *benchmark* para o contexto de ferramentas que podem ser utilizadas também no Ensino Fundamental.

Referências

Anand, G. and Kodali, R. (2008). Benchmarking the benchmarking models. *Benchmarking: An international journal*.

Begosso, L. C., Begosso, L. R., and Christ, N. A. (2020). An analysis of block-based programming environments for cs1. In *2020 IEEE Frontiers in Education Conference*

(FIE), pages 1–5. IEEE.

Burnett, M. M. and McIntyre, D. W. (1995). Visual programming. *COMPUTER-LOS ALAMITOS-*, 28:14–14.

Durak, H. Y. (2020). The effects of using different tools in programming teaching of secondary school students on engagement, computational thinking and reflective thinking skills for problem solving. *Technology, Knowledge and Learning*, 25(1):179–195.

Fronza, I., Corral, L., and Pahl, C. (2019). Combining Block-Based Programming and Hardware Prototyping to Foster Computational Thinking. In *Proceedings of the 20th Annual SIG Conference on Information Technology Education*, SIGITE '19, pages 55–60, Tacoma, WA, USA. Association for Computing Machinery.

Guggemos, J., Seufert, S., and Román-González, M. (2019). Measuring computational thinking-adapting a performance test and a self-assessment instrument for german-speaking countries. In *Proceedings of the 16th International Conference Cognition and Exploratory Learning in the Digital Age*, pages 183–191. ERIC.

Hutchins, N., Biswas, G., Maróti, M., Lédeczi, , Grover, S., Wolf, R., Blair, K., Chin, D., Conlin, L., Basu, S., and McElhaney, K. (2020). C2stem: a system for synergistic learning of physics and computational thinking. *Journal of Science Education and Technology*, 29(1):83–100. cited By 6.

Jocius, R., Joshi, D., Dong, Y., Robinson, R., Cateté, V., Barnes, T., Albert, J., Andrews, A., and Lytle, N. (2020). Code, Connect, Create: The 3C Professional Development Model to Support Computational Thinking Infusion. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, SIGCSE '20, pages 971–977, Portland, OR, USA. Association for Computing Machinery.

Kist, T., Diverio, T. A., de Lima, J. V., and Tollens, L. (2002). Benchmark de ambientes de gerenciamento de cursos a distância. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 1, pages 474–483.

Mørch, A. I., Litherland, K. T., and Andersen, R. (2019). End-User Development Goes to School: Collaborative Learning with Makerspaces in Subject Areas. In Malizia, A., Valtolina, S., Morch, A., Serrano, A., and Stratton, A., editors, *End-User Development*, Lecture Notes in Computer Science, pages 200–208, Cham. Springer International Publishing.

Orlando, S., Gaudioso, E., and De La Paz, F. (2020). Supporting teachers to monitor student's learning progress in an educational environment with robotics activities. *IEEE Access*, 8:48620–48631. cited By 1.

Papadakis, S. and Orfanakis, V. (2018). Comparing novice programing environments for use in secondary education: App inventor for android vs. alice. *International Journal of Technology Enhanced Learning*, 10(1-2):44–72. cited By 12.

PISA (2015). Brasil no pisa 2015: Análises e reflexões sobre o desempenho dos estudantes brasileiros. Disponível em: <https://bit.ly/2DotVQ0>. Acessado em 30 jul. 2020.

PISA (2018). Relatório brasil no pisa 2018, versão preliminar. Disponível em: <https://bit.ly/2ExT3Eh>. Acessado em 02 fev. 2021.

SAEB (2017). Relatório saeb 2017. Disponível em: <https://bit.ly/30cgafW>. Acessado em 02 fev. 2021.

Scaico, P. D., de Lima, A. A., Azevedo, S., da Silva, J. B. B., Raposo, E. H., Alencar, Y., Mendes, J. P., Scaico, A., et al. (2013). Ensino de programação no ensino médio: Uma abordagem orientada ao design com a linguagem scratch. *Revista Brasileira de Informática na Educação*, 21(02):92.

Souza, M. V. R. d. and França, A. C. C. (2013). Um estudo sobre as dificuldades no processo de aprendizagem de programação no curso de análise e desenvolvimento de sistemas na fafica–faculdade de filosofia, ciências e letras de caruaru-pe. *Revista da Escola Regional de Informática*, 2(2):19–27.

Souza Rios, L. K. d., Junior, A. d. O. C., Lima, J. P. F., Guedes, E. B., et al. (2019). Uma análise comparativa entre ambientes de programação em blocos para a interação com o arduino. *Anais do Simpósio Ibero-Americano de Tecnologias Educacionais*.

Vinayakumar, R., Soman, K., and Menon, P. (2018). Fractal geometry: Enhancing computational thinking with mit scratch. In *2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pages 1–6. IEEE.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3):33–35.

Yett, B., Hutchins, N., Stein, G., Zare, H., Snyder, C., Biswas, G., Metelko, M., and Lédeczi, (2020). A Hands-On Cybersecurity Curriculum Using a Robotics Platform. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, SIGCSE '20, pages 1040–1046, Portland, OR, USA. Association for Computing Machinery.