

Uma Proposta de Avaliação da Originalidade do Produto no Ensino de Algoritmos e Programação na Educação Básica

Nathalia da Cruz Alves¹, Christiane Gresse von Wangenheim¹, Matheus Alberto¹,
Lucia Helena Martins Pacheco¹

¹Departamento de Informática e Estatística
Universidade Federal de Santa Catarina (UFSC) – Florianópolis – SC – Brasil

nathalia.alves@posgrad.ufsc.br, c.wangenheim@ufsc.br,
matheusalbertomth@gmail.com, lucia.pacheco@ufsc.br

Abstract. *Creativity has emerged as an important 21st-century competency that can be developed as part of computing education through app development. Currently there is no consensus on the characteristics of product creativity, however originality is considered one of the most important. Therefore, this article presents a model to assess the originality of apps created by students in computing education using the Use-Modify-Create approach. The model has been systematically derived from the theoretical frameworks on creativity defining rubrics as a first step for the assessment of originality in computing education through app development.*

Resumo. *A criatividade é uma competência importante no século XXI e pode ser desenvolvida como parte do ensino de computação por meio do desenvolvimento de apps. Atualmente, não existe um consenso sobre todas as características do produto criativo. No entanto, a originalidade é considerada uma das mais importantes. Assim, este artigo apresenta um modelo para avaliar a originalidade de apps criados por alunos no contexto do ensino de computação usando a abordagem Use-Modifique-Crie. O modelo foi sistematicamente derivado a partir de referenciais teóricos da criatividade, definindo rubricas e fornecendo um primeiro passo na avaliação da originalidade no ensino de computação por meio do desenvolvimento de apps.*

1. Introdução

A criatividade é considerada uma das principais competências no século XXI [Cavallo et al. 2016; Voogt e Roblin 2012] e promover a criatividade dos alunos já na Educação Básica é muito importante. Nesse contexto, a criatividade está inserida em diversos currículos ao redor do mundo [Voogt e Roblin 2012, MEC 2018]. E, embora a criatividade esteja tradicionalmente associada a artes, música e literatura, ela também pode ser desenvolvida como parte da computação segundo a Base Nacional Comum Curricular (BNCC):

“Exercitar a curiosidade intelectual e recorrer à abordagem própria das ciências, incluindo a investigação, a reflexão, a análise crítica, a imaginação e a **criatividade**, para investigar causas, elaborar e testar hipóteses, formular e resolver problemas e **criar soluções (inclusive tecnológicas)** com base nos conhecimentos das diferentes áreas.” [MEC 2018, p. 11].

Criando soluções tecnológicas os alunos aprendem a expressar suas ideias e competências de forma criativa [Clements 1995; Yadav e Cooper 2017], por exemplo

por meio do desenvolvimento de artefatos computacionais, como apps, usando ambientes de programação baseados em blocos visuais, como App Inventor [Lye e Koh 2014]. Uma das estratégias para desenvolver a criatividade progressivamente dentro do ensino de computação é a “Use-Modifique-Crie” (UMC) [Lytle et al., 2019; Lee et al., 2011]. Na UMC os alunos aprendem primeiro “usando” e analisando um determinado artefato computacional, “remixando” e modificando um artefato já existente e, eventualmente, “criando” um novo [Lee et al. 2011]. Assim, os alunos desenvolvem a capacidade de gerar ideias e soluções originais e úteis [Walia 2019] durante as etapas de modificação e criação.

Para estimular o desenvolvimento de criatividade é essencial prover uma avaliação e *feedback* instrucional. A avaliação pode ser feita de diferentes formas, por exemplo, por meio de uma avaliação de desempenho com base no artefato computacional criado pelo aluno como resultado da aprendizagem [Mishra e Henriksen 2013]. Embora a avaliação do produto por si só possa não ser suficiente para avaliar a criatividade como um todo, é um bom ponto de partida que pode ser completado por outros tipos de avaliações [Mishra e Henriksen 2013].

Porém, a avaliação da criatividade focando no produto não é trivial por se tratar de um conceito complexo. Geralmente, a definição da criatividade focando no produto inclui diferentes características como novidade, adequação e condensação [Alves et al. 2020]. Dentro dessa definição, uma das principais características é a originalidade. Geralmente, uma ideia ou produto criativo é considerado original se representa algo novo ou surpreendente que não existia antes [Runco e Jaeger 2012]. Embora a originalidade por si só não seja suficiente para classificar um produto como sendo criativo, ela é uma característica essencial dos produtos criativos [Besemer e Treffinger 1981]. E, apesar da importância da criatividade como competência importante no século XXI e da originalidade como característica essencial dessa competência, até o momento, existem poucas pesquisas sobre como avaliar a originalidade do produto criativo dentro do contexto educacional e especificamente no ensino de computação [Alves et al. 2020]. As abordagens existentes em geral avaliam a originalidade [Scaico et al. 2013; Gal et al. 2017; Mustafaraj et al. 2017; Turbak et al. 2017] ou a divergência [Bennett et al. 2013] em relação ao código-fonte não abordando a originalidade do produto em termos de outras características, como o design visual.

Portanto, o objetivo desta pesquisa é definir um modelo de avaliação da originalidade do produto criado como resultado da aprendizagem, focando em aplicativos móveis desenvolvidos por alunos usando App Inventor. O modelo proposto é aplicado na prática utilizando apps de um contexto real de aprendizagem.

2. Trabalhos relacionados

Resultados de um mapeamento sistemático da literatura demonstram que existem poucas abordagens para avaliar a originalidade de produtos criados pelos alunos dentro do contexto do ensino de computação [Alves et al. 2020]. As abordagens encontradas focam principalmente na análise do código-fonte. Outros aspectos do produto, como o design de interface, o conteúdo e as funcionalidades, tipicamente não são considerados. Dependendo do ambiente de programação adotado, a originalidade do produto é avaliada comparando o produto do aluno com produtos desenvolvidos por outros alunos

em um mesmo contexto educacional [Mustafaraj et al. 2017; Turbak et al. 2017], comparando o artefato do aluno com alguma solução predefinida [Bennett et al., 2013] ou pelo professor pontuando o produto [Scaico et al. 2013] de acordo com uma rubrica [Basu 2019]. Mustafaraj et al. (2017) e Turbak et al. (2017) avaliam a originalidade usando modelos de classificação de aprendizado de máquina. Bennet et al. (2013) avaliam a divergência entre produtos por meio de padrões de pensamento computacional definidos com fórmulas matemáticas. Basu (2019) avalia a novidade de forma limitada por meio de apenas um item em uma rubrica. O *feedback* é apresentado por meio de uma nota final ou uma classificação do produto como ‘original’ ou ‘não original’.

Observou-se, de forma geral, a falta de sistematização das medidas adotadas nas abordagens existentes, bem como a ausência da avaliação de aspectos importantes, como o design visual e funcionalidades do produto. Além disso, a maioria das abordagens não apresenta uma avaliação do modelo proposto, por exemplo, avaliando a eficácia [Mustafaraj et al. 2017] ou a qualidade por meio de um painel de especialistas [Basu 2019]. Portanto, há uma falta de pesquisas focando na avaliação da originalidade como parte da criatividade com base em outras características além do código-fonte de produtos criados por alunos no contexto educacional. Desta forma, este trabalho apresenta uma proposta de avaliação da originalidade incluindo não só o código-fonte, mas também diversos aspectos do produto como o design visual e suas funcionalidades.

3. Método de Pesquisa

O desenvolvimento do modelo de avaliação da originalidade aqui proposto é baseado no método para o desenvolvimento de rubricas proposto por Allen e Knight (2009) e Moskal (2000) adaptado para o contexto de avaliação de aplicativos resultantes de experiências práticas de ensino de computação.

Análise de domínio de conteúdo. Em uma primeira etapa, o domínio do conteúdo é analisado por meio da literatura e a análise de um conjunto de projetos App Inventor resultantes de experiências práticas de ensino de computação [Ferreira et al. 2020, Hauck et al. 2018]. Como resultado, é obtida uma definição clara da avaliação desejada, da abrangência e limites.

Definição da rubrica de avaliação. Com base na análise do domínio, o objetivo da avaliação é definido e decomposto em dimensões a serem medidas adotando a abordagem *Goal/Question/Metric* (GQM) [Basili et al. 1994]. Com base nas dimensões identificadas são definidas as características que precisam estar presentes no produto de um aluno para demonstrar desempenho em relação a originalidade [Moskal 2000; Allen e Knight 2009]. Em seguida, os níveis de desempenho são definidos para cada item. Considerando as diferenças em relação à criatividade da abordagem UMC, são definidas duas rubricas: uma para a etapa Modifique e outra para a etapa Crie. Assume-se que na etapa Use o aluno não possui a liberdade de criação de um produto, usando somente a tecnologia, portanto não se prevê avaliação da criatividade.

Aplicação exemplo. Nesta etapa, o modelo é aplicado exemplarmente de forma exploratória por meio de quatro avaliações de apps desenvolvidos por alunos da Educação Básica em um contexto real de aprendizagem. Esta aplicação exemplo é realizada por professores de computação seguindo as rubricas definidas, e tem como

objetivo obter evidências iniciais da aplicabilidade e corretude das rubricas Modifique e Crie dentro do contexto educacional conforme recomendado por Moskal (2000). Os resultados da aplicação são discutidos e melhorias são propostas.

4. Desenvolvimento do modelo de avaliação da originalidade

4.1. Análise de domínio de conteúdo

Atualmente não existe um conjunto de características do produto criativo consolidado na literatura [Besemer e Treffinger 1981; Walia 2019]. No entanto, resultados de um mapeamento sistemático da literatura [Alves et al. 2020] indicam que as características frequentemente analisadas podem ser agrupadas em três dimensões: novidade (*novelty*), adequação (*appropriateness*) e condensação (*condensation*) (Figura 1).



Figura 1. Características do produto criativo [Alves et al. 2020]

A novidade refere-se a produtos originais que mudam um paradigma. Dentro da novidade, a originalidade refere-se a um produto incomum ou pouco frequente visto em um universo de produtos criados por pessoas com experiência e aprendizagem semelhantes [Jackson e Messick 1964]. A mudança de paradigma refere-se a produtos transformadores que revolucionam uma área. Tendo em vista que no contexto educacional espera-se que alunos não atinjam nível tão elevado de criatividade, tipicamente, a mudança de paradigma não é considerada na avaliação dentro desse contexto. Desta forma, esta pesquisa foca exclusivamente nas características da **originalidade** como parte do produto criativo. As demais características referentes a adequação e condensação também estão fora do escopo deste estudo.

No ensino de computação por meio do desenvolvimento de apps, tipicamente adota-se a abordagem Use-Modifique-Crie (UMC). Nesse contexto, a originalidade pode ser usada como um dos critérios para medir a criatividade. Como a etapa Use consiste no desenvolvimento de um app base, não se espera uma “criação” por parte do aluno. Já na etapa Modifique, o aluno é instigado a modificar alguma característica do app, criando um app “parcialmente seu”. Na etapa Crie, é esperado um nível maior de originalidade, já que o aplicativo criado deve ser muito diferente em relação ao aplicativo base e modificado nas etapas anteriores (Figura 2).

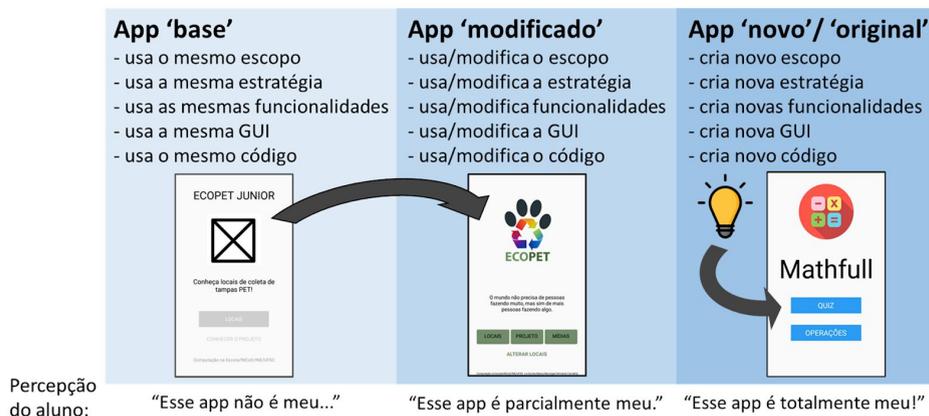


Figura 2. Abordagem UMC no contexto de desenvolvimento de apps

4.2. Definição do modelo de avaliação da originalidade de apps

Com base na análise do domínio do conteúdo, o objetivo da avaliação é definido como: avaliar o aplicativo móvel (criado como resultado da aprendizagem) com relação à originalidade na etapa de Modifique e Crie dentro do contexto de ensino de computação na Educação Básica. Considerando que uma avaliação ampla do produto deve considerar todas as suas dimensões possíveis [Jackson e Messick 1964], outras dimensões além do código-fonte são incluídas. Considerando a importância da estética visual de um app é incluída uma dimensão de design visual. Além disso, dimensões como objetivo e escopo do app são incluídas, pois, apps com objetivo e escopo originais devem ser identificados em uma avaliação do produto criativo. As dimensões e itens são definidos com base no modelo de Garrett (2010) (Figura 3).



Figura 3. Rubrica Modifique e Crie com o Universo de Comparação do Produto

Considerando as diferentes expectativas em relação ao grau de originalidade nos diferentes estágios de Modifique e Crie, são definidas duas rubricas para cada estágio (Figura 3). No estágio Modifique, espera-se que os alunos façam apenas modificações mínimas, assim presume-se que essas modificações não sejam suficientes para alterar os objetivos do produto e sua especificação de conteúdo. Portanto, esses elementos são

avaliados apenas no estágio Crie (Figura 3). Considerando as diferenças dos estágios, é definido o Universo de Comparação do Produto (UnCP) para cada rubrica. No estágio Modifique, a rubrica é aplicada comparando o aplicativo do aluno com o aplicativo base ensinado no estágio Use. Já no estágio Crie, o UnCP consiste em aplicativos feitos por pessoas com experiência e aprendizagem semelhantes, por exemplo, os aplicativos compartilhados na Galeria do App Inventor. Os níveis de desempenho são definidos em uma escala ordinal de três pontos: nenhuma diferença com o UnCP, algumas diferenças com o UnCP ou muitas diferenças com o UnPC.

5. Aplicação Exemplo

Com o objetivo de avaliar preliminarmente a validade do modelo de avaliação, foi realizada uma aplicação exemplo utilizando cada rubrica. Para isso é analisado se a avaliação realizada com as rubricas representa corretamente a originalidade dos apps e se rubricas podem ser aplicadas para avaliar a originalidade de apps. Para melhor representar o contexto de ensino de computação na Educação Básica, foram selecionados os seguintes apps a partir de experiências reais na prática:

O app base ensinado na etapa Use [Ferreira et al. 2020], consiste em um app para visualizar os pontos de coleta de tampinhas de garrafas por uma ONG. As funcionalidades consistem na visualização de locais de coleta e informações sobre o projeto de coleta de tampinhas (Figura 4).

Os apps modificados representam apps criados por alunos na etapa Modifique. Com base em uma avaliação geral abstraindo detalhes dos apps por especialistas em ensino de computação, foram selecionados dois apps. O app menos original, que possui o mesmo objetivo e as mesmas funcionalidades do app base, somente com algumas diferenças no design visual. O app mais original, que possui o mesmo objetivo do app base, porém acrescenta também novas funcionalidades permitindo a alteração de pontos de coleta (incluir, alterar, excluir) e acesso à página do app em redes sociais (Figura 4).

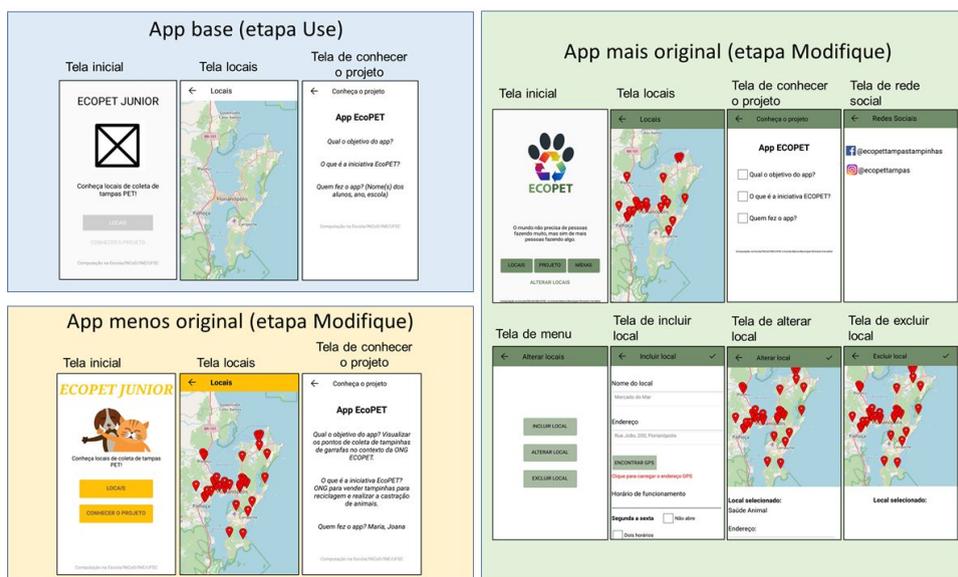


Figura 4. Ilustração dos apps das etapas Use e Modifique.

Os apps “novos” representam apps criados por alunos na etapa Crie. Novamente, com base em uma avaliação por especialistas foram selecionados dois apps: um menos original consistindo em um app com ideias comuns. O app menos original tem como objetivo ajudar estudantes com dificuldades em matemática fornecendo um quiz sobre as 4 operações básicas da matemática e uma explicação sobre cada operação (Figura 5). O app mais original consiste em um app sobre a balneabilidade das praias, com escopo e objetivo mais incomum. As funcionalidades incluem descrição e balneabilidade de praias, mapa de todas as praias e suporte ao usuário (Figura 5).

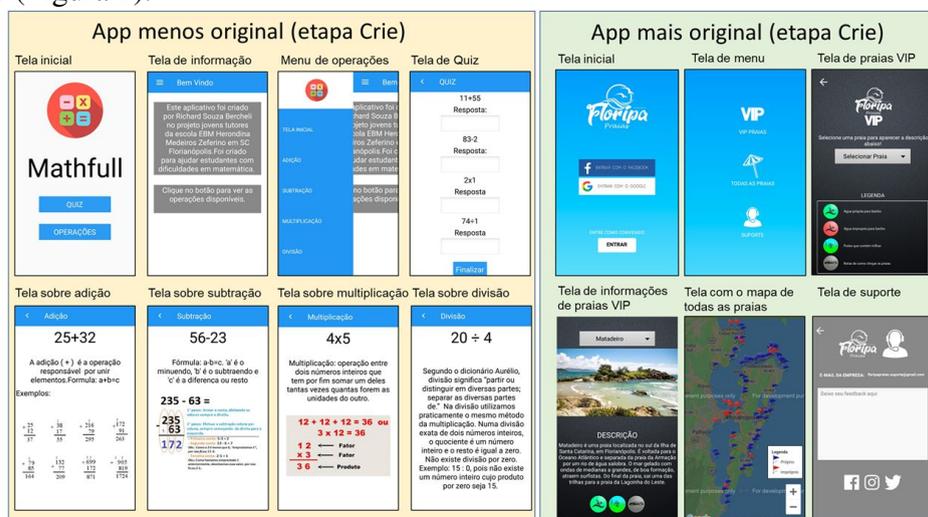


Figura 5. Ilustração dos apps da etapa Crie.

5.1. A avaliação realizada com as rubricas representa corretamente a originalidade de apps?

De forma a identificar se é possível avaliar a originalidade de apps com as rubricas, foram realizadas as avaliações dos apps selecionados. Os apps modificados são comparados com o app base utilizando a rubrica Modifique, e os apps “novos” são comparados com apps da Galeria do App Inventor (<https://appinventor.mit.edu/>).

A aplicação da rubrica Modifique demonstra a diferenciação entre apps menos originais e mais originais (Quadro 1). Todos os itens do app mais original foram avaliados com maior grau de diferença em relação ao app base. Já o app menos original apresenta a avaliação na maioria dos itens como sem diferenças, com diferenças somente no nível de aparência para o elemento de design visual. Isso é esperado já que o app menos original não inclui nenhuma nova funcionalidade em relação ao app base, não muda o esqueleto nem a estrutura. Consequentemente o código-fonte do app também permanece sem diferenças ao app base.

A aplicação da rubrica Crie também demonstra a diferenciação entre apps menos originais e mais originais (Quadro 1). Observando os apps da Galeria do App Inventor, foram encontrados mais de 20 apps com o objetivo de “treinar” ou aprender as operações básicas de matemática. Assim, a avaliação com a rubrica demonstra que a estratégia e o escopo do app menos original possui pouca ou nenhuma diferença com apps existentes. Já em relação ao app mais original, foi encontrado somente um app

similar na Galeria do App Inventor. Além disso, mesmo sendo similar, o app encontrado possui várias diferenças em relação ao app mais original.

Quadro 1. Avaliação do app 'Modificado' e do app 'Novo'

| Rubrica | | | Avaliação do app 'Modificado' (em relação ao app 'Base') | | Avaliação do app 'Novo' (em relação a apps da Galeria App inventor) | | |
|------------|----------------------|--|---|--------------------|---|--------------------|--------------------|
| Dimensão | Elemento | Critério | menos original | mais original | menos original | mais original | |
| Estratégia | Objetivo | Grau de diferença em relação ao objetivo em um contexto especificado. | -- | -- | Algumas diferenças | Algumas diferenças | |
| Escopo | Conteúdo | Grau de diferença em relação ao conteúdo necessário para atender aos objetivos do sistema de software. | -- | -- | Sem diferenças | Muitas diferenças | |
| | Funcionalidade | Grau de diferença entre a funcionalidade/recursos do sistema de software para atender aos objetivos. | Sem diferenças | Muitas diferenças | Algumas diferenças | Algumas diferenças | |
| Design | Estrutura | Design de interação | Grau de diferenças em relação às formas de interação do usuário com o app (p.ex. sensores, mídia, conectividade etc.). | Sem diferenças | Sem diferenças | Algumas diferenças | Algumas diferenças |
| | | Arquitetura de informação | Grau de diferenças em relação ao projeto estrutural do espaço de informações e as informações apresentadas pelo aplicativo. | Sem diferenças | Algumas diferenças | Algumas diferenças | Muitas diferenças |
| | Esqueleto | Design de interface | Grau de diferenças em relação aos componentes da GUI (p.ex., <i>switch</i> em vez de caixa de seleção). | Sem diferenças | Algumas diferenças | Algumas diferenças | Algumas diferenças |
| | | | Grau de diferenças em relação ao posicionamento / organização dos componentes da GUI nas telas. | Sem diferenças | Algumas diferenças | Muitas diferenças | Muitas diferenças |
| | | | Grau de diferenças em relação a novas telas para apresentar novas funcionalidades / informações. | Sem diferenças | Muitas diferenças | Muitas diferenças | Muitas diferenças |
| | | Design de navegação | Grau de diferenças em relação ao fluxo de navegação (p.ex. sem novas telas). | Sem diferenças | Muitas diferenças | Algumas diferenças | Algumas diferenças |
| | Design de informação | Grau de diferenças em relação à maneira como as informações são apresentadas. | Sem diferenças | Sem diferenças | Muitas diferenças | Muitas diferenças | |
| | Aparência | Design visual | Grau de diferenças em relação às cores. | Algumas diferenças | Algumas diferenças | Muitas diferenças | Muitas diferenças |
| | | | Grau de diferenças em relação à tipografia. | Algumas diferenças | Algumas diferenças | Sem diferenças | Algumas diferenças |
| | | | Grau de diferenças em relação a imagens e ícones. | Algumas diferenças | Algumas diferenças | Algumas diferenças | Muitas diferenças |
| Código | Código-fonte | Grau de diferenças em relação ao JSON do projeto (representado como um vetor de <i>features</i>). | Sem diferenças | Muitas diferenças | Muitas diferenças | Muitas diferenças | |

5.2. As rubricas podem ser aplicadas para avaliar a originalidade de apps?

Aplicando as rubricas para avaliação dos apps observou-se de forma geral que as dimensões e itens são aplicáveis a apps desenvolvidos com App Inventor. A avaliação de cada item corresponde a características gerais observáveis encontradas em diferentes apps. Em relação à estrutura da rubrica e decomposição de itens, algumas dimensões estão mais detalhadas, como a dimensão de design. Portanto, alguns elementos dessa dimensão podem ser agrupados de forma a prover uma avaliação balanceada na qual todas as dimensões possuem níveis de decomposição e detalhamento similares. Realizando as avaliações também surgiu a questão a importância da originalidade do código-fonte. Apesar desse tipo de avaliação ser importante em desafios de programação ou maratonas de programação, que possuem problemas bem definidos, no contexto de problemas abertos essa avaliação pode ser questionável. Nesse sentido, a avaliação se o aluno programou um app usando comandos diferente dos demais pode não ser um indicador tão relevante quanto ao grau de originalidade de um app no contexto de atividades educacionais abertas. Isso porque, na dimensão do código, outros

aspectos, como eficiência e funcionalidade, podem ser mais relevantes que a originalidade em si.

Essa primeira aplicação fornece uma indicação preliminar que o modelo de forma geral permite avaliar corretamente a originalidade de apps como resultados de aprendizagem e é aplicável no contexto educacional. As questões levantadas como a importância da dimensão de código, etc. serão analisados em futuras avaliações para obter evidências empíricas robustas acerca das mesmas.

Ameaças à validade. Para minimizar as ameaças à validade do modelo desenvolvido foi adotada uma metodologia sistemática, decompondo o modelo com base em resultados de um mapeamento sistemático da literatura sobre abordagens de avaliação da criatividade e originalidade [Alves et al. 2020]. A aplicação exemplo para uma avaliação preliminar do modelo foi definida com base em exemplos reais de experiências práticas. A aplicação das rubricas foi feita individualmente por cada autor e discutida até se chegar a um consenso. Levando em consideração que a aplicação exemplo apresentada representa somente uma avaliação preliminar, prevê-se a realização de avaliações por meio de um painel de especialistas de diferentes áreas como trabalho futuro.

6. Conclusão

Como resultado deste estudo é apresentado um modelo inédito baseado em rubricas para avaliar a originalidade de apps como produtos resultantes do processo de aprendizagem no contexto do ensino da computação na Educação Básica. O modelo foi sistematicamente derivado e as rubricas foram aplicadas utilizando apps reais criados dentro do contexto educacional demonstrando a aplicabilidade e corretude das rubricas. Com base nesses resultados as rubricas serão completadas em relação a outras características do produto como adequação e condensação de forma a prover um *feedback* mais compreensivo visando contribuir no desenvolvimento da criatividade na Educação Básica. Os resultados apresentados neste artigo podem ser utilizados por educadores e designers instrucionais, bem como por professores, a fim de avaliar a originalidade dos resultados dos estudantes dentro do contexto educacional.

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001 e do Conselho Nacional de Desenvolvimento Científico e Tecnológico – Brasil (CNPq).

Referências

- Allen, S., Knight, J. (2009). A Method for Collaboratively Developing and Validating a Rubric. *International Journal for the Scholarship of Teaching and Learning*, 3(2), 10.
- Alves N. da C., Gresse von Wangenheim, C., Martins-Pacheco, L. H. (2020). Assessing Product Creativity in Computing Education: A Systematic Mapping Study. *Informatics in Education*. (*in-press*)
- Basili, V. R., Caldiera, G., Rombach, H. D. (1994). The goal question metric approach. In *Encyclopedia of Software Engineering*, John Wiley.

- Basu, S. (2019). Using Rubrics Integrating Design and Coding to Assess Middle School Students' Open-ended Block-based Programming Projects. In Proc. of the 50th Technical Symposium on Computer Science Education, 1211–1217. ACM.
- Bennett, V. E., Koh, K. H., Repenning, A. (2013). Computing creativity: divergence in computational thinking. In Proc. of the 44th Technical Symposium on Computer Science Education, 359–364. ACM.
- Besemer, S., Treffinger, D. J. (1981). Analysis of Creative Products: Review and Synthesis. *Journal of Creative Behavior*, 15(3), 158-178.
- Cavallo, D., Singer, H., Gomes, A., Bittencourt, I., Silveira, I. (2016). Inovação e Criatividade na Educação Básica: Dos conceitos ao ecossistema. *Revista Brasileira de Informática na Educação*, 24(2).
- Clements, D. H. (1995). Teaching creativity with computers. *Educational Psychology Review*, 7, 141–161.
- Ferreira, M. N. F., Pinheiro, F. da C., Gresse von Wangenheim, C., Filho, R. M., Hauck, J. C. R. (2020). Ensinando design de interface de usuário de aplicativos móveis no ensino fundamental. *Revista Brasileira de Informática na Educação*, 28.
- Gal, L., Hershkovitz, A., Morán, A. E., Guenaga, M., Garaizar, P. (2017). Suggesting a log-based creativity measurement for online programming learning environment. In Proc. of the 4th Conference on Learning at Scale, 273-277. ACM.
- Garrett, J. (2010). *The Elements of User Experience: User-Centered Design for the Web and Beyond*, (2nd. ed.). New Riders Publishing, USA.
- Hauck, J. C. R., Gresse von Wangenheim, C., Medeiros, G., Filho, R. M., Alves N., Laurentino, S., Santos, V. (2018). Jovens tutores de programação: um relato de experiência. *Revista Eletrônica de Extensão UFSC*, 15(29).
- Jackson, P. W., Messick, S. (1964). The Person, The Product, and the Response. *ETS Research Bulletin Series*, 2, i-27.
- Lye, S. Y., Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12?. *Computers in Human Behavior*, 41, 51–61.
- Moskal, B. M. (2000). Scoring Rubrics: What, When and How?," *Practical Assessment, Research, and Evaluation*, 7(3).
- Mustafaraj, E., Turbak, F., Svanberg, M. (2017). Identifying Original Projects in App Inventor. In Proc. of the 30th Int. Florida Artificial Intelligence Research Society Conference, 567-572.
- Runco, M. A., Jaeger, G. J. (2012). The standard definition of creativity. *Creativity Research Journal*, 24(1), 92-96.
- Scaico, P. D., Lima, A. A., Silva, J. B. B., Azevedo, S., Paiva, L. F., Raposo, E. H., Alencar, Y., Mendes, J. P., Scaico, A. (2013). Ensino de Programação no Ensino Médio: Uma Abordagem Orientada ao Design com a linguagem Scratch. *Revista Brasileira de Informática na Educação*, 21(2).
- Turbak, F., Mustafaraj, E., Svanberg, M., Dawson, M. (2017). Identifying and Analyzing Original Projects in an Open-Ended Blocks Programming Environment. In Proc. of the 23rd Int. Conference on Visual Languages and Sentient Systems, Pittsburgh, PA, USA.
- Voogt, J., Roblin, N. P. (2012). A comparative analysis of international frameworks for 21st century competences. *Journal of Curriculum Studies*, 44(3), 299–321
- Walia, C. (2019). A Dynamic Definition of Creativity. *Creativity Research Journal*, 31(3), 237-247.
- Yadav, A., Cooper, S. (2017). Fostering Creativity Through Computing. *Comm. of the ACM*, 60(2), 31-33.