Análise Automatizada da Originalidade de Aplicativos Android no Contexto Educacional: Um Mapeamento da Literatura

Nathalia da Cruz Alves[1], Matheus A. Pereira[1], Christiane G. von Wangenheim[1]

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

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

Abstract. *Computing can be taught through teaching app development. Yet, the assessment of these apps as learning outcomes must consider 21th century skills, such as the ability to create original and innovative solutions. Typically, manually assessing the originality of an app may require considerable effort and domain experts in order to obtain valid results. Adopting automated approaches for analyzing originality can support the teacher. Thus, this article presents a systematic mapping of the literature on approaches to automatically analyze the originality of apps, seeking to present an overview on this topic and discuss possible paths within computing education.*

Resumo. *O ensino de computação pode ser feito por meio do desenvolvimento de aplicativos. A avaliação desses aplicativos deve ser feita considerando as habilidades do século XXI como a capacidade de criar soluções originais e inovadoras. Tipicamente, a avaliação da originalidade de um aplicativo de forma manual pode requerer um esforço considerável e especialistas no domínio. A adoção de uma abordagem automatizada de análise da originalidade pode auxiliar o professor. Assim, esse artigo apresenta um mapeamento sistemático da literatura de abordagens de análise automatizada da originalidade de aplicativos, buscando apresentar uma visão geral sobre este tema para mostrar possíveis caminhos dentro do ensino da computação.*

1. Introdução

Com o advento da era digital e seus avanços tecnológicos novas habilidades se fazem necessárias para o desenvolvimento do cidadão, conhecidas como habilidades do século XXI [Binkley et al., 2010]. Uma dessas habilidades essenciais para atender às novas demandas do mercado de trabalho é a aplicação do conhecimento de forma criativa gerando soluções originais [Cavallo et al., 2016]. Originalidade refere-se à extensão da novidade do produto, ou seja, qual o nível de novidade de uma solução a um problema apresentado [Alves et al., 2020].

Uma das formas de estimular a competência de propor soluções originais é por meio do ensino da computação [Resnick, 2017]. Nesse contexto, além de apenas ensinar aos alunos a como se tornar consumidores de tecnologia, pode-se ensinar aos alunos a criação de tecnologia [Scaico et al., 2013]. Uma das alternativas comumente adotada é por meio do ensino de desenvolvimento de aplicativos, utilizando, p.ex., App Inventor [Alves et al., 2020]. App Inventor é um ambiente de programação visual baseado em blocos que permite criar aplicativos para Android. Assim, é possível formar futuros criadores inovadores que sejam capazes de projetar sistemas computacionais/aplicativos que contribuam positivamente para a comunidade [Resnick, 2017].

No contexto educacional, acompanhar o desenvolvimento da competência de propor soluções originais envolve a avaliação e o fornecimento de *feedback* ao aluno com base nos próprios artefatos (aplicativos) criados pelo aluno como resultados da aprendizagem [Hattie e Timperley, 2007; Alves et al., 2020]. Porém, esse processo de avaliação tipicamente requer um esforço considerável do professor. Além disso, em se tratando de julgamento humano, especialmente a avaliação de aspectos como originalidade, pode se tornar alvo de influências e preferências variáveis entre cada pessoa [Alves et al., 2021]. Desta forma, uma alternativa pode ser automatizar parte do processo de avaliação para identificar se a solução computacional (aplicativo) criada pelo aluno é original.

Atualmente, existem pesquisas relacionadas à avaliação manual pelo professor sobre a originalidade de jogos [Scaico et al., 2013] e aplicativos [Alves et al., 2020] no contexto educacional por meio de uma análise de similaridade. No entanto, essas pesquisas abordam a avaliação de uma forma manual apesar de já existirem diversas abordagens voltadas à análise automatizada de similaridade de aplicativos. Existem mapeamentos da literatura focando em identificar abordagens para a avaliação da criatividade dentro do ensino de computação [Alves et al., 2021] ou outras áreas [Snyder et al., 2019]. No entanto, não está claro como abordagens automatizadas poderiam ser aplicadas no contexto educacional buscando identificar o grau de originalidade de um aplicativo criado por um aluno. A carência de pesquisas, notadamente no Brasil, acerca da análise automatizada de originalidade de aplicativos no contexto do ensino de computação, motivou a pesquisa apresentada. Assim, é apresentado um mapeamento sistemático da literatura de abordagens de análise automatizada da originalidade de aplicativos, buscando apresentar uma visão geral sobre este tema para mostrar possíveis caminhos dentro do ensino da computação.

2. Definição e execução do estudo de mapeamento sistemático

A fim de identificar o estado da arte é realizado um mapeamento sistemático da literatura seguindo procedimento definido por Petersen et al. (2015).

2.1. Definição do Protocolo de Revisão

Pergunta de pesquisa. Quais abordagens existem para analisar a originalidade de aplicativos de forma automatizada e aplicável no contexto educacional? Essa pergunta de pesquisa é refinada nas seguintes questões de análise:

AQ1. Quais abordagens existem para a avaliação da originalidade de aplicativos e quais suas características?

AQ2. Como os aplicativos são categorizados em relação à originalidade?

AQ3. Quais técnicas são adotadas na análise?

Critérios de seleção. Conforme o foco da pesquisa, os artigos foram selecionados utilizando critérios de inclusão e exclusão de artigos (Quadro 1). Para obter uma visão mais completa, também foram incluídas abordagens que não necessariamente foram projetadas somente para o contexto educacional.

Quadro 1. Critérios de inclusão e exclusão utilizados nesta pesquisa.

Critérios de inclusão	Critérios de exclusão
- Artigos escritos na língua inglesa. - Artigos que apresentam abordagens para a análise da originalidade com base na extração automatizada de *features* de arquivos de aplicativo para dispositivos móveis. - Artigos publicados a partir de 2010, considerando o avanço recente especificamente em relação a aplicativos móveis.	- Artigos que não sejam baseados em arquivos do aplicativo, como descrição do aplicativo, revisões, dados de uso etc. - Artigos que apresentam abordagens específicas para detecção de *malware*/ofuscação por estarem fora do contexto da criatividade. - Artigos que não apresentam informações substanciais, como artigos-resumo ou apenas de uma página.

Fontes. A pesquisa foi realizada nos principais repositórios e bibliotecas digitais no campo da computação, incluindo ACM Digital Library, IEEE Xplore e Scopus com acesso por meio do Portal Capes. Considerando a existência de literatura cinzenta (não indexada por repositórios clássicos), o Google Scholar também foi utilizado por indexar um grande conjunto de dados de diferentes fontes.

***String* de busca**. Com base na pergunta de pesquisa, foram realizadas pesquisas informais para calibrar a *string* de busca, identificando termos de pesquisa relevantes e seus sinônimos. Sinônimos foram utilizados para minimizar o risco de omitir trabalhos relevantes. Após a definição dos termos e seus sinônimos, definiu-se a *string* de busca básica a ser aplicada nas bases de dados:

```
(novelty OR originality OR similar OR categorization) AND (feature OR semantics OR
functionality) AND ("mobile application" OR app OR ios OR android OR "app inventor")
```

A *string* de busca básica foi adaptada conforme a formatação de cada repositório.

2.2. Execução da busca

A busca foi realizada em setembro de 2020 por um dos autores e revisada pelos coautores. A busca inicial resultou em 817 artigos (Quadro 2).

Quadro 2. Número de artigos identificados por repositório e fase de seleção

Fonte	Nº de resultados da busca	Nº de resultados analisados	Nº de documentos potencialmente relevantes	Nº de documentos relevantes
ACM	20	20	0	0
IEEE	219	200	9	8
SCOPUS	263	200	4	3
Google Scholar	315	200	4	4
Total	**817**	**620**	**17**	**15**

A partir do resultado inicial das buscas, foram selecionados artigos potencialmente relevantes de acordo com os critérios de inclusão e exclusão por meio de uma análise do título, resumo e palavra-chave de cada artigo. Foram analisados os primeiros 200 artigos encontrados em cada busca, observando um decréscimo de documentos relevantes a partir dos 100 primeiros resultados. Em seguida, foram analisados os artigos potencialmente relevantes pelo artigo na íntegra. Documentos duplicados foram eliminados. Como resultado final foram identificados 15 artigos relevantes (Quadro 3).

3. Resultados

3.1. Quais abordagens existem para a avaliação da originalidade de aplicativos e quais suas características?

Foram encontrados 15 artigos (Quadro 3) que apresentam abordagens de agrupamento/*clustering* e classificação a fim de avaliar os aplicativos. As abordagens que utilizam técnicas de agrupamento, tipicamente, buscam agrupar em *clusters* distintos aplicativos originais (novos) e não originais (cópias, tutoriais, etc.). Observa-se também que a maioria dos artigos se trata de abordagens para a detecção de similaridade, com exceção de Svanberg (2017), que fez um estudo explorando e comparando diferentes técnicas de agrupamento a fim de encontrar a mais eficiente para a detecção de similaridade em aplicativos criados por alunos.

Quadro 3. Artigos relevantes

Referência	Técnica utilizada	Referência completa
[Kim et al., 2019]	Agrupamento	Kim, B., Lim, K., Cho, S. J., e Park, M. **Romadroid: A robust and efficient technique for detecting android app clones using a tree structure and components of each app's manifest file**. IEEE Access, 7, 2019.
[Svanberg, 2017]	Agrupamento	Svanberg, M. **Using feature vector representations to identify similar projects in App Inventor**. Proc. of IEEE Blocks and Beyond Workshop, 2017. IEEE.
[Linares-Vás quez et al., 2016]	Agrupamento	Linares-Vásquez, M., Holtzhauer, A., e Poshyvanyk, D. **On automatically detecting similar Android apps**. Proc. of 24th International Conference on Program Comprehension. 2016. IEEE.
[Guo et al., 2018]	Agrupamento	Guo, Y., Li, Y., Yang, Z., e Chen, X. **What's inside my app? understanding feature redundancy in mobile apps**. Proc. of the 26th Conference on Program Comprehension. Gothenburg/Suécia. 2018.
[Hu et al., 2020]	Agrupamento	Hu, Y., Xu, G., Zhang, B., Lai, K., Xu, G., e Zhang, M. **Robust App Clone Detection Based on Similarity of UI Structure**. IEEE Access, 8, 2020.
[Mao et al., 2018]	Agrupamento	Mao, J., Bian, J., Ma, H., Jia, Y., Liang, Z., e Jiang, X. **Robust Detection of Android UI Similarity**. Proc. of International Conference on Communications. 2018. IEEE.
[Li et al., 2017]	Agrupamento	Li, L., Bissyandé, T. F., e Klein, J. **Simidroid: Identifying and explaining similarities in android apps**. Proc. of IEEE Trustcom/BigDataSE/ICESS, 2017. IEEE.
[Crussell et al., 2014]	Agrupamento	Crussell, J., Gibler, C., e Chen, H. **Andarwin: Scalable detection of android application clones based on semantics**. IEEE Transactions on Mobile Computing, 14(10), 2014.
[Akram et al., 2018]	Classificação	Akram, J., Shi, Z., Mumtaz, M., e Luo, P. **Droidcc: A scalable clone detection approach for android applications to detect similarity at source code level**. Proc. of IEEE Annual Computer Software and Applications Conference, 2018. IEEE.
[Zhu et al., 2015]	Agrupamento	Zhu, J., Wu, Z., Guan, Z., e Chen, Z. **Appearance similarity evaluation for Android applications**. Proc. of International Conference on Advanced Computational Intelligence, 2015. IEEE.
[Chen et al., 2019]	Agrupamento	Chen, X., Zou, Q., Fan, B., Zheng, Z., e Luo, X. **Recommending software features for mobile applications based on user interface comparison**. Requirements Engineering, 24(4), 2019.
[Mustafaraj et al., 2017]	Agrupamento	Mustafaraj, E., Turbak, F. A., e Svanberg, M. **Identifying Original Projects in App Inventor**. Proc. of FLAIRS Conference, Florida/EUA, 2017.
[Turbak et al., 2017]	Agrupamento	Turbak, F., Mustafaraj, E., Svanberg, M., e Dawson, M. **Work in progress: Identifying and analyzing original projects in an open-ended blocks programming environment**. Proc. of the Int. DMS Conference on Visual Languages and Sentient Systems, Pittsburgh/EUA, 2017.
[Gopalan, 2018]	Classificação	Gopalan, R. **An Assessment Tool to Analyze Code Written in App Inventor**. Dissertation. Utah State University/EUA, 2018.
[Li et al., 2018]	Agrupamento	Li, Y., Pan, Y., Liu, W., e Zhang, X. **An automated evaluation system for App Inventor Apps**. Proc. of IEEE Intl Conf on Dependable, Autonomic and Secure Computing, 2018. IEEE.

Entre as abordagens encontradas, várias são voltadas ao contexto educacional [Svanberg, 2017; Zhu et al., 2015; Mustafaraj et al., 2017; Turbak et al., 2017; Gopalan, 2018; Li et al., 2018]. Cabe ressaltar que abordagens voltadas a diversos contextos, como o mercado de aplicativos, são aplicáveis ao contexto educacional, especialmente no que se refere a aspectos relacionados à detecção de plágio. Observa-se que com a exceção de Crussell et al. (2014), todos os artigos encontrados foram publicados nos últimos cinco anos, confirmando que a discussão desse assunto é recente (Figura 1).

Figura 1. Quantidade de publicações relevantes por ano sobre abordagens de avaliação de originalidade de apps

3.2. Como os aplicativos são categorizados em relação à originalidade?

Em relação à fonte alvo de análise, todas as abordagens baseiam-se em arquivos do aplicativo, incluindo tanto arquivos de código-fonte como arquivos de propriedades do aplicativo. A maioria das abordagens usa como entrada arquivos Android (.apk). Somente as abordagens voltadas ao contexto educacional utilizam como entrada arquivos de projetos criados com App Inventor (.aia). Nenhuma abordagem utiliza capturas de tela para avaliar a originalidade/similaridade da GUI de apps. As abordagens que avaliam originalidade em relação à GUI analisam exclusivamente arquivos de codificação da GUI (XML) para inferir se diferentes aplicativos utilizam os mesmos componentes.

Observou-se que a maioria das abordagens caracterizam os aplicativos em relação à originalidade adotando um fator/métrica de similaridade. Somente a abordagem proposta por Gopalan (2018) avalia a originalidade focando no construto da criatividade incluindo uma análise se o aplicativo do aluno está correto e atende a alguns pré-requisitos. Caso positivo, Gopalan (2018) então analisa a originalidade comparando o aplicativo do aluno com um *template* básico com o objetivo de identificar se os dois têm algo em comum.

Observa-se que o resultado da análise da maioria das abordagens encontradas é apresentado de forma numérica (Figura 2), como Hu et al. (2020), que representam o nível de similaridade de um aplicativo por meio de um número dentro do intervalo entre 0 e 1. Por outro lado, algumas abordagens como a de Akram et al. (2018) usam valores nominais de conjuntos como {tipo1, tipo2, tipo3} para determinar qual é o tipo de um aplicativo. Foram encontradas abordagens que mesclam tanto a apresentação de um valor numérico e nominal, p.ex. Linares-Vásquez et al. (2016) usam valores numéricos que correspondem a um significado dentro do conjunto {completamente diferentes, na

sua maior parte diferentes, na sua maior parte semelhantes e completamente semelhantes} para indicar o nível de similaridade entre aplicativos.

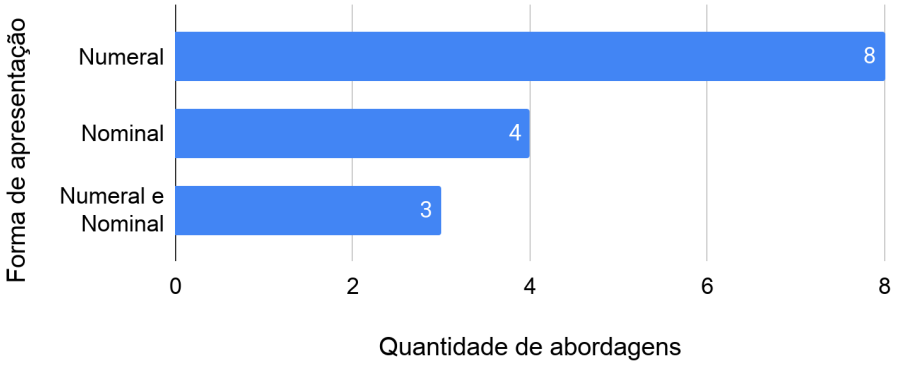

Figura 2. Apresentação do resultado da análise pelas abordagens

Dentro do contexto de avaliação da originalidade a categorização de um aplicativo em relação a sua originalidade depende do conjunto de soluções conhecidas para comparação, também conhecido como Universo de Produtos para Comparação (UPC). As abordagens apresentadas utilizam UPCs diferentes, sendo que algumas abordagens voltadas ao contexto educacional usam aplicativos baseados em tutoriais como UPC e consideram um aplicativo original caso seja diferente do tutorial ensinado anteriormente. Consequentemente, os UPCs também variam em relação ao tamanho (Figura 3). Há uma grande variação em relação à quantidade de dados, com estudos utilizando de 100 a 1000 aplicativos, como Gopalan (2018) que usa 111 aplicativos, até estudos utilizando conjuntos de aplicativos com grande quantidade de dados, como Akram et al. (2018) com 30 milhões de aplicativos. Além disso, Turbak et al. (2017), que apresentam uma abordagem voltada ao contexto educacional, relatam somente que o UPC consiste em aplicativos criados por 6.012 usuários de App Inventor e não diretamente a quantidade de aplicativos, já que um mesmo usuário pode criar 1+ aplicativos. No contexto educacional, uma grande quantidade de aplicativos como 30 milhões pode não ser viável, pois deseja-se comparar o aplicativo criado pelo aluno com outros aplicativos criados por outros alunos em contextos similares. Desta forma, pode ser necessária uma calibração do UPC voltado ao contexto educacional.

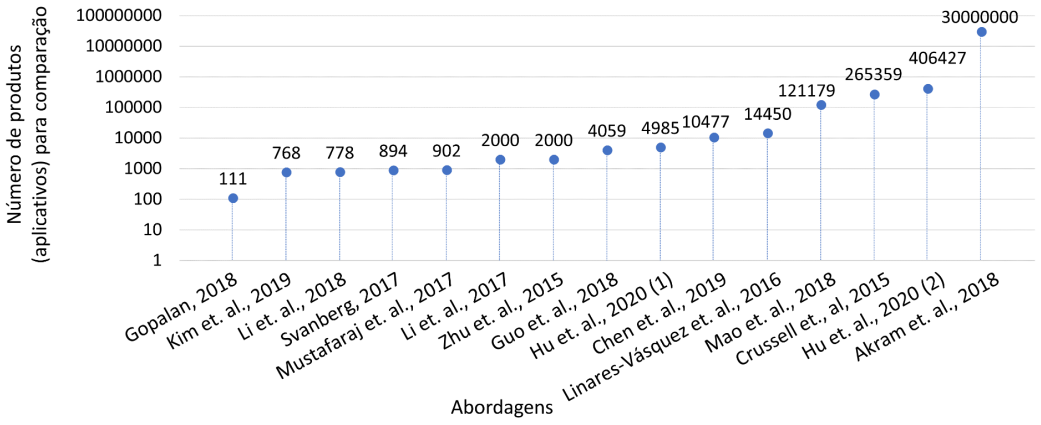

Figura 3. Tamanho do Universo de Produtos para Comparação das abordagens

3.3. Quais técnicas são adotadas na análise?

Como parte da análise de originalidade, tipicamente as abordagens extraem *features* do aplicativo de forma automatizada. A maioria utiliza algoritmos próprios, como Kim et al. (2019), Mao et al. (2018), Crussell et al. (2014), que extraem *features* selecionadas pelos autores. Algumas abordagens utilizam ferramentas de extração existentes (Figura 4), como Guo et al. (2018). Entre as ferramentas usadas estão "apktool", que faz engenharia reversa de aplicativos fechados e binários para Android, "AMtool", que é uma ferramenta de linha de comando fornecida pelo Android, e "UIAutomator", que permite a extração de informações de *widgets* e visualização de sua hierarquia [Hu et al., 2020]. Algumas abordagens utilizam em conjunto ferramentas existentes e soluções próprias, como Linares-Vásquez et al. (2016), Hu et al. (2020) e Akram et al. (2018).

Figura 4. Ferramentas de extração de *features*

O resultado da extração de *features* é analisado utilizando técnicas/algoritmos distintos. Diversas abordagens que analisam o código-fonte utilizam algoritmos baseados em análise léxica, buscando identificar quantas vezes determinado comando, componente, etc., aparece em um documento ou em uma coleção de documentos, como Svanberg (2017). Outras abordagens utilizam algoritmos baseados em análise sintática, criando a estrutura de árvore sintática de um programa de forma a identificar seus nodos, como Kim et al. (2019). Algumas abordagens utilizam mapas, definindo como chave o aplicativo e como índices os principais recursos (ex.: permissões) do código-fonte dos arquivos APK, como Akram et al. (2018).

Para a análise de similaridade, a maioria das abordagens utiliza medidas e algoritmos existentes já conhecidos (Figura 5). Por exemplo, Linares-Vásquez et al. (2016) usam a Medida de Cosine, que mede a distância entre dois vetores de dimensão n, Mustafaraj et al. (2017) usam o Índice de Jaccard, que mede a similaridade entre conjuntos de amostras finitas, Mao et al. (2018) utilizam o algoritmo K-Means, que agrupa aplicativos similares dentro de um grupo mais próximo da média, e Kim et al. (2019) utilizam o algoritmo *Longest Common Subsequence*, que gera as diferenças (`diff`) entre dois aplicativos. Por outro lado, algumas abordagens também propõem medidas próprias, como cálculo de redundância elaborado por Guo et al. (2018) e o *Creativity Score* elaborado por Gopalan (2018), porém são menos frequentes.

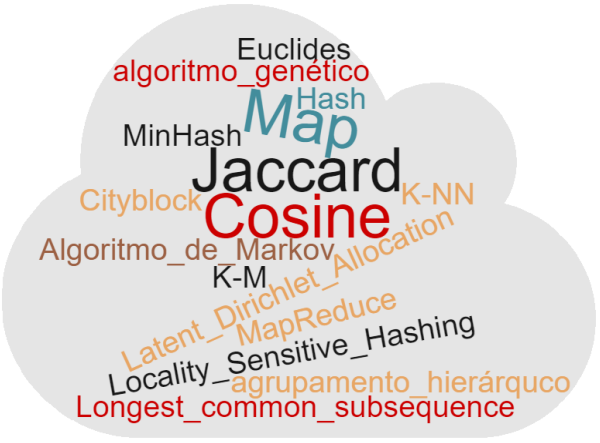

Figura 5. Técnicas e algoritmos adotados nas abordagens

4. Discussão

Em geral, grande parte da pesquisa sobre originalidade tem se baseado em julgamentos subjetivos de avaliadores humanos para avaliar aspectos de originalidade e criatividade. Tipicamente, as avaliações manuais são feitas utilizando técnicas como Análise Consensual (Amabile, 1982) em que o avaliador relata sua impressão de originalidade de um artefato, sem basear-se em uma rubrica ou definição formal, necessitando de um avaliador experiente. No entanto, visando a introdução ampla do ensino de computação em escolas brasileiras, atualmente caracterizadas pela falta de professores formados na área de computação e sem tempo livre no contexto de turmas grandes, a adoção de abordagens automatizadas como parte da avaliação pode auxiliar em diversos aspectos como rapidez, redução de esforço e objetividade, além de prover evidências tangíveis.

As abordagens encontradas utilizam modelos distintos, porém apresentam resultados da análise de forma similar e aplicáveis em um contexto educacional. Neste contexto, um valor, p.ex., numeral no intervalo [0, 1] ou nominal sobre a originalidade de um aplicativo, fornece evidências tangíveis para o professor decidir se um aplicativo é suficientemente original. Além disso, abordagens que apresentam agrupamentos permitem ao professor identificar que tipos de aplicativo são mais frequentes dentro do universo de aplicativos criados por alunos.

Uma questão em aberto é a interpretabilidade de resultados de abordagens que apresentam modelos de Inteligência Artificial (IA) mais complexos, como o modelo proposto por Akram et al. (2018). Especialmente no contexto educacional, a razão pela qual um aplicativo é considerado original ou não original é relevante no momento de prover um *feedback* construtivo mais completo ao aluno. Ao avaliar o aplicativo do aluno dando somente uma classificação "não original" pode frustrá-lo, não o motivar e ajudá-lo a melhorar a aprendizagem, especialmente considerando que criar é difícil [Vygotsky, 2014]. Ao receber uma avaliação indicando que o aplicativo criado não é original sem prover um *feedback* explicando as razões para essa avaliação pode gerar um sentimento de sofrimento também conhecidos como "tormentos da criação" (Vygotsky, 2014). No entanto, cabe ressaltar que essa é uma questão em aberto em relação ao *feedback* educacional em geral. Neste contexto, uma oportunidade de pesquisa é o desenvolvimento de uma abordagem de avaliação automatizada da

originalidade de aplicativos que além de prover um resultado, também apresenta um *feedback* construtivo e motivador.

4.1. Ameaças à validade

Mapeamentos sistemáticos podem ter a tendência de que resultados positivos são mais prováveis de serem publicados do que negativos. Entretanto, consideramos que para este mapeamento as abordagens de análise automatizada da originalidade de aplicativos Android no contexto educacional são mais relevantes do que a análise de seus resultados, assim consideramos esse risco mínimo. Também corremos o risco de ter omitido algum resultado relevante. Para mitigar esse risco, construímos cuidadosamente a *string* de busca para ser o mais inclusiva possível, considerando não apenas os conceitos principais, mas também seus sinônimos. Os primeiros 200 resultados foram analisados em todos os repositórios de buscas, nos quais observou-se a diminuição da relevância dos resultados a partir dos 100 primeiros artigos. Ameaças à seleção de abordagens relevantes e extração de dados foram mitigadas fornecendo uma definição detalhada de critérios de inclusão/exclusão. Definimos e documentamos um protocolo rígido para a seleção dos estudos, discutindo a seleção até alcançar um consenso.

Ameaças à análise. Em relação à aplicação deste tipo de abordagem no contexto educacional, surgem diversas pesquisas na literatura criando um corpo conceitual sobre a adoção de IA na avaliação da originalidade enfocando a criatividade, como Beghetto (2019), que diz "o aprendizado de máquina e a IA podem ser capazes de avaliar a criatividade usando abordagens atualmente usadas e talvez levem a novas maneiras de avaliar a criatividade."

5. Conclusão

Como resultado desta pesquisa é apresentado um mapeamento sistemático sobre abordagens de análise automatizada da originalidade de aplicativos Android fornecendo uma visão geral sobre este tema visando a sua aplicação no contexto do ensino de computação. Com base nesses resultados, espera-se contribuir para o debate de como adotar abordagens automatizadas para auxiliar na avaliação da originalidade de aplicativos criados por alunos como resultados da aprendizagem. Considerando o progresso recente na área de IA, apropriar-se de tais abordagens de avaliação automatizada pode mitigar limitações relacionadas à subjetividade e tempo despendido na análise manual dos aplicativos dos alunos. Dado que em um contexto educacional os professores podem ter que avaliar um número grande de artefatos criados por uma turma levando à fadiga e até mesmo impossibilidade prática de avaliar o aspecto de originalidade, adotar uma abordagem automatizada pode auxiliar na ampla adoção de avaliações objetivas sobre a originalidade de aplicativos. Além disso, pode fornecer uma visão geral do estado da arte indicando também oportunidades de pesquisa. Os resultados apresentados neste artigo podem ser utilizados por educadores e designers instrucionais, bem como por professores, a fim de enfatizar o ensino dessa habilidade do século XXI e avaliar a originalidade dos resultados dos estudantes dentro do contexto educacional de uma forma viável no contexto de escolas brasileiras.

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

Alves, N. da C., Gresse von Wangenheim, C., Alberto, Matheus, Martins-Pacheco, L. H. Uma Proposta de Avaliação da Originalidade do Produto no Ensino de Algoritmos e Programação na Educação Básica. In: Anais do Simpósio Brasileiro de Informática na Educação, Online. 2020.

Alves, N. da C., Gresse von Wangenheim, C., Martins-Pacheco, L. H. Assessing Product Creativity in Computing Education: A Systematic Mapping Study. Informatics in Education, 20(1), 19-45, 2021.

Alves, N. da C., Gresse von Wangenheim, C., Martins-Pacheco, L. H., Borgatto, A. F. Existem concordância e confiabilidade na avaliação da criatividade de resultados tangíveis da aprendizagem de computação na Educação Básica? In: Anais do Simpósio Brasileiro de Educação em Computação, Jataí, Goiás, 2021.

Amabile, T. M. Social psychology of creativity: A Consensual Assessment Technique. Journal of Personality and Social Psychology, 43, 997-1013, 1982.

Beghetto, R. A. Large-scale assessments, personalized learning, and creativity: Paradoxes and possibilities. ECNU Review of Education, 2(3), 311-327, 2019.

Binkley, M., Erstad, O., Herman, J., Raizen, S., Ripley, M. and Rumble, M. Defining 21st Century Skills. Draft paper. Melbourne: The University of Melbourne, 2010.

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).

Hattie, J., Timperley, H. The Power of Feedback. Review of Educational Research, 77(1), 81-112, 2007.

Petersen, K., Vakkalanka, S., Kuzniarz, L. Guidelines for conducting systematic mapping studies in software engineering: An update. Information and Software Technology, 64, 1-18, 2015.

Resnick, M., Robinson, K. Lifelong kindergarten: Cultivating creativity through projects, passion, peers, and play. MIT press, 2017.

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.. Ensino de Programação no Ensino Médio: Uma Abordagem Orientada ao Design com a linguagem Scratch. Revista Brasileira de Informática na Educação, 21(2), 2013.

Snyder, H. T., Hammond, J. A., Grohman, M. G., Katz-Buonincontro, J. Creativity measurement in undergraduate students from 1984–2013: A systematic review. Psychology of Aesthetics, Creativity, and the Arts, 13(2), 133–143, 2019.

Vygotsky, L. S. Imaginação e criatividade na infância. Tradução de João Pedro Fróis. São Paulo: WMF Martins Fontes, 2014.