UNIVERSIDADE FEDERAL OF SANTA CATARINA

CENTRO TECNOLÓGICO

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Nathalia da Cruz Alves

Assessing the Creativity of Mobile Applications in Computing Education

Florianópolis

2023

Nathalia da Cruz Alves

Assessing the Creativity of Mobile Applications in Computing Education

Tese submetida ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Santa Catarina como requisito parcial para a obtenção do título de Doutora em Ciência da Computação.

Orientador: Prof.ª Dr.ª rer. nat. Christiane Gresse von Wangenheim, PMP.

Florianópolis

2023

Ficha de identificação da obra

da Cruz Alves, Nathalia

Assessing the creativity of mobile applications in computing education / Nathalia da Cruz Alves ; orientadora, Christiane Gresse von Wangenheim, 2023.

140 p.

Tese (doutorado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Ciência da Computação, Florianópolis, 2023.

Inclui referências.

1. Ciência da Computação. 2. Computing Education. 3. Creativity. 4. Assessment. 5. App Inventor. I. Gresse von Wangenheim, Christiane. II. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Ciência da Computação. III. Título.

Nathalia da Cruz Alves

Assessing the Creativity of Mobile Applications in Computing Education

O presente trabalho em nível de Doutorado foi avaliado e aprovado, em três de agosto de 2023, pela banca examinadora composta pelos seguintes membros:

Prof. James Corey Kaufman, Dr.
University of Connecticut

Prof. Rafael de Santiago, Dr.
Universidade Federal de Santa Catarina

Prof. Roberto Almeida Bittencourt, Dr.
Universidade Estadual de Feira de Santana

Certificamos que esta é a versão original e final do trabalho de conclusão que foi julgado adequado para obtenção do título de Doutora em Ciência da Computação.

GOV.BR
Documento assinado digitalmente
PATRICIA DELLA MEA PLENTZ
Data: 30/10/2023 18:19:42-0300
CPF: ***.243.700-**
Verifique as assinaturas em https://v.ufsc.br

Prof.ª Patricia Della Méa Plentz, Dr.ª
Coordenação do Programa de Pós-Graduação

ICP-Edu
Documento assinado digitalmente
Christiane Anneliese Gresse Von Wangenheim
Data: 30/10/2023 16:39:20-0300
CPF: ***.218.819-**
Verifique as assinaturas em https://v.ufsc.br

Prof.ª Christiane Gresse von Wangenheim, Dr.ª
Orientadora

Florianópolis, 2023.

ACKNOWLEDGEMENTS

I would like to thank everyone who contributed in one way or another to the planning, execution and conclusion of this work. I find myself fortunate and grateful for the network of support that has surrounded and propelled me on this journey.

I am extremely grateful to Prof. Christiane for her guidance, helpfulness, diligence, availability, consistency, and support, not only throughout this work but during our ten-year partnership since I was an undergraduate student. I am thankful to Prof. Jean for all his support in carrying out intricate and technical activities. I am thankful to Prof. Kaufman for his guidance, mentorship and insightfulness during the period in the US. I am thankful to Prof. Plentz for all her support in academic procedures. I am thankful to Prof. Roberto and Prof. Rafael, who agreed to participate as members of the committee and provided insightful and constructive feedback.

I am thankful to all the current and former members of the Software Quality Group and Computing at School Initiative at UFSC. They effectively collaborated and contributed to the completion of this work. I am thankful to the App Inventor Foundation for sharing essential data for this work. I am thankful to the study participants and anonymous reviewers who also provided constructive feedback during this work.

I am extremely grateful to my beloved parents, Cida and Valmir, who provided essential support and encouragement during this work. I am thankful to my inner circle and closest friends, the Biruleibes team, and my colleagues at UConn for all their support.

I would like to thank CAPES for the financial support throughout the development of this work. This work was carried out with the support of the *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior* – Brazil (CAPES) – Financing Code 001.

Not everything that can be counted counts, and not everything that counts can be counted.

– William Bruce Cameron

RESUMO

A criatividade é uma habilidade importante do século XXI. Embora seja tradicionalmente associada às artes e à literatura, ela também pode ser desenvolvida como parte do ensino da computação. Uma das formas de estimular a criatividade é por meio do ensino de desenvolvimento de aplicativos móveis com o App Inventor, um ambiente de programação baseado em blocos tipicamente usado na Educação Básica. Nesse contexto, a avaliação desempenha um papel crucial na avaliação da aprendizagem e do progresso dos alunos. Embora existam modelos de avaliação da criatividade, sua avaliação enfocando aplicativos móveis na educação em computação permanece relativamente inexplorada. Nesse contexto, esta tese apresenta um modelo de avaliação da criatividade de aplicativos móveis desenvolvidos com o App Inventor. De acordo com a definição de criatividade, o modelo inclui as dimensões de originalidade, flexibilidade e fluência. Essas dimensões são avaliadas em relação às funcionalidades, componentes, programação, tópicos e tags, computando suas frequências dentro do aplicativo e comparando-as com um universo de referência de aplicativos existentes. Com o objetivo de analisar a capacidade do modelo em avaliar a criatividade de aplicativos móveis foram realizadas diversas análises estatísticas. Os resultados demonstram que o modelo é capaz de diferenciar aplicativos considerados criativos de acordo com avaliadores humanos, indicando a eficácia do modelo. Além disso, as análises estatísticas confirmaram a confiabilidade, a validade de construto e a qualidade do modelo de avaliação. O uso de um universo de referência pelo modelo fornece uma base padronizada e objetiva para medir a criatividade, facilitando comparações e interpretação dos resultados. Ao introduzir um modelo de avaliação da criatividade de aplicativos móveis desenvolvidos com o App Inventor, visa-se estimular e nutrir a criatividade entre os estudantes no contexto do ensino da computação. O modelo desenvolvido contribui para o estabelecimento de uma ferramenta de avaliação para avaliar a criatividade, beneficiando tanto educadores como estudantes.

Palavras-chaves: Criatividade. Aplicativos Móveis. Avaliação. App Inventor. Educação Básica, Educação em Computação.

ABSTRACT

Creativity is an important skill of the 21st century. Although it is traditionally associated with arts and literature, it can also be developed as part of computing education. Creativity can be stimulated by teaching students to develop their mobile applications with App Inventor, a block-based programming environment typically used in K-12. As an essential component of effective teaching, assessment plays a crucial role in gauging student learning and progress. While there exist established assessment models for creativity, the assessment of creativity of mobile applications in computing education remains relatively uncharted. This work introduces a comprehensive assessment framework for assessing the creativity of mobile applications developed with App Inventor. In accordance with the definition of creativity, the proposed framework assesses the originality, flexibility, and fluency of mobile applications. These dimensions are assessed with regard to the functionalities, components, programming blocks, topics, and tags, by computing their frequencies within the mobile application and comparing them to a reference universe of existing mobile applications. Statistical analyses were conducted to evaluate the framework's capability in assessing the creativity of mobile applications. The results demonstrate that the framework is capable of differentiating mobile applications considered creative according to human raters, indicating its effectiveness. Furthermore, the statistical analyses confirmed the reliability, construct validity, and quality of the assessment framework. The framework's use of a reference universe provides a standardized and objective basis for measuring creativity, facilitating meaningful comparisons and interpretation of results. By introducing a comprehensive assessment framework for assessing the creativity of mobile applications developed with App Inventor, this model aims to stimulate and nurture creativity among students in computing education, benefiting both educators and students.

Keywords: Creativity. Mobile Application. Assessment. App Inventor. K-12. Computing Education.

RESUMO EXPANDIDO

Introdução

Na sociedade digital de hoje em constante mudança, a criatividade é considerada uma das principais competências do século XXI, essencial para o sucesso profissional e pessoal. Consequentemente, o desenvolvimento da criatividade dos estudantes na Educação Básica tornou-se uma preocupação dominante. Embora a criatividade seja tradicionalmente associada às artes, música e literatura, ela também pode ser desenvolvida como parte de outras áreas de conhecimento, como a computação, para as quais são necessários design, pesquisa e inovação. Considerando que a criatividade pode ser ensinada como parte do ensino de computação, existem várias propostas de unidades instrucionais que ensinam computação e criatividade por meio do desenvolvimento de aplicativos móveis. Ao adotar uma estratégia de ação computacional, os estudantes aprendem conceitos básicos e como criar um aplicativo móvel para resolver um problema relacionado às suas vidas e comunidades usando o App Inventor. Embora já existam várias propostas de avaliação da aprendizagem de conceitos e práticas de computação, a avaliação da criatividade nesse contexto é escassa, uma vez que a avaliação da criatividade é desafiadora. Portanto, a pergunta que orienta esta pesquisa é: é possível avaliar automaticamente a criatividade de aplicativos móveis como resultados de aprendizagem no ensino da computação de uma forma fiável e válida?

Objetivos

O principal objetivo desta pesquisa é desenvolver e avaliar uma abordagem automatizada para avaliar a criatividade de aplicativos móveis como resultado de projetos autênticos no contexto do ensino de computação. Para alcançar esse objetivo, é desenvolvido um modelo conceitual que especifica critérios para avaliar a criatividade de aplicativos móveis criados com o App Inventor. Além disso, a fim de facilitar a aplicação do modelo na prática educacional, é implementado um módulo de software para avaliação automatizada. O modelo de avaliação é avaliado estatisticamente em termos de confiabilidade e validade. Para alcançar o objetivo principal, os seguintes objetivos específicos são identificados: O1. Sintetizar as abordagens do estado da arte para avaliar a criatividade de programas de computador no contexto da educação em computação; O2. Desenvolver um modelo conceitual para avaliar a criatividade de aplicativos móveis no contexto da educação em computação com base nas dimensões identificadas na literatura; O3. Desenvolver um módulo de software para avaliar automaticamente os critérios do modelo conceitual, adotando técnicas estatísticas e de inteligência artificial; O4. Integrar o módulo de avaliação automatizada na ferramenta de avaliação CodeMaster; e O5. Avaliar o modelo por meio de uma análise estatística de sua confiabilidade e validade.

Metodologia

A metodologia de pesquisa é definida com base na proposta de Saunders, Lewis e Thornhill (2019) para classificar diferentes camadas da pesquisa. Dado que a propriedade de interesse é a criatividade em aplicativos móveis, há uma ênfase em soluções práticas e resultados para o ensino de computação. Em termos de natureza, esta pesquisa é caracterizada como pesquisa aplicada e os dados são analisados qualitativamente e quantitativamente. O horizonte temporal é transversal, uma vez que são coletados dados de eventos dissociados no tempo. A pesquisa é dividida em quatro etapas, começando com a análise do estado da arte, seguida pelo desenvolvimento do modelo de avaliação conceitual adotando-se o framework *Evidence-Centered Design* (ECD). Para a implementação do modelo de avaliação automatizado, é seguido um processo de desenvolvimento iterativo e incremental, bem como um processo de aprendizado de máquina iterativo centrado no ser humano. Para avaliar a confiabilidade e validade do modelo de avaliação, são realizadas análises estatísticas em um estudo de caso seguindo a abordagem *Goal-Question-Metric* (GQM).

Resultados e Discussão

Os resultados das análises estatísticas fornecem evidências de que a nota de criatividade gerada pelo modelo é mais alta para aplicativos móveis criativos em comparação com aplicativos móveis não criativos, de acordo com avaliadores humanos. Em geral, a análise demonstra uma associação positiva entre a nota de criatividade gerada pelo modelo e a criatividade percebida dos aplicativos móveis, apoiando a ideia de que o modelo captura os aspectos criativos dos aplicativos móveis. A análise de confiabilidade, avaliada usando o coeficiente ômega, revelou uma boa confiabilidade geral ($\omega = 0.86$). Em termos de validade convergente, a matriz de correlação indicou que os itens de flexibilidade e fluência possuem fortes correlações internas, enquanto a dimensão de originalidade exibiu correlações internas mais fracas. Além disso, as correlações entre as dimensões indicaram que flexibilidade e fluência se correlacionaram entre si e com itens da dimensão de originalidade. A análise fatorial exploratória revelou a presença de fortes cargas fatoriais para dois e três fatores subjacentes no modelo, indicando um ajuste razoavelmente bom para esses fatores. Os resultados da análise usando a Teoria da Resposta ao Item, indicam que a maioria dos itens foi calibrada adequadamente, com parâmetros de inclinação acima de 1, sugerindo um bom poder discriminatório. No geral, a análise fornece evidências de qualidade no modelo, pois demonstra a capacidade do modelo de discriminar entre diferentes níveis do construto. No entanto, estudos adicionais e refinamentos podem ser necessários para aprimorar a qualidade geral do modelo.

Considerações Finais

O framework e modelo de avaliação propostos oferecem uma abordagem para avaliar a criatividade de aplicativos móveis desenvolvidos como resultados de aprendizagem na educação em computação, considerando dimensões de originalidade, flexibilidade e fluência. Por meio de análises estatísticas e da comparação das características de aplicativos móveis com um universo de referência de aplicativos móveis existentes, o modelo demonstra a capacidade de diferenciar aplicativos móveis criativos de forma positiva em comparação com aplicativos móveis não criativos, de acordo com avaliadores humanos. Com o objetivo de automatizar a avaliação da criatividade em aplicativos móveis, esta pesquisa expande as fronteiras da automação em um domínio tradicionalmente associado ao julgamento humano e à avaliação subjetiva, avançando o campo da ciência da computação e abrindo novos caminhos para explorar a complexa relação entre a criatividade humana e sistemas computacionais. Ao fornecer uma ferramenta automatizada de avaliação da criatividade, esta pesquisa democratiza o acesso à avaliação da criatividade, permitindo que estudantes e professores de diferentes contextos possam cultivar e aprimorar suas habilidades criativas com base em evidências. Ao fornecer *feedback* ao aluno, espera-se apoiar seu progresso de aprendizado, bem como fornecer *feedback* ao professor, permitindo a melhoria do ensino. Além disso, a automatização do modelo auxilia no fornecimento ágil de evidências para a avaliação da criatividade de aplicativos móveis, que pode ser complementada pela avaliação humana.

Palavras-chave: Criatividade. Aplicativos móveis. Avaliação. App Inventor, Educação Básica, Ensino de computação.

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

BNCC	*Base Nacional Comum Curricular*
CAT	Consensual Assessment Technique
DT	Divergent Thinking
ECD	Evidence-Centered Design
GQM	Goal-Question-Metric
GRM	Graded Response Model
IRT	Item Response Theory
MOOC	Massive Open Online Courses
PADI	Principled Assessment Designs for Inquiry
PISA	Programme for International Student Assessment
SE	Software Engineering
TTCT	Torrance Tests of Creative Thinking
UI	User Interface

SUMMARY

1 INTRODUCTION

1.1 CONTEXTUALIZATION

In today’s rapidly changing digital society, creativity is considered one of the main 21st century competencies essential for professional and personal success (KAUFMAN and BEGHETTO, 2009; VAN LAAR, VAN DEURSEN, *et al.*, 2020). Consequently, developing students’ creativity from an early age has become a dominant concern (VOOGT and ROBLIN, 2012; BEGHETTO, 2010). Diverse curriculum frameworks also explicitly express the need for schools to foster creativity (P21, 2020; ISTE, 2020; VOOGT and ROBLIN, 2012). Supplying students with opportunities to engage in creative ways can help them to develop the capacity to undertake work that cannot be easily automated and address increasingly complex challenges with out-of-the-box solutions (STERNBERG, 2015).

Although creativity is traditionally associated with arts, music, and literature, it can also be developed as part of other knowledge areas, such as computing, for which design, research, and innovation are required (BENNETT, KOH, and REPENNING, 2011). Teaching students to solve computational problems by creating novel and appropriate computer programs can allow students to express their ideas and thus their creativity (ROMERO, LEPAGE, and LILLE, 2017). Computing may also nurture competencies, such as imagination, visualization, and abstraction, to solve problems creatively (CLEMENTS and GULLO, 1984; YADAV and COOPER, 2017; GROVER and PEA, 2013) while, on the other hand, creative skills enhance solving algorithmic problems, creating computational artifacts, and developing new knowledge (SHELL, HAZLEY, *et al.*, 2014).

In computing education, creating computational artifacts is typically introduced using active learning strategies also known as “learning-by-making” (RODE, WEIBERT, *et al.*, 2015). By adopting a complexity increasing approach, such as the use-modify-create cycle (LEE, MARTIN, *et al.*, 2011; LYTLE, CATETÉ, *et al.*, 2019), students first learn to “use” and analyze a given computational artifact, then to remix and modify an existing artifact, until eventually to “create” a new one. This leads students to develop the ability to generate original and useful ideas and solutions, especially during the creation stage. In order to stimulate creativity, learning activities are often posed as open-ended ill-defined problems in a constructivist context adopting a problem-based learning strategy with authentic tasks (KIESLER, 2022).

Considering that creativity can be taught as part of computing education, there are several proposals for instructional units teaching computing and creativity through the development of computer programs, such as mobile apps, short for applications (BASU, 2019; PATTON, TISSENBAUM, and HARUNANI, 2019). By adopting a strategy of computational action (TISSENBAUM, SHELDON, and ABELSON, 2019), students learn basic concepts and how to create a mobile app to solve a problem related to their lives and community using App Inventor (MIT, 2012). In this context, the mobile app is developed using a problem-based constructivist approach, engaging students in the construction of digital and tangible computing artifacts through the use of technologies (HAUCK, GRESSE VON WANGENHEIM, *et al.*, 2018; FERREIRA, GRESSE VON WANGENHEIM, *et al.*, 2019).

As part of the learning process, it is important to provide feedback through the assessment expressed as grades as well as to enable the provision of feedback to guide the learning and the teaching process. Despite the recognition of the importance of measuring and assessing creativity, however, there are only a few approaches for measuring creativity in mobile apps (ALVES, GRESSE VON WANGENHEIM, and MARTINS-PACHECO, 2021a).

1.2 PROBLEM

Although there exist already several proposals for assessing the learning of computing concepts and practice, so far, the assessment of creativity in this context is scarce (ALVES, GRESSE VON WANGENHEIM, and MARTINS-PACHECO, 2021a) as assessing creativity is challenging (HENRIKSEN, MISHRA, and MEHTA, 2019). While initial scholarship has explored the scoring of creativity in computer programming (ALVES, GRESSE VON WANGENHEIM, and MARTINS-PACHECO, 2021a), the focus has primarily been on open-ended well-structured tasks (KOH, BENNETT, and REPENNING, 2011; BENNETT, KOH, and REPENNING, 2013; MANSKE and HOPPE, 2014; HERSHKOVITZ, SITMAN, *et al.*, 2019). The use of well-structured programming activities with multiple levels is a commonly employed educational approach for promoting structured learning. This method enables students to incrementally advance in complexity, thereby establishing a solid knowledge foundation (UYSAL, 2014). However, one potential limitation of this approach is that it may restrict students from exploring their ideas and developing computer programs based on their interests, and thus the expression of creativity.

One alternative to well-structured tasks is to adopt a problem-based learning approach with authentic tasks and projects (KIESLER, 2022) in which students develop open-ended free-choice projects reflecting students' world and concerns. In such scenarios, the assessment typically follows a performance-based approach, in which the computer program representing the outcome of the learning process by the students is used for the assessment (RITCHIE, 2001). Such assessments typically measure whether desired properties are present in the outcomes and to what degree.

Most of the existing approaches for assessing creativity in open-ended free-choice computing projects are based on subjective criteria, as proposed by Grover et al. (2018), wherein instructors assess computer programs using an ordinal scale that includes descriptors such as "not very novel," "some novelty," or "very novel" (BASU, 2019). Often, multiple raters will assess these programs and they tend to show agreement (e.g., BASU, 2019), consistent with the Consensual Assessment Technique (CAT) (AMABILE, 1996; KAUFMAN and BAER, 2012). However, these methods can be time-consuming and resource-intensive, posing challenges in scenarios such as large classes or Massive Open Online Courses (MOOCs). While expert judgments offer contextually tailored results in creativity assessment, providing consistent and timely feedback may be unfeasible (BEATY and JOHNSON, 2020). Consequently, these factors restrict the effectiveness of manual approaches as the sole assessment option.

Despite the recognition of the importance of assessing creativity with respect to open-ended free-choice computer programs, such as mobile apps, there is a lack of valid and reliable approaches that provide systematic support in the definition, execution, and analysis of the creativity of the student's mobile apps as part of computing education (ALVES, GRESSE VON WANGENHEIM, and MARTINS-PACHECO, 2021a). Thus, the question guiding this research is:

Research question: Is it possible to automatically assess the creativity of mobile apps as learning outcomes in computing education in a reliable and valid manner?

1.3 OBJECTIVES

The main objective of this research is to develop and evaluate an automated approach for assessing the creativity of open-ended free-choice mobile apps as a result of ill-defined activities and authentic projects in computing education. To achieve this objective, a conceptual model that specifies criteria for assessing the creativity of mobile apps created with App Inventor is developed. Furthermore, to facilitate the application of the model in educational practice, a software module for automatized assessment is implemented. The model is integrated into CodeMaster[1], an automated tool for assessing computational thinking, interface design, and aesthetics of apps (GRESSE VON WANGENHEIM, HAUCK, *et al.*, 2018).

The proposed assessment model is evaluated statistically in terms of reliability and validity.

In order to achieve the main objective, the following specific objectives are identified:

O1. Synthesize the state-of-the-art approaches used to assess the creativity of computer programs in a computing education context.

O2. Develop a conceptual model for assessing the creativity of mobile apps in the context of computing education based on the dimensions identified in the literature.

O3. Develop a software module to automatically assess the criteria of the conceptual model adopting machine learning and statistical techniques.

O4. Integrate the automated assessment module into the CodeMaster assessment tool.

O5. Evaluate the automated assessment module by performing a statistical analysis regarding its reliability and validity.

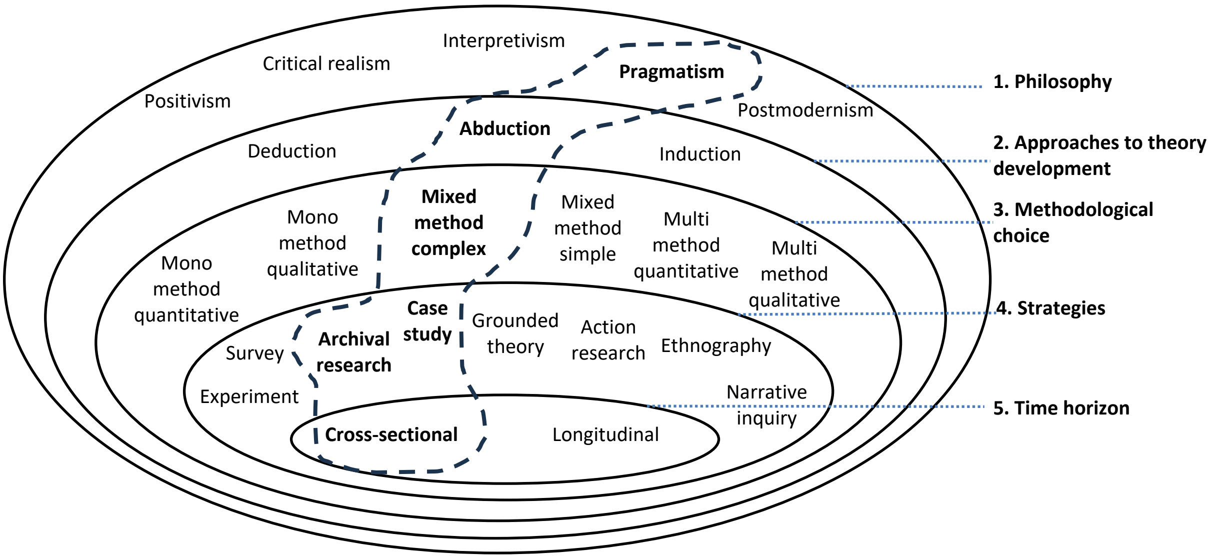
1.4 RESEARCH METHODOLOGY

1.4.1 Research context

The research methodology is defined based on the research ‘onion’ proposed by Saunders, Lewis, and Thornhill (2019) for classifying different aspects (layers) of research (Figure 1).

[1] http://apps.computacaonaescola.ufsc.br/codemaster/

Figure 1 – Research layers personalized for this research



Source: Elaborated by the author based on Saunders, Lewis, and Thornhill (2019).

Given that the property of interest in this research is creativity in mobile apps, there is an emphasis on practical solutions and outcomes for computing education, i.e., there exists a practical meaning of knowledge for a specific context. According to the classic classification of research, this research is characterized, in terms of nature, as applied research (HEDRICK, BICKMAN, and ROG, 1993). This research starts with the problem of identifying creativity properties in mobile apps and aims to contribute practical solutions that inform future practice based on a pragmatic philosophy.

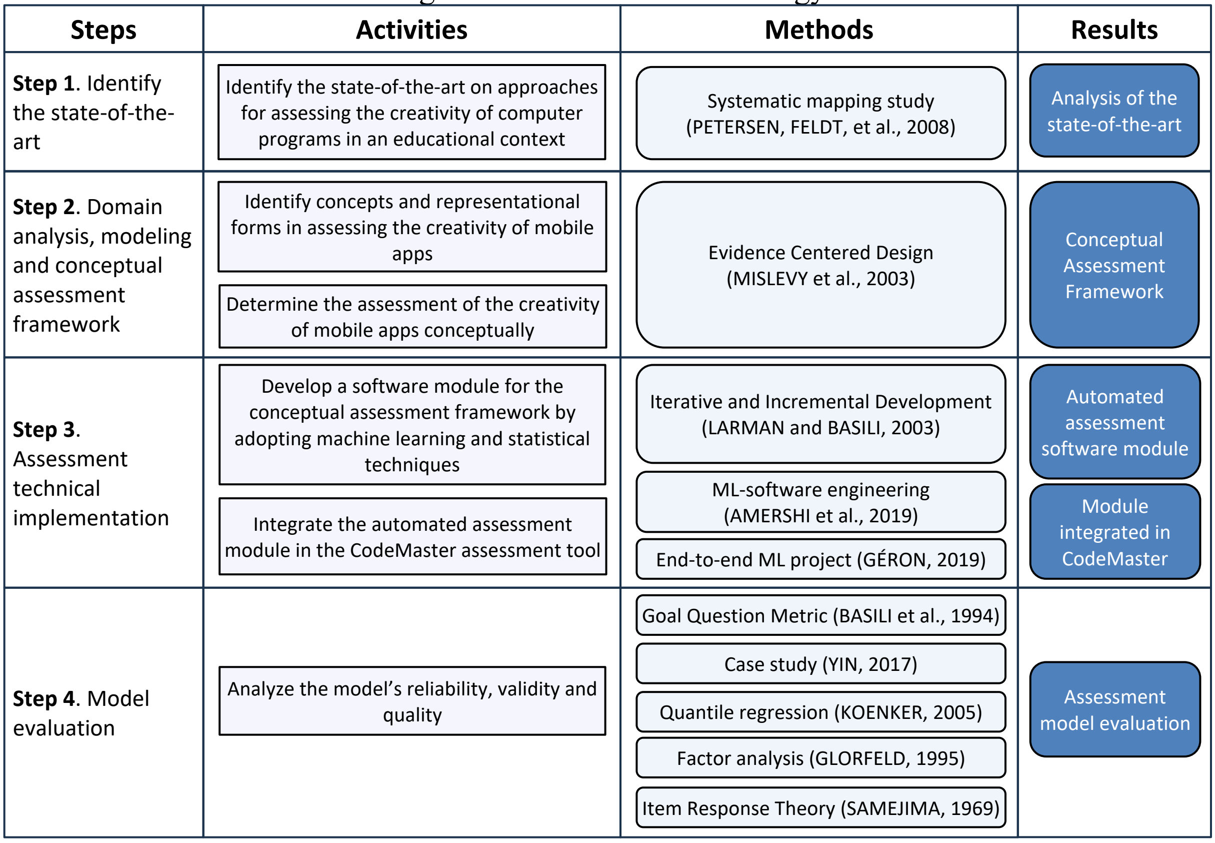
The property of interest in this research is not easy to measure, i.e., there is no scale to put a mobile app that will give a direct and precise measure of its creativity. Therefore, it is necessary to develop an indirect measure that is both observable and recordable (FINCHER and ROBINS, 2019), making generalizations between the specific and the general interactions, in an abduction approach.

Considering different research steps, data can be analyzed qualitatively or quantitatively or using quantitative techniques to analyze qualitative data and vice-versa, e.g., when performing a case study. Therefore, in this research, several research methods are combined in a complex way. Similarly, different strategies are adopted depending on the research specific objectives, e.g., archival research for analyzing the state-of-the-art (PETERSEN, VAKKALANKA, and KUZNIARZ, 2015) and a case study for exploring a research topic or phenomenon (YIN, 2017). The time horizon is cross-sectional as this research collects data on events dissociated in time (SAUNDERS, LEWIS, and THORNHILL, 2019).

1.4.2 Research steps

This research is divided into 4 steps (Figure 2) starting with the analysis of the state-of-the-art (PETERSEN, FELDT, *et al.*, 2008), then developing the conceptual assessment model following the Evidence-Centered Design (ECD) framework (MISLEVY, ALMOND, and LUKAS, 2003). For the implementation of the automated assessment model, an iterative and incremental development process is followed (LARMAN and BASILI, 2003) as well as a human-centric iterative Machine Learning process (AMERSHI, BEGEL, *et al.*, 2019; GÉRON, 2019). In order to evaluate the reliability and validity of the assessment model, statistical analyses are performed (KOENKER, 2005; GLORFELD, 1995; SAMEJIMA, 1969) in a case study (YIN, 2017) following the Goal-Question-Metric (GQM) approach (BASILI, CALDIERA, and ROMBACH, 1994).

Figure 2 – Research methodology



Steps	Activities	Methods	Results
Step 1. Identify the state-of-the-art	Identify the state-of-the-art on approaches for assessing the creativity of computer programs in an educational context	Systematic mapping study (PETERSEN, FELDT, et al., 2008)	Analysis of the state-of-the-art
Step 2. Domain analysis, modeling and conceptual assessment framework	Identify concepts and representational forms in assessing the creativity of mobile apps Determine the assessment of the creativity of mobile apps conceptually	Evidence Centered Design (MISLEVY et al., 2003)	Conceptual Assessment Framework
Step 3. Assessment technical implementation	Develop a software module for the conceptual assessment framework by adopting machine learning and statistical techniques Integrate the automated assessment module in the CodeMaster assessment tool	Iterative and Incremental Development (LARMAN and BASILI, 2003) ML-software engineering (AMERSHI et al., 2019) End-to-end ML project (GÉRON, 2019)	Automated assessment software module Module integrated in CodeMaster
Step 4. Model evaluation	Analyze the model's reliability, validity and quality	Goal Question Metric (BASILI et al., 1994) Case study (YIN, 2017) Quantile regression (KOENKER, 2005) Factor analysis (GLORFELD, 1995) Item Response Theory (SAMEJIMA, 1969)	Assessment model evaluation

Source: Elaborated by the author.

Step 1. Identify the state-of-the-art. In order to identify the state-of-the-art creativity assessment approaches, a systematic mapping study is performed (PETERSEN, FELDT, *et al.*, 2008; PETERSEN, VAKKALANKA, and KUZNIARZ, 2015). The analysis of the state-of-the-art aims at identifying existing approaches (methods, models, frameworks, scales) to systematically assess the creativity of computer programs. The systematic mapping process is divided into definition, execution, and analysis. In the definition phase, research objectives are identified and a systematic review protocol is defined. The protocol specifies the central research questions and the procedures that will be used to conduct the review, including the definition of inclusion/exclusion criteria, quality criteria, data sources, and search strings. The execution phase consists of the search and identification of relevant studies, and their selection following the inclusion/exclusion and quality criteria established in the protocol. Once identified, data related to the research question(s) are extracted from the relevant studies, analyzed, and synthesized during the analysis phase.

Step 2. Domain analysis, modeling, and conceptual assessment framework. Following ECD, the conceptual model is developed. First, the domain is analyzed based on curriculum guidelines for computing education, focusing on K-12 (CSTA, 2016; P21, 2020; MEC, 2018; SBC, 2018). In this step, information is extracted about how creativity in mobile apps is perceived and communicated, as well as concepts, terminology, tools, representation forms, and analyses of information use (MISLEVY, ALMOND, and LUKAS, 2003). In order to model the domain, the assessment is expressed in narrative form using principled assessment designs for inquiry design patterns (SEERATAN and MISLEVY, 2008). Here, the student's model is defined considering focal and additional knowledge, skills, and abilities; the evidence model describing potential artifact observations; and the task model, defining potential work artifact, characteristic and variable features (MISLEVY and HAERTEL, 2006). The conceptual model is developed using Principled Assessment Designs for Inquiry (PADI) design patterns (RICONSCENTE, MISLEVY, and HAMEL, 2005).

Step 3. Technical implementation of the assessment. Based on the conceptual model, the automation of the assessment is developed by adopting an iterative and incremental approach (LARMAN and BASILI, 2003), including requirements analysis, design, modeling, development, and testing. In accordance with the specific requirements for the automation of the assessment of each of the dimensions, machine learning and statistical approaches are adopted following a human-centric iterative machine learning process (AMERSHI, BEGEL, *et al.*, 2019) using an end-to-end project methodology (GÉRON, 2019). The model is integrated into the CodeMaster tool (CNE, 2023) and tested.

Step 4. Model evaluation. The assessment model is evaluated regarding its capacity to differentiate creative mobile apps from non-creative mobile apps according to human raters by performing a case study (YIN, 2017). First, the research question and objectives are defined as well as the focus of the study adopting GQM (BASILI, CALDIERA, and ROMBACH, 1994). The overall design and structure of the case study are determined, including the time frame and data collection methods. The data is collected using observations and artifacts, specifically, data is collected from a real-world context by downloading mobile apps and outcome information (winner/non-winner) from the App of the Month contest from 2016-2022 with support from MIT App Inventor Foundation (2023). The collected data is analyzed to identify patterns and key findings. Quantile regression (KOENKER, 2005) is performed to explore the relationship between the groups (winners/non-winners). Reliability is analyzed in terms of internal consistency using the Omega coefficient (HAYES and COUTTS, 2020). The validity is analyzed via factor analysis (GLORFELD, 1995) and the quality is measured via Item Response Theory (SAMEJIMA, 1969).

1.5 ORIGINALITY OF THIS RESEARCH

Based on the results of the state-of-the-art analysis, existing studies for assessing creativity in computing education are scarce (ALVES, GRESSE VON WANGENHEIM, and MARTINS-PACHECO, 2021a). Most of the studies only focus on originality as an essential dimension of creativity, not including other dimensions. Furthermore, most do not present an evaluation of the assessment instrument proposed.

Considering the scoring strategy, many studies rely on an entirely subjective scoring method, which is useful, however, this method can have several limitations related to labor cost and subjectivity (BEATY and JOHNSON, 2020). Especially as it often involves having multiple human raters evaluate the responses, considering that each rater may vary in their perceptions and preferences, it may result in a lack of reliability and validity (BEATY and JOHNSON, 2020; ALVES, GRESSE VON WANGENHEIM, *et al.*, 2021c). On the other hand, the existing automated approaches use objective scoring focusing on computer programs as solutions to well-defined tasks. This type of task contains more constraints than open-ended free-choice projects, and the approaches are specialized for each task (KOH, BENNETT, and REPENNING, 2011; BENNETT, KOH, and REPENNING, 2013), therefore they cannot be applied to measure the creativity of mobile apps created as a result of open-ended free-choice projects.

Based on the results of the state-of-the-art, there is a lack of assessment models systematically developed to assess creativity in computing education through the development of mobile apps. Thus, this research provides an original contribution in terms of the systematic design and evaluation of a new assessment model for mobile apps based on the current literature definition of creativity, providing comprehensive support for assessing creativity in computing education. The assessment model is composed of a theoretical model and a software implementation integrated into the existing CodeMaster tool (GRESSE VON WANGENHEIM, HAUCK, *et al.*, 2018) as well as a systematic evaluation of its reliability and validity. To the best of our knowledge, such a model has not been found in the literature to systematically assess the creativity of mobile apps. In terms of scientific impact, innovative knowledge is created in the process of developing a model to assess creativity using mobile apps as input.

1.6 CONTRIBUTIONS

This research, at the doctoral thesis level, has scientific, technological, and social contributions. Some contributions are the result of collaborative work with other colleagues of the Software Quality Group/UFSC.

Scientific contributions. Aiming to automate the assessment of creativity in mobile apps, this research makes a significant scientific contribution to computer science by addressing the fundamental question underlying all of computing: "What can be (efficiently) automated?" (COMER, GRIES, *et al.*, 1989, p. 12). By developing algorithms and a software module that can evaluate and quantify creativity in mobile apps, this research expands the frontiers of automation in a domain traditionally associated with human judgment and subjective evaluation. Automating creativity assessment enables the efficient analysis of creative outputs, thereby saving time and resources. This research not only advances the field of computer science but also opens up new avenues for exploring the intricate relationship between human creativity and computational systems. Ultimately, it paves the way for innovative applications and technologies that can harness and enhance human creativity in novel ways, amplifying our understanding of what is possible in the digital realm.

Technological contributions. The main technical contribution of this research is the implementation of the automated assessment model as a software module and pip package (ALVES, 2023). By packaging the automated assessment model as a pip module, it becomes easily installable and usable for developers and mobile app creators, enhancing their ability to assess the creative aspects of their mobile apps. This research not only pioneers the development of an innovative algorithmic approach for assessing the creativity of mobile apps but also provides a practical and accessible solution by integrating it into the CodeMaster tool (GRESSE VON WANGENHEIM, HAUCK, *et al.*, 2018). The integration of the assessment package into the CodeMaster tool further streamlines the workflow, enabling seamless assessment and feedback loops during the development process of mobile apps. It also enables users to use the results of this research enabling them to assess the creativity of their mobile apps created with App Inventor anytime using the online tool.

Social contributions. The social contribution of this research is the support for the development of creativity as part of computing education. In today's rapidly evolving digital landscape, fostering creativity is paramount, and this research addresses this need directly. By providing an automated creativity assessment tool, this research democratizes access to the assessment of creativity, empowering students and teachers of all backgrounds to cultivate and refine their creative abilities, a crucial skill in the 21st century. By providing feedback to the learner, it is expected to support their learning progress as well as feedback to the teacher allowing the improvement of teaching. Furthermore, automating the model is expected to reduce the assessment effort and provide evidence for assessing the creativity of mobile apps. Additionally, by providing a user interface publicly available online in Brazilian Portuguese, this research fosters inclusivity and accessibility, catering to a user base for the computing education community in Brazil.

1.7 ADHERENCE TO THE GRADUATE PROGRAM IN COMPUTER SCIENCE

The present work adheres to the graduate program in Computer Science by aligning with the Software Engineering (SE) research line of the Computer Science Graduate Program (PPGCC) as part of the subtopic of Computing in School[2]. Specifically, this research focuses on assessing creativity based on the software artifacts created as a result of the learning process, emphasizing the SE area as a social discipline. According to Sjøberg et al. (2008), the study of SE can be approached from a social perspective, where actors apply technologies to perform activities on existing or planned software systems. In this thesis, the archetype classes of actors, technology, activity, and software systems are explored to understand their relative effects and measure them in terms of different characteristics. Within the educational context, the primary focus is on assessing the creativity of computer programs (i.e., mobile apps) developed for

[2] https://ppgcc.posgrad.ufsc.br/linhas-de-pesquisa-2/

educational purposes. By examining creativity within the SE framework, this research delves into the core aspects of software design, development, and evaluation, while also incorporating the social dimensions inherent in the educational context.

1.8 STRUCTURE OF DOCUMENT

The structure of the document is organized into several sections, each addressing specific aspects of the research. The *THEORETICAL BACKGROUND* (chapter 2) synthesizes the concepts regarding creativity and teaching the development of mobile apps. It also explores creativity in computing education, including the measurement of creativity. The *STATE-OF-ART* (chapter 3) reviews relevant literature and addresses various aspects such as the definition and analysis of creativity, context, sample size, and evaluation of the existing approaches. Next, the document presents the *CONCEPTUAL ASSESSMENT MODEL* (chapter 4), in which the *DOMAIN MODELING* subsection covers student competencies, the evidence model, and the tasks model. The *CONCEPTUAL ASSESSMENT FRAMEWORK* section is then introduced, encompassing the assessment model for creativity. The *TECHNICAL IMPLEMENTATION OF THE ASSESSMENT MODEL* section explains the software module and its integration into CodeMaster. Following this, the *EVALUATION OF THE MODEL* (chapter 5) presents the results of statistical analysis on the model's capability to differentiate between creative and non-creative mobile apps according to human raters. The *LIMITATIONS* (chapter 6) identify the limitations of the study. The *CONTRIBUTIONS* (chapter 7) section highlights the specific contributions made by the research. The document concludes with a final section (chapter 8), summarizing the key findings and suggesting possible avenues for future research. At the end, a reference list is provided.

2 THEORETICAL BACKGROUND

This section presents the construct and the measurement of creativity. It also presents the structure of computer programs and their idiosyncrasies considering mobile apps created with App Inventor focusing on the computing education context. This elucidation not only serves to provide a robust foundation for understanding the multifaceted nature of creativity in the context of mobile app development but also underscores the critical role that these mobile apps play in shaping contemporary pedagogical approaches within the domain of computing education.

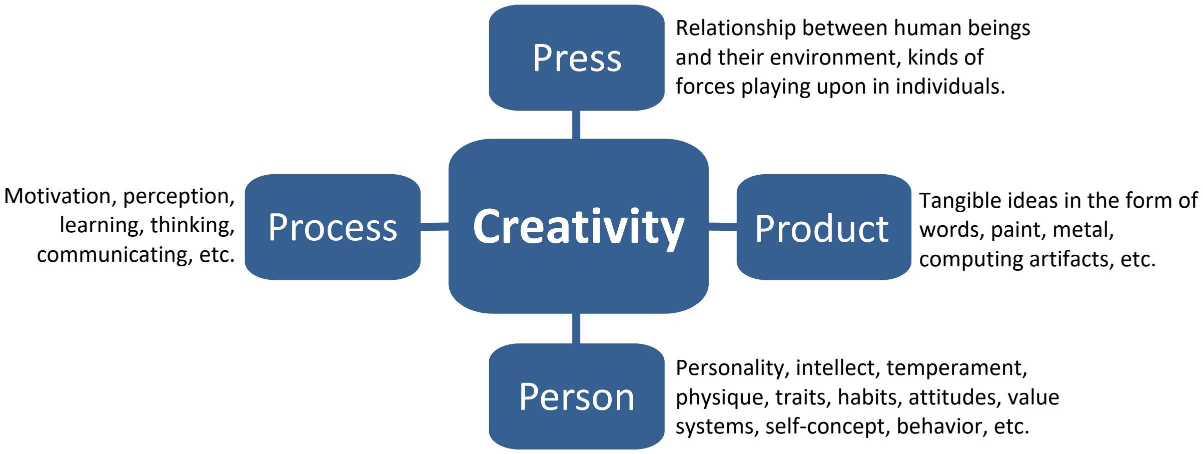
2.1 CREATIVITY

Creativity is a multidimensional construct that can be represented from different perspectives (WALIA, 2019). There are many definitions of creativity in the literature as the field historically suffered from the "definition problem" (PLUCKER, BEGHETTO, and DOW, 2004). Nevertheless, it is imperative to acknowledge that amidst the inherent complexity of the multifaceted construct of creativity, certain attributes have attained a consensus among experts in the field. These attributes encompass characteristics pertaining to the creative individual, the creative processes undertaken, the resultant creative products, and the environmental factors that influence and facilitate creativity. The recognition and categorization of these fundamental facets serve as a critical foundation in the scientific study of creativity, offering a common framework upon which to build an understanding of this construct and its definition.

> Creativity is the interaction among *aptitude*, *process*, *and environment* by which an individual or group produces a *perceptible product* that is both *novel* and *useful* as defined within a *social context*. (PLUCKER, BEGHETTO, and DOW, 2004, p. 90).

Aiming at a more comprehensive and in-depth exploration of the construct, researchers have proposed conceptual schemas to structure the definition of creativity. Among those is the widely recognized Four P's framework for creativity, as introduced by Rhodes (1961). This framework offers a structured approach to comprehending creativity by categorizing its essential components into four strands: person, process, product, and press (environment). The utilization of such conceptual schemas aims to establish a systematic foundation for the study of creativity, facilitating a deeper and more nuanced exploration of the construct (Figure 3).

Figure 3 – Four Ps: Creativity strands


Press
Relationship between human beings and their environment, kinds of forces playing upon in individuals.
Process
Motivation, perception, learning, thinking, communicating, etc.
Creativity
Product
Tangible ideas in the form of words, paint, metal, computing artifacts, etc.
Person
Personality, intellect, temperament, physique, traits, habits, attitudes, value systems, self-concept, behavior, etc.

Source: Elaborated by the author based on Rhodes (1961).

In Rhode's Four Ps, the person refers to the individual who is performing the creative act. This includes the personality and various traits and attitudes of the creative individual as well as the individual's creative potential. Process refers to the mental processes during creative endeavors involving the learning, thinking, and communication of ideas as well as the tools and strategies employed. Product refers to the tangible outcomes, i.e., artifacts, of the creative process, such as works of art, writings, computer programs, etc. The press refers to the environment or setting in which creativity takes place, as well as whether the environment favors the relationship with the individuals regarding creativity.

More recently, another conceptual schema providing an all-encompassing view of creativity has been proposed by Glăveanu (2013). The Five A's framework provides five strands considering the social and cultural context: actor, action, artifact, audience, and affordances. The term actor is used to recognize people as social beings who are shaped by their sociocultural context and who act within it, in coordination with others, to change and shape this context in appropriate ways. The action focuses on cognitive processes to capture an essential part of its manifestation acknowledging the internal psychological dimension and the external behavioral dimension. Artifacts highlight the cultured nature of products and the cumulative character of creation in human groups and societies. Audience and affordances refer to the interdependence between creators and the physical and social world.

In the context of this research, the primary focus lies on the assessment of creativity based on the artifacts created by students as learning outcomes, particularly mobile apps. By emphasizing the assessment of these mobile apps, this research aims to provide a comprehensive understanding of the students' creativity in the field of computing education.

Mobile apps serve as tangible representations of the students' learning and problem-solving abilities within the realm of mobile app development.

The nature of creativity can be studied in different levels of magnitude degrees, e.g., the eminent creativity focusing on the creative genius and creative works that may last forever, and everyday creativity focusing on creative activities that the average person may engage in (KAUFMAN and BEGHETTO, 2009). Considering these distinctions between the various levels of creative magnitude, the Four Cs of creativity has been proposed by Kaufman and Beghetto (2009) introducing four ways that creativity can be conceptualized: mini-c, little-c, Pro-C, and Big-C. The concept of mini-c refers to the creative insights and interpretations that emerge during the process of learning. It encompasses the generation of unique and personally significant understandings of experiences, actions, and events. The little-c is more focused on everyday activities in which the nonexpert may participate each day and be recognized as creative by peers. The Pro-C category is for individuals who are professional creators but have not reached eminent status but made an impact in a domain, typically to reach the Pro-C, people will undergo a formal apprenticeship for approximately 10 years, e.g., through academic institutions, or tinkering in a domain and improving through experimentation without a structured mentorship. As for Big-C, the level of achievement is so legendary that the person becomes a representation of their field, and many people will know about their accomplishments (BEGHETTO and KAUFMAN, 2007).

Creativity can also be studied both in a domain-general and domain-specific approach. The degree of specificity or generality of creativity is known to vary with the social context and evolves as individuals progress from childhood to adulthood (PLUCKER and BEGHETTO, 2004). Even though the most common measures of creativity are primarily domain-general, such as the Torrance Tests (TORRANCE, 2008), the study of creativity is inherently intertwined with the study of various domains and disciplines (HOLINGER, GLăVEANU, *et al.*, 2017). Such a domain-specific approach can encompass general thematic areas, domains, and microdomains (BAER and KAUFMAN, 2005).

In a K-12 context, educators can expect to encounter different levels of creativity based on the mini-c magnitude of creative expression. In the context of this research, considering the computing education context, creativity is nurtured in a domain-specific perspective. By understanding and implementing these frameworks, educators can create an environment that supports the development of creative thinking, encourages self-expression, and empowers students to make unique contributions to computing education.

2.1.1 Creativity in computing education

Computing is recognized as a creative human activity that allows the exploration and creation of knowledge, enables innovation, and allows individuals to deploy technology toward creating novel artifacts (MISHRA and YADAV, 2013). Teaching computing in school typically focuses on computational thinking, aiming at expressing solutions as computational steps or algorithms that can be carried out by a computer (CSTA, 2016). It involves solving problems, designing systems, and understanding human behavior by drawing on the concepts fundamental to computer science (WING, 2006).

The creative use of digital technologies to solve diverse problems engages students in an active design and creation process using computational concepts and methods to create computing artifacts (ROMERO, LEPAGE, and LILLE, 2017). By moving from computational thinking to computational making (RODE, WEIBERT, *et al.*, 2015) students learn to create, test, and refine computer artifacts (CSTA, 2016; LYE and KOH, 2014; SHUTE, SUN, and ASBELL-CLARKE, 2017), enabling them to creatively express themselves, concretize their ideas, and develop diverse and innovative ways to build and learn (CLEMENTS, 1995; GROVER and PEA, 2013). In this respect, creativity is one of the keys to respond common challenges in the development of computer programs today (ROBERTSON, 2005), as computing is not only about writing computer programs but also about competencies to analyze context and requirements (ROBERTSON, 2005), to ideate novel, useful, and technically feasible solutions (ROMERO, LEPAGE, and LILLE, 2017), to design a computer program by modeling data and architecture (GU and TONG, 2004), to design a usable and visually aesthetic user interface (RODE, WEIBERT, *et al.*, 2015; FERREIRA, GRESSE VON WANGENHEIM, *et al.*, 2019) as well as to implement and test code (GLASS, 1995).

In this context, active methodologies placing the student at the center of the knowledge acquisition process play a pivotal role in teaching students to create computer programs. Such methodologies engage in hands-on and experiential learning experiences. One of the pedagogic strategies that implement active learning approaches is the use-modify-create cycle (LEE, MARTIN, *et al.*, 2011; LYTLE, CATETÉ, *et al.*, 2019). At the use stage, students use and analyze a given computational artifact, then at the modify stage, students can remix and/or modify an existing computational artifact, until, at the creation stage, students create a new computational artifact. In order to provide students with the opportunity to do computing in ways that have a direct impact on their lives and their communities, often a perspective of computational action (TISSENBAUM, SHELDON, and ABELSON, 2019) is adopted focusing on real-world problems, usually in an interdisciplinary way (DOUSAY, 2018). In this context, programming activities are posed as open-ended ill-defined problems in a constructivist context with authentic tasks (KIESLER, 2022). These activities aim to stimulate the development of higher-order thinking skills not prescribing a correct or best solution in advance and give students more freedom to choose abstract concepts for creating a solution. As a result, students create computer programs to solve real-world problems providing opportunities for students "to extend their creative expression to solve problems, create computational artifacts" (YADAV and COOPER, 2017, p. 31).

2.1.2 Measurement of creativity

In the field of creativity research, a common distinction is made between convergent thinking and divergent thinking (DT). Convergent thinking focuses on finding a singular or optimal solution to a problem. On the other hand, DT involves generating multiple ideas in response to open-ended prompts, without a single or objectively correct solution. Even though DT is not equivalent to creativity, it is often examined as a measure of the potential for creative output (KAUFMAN, PLUCKER, and BAER, 2008). In the classic definition, DT was characterized by three key components: fluency, flexibility, and originality (GUILFORD, 1950).

Fluency refers to the ability to come up with a large number of ideas, possibilities, consequences, and objects (RENZULLI, 2018). It is easily quantifiable by tallying the number of uses generated by participants in tasks such as the alternative-uses task, where individuals are prompted to envision different applications for an object, such as a bowl or tire (DUMAS, ORGANISCIAK, and DOHERTY, 2021). As a result, fluency stands out as the most frequently assessed aspect of DT (DUMAS, 2018).

Flexibility refers to the ability to adapt numerous approaches or strategies to solve a problem (RENZULLI, 2018). It enables individuals to break free from patterns and routines when tackling problems, which not only fosters creative problem-solving but also relates to adaptability and the ability to shift perspectives while addressing challenges. Flexibility is typically measured in tasks, such as the alternative-uses task, by categorizing each response in advance based on a conceptual schema. Then it is scored by counting the number of categories employed by the individual (BEKETAYEV and RUNCO, 2016).

Originality refers to an unusual or infrequent artifact seen in a universe of artifacts made by people with similar experience and training (JACKSON and MESSICK, 1964). There exist several definitions for originality in the context of creativity research and, typically, when measuring originality, authors use diverse concepts, such as unusualness, infrequency, rarity, divergence, etc. Unusualness includes "a comparison of the product in question with other products of the same class and a counting of those comparisons that yield similar or identical products" (JACKSON and MESSICK, 1964, p. 5). Infrequency refers to "how rare or uncommon this object or objects like it are in this set of things" (WARD and WARREN, 1971, p. 212), similar to the rarity that is determined "by counting the number of times an idea occurs in a set of ideas" (DEAN, HENDER, *et al.*, 2006, p. 658). Divergence, especially in the computing education context, "could be assessed through the evaluation of the different approaches each student employed within the specified design parameters" (BENNETT, KOH, and REPENNING, 2013, p. 361). Since the rarity of a concept can more easily be quantified, other research explicitly reports the rarity use as a substitute for originality (JANSSON and SMITH, 1991). Yet, to apply consistently such a relative frequency criterion, some standard must be established to decide how much is "few". Independent of what concept is used for measuring originality, a reference universe must be taken into account. Here, the reference universe refers to a set composed of artifacts created in a similar context (JACKSON and MESSICK, 1964), for example, mobile apps created by students in a classroom or a pool of mobile apps publicly available, e.g., App Inventor Gallery. It is important to note though that different results may be obtained if different reference universes are used for measuring originality.

One of the most used tests for measuring DT is the Torrance Tests of Creative Thinking (TTCT) (TORRANCE, 1981). TTCT incorporates various tasks, both verbal and figural, designed to evaluate distinct facets of DT, including fluency (the ability to generate a large number of ideas), flexibility (shifting between different categories or perspectives), and originality (producing unique and unconventional ideas). While DT tasks typically encompass verbal and figural domains, they are generally considered domain-general, yet criticized for not fully capturing the intricacies of real-world creativity (BAER, 2015).

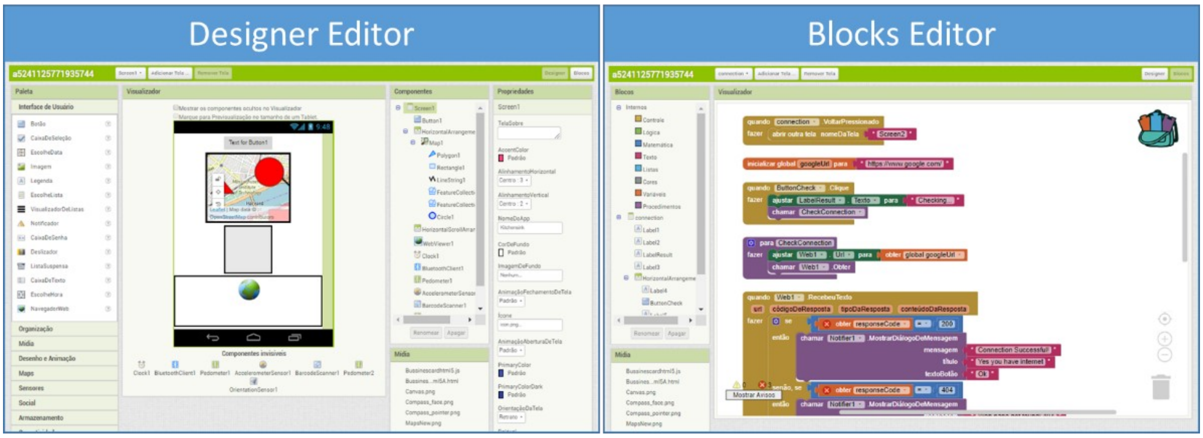
An alternative method for measuring creativity is through a more domain-specific approach, wherein individuals are tasked with creating tangible artifacts that are subsequently assessed for their level of creativity by human raters. This method, known as the Consensual Assessment Technique (CAT) (AMABILE, 1996), involves convening a panel of experts who evaluate the creativity of the produced artifacts (BAER, KAUFMAN, and GENTILE, 2004). However, this approach can be demanding in terms of time and resources, difficult to adopt in an educational context, and some argue that the level of agreement among raters may vary (BARBOT, 2018). In the context of open-ended assignments in problem-based learning, these judgments become even more complex and difficult, leaving educators to subjective assessments (DOUSAY, 2018). This may lead to inaccurate results, especially by interdisciplinary educators who may lack competence for accurate assessment of other areas, such as computing.

In everyday life, assessing creativity happens naturally, but in the classroom, it must move beyond such subjective measurements (MISHRA and HENRIKSEN, 2013). Especially when considering assessment in active learning environments using problem-based strategies following a constructionist theory. It becomes clear that measuring the creativity of the outcomes of these practical learning experiences plays a crucial role and has the potential to be highly authentic (BIALIK, MARTIN, *et al.*, 2016).

2.2 TEACHING THE DEVELOPMENT OF MOBILE APPS

A popular approach to teaching computing through the development of computational artifacts is by teaching the development of mobile apps (PATTON, TISSENBAUM, and HARUNANI, 2019). By adopting the computational action strategy (TISSENBAUM, SHELDON, and ABELSON, 2019), students learn basic computing concepts while developing their own mobile apps. Among the programming environments used to teach the development of mobile apps is App Inventor, especially in K-12. App Inventor (MIT, 2012) is a visual block-based programming environment that allows the creation of a mobile app by drag-and-drop programming graphic blocks. It is an open-source project that was created at Google and is currently maintained by the Massachusetts Institute of Technology (MIT). The current version is App Inventor 2.0, as the App Inventor Classic was retired from production in 2015.

Figure 4 – App Inventor programming environment



Source: Elaborated by the author based on https://appinventor.mit.edu/.

Using App Inventor, a mobile app can be created in two stages (Figure 4). First, the user interface components are defined in the Designer Editor. The Designer Editor also allows specifying non-visual components, such as sensors, social, and media components that access resources from the phone or other applications (Table 1).

Table 1 – Designer Editor components

Category	Description	Components examples
User Interface	Visual and interactive components for creating the user interface.	Button, Checkbox, Image, Label, Notifier, etc.
Layout	Components that assist in organizing the visible components on the screen.	HorizontalArrangement, TableArrangement, etc.
Media	Components that allow to work with various types of multimedia elements such as images, sounds, and videos.	Camcorder, Camera, Player, ImagePicker, Sound, etc.
Drawing and Animation	Components that allow the user to draw and view animations.	Ball, Canvas, ImageSprite
Maps	Components that allow the integration of mapping and location-based functionality into the mobile app.	Circle, Map, Marker, Polygon, etc.
Sensors	Components that allow the mobile app to interact with various sensors built into the device.	AccelerometerSensor, GyroscopeSensor, etc.
Charts	Visual components that allow the display of data graphically in a more visually appealing and comprehensible manner.	Chart, ChartData2D
Social	Components that allow the mobile app to communicate with other social media platforms and services.	ContactPicker, EmailPicker, Sharing, etc.
Storage	Components that allow the mobile app to work with various storage options and manage data	File, FusiontablesControl, TinyDB, TinyWebDB
Connectivity	Components that allow various communication, networking features, and access to external data sources.	ActivityStarter, BluetoothClient, Web
Lego Mindstorms	Specialized components for controlling LEGO® MINDSTORMS® NXT robots using Bluetooth.	NxtDirectCommands, NxtLightSensor, etc.
Experimental	Components that are not part of the core set of standard components and are not as extensively tested or documented as the standard components.	CloudDB, FirebaseDB

Source: Elaborated by the author.

In the second stage, in the Blocks Editor, the mobile app's behavior is specified via programming, by connecting visual blocks (Figure 4). These blocks, which utilize a visual programming paradigm, represent a wide range of programming concepts and constructs, including conditional statements, loops, variables, and event handlers (Table 2).

Table 2 – Programming blocks in App Inventor

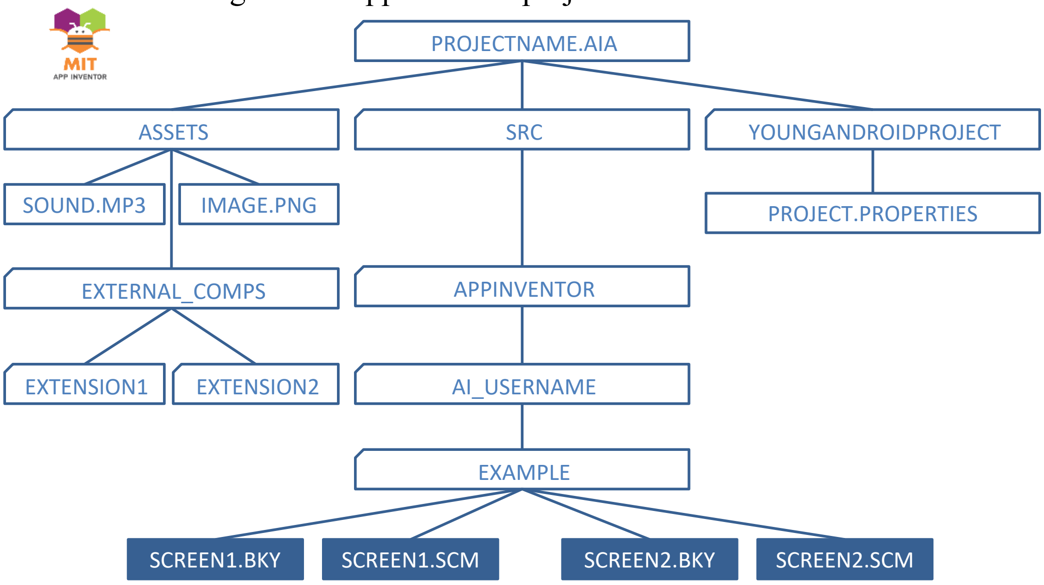
Category	**Description**
Control	Blocks for controlling the flow of the mobile app, including important blocks like loops and conditionals. Examples: `while`, `if`, etc.
Logic	Blocks for logic operations on variables including relational and Boolean operators. Examples: `and`, `or`, etc.
Math	Blocks for performing basic and advanced math operations. Examples: `add`, `cos`, etc.
Text	Blocks for creating and manipulating text strings. Examples: `join`, `length`, etc.
Lists	Blocks for creating and manipulating lists. Examples: `create_list`, `insert`.
Dictionaries	Blocks for creating and manipulating data structures that store key-value pairs. Examples: `create_empty_dictionary`, `remove_entry_for_key`, etc.
Colors	Blocks for creating and manipulating colors. Examples: `make_color`, `red`, `blue`, etc.
Variables	Blocks for creating and manipulating variables. Examples: `initialize_variable`, `get`, etc.
Procedures	Blocks for defining and calling a sequence of blocks together into a group. Examples: `procedure_do`, `procedure_result`, etc.
Components	Blocks for manipulating Designer Editor components (as defined in Table 1). They can be event blocks for specifying how a component responds to certain events, such as a button that has been pressed, blocks for changing the properties of components, call methods for performing complex tasks, or blocks of instances of a specific component. Examples: `when_button_click`, `set_label_text`, etc.
Screen	Blocks for controlling and changing the screen properties of the mobile app. Examples: `when_screen_initialize`, `set_screen_background_color`, etc.
Helper	Blocks for operating with constants, such as special strings or numbers can be related to an asset or component. Examples: `permission_access_camera, image.png`, etc.
Extensions	Blocks for additional programming functionalities that extend the capabilities of App Inventor beyond its built-in components and blocks.

Source: Elaborated by the author.

App Inventor allows to develop a wide range of mobile apps using available phone features, such as sensors, camera, recorder, etc. (Table 1). It is also possible to develop games using drawing and animation components, and mobile apps with more advanced components using API extensions and, in this way, provide support for the development of creativity as part of the development of mobile apps.

App inventor projects can be exported as .aia files (Figure 5), which is a compressed (ZIP) collection of files that includes a project properties file, media files used by the mobile app, and, for each screen of the mobile app, two main files: a .bky file and a .scm file. The .bky file encapsulates an XML structure with all the programming blocks used in the mobile app. The .scm file encapsulates a JSON structure that contains all the visual components used in the mobile app (TURBAK, MUSTAFARAJ, *et al.*, 2017).

Figure 5 – App Inventor project AIA file structure



Source: Elaborated by the author based on Mathijssen (2019).

3 STATE-OF-ART

In order to elicit state-of-the-art approaches for assessing the creativity of computing artifacts based on the analysis of computer programs developed by students in an educational context as an outcome of the learning process, a systematic mapping following the procedure defined by Petersen et al. (2015; PETERSEN, FELDT, *et al.*, 2008) was performed. The results of this review have also been published in Alves et al. (2021a).

3.1 DEFINITION OF THE REVIEW PROTOCOL

Research Question. Which studies exist for the assessment of the creativity of computer programs in the educational context?

This research question was refined into the following analysis questions:

AQ1. Which studies exist and for what kind of computer programs and educational stages?

AQ2. What is the definition of the creativity dimensions being assessed?

AQ3. How are these creativity dimensions analyzed?

AQ4. What is the context and sample size of the application of the approach?

AQ5. If, and how the approach has been evaluated?

Data source. All published English-language articles that are available on Scopus, including publications from ACM, Elsevier, IEEE, and Springer with free access through the Capes Portal[3] were examined.

Inclusion/exclusion criteria. The articles included are peer-reviewed English-language articles that present a form of measurement or assessment of the creativity (or originality) of computer programs. In this context, studies that present substantial information to enable the extraction of relevant information regarding the analysis questions are included, i.e., studies in which creativity assessment or measurement are mentioned in passing only and not explored to any thoroughness are excluded. Studies that do not focus on the product strand according to Rhode's (1961) 4P's, i.e., studies focusing exclusively on the press, person, and process creativity are excluded. In addition, studies that assess creativity based on artifacts other

[3] A portal for accessing scientific works, managed by the Brazilian Ministry of Education, available to authorized institutions, such as universities and research agencies (www.periodicos.capes.gov.br).

than computer programs, i.e., requirements, wireframes, etc. are excluded. Studies within an educational context that have been published until May 2023 are included.

Definition of the search string. Following the research objective, the search string has been defined by identifying core concepts and also considering synonyms, as indicated in Table 3. The term creativity has been chosen as it expresses the main concept to be searched. As originality is one dimension widely accepted in the field (RUNCO and ALBERT, 2010), this term was chosen as a synonym for creativity to broaden the search results. Although originality alone is not sufficient to classify an artifact as being creative, independently of what other positive qualities it may have, it is generally considered an important dimension for a creative artifact to possess (JACKSON and MESSICK, 1964). Other synonyms of originality, such as novel, infrequency, or unusualness, are not used, as these terms are used in many contexts with meanings unrelated to creativity. Considering the focus on the assessment, synonyms that are commonly used in the educational context, such as measuring, evaluation, and analysis are used. Keywords related to educational context are chosen to limit results to this specific context. Considering the focus on computing education, specifically the assessment of creativity based on computer programs, terms related to this domain are also included. In this sense, computational thinking is used as a synonym for programming/coding. And, although computational thinking covers a much wider field than just programming and coding, it is frequently used as a synonym for these terms in the literature (ARMONI, 2016). Wildcard characters are used to cover as many variations of the terms as possible, such as creativ* representing "creative" and "creativity".

Table 3 – Keywords used in the search string

Concept	**Keywords and synonyms**
Creativity	creativ*; original*
Assessment	assess*; measur*; evaluat*; analy*
Educational context	K-12; school; education; learning
Programming artifact oriented	Programming; coding; computational thinking

Source: Elaborated by the author.

Using these keywords, the search string has been calibrated and adapted in conformance with the specific syntax of the source:

```
        TITLE-ABS-KEY(( creativ*  OR  original* )  AND  ( assess*  OR  measur*  OR
evaluat*  OR  analy* )  AND  ( "K-12"  OR  school  OR  education  OR  learning )
AND  ( coding  OR  programming  OR  "computational thinking" ))
```

3.2 EXECUTION OF THE SEARCH

The search was executed in May 2023 by the author and revised by the advisor using Scopus. As Scopus allows filtering results based on the field, works on unrelated fields, such as medicine, biology, etc, were excluded. In the first analysis stage, titles, abstracts, and keywords of all filtered search results (3,176 articles) were reviewed to identify articles that matched the exclusion criteria, resulting in 149 potentially relevant articles. In the second stage, the full-text of the pre-selected articles was analyzed. A total of 21 articles that analyze the creativity of artifacts based on computer programs created by the students were selected as relevant. The selection process and the selection of papers were discussed until a consensus was reached (Table 4).

Table 4 – Quantity of articles per selection stage

Search results	**Potentially relevant articles selected based on the analysis of the title and abstract**	**Selected articles based on full-text analysis**
3,176	149	21

Source: Elaborated by the author.

Many articles have been excluded based on the analysis of their abstracts as they are related to other fields such as video coding, artificial intelligence, and deep learning. This is due to the fact that ‘original’ is a term widely used to describe datasets used in the studies of these areas. In addition, a large number of articles using the term ‘originality’ to indicate the novelty of the study have been excluded. During the full-text analysis of the remaining articles, several have been excluded as they present approaches exclusively analyzing creativity concerning other strands that are outside of the focus of our research, such as press (ENGELMAN, MAGERKO, *et al.*, 2017), process (PEREZ-POCH, OLMEDO, *et al.*) or person (ENGELMAN, MAGERKO, *et al.*, 2017). Some studies also mention measuring creativity but the focus is not on the assessment (YANG, 2022), thus they do not present substantial information to enable the extraction of information regarding the analysis questions. As a result, a total of 21 articles were identified as relevant to the research objective (Table 5). All selected articles were published within the last twelve years, which also indicates the recent importance of this topic.

Table 5 – Articles selected

#	**Reference**
1	BASU. S. Using Rubrics Integrating Design and Coding to Assess Middle School Students' Open-ended Block-based Programming Projects. Proc. of the 50th ACM Technical Symposium on Computer Science Education, p. 1211–1217, 2019. ACM, New York, NY, USA.

#	**Reference**
2	BENNETT, V.; KOH, K. H.; REPENNING, A. Computing creativity: Divergence in computational thinking. Proc. of the 44th ACM Technical Symposium on Computer Science Education, p. 359-364, 2013, New York, NY, USA. ACM.
3	GAL, L. HERSHKOVITZ, A., MORÁN, A., GUENAGA, M., GARAIZAR, P. Suggesting a Log-Based Creativity Measurement for Online Programming Learning Environment. Proc. of the Fourth ACM Conference on Learning @ Scale, p. 273–277, 2017. New York, NY, USA. ACM.
4	GROENEVELD, W.; MARTIN, D.; PONCELET, T.; AERTS, K. Are Undergraduate Creative Coders Clean Coders? A Correlation Study. Proc. of the 53rd ACM Technical Symposium on Computer Science Education, 314–320, 2022. ACM, New York, NY, USA.
5	GROVER, S.; BASU, S.; SCHANK, P. What We Can Learn About Student Learning From Open-Ended Programming Projects in Middle School Computer Science. Proc. of the 49th ACM Technical Symposium on Computer Science Education, p. 999–1004, 2018. ACM, New York, NY, USA.
6	HERSHKOVITZ, A.; SITMAN, R.; ISRAEL-FISHELSON, R.; EGUÍLUZ, A.; GARAIZAR, P.; GUENAGA, M. Creativity in the acquisition of computational thinking. Interactive Learning Environments, 27(5-6), 628-644, 2019.
7	ISRAEL-FISHELSON, R.; HERSHKOVITZ, A.; EGUÍLUZ, A.; GARAIZAR, P.; GUENAGA, M. Computational Thinking and Creativity: A Test for Interdependency. Proc. of the 4th International Conference on Computational Thinking Education, 2020.
8	ISRAEL-FISHELSON, R.; HERSHKOVITZ, A.; EGUÍLUZ, A.; GARAIZAR, P.; GUENAGA, M. A log-based analysis of the associations between creativity and computational thinking. Journal of Educational Computing Research, 59(5), 926-959, 2021.
9	ISRAEL-FISHELSON, R.; HERSHKOVITZ, A.; EGUÍLUZ, A.; GARAIZAR, P.; GUENAGA, M. The associations between computational thinking and creativity: The role of personal characteristics. Journal of Educational Computing Research, 58(8), 1415-1447, 2021.
10	KERSHAW, T. C.; CLIFFORD, R. D.; KHATIB, F.; EL-NASAN, A. An initial examination of computer programs as creative works. Psychology of Aesthetics, Creativity, and the Arts. Advance online publication, 2022.
11	KHAWAS, P.; TECHAPALOKUL, P.; TILEVICH, E. Unmixing Remixes: The How and Why of Not Starting Projects from Scratch. Proc. of the IEEE Symposium on Visual Languages and Human-Centric Computing, pp. 169-173, 2019. Memphis, TN, USA.
12	KOH, K. H.; BENNETT, V.; REPENNING, A. Computing indicators of creativity. Proc. of the 8th ACM conference on Creativity and cognition, p. 357-358, 2011, New York, NY, USA. ACM.
13	KOVALKOV, A.; PAAßEN, B.; SEGAL, A.; PINKWART N.; GAL, K. Automatic Creativity Measurement in Scratch Programs Across Modalities. IEEE Transactions on Learning Technologies, 14(6), pp. 740-753, 2021.
14	KOVALKOV, A.; SEGAL, A.; GAL, K. Inferring Creativity in Visual Programming Environments. In Proceedings of the Seventh ACM Conference on Learning @ Scale, 269–272, 2020. ACM, New York, NY, USA.
15	LUO, J.; LU, F. WANG, T. A Multi-Dimensional Assessment Model and Its Application in E-learning Courses of Computer Science. Proc. of the 21st Annual Conference on Information Technology Education, p. 187–193, 2020. ACM, New York, NY, USA.
16	MANSKE, S.; HOPPE, H. U. Automated Indicators to Assess the Creativity of Solutions to Programming Exercises. Proc. of the 14th IEEE International Conference on Advanced Learning Technologies, p. 497-501, 2014. Athens, Greece.
17	MUSTAFARAJ, E.; TURBAK, F.; SVANBERG, M. Identifying Original Projects in App Inventor. Proc. of the 30th International Florida Artificial Intelligence Research Society Conference, p. 567-572, 2017. Florida: Association for the Advancement of Artificial Intelligence.
18	ROMERO, M.; LEPAGE, A.; LILLE, B. Computational thinking development through creative programming in higher education. International Journal of Educational Technology in Higher Education, 14, 42, 2017.
19	TURBAK, F.; MUSTAFARAJ, E.; SVANBERG, M.; DAWSON, M. Work in progress: Identifying and analyzing original projects in an open-ended blocks programming environment. Proc. of the 23rd International DMS Conference on Visual Languages and Sentient Systems, 2017.

#	Reference
20	UNAHALEKHAKA, A.; BERS, M. U. Evaluating young children’s creative coding: rubric development and testing for ScratchJr projects. Education and Information Technologies, 27(5), 6577–6597, 2022.
21	ZHONG, B.; WANG, Q.; CHEN, J.; LI, Y. An Exploration of Three-Dimensional Integrated Assessment for Computational Thinking. Journal of Educational Computing Research, 53(4), p. 562–590, 2016.

Source: Elaborated by the author.

3.3 DATA ANALYSIS

3.3.1 Which studies exist and for what kind of computer programs and educational stages?

A total of 21 articles describing 17 approaches were found, as some articles present the same approach from a different perspective (KOH, BENNETT, and REPENNING, 2011; BENNETT, KOH, and REPENNING, 2013; ISRAEL-FISHELSON, HERSHKOVITZ, *et al.*, 2020; ISRAEL-FISHELSON, HERSHKOVITZ, *et al.*, 2021; ISRAEL-FISHELSON, HERSHKOVITZ, *et al.*, 2021; HERSHKOVITZ, SITMAN, *et al.*, 2019). The approaches analyze a student’s computer program created as an outcome of the learning process. Some approaches focus on creativity assessment, while others study the relationship between creativity with other constructs, such as computational thinking (HERSHKOVITZ, SITMAN, *et al.*, 2019).

The artifacts are a result of a well-defined activity (with a known solution in advance) or an ill-defined activity (without or with more than one known solution known in advance). The type of computer programs assessed include games (KOH, BENNETT, and REPENNING, 2011), solutions to programming tasks (MANSKE and HOPPE, 2014; GAL, HERSHKOVITZ, *et al.*, 2017; KERSHAW, CLIFFORD, *et al.*, 2022; LUO, LU, and WANG, 2020; KOVALKOV, SEGAL, and GAL, 2020), projects as results of creative programming activities (ROMERO, LEPAGE, and LILLE, 2017) as well as free-choice open-ended projects (GROVER, BASU, and SCHANK, 2018; BASU, 2019; GROENEVELD, MARTIN, *et al.*, 2022). Only two approaches were found for evaluating the originality of mobile apps developed in a university context and published in public galleries (MUSTAFARAJ, TURBAK, and SVANBERG, 2017; TURBAK, MUSTAFARAJ, *et al.*, 2017).

The analysis of creativity based on the students’ computer programs is provided for diverse programming environments and languages. Some are for block-based visual programming environments, which are typically used for computing education in K-12, such as Scratch (KHAWAS, TECHAPALOKUL, *et al.*, 2019), ScratchJr (UNAHALEKHAKA and BERS, 2022), App Inventor (MUSTAFARAJ, TURBAK, and SVANBERG, 2017; TURBAK, MUSTAFARAJ, *et al.*, 2017), and Kodetu (HERSHKOVITZ, SITMAN, *et al.*, 2019). Some studies also cover more than one programming environment, e.g., Basu (2019) or Grover et al. (2018).

The approaches target different educational stages. Some studies were designed for some stage in K-12 education (BENNETT, KOH, and REPENNING, 2013; BASU, 2019; GROVER, BASU, and SCHANK, 2018), while others target higher education (ROMERO, LEPAGE, and LILLE, 2017; MUSTAFARAJ, TURBAK, and SVANBERG, 2017; TURBAK, MUSTAFARAJ, *et al.*, 2017; KERSHAW, CLIFFORD, *et al.*, 2022). Some approaches use data from galleries that contain a set of well-defined programming tasks (MANSKE and HOPPE, 2014) or free-choice open-ended projects (KHAWAS, TECHAPALOKUL, *et al.*, 2019).

3.3.2 What is the definition of the creativity dimensions being assessed?

Detailing the specific dimensions of artifact creativity that are assessed, the most analyzed dimension is originality, analyzing the newness of the artifact. Although, the authors use different terms, such as novelty (BASU, 2019; GROVER, BASU, and SCHANK, 2018) or divergence (BENNETT, KOH, and REPENNING, 2013; KOH, BENNETT, and REPENNING, 2011), they refer to the same concept of originality.

Typically, originality is assessed by comparing the student's computer program with a specific set of computer programs. This set can contain all other student's computer programs (GAL, HERSHKOVITZ, *et al.*, 2017), or pre-programmed solutions and patterns for well-defined activities (KOH, BENNETT, and REPENNING, 2011). The indicator of originality, novelty, or divergence is then measured by the extent to which the student's computer program is different from this set. Originality is also assessed by using more subjective criteria, as proposed by Grover et al. (2018) and Basu (2019), having the instructor assess computer programs on an ordinal scale as "not very novel, some novelty, or very novel" (BASU, 2019).

Inspired by the Torrance Tests (TORRANCE, 2008), the flexibility dimension is also analyzed. Kovalkov et al. (2021) measure flexibility by counting the number of distinct concepts in the code, images, and sounds of the project. In another study, Kovalkov et al. (2020)

measure the flexibility of Scratch projects based on the diversity that is embedded in the textual and visual outputs of the project.

Considering the classic definition of DT for fluency, referring to the number of generated ideas, Kovalkov et al. (2021) measure fluency focusing on Scratch projects by defining the fluency score as the distance of the project being assessed to an empty Scratch project. In another study, Kovalkov et al. (2020) measure elaboration by a tally of the occurrences of elements within each of the categories of Scratch that are present in a project. Subsequently, they count the number of scripts contained within the project, along with their maximum depth.

Some studies also measure other dimensions, such as the quality of the computer program. Different from originality, flexibility, and fluency, which can be somewhat agreed on independently from the domain, terms related to quality, such as sophistication and elegance are very domain-specific without a general agreement outside specific domains. In this regard, in the context of software engineering, elegance, for example, is measured using software metrics where “experts infer a weighting and interpretation to these metrics” (MANSKE and HOPPE, 2014, p. 498). Other terms related to quality, such as completeness and standardization are more straightforward to assess. For example, completeness is measured by verifying if the computer program is completed and standardization if the computer program follows a defined pattern or formatting rule (ZHONG, WANG, *et al.*, 2015).

The usefulness and correctness are assessed by a few approaches. Manske and Hoppe (2014), for example, use the term usefulness and define that it is achieved if the student’s computer program is correct for the activity for which it was submitted. Basu (2019) uses the term correctness in a similar way and defines more subjective assessment criteria, ranging from “the programs contain several errors” to "program runs correctly without error and the output is appropriate" (BASU, 2019, p. 1213).

3.3.3 How are these creativity dimensions analyzed?

The approaches vary largely concerning the type of assessment, methods, and techniques used. With respect to who performs the assessment, some rely on instructor assessment using rubrics (GROVER, BASU, and SCHANK, 2018; BASU, 2019), expert assessment based on personal knowledge as input to automated assessment (MANSKE and HOPPE, 2014) as well as automated assessment based on techniques from computer science and mathematics (KOH, BENNETT, and REPENNING, 2011; BENNETT, KOH, and REPENNING, 2013; GAL, HERSHKOVITZ, *et al.*, 2017; MUSTAFARAJ, TURBAK, and SVANBERG, 2017; TURBAK, MUSTAFARAJ, *et al.*, 2017).

Some studies use automated assessment and additional measurements. For example, Hershkovitz et al. (2019) use the Torrance Tests of Creative Thinking (TTCT) for measuring DT. The output from the TTCT is then compared with the automated assessment based on the statistical infrequency of solutions to programming exercises created by the students. Kershaw et al. (2022) used Cropley and Kaufman's (2012) Creative Solution Diagnosis Scale (CSDS) for consensual assessment of functional creativity using nonexpert judges. CSDS is composed of five subscales corresponding to five factors, namely relevance and effectiveness, problematization, propulsion, elegance, and genesis.

One way of assessing the creative artifact is by the instructor manually assessing the outcome created by the students using rubrics. Rubrics consist of a matrix of criteria and performance levels for each criterion. Such rubrics typically contain one or more criteria related to creativity or its dimensions, e.g., novelty or originality and quality or engagement, along with the respective performance levels (GROVER, BASU, and SCHANK, 2018; BASU, 2019; ZHONG, WANG, *et al.*, 2015).

Several approaches, envisioning the automation of the assessment, adopt metrics with machine learning models to assess creativity (MANSKE and HOPPE, 2014), to identify original projects (TURBAK, MUSTAFARAJ, *et al.*, 2017; MUSTAFARAJ, TURBAK, and SVANBERG, 2017) and to measure flexibility (KOVALKOV, SEGAL, and GAL, 2020). The models range from regression methods, such as linear regression and support vector regression (MANSKE and HOPPE, 2014), clustering methods, such as the Markov cluster algorithm (TURBAK, MUSTAFARAJ, *et al.*, 2017), the K-Nearest Neighbors algorithm (MUSTAFARAJ, TURBAK, and SVANBERG, 2017), and the K-Means clustering algorithm (KOVALKOV, SEGAL, and GAL, 2020). The input for these algorithms includes features gathered using statistical concepts, such as term frequency-inverse document frequency (TF-IDF) (TURBAK, MUSTAFARAJ, *et al.*, 2017), the Jaccard index of similarity (MUSTAFARAJ, TURBAK, and SVANBERG, 2017), and categories of images classified using a ResNet50 convolutional neural network (KOVALKOV, SEGAL, and GAL, 2020). Some features were defined using software engineering metrics, such as effective lines of code, visited lines of code, and cyclomatic complexity (MANSKE and HOPPE, 2014) as well as using the categories of the programing blocks (KOVALKOV, SEGAL, and GAL, 2020).

Abstract language tokens (obtained during the lexical analysis phase) were also used for comparing distances between artifacts using string metrics (MANSKE and HOPPE, 2014). In general, supervised learning methods were used to train machine learning models to assess creativity (MANSKE and HOPPE, 2014), while unsupervised learning methods were adopted for the identification of the originality and flexibility dimension in projects.

Some approaches also use automated analysis based on a framework especially defined for the programming tasks, such as the Computational Thinking Pattern Analysis (BENNETT, KOH, and REPENNING, 2013; KOH, BENNETT, and REPENNING, 2011) to analyze creativity divergence in the students' programming solutions compared to patterns previously defined. Another example is the automated identification of originality in App Inventor projects (TURBAK, MUSTAFARAJ, *et al.*, 2017; MUSTAFARAJ, TURBAK, and SVANBERG, 2017).

Regarding instructional feedback and grading, the approaches typically calculate a score for the student's computer program. Depending on the dimension being assessed, some approaches use rating scales with performance levels specifying more complex artifact characteristics as the score increases. Others, such as Koh et al. (2011) and Bennett et al. (2013), calculate the scores using a math formula for assessing a score on divergence. On the one hand, approaches adopting machine learning classification models provide a result that indicates if the computer program was classified as original or unoriginal, without assigning a score. On the other hand, machine learning regression models provide a score based on the expert rating scale in the datasets.

3.3.4 What is the context and sample size of the application of the approach?

The majority of the approaches were applied in K-12, such as a middle school in a large urban school district in the Western US (GROVER, BASU, and SCHANK, 2018), a primary school in Spain (HERSHKOVITZ, SITMAN, *et al.*, 2019), and primary school in Changshu City of China (ZHONG, WANG, *et al.*, 2015). Blended applications with face-to-face and online classes were applied in the middle school context (BENNETT, KOH, and REPENNING, 2013; KOH, BENNETT, and REPENNING, 2011). Face-to-face university classes included a course at Wellesley College in the United States (MUSTAFARAJ, TURBAK, and SVANBERG, 2017) and the Université Laval in Canada (ROMERO, LEPAGE, and LILLE, 2017).

Some of the approaches have also been evaluated by using projects shared by students in public repositories. In these studies, the solutions created by students can be downloaded and analyzed. Here, specifically, the App Inventor Gallery was used (TURBAK, MUSTAFARAJ, *et al.*, 2017) as well as the Project Euler (MANSKE and HOPPE, 2014) to obtain thousands of students' projects.

The sample size varies from small samples in the university context (MUSTAFARAJ, TURBAK, and SVANBERG, 2017), face-to-face classes (HERSHKOVITZ, SITMAN, *et al.*, 2019) to large samples obtained from online public project galleries, with more than 200 thousand projects (MANSKE and HOPPE, 2014).

3.3.5 If, and how the approach has been evaluated?

Most articles do not present an evaluation of the approach, as this may have been outside the scope of the articles. Hershkovitz et al. (2019) assume the reliability and validity of the TTCT as the test has been widely evaluated beforehand. Kershaw et al. (2022) used the CSDS which shows a high degree of reliability (Cronbach's alpha = 0.96). They also presented the interrater and scale reliability scores for self-rating, peer rating, and expert rating, as well as, factor analysis considering the scores by human raters.

Basu (2019) presented an evaluation of the defined rubric through inter-rater reliability, Manske and Hoppe (2014) and Mustafaraj et al. (2017) presented performance metrics regarding machine-learning models, and Kovalkov et al. (2020) presented reliability analysis. Basu (2019) used Cohen's kappa coefficient to analyze the inter-rater reliability of the proposed rubric. A high inter-rater reliability value of 0.92 was found. After computing the coefficient, teachers also scored additional projects independently providing additional opportunities to refine the rubric based on their feedback.

Manske and Hoppe (2014) performed a reliability evaluation on the agreement of expert assessments, which were used as input to the proposed machine learning model. Yet, the evaluation of the machine learning model did not provide meaningful results due to the lack of agreement between raters. As raters used individual creativity definitions in a not consistent way, this resulted in two different groups of agreement measured via Krippendorff's alpha coefficient, with low values (below 0.3) indicating no agreement between raters. However, they found a high agreement within the specific theorists-group from the educational context (Krippendorff's coefficient 0.729) and medium agreement within software engineering related experts from industry (Krippendorff's coefficient 0.552), indicating that the two groups can be separated in terms of assessing artifact creativity.

Kovalkov et al. (2020) used the Kendall Rank Correlation Coefficient to measure the intra-rater reliability between two human raters as well as the reliability between the ranking by the proposed model and the ranking by the two human raters. They found intra-rater reliability between the two human raters of $\tau = 0.59$ and a ranking agreement of $\tau = 0.435$ between the model and one human rater, indicating some degree of positive ranking agreement. In another study, Kovalkov et al. (2021) also computed Kendall's t among five human raters and the proposed model. They found substantial differences in the human rankings suggesting that that they differ in their interpretation of creativity. As for the comparison of the ranking by the model and the ranking by the human raters, they found a similar result of τ above 0.41.

Mustafaraj et al. (2017) analyzed the accuracy of the classification of original and unoriginal projects regarding the Jaccard distance. They found a good accuracy of 89% for both classes using a 0.4 Jaccard distance. A value less or greater than this results in diminishing the accuracy of one class, thus labeling it wrong, e.g., more than 11% of original projects may be labeled unoriginal if the value for distance is not 0.4.

Grover et al. (2018) reported joint discussions to establish interrater reliability without providing further information. Koh et al. (2011) stated that the divergence calculation used in their approach is supported by other data sources and that the validity of the approach demonstrates uniqueness in separate learning conditions.

Hershkovitz et al. (2019) compared the results from the TTCT test for creative thinking with the assessment of the originality of the students' computer programs to well-defined problems. They found significant correlations between the two types of creativity measures and that in some cases "creativity in programming is positively associated with the broad construct of creativity" (HERSHKOVITZ, SITMAN, *et al.*, 2019, p. 638).

3.4 DISCUSSION

Considering the importance of creativity as a 21st century skill, only very few assessment approaches have been encountered in the context of computing education with active learning strategies for assessing the student's creative artifact. Although there exist already a considerable number of approaches for assessment in computing education in general, these mostly focus exclusively on computational thinking concepts and practices (MORENO-

LEÓN and ROBLES, 2015; GRESSE VON WANGENHEIM, HAUCK, *et al.*, 2018). Only very few of them also include assessment criteria related to creativity on the strand of the artifact (KOH, BENNETT, and REPENNING, 2011; BASU, 2019), some of them using subjective criteria to be judged manually by the instructor or peers. Automated assessment tools for assessing the outcomes created with block-based programming languages such as Dr. Scratch (MORENO-LEÓN and ROBLES, 2015) or CodeMaster (GRESSE VON WANGENHEIM, HAUCK, *et al.*, 2018) also do not assess creativity.

Analyzing the approaches for assessing the artifact creativity of computer programs created by students as part of computing education, it becomes clear that the definition of creativity strongly influences how the approaches assess the artifact. Considering that the construct can be analyzed through many dimensions, it is not sufficient to only define which strand is being analyzed. This issue is further complicated through context-dependency as dimensions analyzed can vary, as well as have different meanings for the same terms in different types of computer programs. For example, on the one hand, the usefulness of a mobile app can be understood as if the mobile app allows the user to perform the desired tasks, while on the other hand, the usefulness of a digital game can be seen as if the game is fun to play. In this context, it seems that originality is one of the few well-established dimensions in the literature regarding artifact creativity.

Some of the approaches employ the term "creativity" without presenting a comprehensive definition derived from the existing literature regarding the dimensions of creativity assessment. Some of the approaches focus on one dimension as a way of assessing creativity, excluding other dimensions typically considered for the general assessment of the creativity of artifacts. As the search string also included “originality”, some approaches that focus exclusively on originality do not assess creativity itself and aim at assessing originality as a singular construct, which may not provide an in-depth analysis of creativity. Approaches that assess quality include those that define the term using software engineering metrics and differ in terms, such as completeness, elegance, sophistication, and engagement. The usefulness is defined only by two approaches using the terms correctness and appropriateness.

As the definition of creativity also depends on the specific context, some approaches tailor well-known dimensions of the creativity assessment of artifacts to the context of computing education. Thus, the definition of each dimension assessed is related to computing concepts. Originality, for example, is typically customized by comparing the students’ computer programs to identify the frequency of different solutions.

Considering that only a few approaches explicitly define creativity, this shows that it is imperative to move towards a more precise definition of creativity in the context of computing education. This would provide a shared understanding of the construct as a basis for the design and development of reliable and valid assessments.

Half of the approaches propose manual assessments, yet, these may be biased and time-intensive to complete. This becomes especially problematic in the context of large classes or Massive Open Online Courses (MOOCs). Even if manual assessments provide a context-tailored result, it may be impossible to provide constant timely feedback throughout the learning process. And, although such manual assessments can also rely on peers with results that align with instructor assessments reducing the instructor's assessment effort, they are still subjective and require substantial time and organization (MILLER, BAILEY, and KIRLIK, 2014). Observing also a lack of computing teachers in K-12 that are formally trained in this knowledge area, these reasons may limit the utility of manual approaches as the sole assessment alternative.

Yet, as creativity is complex and multi-dimensional, it can be expressed in diverse ways, just as an automated assessment of computer programs may not be sufficient as a single way to account for all its facets. Artifact-oriented approaches for the assessment of creativity are sometimes criticized for under-representing the creativity of individuals (COUGER and DANGATE, 1996). Thus, to capture creativity in a more comprehensive way, it may be beneficial to adopt diverse approaches, e.g., completing an automated objective assessment of the artifact and manual subjective assessment by peers and/or instructors, such as in Kershaw et al. (2022). Or comparing the results of artifact assessment with the results of the TTCT test, such as in Hershkovitz et al. (2019).

In order to properly assess creativity, it is essential to provide a robust assessment model. Considering the complexity of the assessment of the creativity of artifacts, a further shortcoming observed is the lack of scientific and systematic evaluations concerning the reliability and validity of the proposed assessment approaches. Although some studies are based on large samples, these may not provide a detailed evaluation of the approach.

Given that human raters are critical in manual assessments, understanding who they are and their level of expertise is also important as it has a direct impact on inter-rater agreement and reliability. As the human judgment of creativity may be a subjective process, it is necessary to study to which degree the perception of creativity of computer programs is consistent and not an idiosyncratic result of an assessor's subjective judgment. In this regard, Manske and Hoppe (2014) found that assessors with an educational theoretical background are more likely to provide a consistent assessment of artifact creativity in the context of computing education. Taking into consideration that currently teachers formally trained in computer science are scarce in K-12, as well as self- or peer-assessment conducted by students that are still learning computing, this question has to be considered carefully in the design of the assessment instruments. This is particularly important to assure consistency regarding the meaning of assessment criteria used by the assessors.

In general, the approaches indicate a score or performance level as a result of the assessment, typically on an ordinal scale. These scales are developed using Classical Test Theory, representing the creativity of the artifact as the sum of the score, e.g., in Zhong et al. (2015). Alternatives such as a definition or evaluation of a scale based on Item Response Theory may be a more appropriate way of creating a construct for assessing artifact creativity. However, none of the encountered approaches uses Item Response Theory. Myszkowski and Storme (2019) argue that Item Response Theory-based scoring can lead to a more appropriate and accurate estimation of the latent trait (the creative value of the artifact), questioning also common practices regarding the aggregation of ratings.

Most of the approaches do not propose how to use these results as part of a summative assessment for grading. For example, Manske and Hoppe (2014) propose to use the score to classify the student's computer program. Different from other automated approaches in the context of computing education such as Dr. Scratch (MORENO-LEÓN and ROBLES, 2015) or CodeMaster (GRESSE VON WANGENHEIM, HAUCK, *et al.*, 2018), none of the approaches uses any kind of ludic representation of the results of the assessments (such as badges, ninja belts, etc.) to motivate students, especially in K-12.

Considering that creativity is a central competence of the 21st century, the lack of wider research on the assessment of creativity on the strand of the artifact as part of computing education is surprising. Although the study of creativity in computing education on the strand of the process seems to be more emphasized, this indicates a need for future work in the area of the artifact to effectively and efficiently support creativity as part of computing education. The availability of reliable and valid approaches is also essential to systematically create a body of empirical evidence supporting the assumption that computing education also contributes to the development of creativity, especially on the strand of the artifact, as systematic research on this issue is still scarce, mostly dating back to the 1980s and 1990s (CLEMENTS, 1995), with only a more recent study aiming at establishing that computer programs are creative works (KERSHAW, CLIFFORD, *et al.*, 2022).

4 CONCEPTUAL ASSESSMENT MODEL

This chapter proposes a conceptual model for assessing the creativity of computing artifacts focusing on App Inventor projects. Aiming at developing a coherent assessment, the Evidence Centered Design framework (MISLEVY, ALMOND, and LUKAS, 2003) is adopted for analyzing the context and designing a conceptual model.

4.1 DOMAIN ANALYSIS

In this section, definitions and details on assessing artifact creativity are presented considering computational artifacts in the context of computing education.

4.1.1 Creativity in computing education curricula/frameworks/guidelines

Computing is a fundamentally creative discipline (FINCHER and ROBINS, 2019). In the context of computing education, creativity is one of the keys to respond common challenges in the development of computer programs today as well as to preparing students to learn to create, test, and refine computer programs (SHUTE, SUN and ASBELL-CLARKE, 2017). Creativity can be analyzed based on different strands (person, product, process, press) (RHODES, 1961). In the educational context, the work product created by the student and its assessment can be an alternative to nurture creativity, especially considering that "whatever we as teachers want our students to learn, their perceptions of what is significant (and therefore where and how to direct their efforts) are strongly shaped by the lens of assessment." (FINCHER and ROBINS, 2019, p. 539).

Several curricula/frameworks/guidelines for computing education include some aspects of creativity in creating computing artifacts. For example, CSTA (2016) considers that "more than just a tool, computers are a readily accessible medium for creative and personal expression." (CSTA, 2016, p. 11). In this context, along with computing practices, the CSTA framework explicitly includes creativity as part of the process of creating a computational artifact (Table 6), considering that "a teacher can also explicitly present problems and ask for children's creative solutions." (CSTA, 2016, p. 190).

Table 6 – Creativity in computing curricula/framework/guidelines for K-12 education

Curricula/ Framework/ Guideline	Creativity for creating computing artifacts in K-12
Computer Science Teachers Association (CSTA, 2016)	**"**Creating Computational Artifacts: The process of developing computational artifacts embraces both **creative expression and the exploration of ideas to create prototypes and solve computational problems** [emphasis added]. Students create artifacts that are personally relevant or beneficial to their community and beyond. Computational artifacts can be created by combining and modifying existing artifacts or by developing new artifacts. Examples of computational artifacts include programs, simulations, visualizations, digital animations, robotic systems, and apps." (CSTA, 2016, p. 80)
Partnership for 21st Century Learning - A Network of Battelle for Kids (P21, 2020)	"Learning & Innovation Skills - Implement Innovations: **Act on creative ideas to make a tangible** [emphasis added] and useful contribution to the field in which the innovation will occur." (BATTELLE FOR KIDS, 2019, p. 4)
Base Nacional Comum Curricular (MEC, 2018)	"Exercising intellectual curiosity and using the proper approach to science, including research, reflection, critical analysis, imagination, and **creativity** [emphasis added], to investigate causes, develop and test hypotheses, formulate and solve problems and **create solutions (including technological)** [emphasis added] with based on knowledge of different areas." (MEC, 2018, p. 9) – translated.
Guidelines for Teaching Computing in K-12 Education (SBC, 2018) in Brazil	Emphasize the importance of creativity superficially. Does not explicitly include creativity as one key practice, competency, ability, or skill to create computing artifacts.

Source: Elaborated by the author based on cited references.

Other curricula for K-12, such as the Framework for 21st Century Learning (P21, 2020), include creativity as a main topic, inside the "Learning & Innovation Skills" with the same level of importance of key subjects, such as reading, mathematics, science, geography, and history. P21 considers that learning and innovation skills are what will separate those who are prepared for 21st century life and work challenges from those who are not. Thus, focusing on creativity and other learning and innovation skills is essential to prepare students for building the skills needed not only in school but also in life (P21, 2020).

Other international frameworks, such as the Creative Thinking Framework created by the Programme for International Student Assessment (PISA - OECD, 2021) explicitly include creativity. Although focusing on creative thinking, the framework also includes dimensions of artifact creativity in general, yet not focusing on computing education. For example, PISA considers that "creative engagement is a means to a 'better end', and it can thus be characterized by generating solutions that are original, innovative, effective and efficient." (PISA - OECD, 2021, p. 19). This shows that creativity is being recognized as an important 21st century competency to be developed also as part of K-12 education, including computing education.

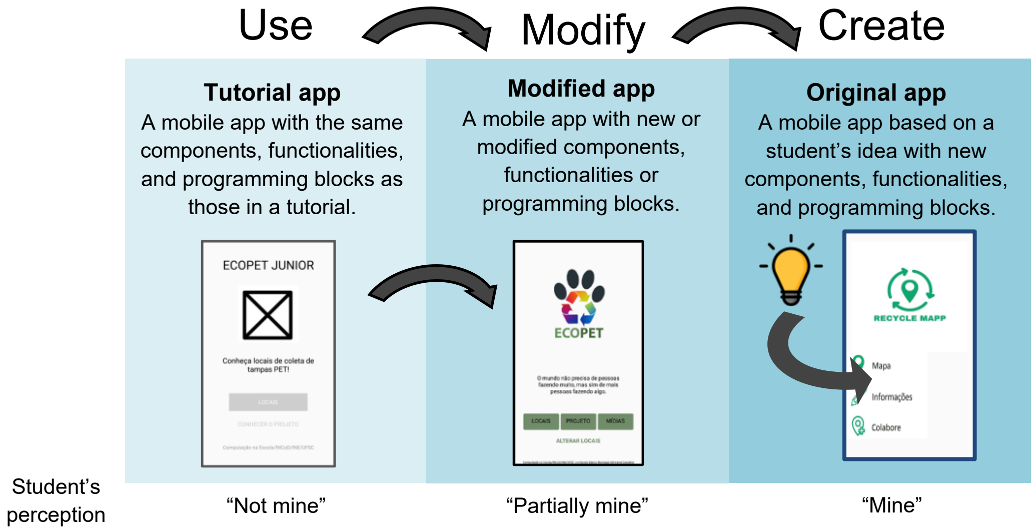
The Brazilian curricula *Base Nacional Comum Curricular* (BNCC) for K-12 education, also includes creativity as part of one of the key competencies for K-12 students (MEC, 2018). BNCC considers that in the new world scenario, competencies such as being creative are essential and require much more than just 'accumulating' information. This is emphasized by considering one of the learning and development rights in early childhood education as to be able to express themselves, as a dialogical, creative, and sensitive person. However, the guidelines for teaching computing in K-12 by the Brazilian Computing Society only cite the importance of creativity and do not explicitly include it as a competency, ability, or skill (SBC, 2018).

4.1.2 Pedagogic approach for artifact creativity of mobile apps

The inclusion of artifact creativity in computing education has several implications for practice, especially regarding the choice of pedagogic approaches supporting the development of creative mobile apps. In this context, one of the alternatives is an active learning approach, which is a "general term to describe a range of practices where students are involved in actively doing and reflecting to facilitate their learning." (FALKNER and SHEARD, 2019, p. 584). The development of creative mobile apps not only includes the "active" involvement of students but also encompasses opportunities for reflection when considering all the processes of creating a mobile app using, e.g., the design thinking process in a problem-based strategy (FERREIRA, GRESSE VON WANGENHEIM, *et al.*, 2019).

Considering an active learning scenario, one of the approaches to progressively develop creativity is the use-modify-create cycle (LEE, MARTIN, *et al.*, 2011; LYTLE, CATETÉ, *et al.*, 2019) by adopting a problem-based strategy. Following use-modify-create, students learn first by "using" and analyzing a given computational artifact, "remixing" and modifying an existing artifact, until eventually "creating" a new one (LEE, MARTIN, *et al.*, 2011). Thus, students develop the ability to generate creative computing artifacts during the modification and creation steps (Figure 6).

Figure 6 – Use-modify-create in the context of mobile app development

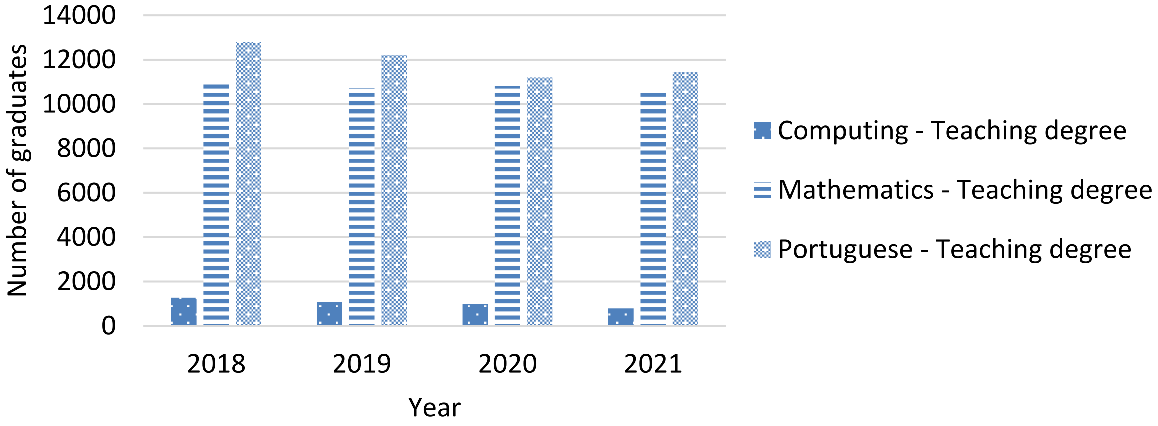

Use
Modify
Create
Tutorial app
A mobile app with the same components, functionalities, and programming blocks as those in a tutorial.
Modified app
A mobile app with new or modified components, functionalities or programming blocks.
Original app
A mobile app based on a student's idea with new components, functionalities, and programming blocks.
ECOPET JUNIOR
ECOPET
RECYCLE MAPP
Mapa
Informações
Colabore
Student's perception
"Not mine"
"Partially mine"
"Mine"

Source: Elaborated by the author.

Focusing on the create stage, students can follow different processes to create a mobile app based on their ideas. One alternative is following a design thinking process using a systematic human-centered approach to explore the definition of the problem and synthesize a solution. This process encompasses inspiration, ideation, and implementation to satisfy the needs of the end-users to arrive at a strategy that is technologically feasible and viable (BROWN, 2008). By incorporating design thinking into the create stage, it is possible to better prepare students to shift from "technology-centered design" to "human-centered design" (BRENNER and UBERNICKEL, 2016) allowing them to express themselves creatively.

In the context of this research, the assessment of artifact creativity focuses on mobile apps created by students in a computing education context. Considering the K-12 context in Brazil, most schools do not have teachers with teaching degrees in computing. One of the reasons is that the number of graduates with teaching degrees focusing on computing is much lower than the number of graduates with teaching degrees for other subjects, such as Mathematics and Portuguese (Figure 7).

Figure 7 – Number of graduates with teaching degrees from 2018 to 2021 in Brazil



Source: Elaborated by the author based on the reports from the Ministry of Education (INEP; MEC, 2019; 2020; 2021; 2022).

Introducing computing in K-12 on a larger scale in Brazil, therefore, depends on non-computing teaching degree professionals, making computing education inherently interdisciplinary. By involving professionals from diverse disciplines such as mathematics, science, languages, and social sciences, computing education concepts can be effectively incorporated into existing curricula. However, this can make it more difficult, for example, to assess student learning as formally trained computing teachers in K-12 are scarce. Various initiatives aim at training in-service teachers from other areas, covering basic computing competencies as well as pedagogical and technological knowledge to enable the teaching of computing in a multidisciplinary way in their respective disciplines (ALVES, KRETZER, *et al.*, 2020).

4.2 DOMAIN MODELING

In this section, the elements that are needed in the assessment of creativity are identified considering computational artifacts in the context of computing education using Principled Assessment Designs for Inquiry (PADI) design patterns (SEERATAN and MISLEVY, 2008). Thus, a conceptual model is designed including (i) student competencies, (ii) evidence model, and (iii) tasks (MISLEVY, ALMOND, and LUKAS, 2003).

With respect to student competencies, the student model is designed considering focal knowledge, skills, and abilities (MISLEVY and HAERTEL, 2006) as well as other knowledge, skills, and abilities that may be required in the context of artifact creativity. For the evidence model, potential observations are defined, i.e., possible things one could see students doing that would give evidence about their knowledge, skills, and abilities (MISLEVY and HAERTEL, 2006). The task model includes (i) potential work products, (ii) characteristic features evidencing aspects of assessment situations that are likely to evoke the desired evidence, and (iii) variable features for describing aspects of assessment situations that can be varied to shift difficulty or focus (MISLEVY and HAERTEL, 2006).

4.2.1 Student competencies

Considering a use-modify-create strategy, at the create level, the general learning objective is that the student should be able to create a creative mobile app (Table 7). In addition, students also need to be able to develop mobile apps with App Inventor, covering basic computing concepts and practices.

Table 7 – Student model for creativity assessment of mobile applications

Component	**Value**
Title	Creativity assessment
Summary	This design pattern concerns working with mobile apps created by students using App Inventor.
Rationale	Creativity is one of the key skills of the 21st century that can be also nurtured through computing education.
Focal Knowledge, Skills, & Abilities	- Create technological solutions using the proper approach to science, including research, reflection, critical analysis, imagination, and creativity (MEC, 2018). - Demonstrates originality and inventiveness in work (BATTELLE FOR KIDS, 2019). - Acts on creative ideas to make a tangible and useful contribution to the field in which the innovation occurs (BATTELLE FOR KIDS, 2019).
Additional Knowledge, Skills, & Abilities	- Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests (CSTA, 2017). - Create clearly named variables that represent different data types and perform operations on their values (CSTA, 2017). - Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals (CSTA, 2017). - Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions (CSTA, 2017). - Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs (CSTA, 2017). - Incorporate existing code, media, and libraries into original programs, and give attribution (CSTA, 2017). - Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs (CSTA, 2017).

Source: Elaborated by the author.

4.2.2 Evidence model

Considering the focus on the creativity of mobile apps, the potential observation relies on the analysis of the students' mobile apps. The analysis objective is to determine the degree to which the student has acquired the necessary focal and additional knowledge, skills, and abilities considering different *aspects* of mobile apps. An aspect, within the context of this research, is defined as *a discrete entity characterized by explicitly specified criteria for its systematic extraction and identification*. Aspects can encompass various elements of mobile apps created with App Inventor. They include components, programming blocks, functionalities, topics, and tags (see section 4.2.2.1).

The creativity construct is decomposed into three dimensions based on DT dimensions (GUILFORD, 1950), originality (rarity of ideas), fluency (number of ideas), and flexibility (range of ideas). Considering that the main objective is to quantify the creative essence of a singular artifact (i.e., the mobile app), the conventional definitions of DT are adapted as follows: originality is measured through the rarity of aspects, fluency is measured through the number of aspects, and flexibility is measured through the range of aspects. Expanding these definitions to the context of mobile apps, originality refers to the extent to which the aspects of the mobile app deviate from a reference universe of mobile apps created in a comparable context; fluency refers to the extent to which various aspects are employed within the mobile app in contrast to an entirely empty mobile app; and flexibility refers to the breadth of categories of the aspects used within the mobile app (Table 8).

Table 8 – Evidence model for the creativity of mobile apps in computing education

Component	**Value**		
Title	**Originality**	**Fluency**	**Flexibility**
Potential Observation	Determining the degree to which the mobile app aspects are different from a reference universe of mobile apps created in a similar context.	Determining the degree of utilization of aspects within the mobile app compared to an entirely empty mobile app.	Determining the degree of diversity of the categories of aspects used in the mobile app.

Source: Elaborated by the author.

4.2.2.1 Aspects of mobile apps

There are five aspects of mobile apps defined in the context of this research: (i) components, (ii) programming blocks, (iii) functionalities, (iv) topic, and (iv) tags. Components and programming blocks follow the structure presented in Table 1 and Table 2 respectively. As part of the components, a set of User Interface (UI) components (Table 9) has also been defined through an analysis of which UI components are visible to the user (KREUCH, 2022; SOUZA, 2022). This is relevant because UI components do not necessarily need programming blocks to be useful on the screen, for example, a label on the screen possesses inherent value simply by its presence and visibility on the user interface.

Table 9 – UI components in App Inventor

UI component	Description
BackgroundImage	An image is displayed in the background of the screen.
Button	A component designed to register clicks.
CheckBox	A component that triggers an event when the user clicks on it.
ContactPicker	A button that, upon clicking, presents a list of contacts for the user to select from.
DatePicker	A button that, upon being clicked, opens a popup dialog, enabling the user to choose a date
EmailPicker	A type of text input field where users can enter the name or email address of a contact, and the phone will display a dropdown menu of suggested options to assist in completing the entry.
Image	A component for displaying images.
ImagePicker	A specialized button that opens the device's image gallery for user image selection.
Label	A component for displaying text, which content is determined by the Text property.
ListPicker	A button that, upon clicking, presents a list of text options for the user to select from.
ListView	A component for presenting a list of text and image elements.
Map	A two-dimensional container that displays map tiles as a background and supports the placement of multiple Marker elements to pinpoint locations on the map.
Notifier	A component for presenting alert dialogs, messages, and temporary notifications.
PasswordTextBox	A password input field.
PhoneNumberPicker	A button that, upon clicking, shows a list of phone numbers from the contacts to select.
Slider	A progress bar with a draggable thumb.
Spinner	A component that presents a popup containing a list of items.
Switch	A component that allows users to toggle between two states, generating an event when switched.
TextBox	A text input field for user text entry and display text.
TimePicker	A button that, upon clicking, opens a popup dialog for user time selection.
VideoPlayer	A multimedia component with the ability to play videos.
WebViewer	A component for displaying web pages.

Source: Elaborated by the author.

The functionalities aspect is defined and extracted through a systematic and rule-based approach (Table 10). Essentially, a set of predefined rules is adopted, so that when satisfied by the mobile app, it triggers the identification of specific functionalities: *IF the rule for a functionality is satisfied THEN identify the functionality for the mobile app*. A rule defines one or more blocks as evidence for identifying functionalities. Blocks refers to the set of programming blocks available in App Inventor, and text block refers specifically to the string block. This systematic method ensures that the framework encompasses not only the structural components and programming blocks of the mobile app but also the diverse range of functionalities it offers.

Table 10 – Extracted functionalities

Functionality	**Extraction rule**
Access website	Block contains WebViewer OR block contains Web OR text block contains “http://www.”
Animation	Block contains ImageSprite OR block contains Ball
Calculator	Block contains math_add AND Block contains math_subtract AND Block contains math_multiply AND Block contains math_division AND Block contains math_number AND Block contains Label AND Block contains Button
Canvas	Block contains Canvas
Choose an image from the gallery	Block contains ImagePicker
Convert speech to text	Block contains SpeechRecognizer
Convert text to speech	Block contains TextToSpeech
Count steps	Block contains Pedometer
Detect acceleration	Block contains AccelerometerSensor
Display information	Block contains Label with characters >= 20 OR block contains TextBox with characters >= 20
Display information on a list	Block contains ListView
Login	(Block contains File OR block contains TinyDB OR block contains FirebaseDB OR block contains TinyWebDB OR block contains CloudDB) AND block contains PasswordTextBox
Make a call	Block contains PhoneCall
Mark the position on a map	Block contains Circle OR block contains FeatureCollection OR block contains LineString OR block contains Marker OR block contains Polygon OR block contains Rectangle
Measure air pressure	Block contains Barometer
Measure angular velocity	Block contains GyroscopeSensor
Measure light level	Block contains LightSensor
Measure magnetic field	Block contains MagneticFieldSensor
Measure proximity	Block contains ProximitySensor
Measure relative air humidity	Block contains Hygrometer
Measure temperature	Block contains Thermometer
Paint	Block contains Canvas_Clear OR block contains Canvas_DrawArc OR block contains Canvas_DrawCircle OR block contains Canvas_DrawLine OR block contains Canvas_DrawPoint OR block contains Canvas_DrawShape OR block contains Canvas_DrawText OR block contains Canvas_DrawTextAtAngle OR block contains Canvas_SetBackgroundPixelColor

Functionality	Extraction rule
Play sound	Block contains Sound OR block contains Player
Play video	Block contains VideoPlayer
Record audio	Block contains SoundRecorder
Record data	Block contains TextBox >= 3 AND (Block contains File OR Block contains TinyDB OR Block contains FirebaseDB OR Block contains TinyWebDB OR Block contains CloudDB)
Record video	Block contains Camcorder
Save data in the cloud	Block contains FirebaseDB OR block contains TinyWebDB OR block contains CloudDB
Save data locally	Block contains File OR block contains TinyDB
Scan QRCode	Block contains BarcodeScanner
Share artifact	Block contains Sharing OR block contains Texting OR block contains Twitter
Show device spatial orientation	Block contains OrientationSensor
Show geolocation	Block contains LocationSensor
Show image	Block contains Image_set_Visible
Take a picture	Block contains Camera
Timer	Block contains Clock
Translate	Block contains YandexTranslate
Use API from another app	Block contains ActivityStarter
Use Bluetooth	Block contains BluetoothClient OR block contains BluetoothServer
Use data from forms	Block contains TextBox >= 3
View contact list	Block contains ContactPicker OR block contains EmailPicker OR block contains PhoneNumberPicker
View map	Block contains Map OR (block contains “http://www.” AND block contains “maps”)

Source: Elaborated by the author.

The set of functionalities was defined through an iterative approach. To encompass functionalities that could be extracted through static analysis of App Inventor projects, a thorough examination of various App Inventor projects was performed to identify and catalog a comprehensive range of functionalities that mobile apps created with App Inventor could exhibit. This iterative process involved continuously refining and expanding the set of functionalities to ensure its good coverage.

The aspect of topics has been defined based on the mapping of categories used in the main online app stores, such as Google Play and Apple App Store, and adjusted to the characteristics of the App Inventor scenario. Some categories were merged because they correlated with each other, such as environment and botany; design, painting, and photography; finance and work; tourism and geography; and weather and meteorology. New categories were created to generalize their scope: cars, vehicles, and transport; beauty and fashion; spirituality, belief, and divination; citizenship and social issues; engineering, physics, and construction. New categories were also created to accommodate mobile apps that do not fit within the scope of other categories: math; music; robotics, physical computing, and automation (Table 11).

Table 11 – Set of topics of mobile apps

#	**Topic**	**Description**
1	Animals and pets	Apps for animal care, animal identification, or animal-related.
2	Beauty and Fashion	Apps that provide information on fashion history or on scheduling for beauty salons or beauty tutorials.
3	Cars, vehicles, and transport	Apps related to automobiles, bicycles, or other types of vehicles and transportation.
4	Citizenship and social issues	Apps to deal with civic issues, complaints, and transparency of public resources.
5	Communication	Apps that provide support for communication such as access to chats or chat applications, contacts, phones, etc.
6	Design, painting, and photography	Apps for sketching, painting, designing, coloring books, or helping capture, edit, manage, store, or share photos.
7	Education	Apps for learning a skill, or specific subject or focused on study matters, such as test scores, classrooms, educational institutions.
8	Engineering, physics, and construction	Apps related to construction, planning/calculating/measuring construction materials, or information on engines and machines.
9	Entertainment	Apps that inform the user about some event or present visual entertainment or other entertainment content.
10	Environment and botany	Apps for recycling, information about the environment/plants, plant identification, and plant care.
11	Finance and work	Apps that carry out financial transactions or assist the user in commercial, professional, or personal financial matters, or advertise and help to find jobs, openings, portfolios, etc.
12	Food and drinks	Apps that provide recommendations, instructions, recipes, or reviews related to preparing, consuming, or reviewing foods or liquids/juices.
13	Healthy life and sport	Apps related to healthy living or practicing a sport, training, diet, nutrition, stress management, physical conditioning, and weight indicators (e.g., BMI).
14	Math	Apps that provide information or perform math calculations, such as algebraic, trigonometric, and geometric calculations.
15	Medicine and health	Apps focused on medical education, management of health information for patients or healthcare professionals, and baby and child care.
16	Mobile tools	Apps for flashlights, code readers, etc.
17	Music	Apps for listening to the radio, music, etc.
18	Productivity	Apps that make a specific process or task more organized or efficient.
19	Robotics, physical computing, and automation	Apps for controlling robots, communicating with physical boards (e.g., Arduino), and home/building automation.
20	Spirituality, belief, and divination	Apps for fortune-telling, horoscope, religious, philosophical, and existential issues
21	Tourism and geography, weather and meteorology	Apps that provide information or maps about a location or points of interest, or provide forecasts, alerts, and information related to the weather conditions of a location

Source: Elaborated by the author.

The aspect of tags is related to the keywords that can be extracted from the textual content embedded within the mobile app. In essence, tags serve as markers or descriptors for the mobile app. Given that the textual content of every mobile app is distinct and tailored to its specific purpose and audience, it follows that the assortment of tags associated with each mobile app is likewise one-of-a-kind and a reflection of its identity.

4.2.3 Tasks model

For the task model, the potential work artifact is the mobile app (`.aia` file) created by the learners using App Inventor (Table 12). Based on this artifact, evidence is extracted in order to get potential observations defined in Section 4.2.2. Considering the aspects of the assessment task, the characteristic features comprise the development of an open-ended free-choice mobile app following a problem-based strategy in an active learning context, e.g., during the create stage of the UMC cycle. Considering that creating a mobile app is not only about programming it but also following the whole design thinking process, students follow all the steps for empathizing, defining, ideating, prototyping, and testing the mobile app.

Table 12 – Design pattern for creativity

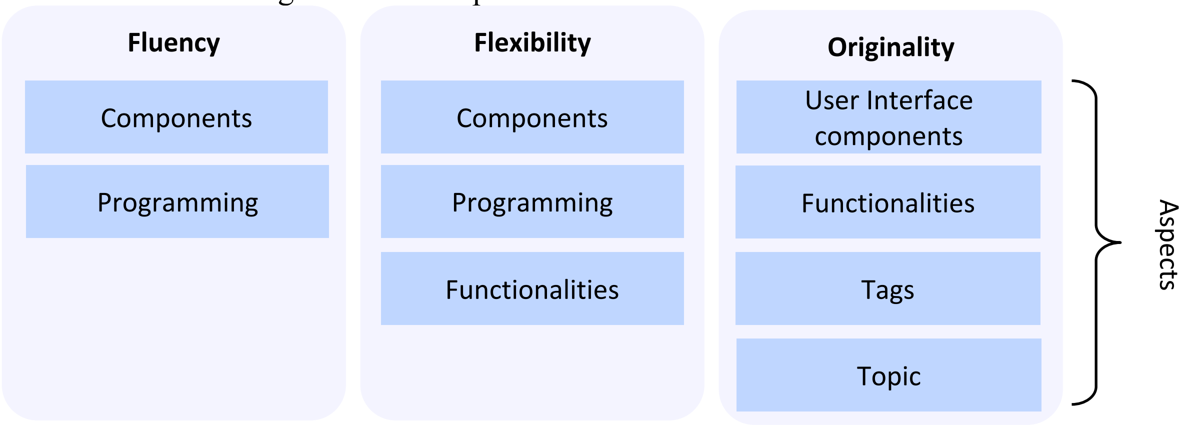
Component	**Value**
Title	**Creativity**
Potential Work Product	Mobile app (.aia file) created with App Inventor in Brazilian Portuguese.
Characteristic Features	Active learning context, focusing on the create stage in the UMC cycle, on which students need to create a mobile app using design thinking to solve a problem in a problem-based strategy. Games apps are excluded.
Variable Features	Mobile app created by the student(s) vs. mobile app created by other students(s) in a similar context (e.g., mobile apps in App Inventor Gallery).

Source: Elaborated by the author.

4.3 CONCEPTUAL ASSESSMENT FRAMEWORK

In this section, the conceptual framework for assessing originality, flexibility, and fluency dimensions is presented (Figure 8). All measures are based on the aspects of mobile apps. The assessment of fluency measures the overall number of the aspects of components and programming blocks within the mobile app, seeking to capture the extent to which these aspects are utilized. In essence, it serves as a quantitative measure of how many components and programming blocks are employed within the mobile app. With respect to flexibility, the diversity of categories of the aspects of components, programming blocks, and functionalities are measured. Regarding originality, it involves identifying which functionalities are present in the mobile app, which user interface components are used, which topic best describes the mobile app, and extracting relevant tags. Here rarity is measured considering a reference universe of existing mobile apps.

Figure 8 – Conceptual assessment framework overview



Source: Elaborated by the author.

In Brazil, it is common practice to use a scale ranging from 0 (insufficient) to 10 (excellent) for grading students' work, which is therefore adopted as a grading scale in the model. This scale allows for a clear and standardized assessment of students' performance and provides a broader range of evaluations compared to other grading systems. Assigning a numerical grade on a scale of 0 to 10 enables educators and students to more accurately measure progress, identify areas of improvement, and gauge the level of mastery attained. By aligning the model's grading system with the widely used scale in Brazil, consistency and familiarity are ensured for both students and educators, facilitating a seamless integration of the model's assessments within the existing educational framework.

4.3.1 Conceptual assessment model of originality

Besides ‘exactly replica’ (non-original) or ‘different’ (original), originality can have different degrees depending on the elements present in the reference universe. This is related to the distinction made between Kaufman and Beghetto’s Four-C model: mini-c (impact on individual) and little-c (impact on individual’s community), as opposed to Pro-c (impact on organization or field) and Big-C (impact on culture and society). Considering the educational context, a creative artifact definition is adopted contextualized to the little-c and mini-c creativity, encompassing the creativity inherent in the everyday and the learning process

(KAUFMAN and BEGHETTO, 2009). Thus, when referring to an original artifact, Pro-c and Big-C are purposely excluded, as on the one hand, it is not expected for the student to create something so revolutionary inside the classroom context, but on the other hand, the creative insights experienced by students should not be overlooked. This is calibrated by inserting into the reference universe only mobile apps created in a similar educational context.

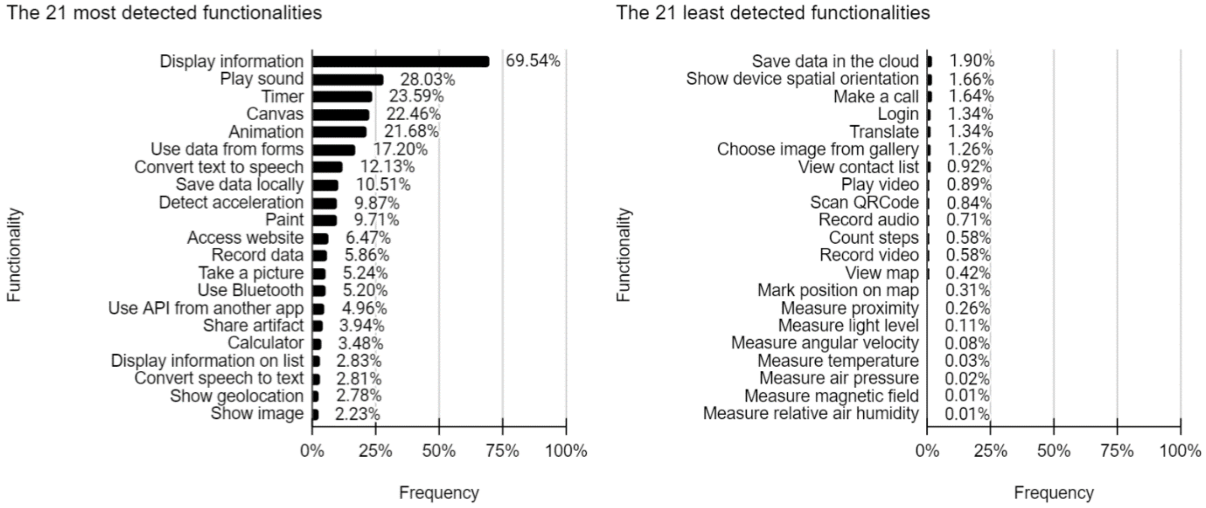
The analysis of originality involves computing the frequencies of the aspects of functionalities, components, topics, and tags within the mobile app being assessed. This helps to capture the unique aspects and characteristics of the mobile app quantitatively. By computing the frequencies and combinations, the model can account for the distinctiveness and the extent to which the app deviates from common patterns. To give a score for originality, the computed frequencies and combinations are compared with a reference universe. The reference universe represents a collection of existing mobile apps that serve as a benchmark for assessing originality. By comparing the app under assessment with this reference universe, the model can determine the extent to which the app introduces novel and unique elements. This comparative analysis helps identify overlaps, similarities, or deviations from established norms, allowing for a more holistic evaluation of the app's originality.

4.3.1.1 Originality of functionalities

Functionalities play a pivotal role in the overall perceived creativity of a mobile app, making them a crucial aspect of the assessment. They represent the core features, capabilities, and interactions that an app offers to its users. Assessing the originality of functionalities enables comparisons with similar apps and helps identify unique or innovative features that set the app apart from others in its category.

An analysis of 100,000 mobile apps created by people from all over the world, available at the App Inventor Gallery, was performed. Results show that no functionality was detected in 1,2K apps, according to the defined extraction rules (Table 10). Most of the apps in which no functionality was detected are apps without programmatic functionality, that is, apps composed only of screens with design components and no programming blocks or completely blank projects, without any design components or programming blocks. Considering 98,770 apps in which at least one functionality was detected, the functionality that appears the most is *display information* in more than 69.54% of the apps (Figure 9).

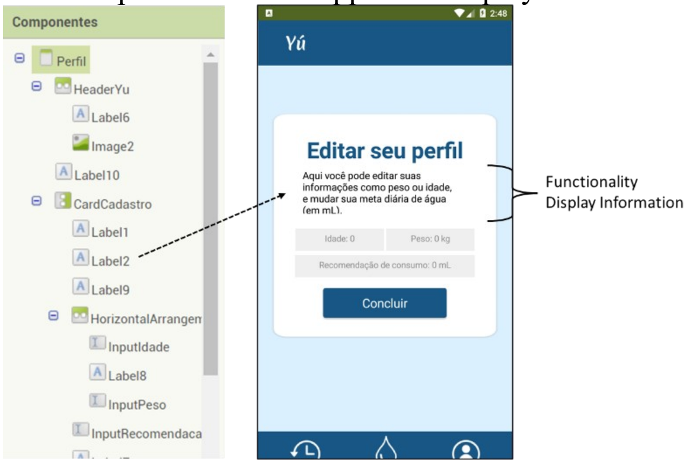
Figure 9 – Detected functionalities in mobile apps (n=98,770) of the App Inventor Gallery



Source: Elaborated by the author.

The most detected functionality, *display information*, comprises a basic user interface design with labels or text boxes used to show information on the screen to the user, including help information about using the mobile app (Figure 10). It is also common for teaching materials to present the App Inventor programming environment through tutorials that display information on the screen in order to demonstrate to the learner an app similar to the classic "Hello World", i.e., showing "Hello World" on the screen. Therefore, this functionality can be used with different goals in mind.

Figure 10 – Example of a mobile app with Display Information



Aqui você pode editar suas informações como peso ou idade, e mudar sua meta diária de água (em mL).

Source: Elaborated by the author based on projects of the App Inventor Gallery.

The second functionality that appears the most is *play sound* in 28.03% of the mobile apps. Considering the defined extraction rules (Table 10), this includes apps for listening to music or sound effects, such as the Hello Codi tutorial (MIT APP INVENTOR, 2022b), which teaches how to create an app that plays the sound of a bee when clicking a button. The functionality *timer* appears in 23.59% of the apps (Figure 9). This can be explained by the fact that, typically, time stamping is used for synchronizing screen effects, events, or other functionalities programmed in the app.

The functionality *canvas* appears in 22.46% of mobile apps and can be used for a variety of purposes, including drawing, displaying, and editing images or sprites, which are components of the canvas. The functionality *paint* appears in 9.71% of the projects and includes drawing by dragging one finger across the screen, using buttons, or dragging sprites from a canvas. It is also possible to use sprites to show animations programmed via blocks as defined by the *animation* functionality, which appears in 21.68% of the apps (Figure 9). Animations can also be used for a game, such as in the popular tutorial for creating a Space Invaders-like game (MIT APP INVENTOR, 2022a).

Some data-related functionalities appear in several projects, for example, *use data from forms*, which appears in 17.20% of the projects and refers to the mobile apps with at least three text fields for entering data and using this data in a programmatic way. Other data-related functionalities include *save data locally*, which appears in 10.51% of the apps, *save data in the cloud*, which is less used and appears in 1.90% of the apps, and *record data* (creating data logging), which appears in 5.86% of the apps (Figure 9).

Some functionalities are related to the usability aspects of mobile apps. For example, *convert text to speech* is one of the most detected functionalities and appears in 12.13% of projects. One of the reasons for this expressive use is the TextToSpeech component of App Inventor and the Talk to Me tutorial (MIT APP INVENTOR, 2022c), one of the most popular beginner tutorials of App Inventor, available directly on the website, teaching how to create an app that can “talk”. Although the TextToSpeech component is only available in English, German, Spanish, French, and Italian, it can also be used with texts in other languages. Projects with this functionality can be used to help blind or illiterate people as the information on the screen is spoken to assist in using the app. In this context, *detect acceleration* can be used with the same goal of accessibility, allowing the interaction with the app to be carried out through gestures, for example, an app that speaks the result of an action triggered after shaking the phone. The functionality to *convert speech to text* appears in 2.81% of the apps (Figure 9) and allows text-type data entries to be made via voice commands. It can also be used to facilitate the usability of mobile devices.

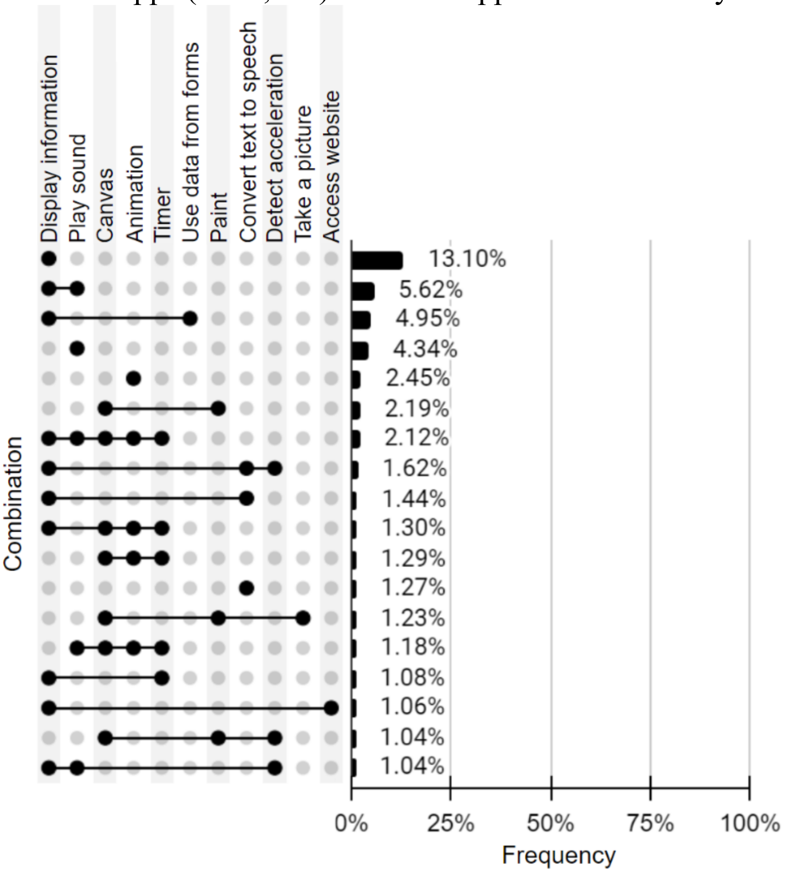
Some functionalities that include the connection of the mobile app with external smartphone components and are among the 21 most used functionalities are *access website*, *use*

Bluetooth, *use API from another app*, and *share artifact* (Figure 9). These functionalities are typically used to access specific resources, for example, accessing a store's website, connecting with a physical device, and accessing messaging applications such as WhatsApp. The functionality of sharing artifacts allows the smartphone operating system to share anything through the apps installed on the smartphone.

Among the less detected functionalities are the ones that refer to something specific, such as measurement functionalities, for example, *measure proximity*, *light level*, *angular velocity*, *temperature*, *air pressure*, *magnetic field*, *relative air humidity*, etc. (Figure 9). The implementation of these functionalities is only possible because App Inventor has purpose-specific components that exchange messages with the smartphone's hardware components. The implementation of these functionalities, however, also depends on the smartphone having support for it, for example, a smartphone without a barometer will not allow implementing *measure air pressure* as it depends on that component, which is only present in some smartphones.

Considering 98,770 mobile apps in which at least one functionality was detected, in total, 6,905 combinations of functionalities were detected. Few combinations are predominant, with only 18 combinations appearing in more than 1% of the projects. Furthermore, the predominant combinations (>1% of the projects) are composed of 1 to 5 functionalities (Figure 11). Most combinations appear in only a few projects, 4,432 combinations consisting of 2 to 29 functionalities appear only in one app, that is, 4,432 apps with unique combinations that only appear in them.

Figure 11 – Frequency of functionalities combinations appearing in more than 1% of the mobile apps (n=98,770) from the App Inventor Gallery



Source: Elaborated by the author.

Combinations of one functionality. Combinations appearing in more than 1% of mobile apps composed of one functionality include the most frequent combination *display information* and appear in 13.10% of the projects. This result is consistent with the frequency of this functionality overall, which is present in 69.54% of the projects, a number that includes it alone or together with other features (Figure 9). The combination of apps that have only *play sound* is represented by 4.34%. This usually encompasses projects like soundboards with one or more buttons that play a sound, such as the Hello Codi tutorial (MIT APP INVENTOR, 2022b), or simple projects that use sound signals to indicate some action. Projects with only *animation* represent 2.45% of the apps and are related to animations of sprites moving on a canvas, like the Ball Bounce tutorial (MIT APP INVENTOR, 2022c), or sprites being controlled by the user, such as in games. Projects with only *convert text to speech* represent 1.27% of the apps (Figure 11) and are usually inspired by or identical to the Talk to Me tutorial (MIT APP INVENTOR, 2022c).

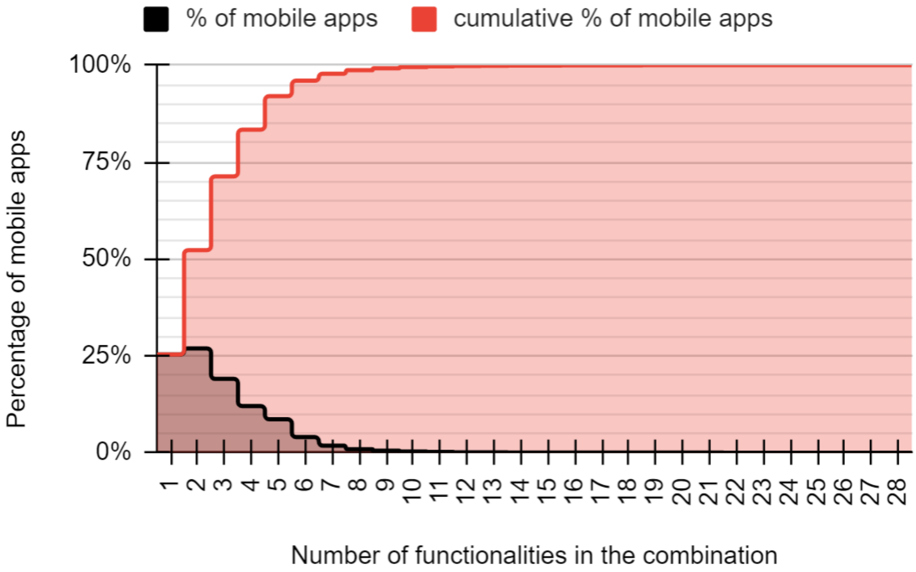
Combinations of two functionalities. The most frequent combination of two functionalities is *display information* and *play sound* appearing in 5.62% of projects (Figure 11). Typically, these projects are remixes of the tutorials available on the App Inventor website. This also includes the combination of *display information* and *use data from forms* representing 4.95% of the projects. This combination normally includes projects that perform some action based on data entered by the user through a TextBox component, for example, calculating the BMI based on the weight and height entered. The combination composed by the functionality *canvas* and *paint* represents 2.19% of the projects. These projects are very similar to the classic Paint software, providing an area for drawing, and buttons to erase the drawing or to select colors and geometric shapes such as squares, rectangles, etc. The combination composed by *converting text to speech* and *display information* is generally very similar to the Talk to Me tutorial (MIT APP INVENTOR, 2022c) with the inclusion of more information on the screen and represents 1.44% of the projects. The combination of *display information* and *timer* represents 1.08% of the projects and includes apps with stopwatches for various activities, such as synchronizing interface events, for example, enabling a button after a few seconds. The combination of *access website* and *display information* appears in 1.06% of the projects and, in general, they include very simple apps for accessing one specific website without using more complex programming blocks.

Combinations of three functionalities. Five combinations with cardinality three were detected in more than 1% of the mobile apps (Figure 11). Three of them include *detect acceleration* for performing some action when changing a device orientation, for example, displaying some information. The *canvas* also appears three times in these combinations, for example, the combination of *paint*, *canvas,* and *timer* appears in 1.29% of the apps, and, in general, apps with that combination are similar to the game developed in the Space Invaders tutorial (MIT APP INVENTOR, 2022a). The combination of *canvas*, *detect acceleration*, and *paint* appears in 1.04% of the apps and includes projects with an area for drawing and actions through gestures, for example, erasing the drawing when shaking the smartphone.

Combinations of four functionalities. Only two combinations are composed of four functionalities and appear in 1% of the mobile apps (Figure 11). Both combinations include the functionality of *animation*, *canvas*, and *timer*. These three functionalities together are related to games, such as Mini Golf games (MIT APP INVENTOR, 2022d), in which the information refers to game instructions. Alternatively, they may have sound effects when the functionality *play sound* is included, and in this case, in general, this combination is similar to Mole Mash-type games, including the tutorial available on the App Inventor website (MIT APP INVENTOR, 2022e).

Combinations of five functionalities. Only the combination formed by five functionalities *animation*, *canvas*, *display information*, *play sound* and *timer* appear in more than 1% of the projects. In general, they are similar games or derivatives of the Mole Mash tutorial (MIT APP INVENTOR, 2022e) with some modifications.

Figure 12 – Frequency of projects by functionality combination cardinality



Source: Elaborated by the author.

More than 50% of the mobile apps have up to two functionalities and more than 75% have up to 4 functionalities. Less than 1% of apps have more than 8 functionalities. Considering that apps usually have some specific objective, this small number of functionalities per app is expected, given that an app with many functionalities may not have a clear focus. No app with more than 28 functionalities was identified (Figure 12).

Utilizing a reference universe of 100K mobile apps provides a substantial and diverse pool for measuring and assessing the originality of the functionalities of a new app. By drawing from such an extensive collection, it is possible to compare the functionalities of a new app with a wide range of existing apps, providing a robust basis for assessing originality. The large reference universe also increases the likelihood of encountering similar or related functionalities, allowing the identification of distinctive features and innovative approaches that differentiate the new app from others. Furthermore, the sheer scale of the reference universe enhances the statistical significance of the assessment, enabling more accurate measurements of originality by considering a substantial number of comparable apps.

Grading originality of functionalities. The originality of functionalities grade is divided into the originality of each of the functionalities and the originality of the combination of functionalities, which are calculated considering the set F of all functionalities that can be extracted from a mobile app (Table 10).

$$F = \{f \mid f \; has \; a \; rule \; of \; extraction\} \tag{1}$$

Originality of a functionality. The detection d of a functionality $f \in F$ in a mobile app a is 1 if the app contains the functionality, and 0 otherwise:

$$d_{f,a} = 1, if \; f \in a \tag{2}$$

$$d_{f,a} = 0, if \; f \notin a \tag{3}$$

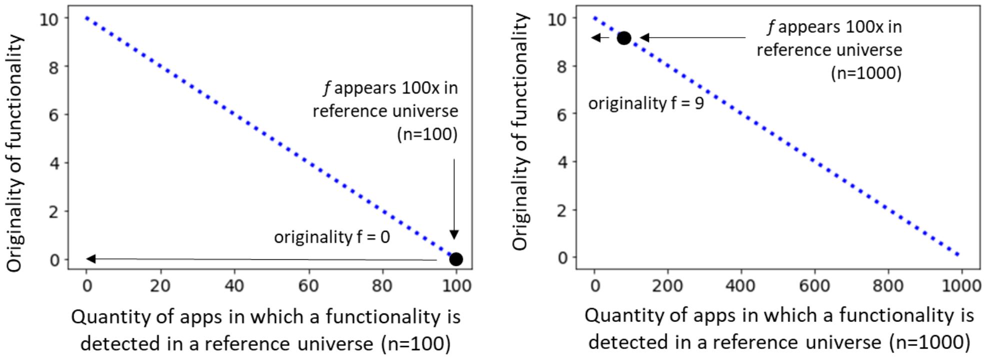
The originality of a functionality f depends on the number of mobile apps in which f is detected in the reference universe $U = \{a_1, a_2, \dots, a_n\}$. A reference universe must contain $n > 0$ apps, i.e., at least one app. The reference universe is composed of apps with at least one functionality and no duplicated apps. The multiplication by 10 is used solely for scaling the originality of the functionality on a scale [0, 10].

$$o_{f,U} = \left(1 - \frac{1}{n} \sum_{a \in U} d_{f,a}\right) * 10 \tag{4}$$

The more the functionality is detected in the apps of the reference universe, the less original it is due to the sum of these detections. For example, if a functionality f is detected in all apps of the reference universe, the sum is equal to n as $\sum_{a \in U} d_{f,a} = n, \; if \; d_{f,a} = 1 \; \forall a \in U$. In this case, the division $(\frac{1}{n} \sum_{a \in U} d_{f,a})$ equals 1, resulting in 1 – 1 = 0, i.e., zero originality for that functionality.

The originality of functionalities is also dependent on the reference universe, thus using reference universes with different mobile apps or a different number of apps, directly influences the value of the individual functionality (Figure 13). For example, if a functionality appears in 100 apps of a reference universe with 100 apps, it is considered unoriginal, i.e., $o_{f,U} = 0$. However, if a functionality appears in 100 apps of a reference universe with 1000 apps, it is considered original, i.e., $o_{f,U} = 9.0$.

Figure 13 – Possible originality values for a functionality appearing in hypothetical reference universes with different numbers of mobile apps



Source: Elaborated by the author.

Originality of the combination of functionalities. The originality of combinations of functionalities considers the set of combinations (without repetition) of functionalities cf that appear in the reference universe.

$$cf = \{f \mid f \ appears \ sole \ or \ together \ with \ other(s) \ f \ and \ f \in F\} \qquad (5)$$

The detection d of a combination of functionalities in a mobile app a is 1 if the app contains the combination of the functionalities, and 0 otherwise:

$$d_{cf,a} = 1, if \ cf \in a \qquad (6)$$

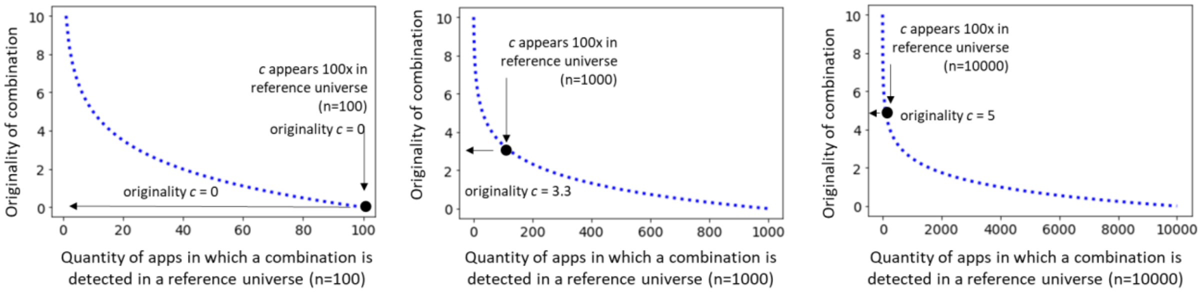
$$d_{cf,a} = 0, if \ cf \notin a \qquad (7)$$

The actual number of possible combinations is lesser than the theoretical number because many functionalities do not appear together with other functionalities. The originality of a combination of functionalities also depends on the number of apps in the reference universe $U = \{a_1, a_2, \ldots, a_n\}$ in which cf is detected. The more the combination is detected in the apps of the reference universe, the less original it is. Considering that a combination is rarer to detect than a functionality (Figure 11) as it depends on the detection of other functionalities as well, to compensate for this rarity the logarithm is used. The multiplication by 10 is used solely for scaling the originality on a scale [0, 10]. If there is no app in the universe with the combination, the originality of the combination is set to 10.0 without using formula 8, as the sum would be zero and the logarithm of zero is undefined.

$$o_{cf,U} = 10 - \left(log_n \sum_{a \in U} d_{cf,a} \right) * 10 \qquad (8)$$

Considering a hypothetical reference universe of 100,000 mobile apps, a combination could appear [0, 100,000] times. The originality value greatly decreases the more the combination appears (Figure 14). For example, if a combination appears in 100 apps of a reference universe with 100 apps, it is considered unoriginal, i.e., $o_{cf,U} = 0$. If the combination appears in 100 mobile apps of a reference universe with 1000 mobile apps, it is still considered not quite original, i.e., $o_{cf,U} = 3.3$. The 'logarithm penalty' is used because combinations are rarer to detect than a single functionality (Figure 11), without this penalty almost all combinations would be considered original even though they are not.

Figure 14 – Possible originality values for a combination appearing in hypothetical reference universes with different numbers of mobile apps



Source: Elaborated by the author.

Originality of functionalities of the mobile app. The overall originality of functionalities fs is calculated from the average of the originality of the m detected functionalities plus its combinations divided by 2.

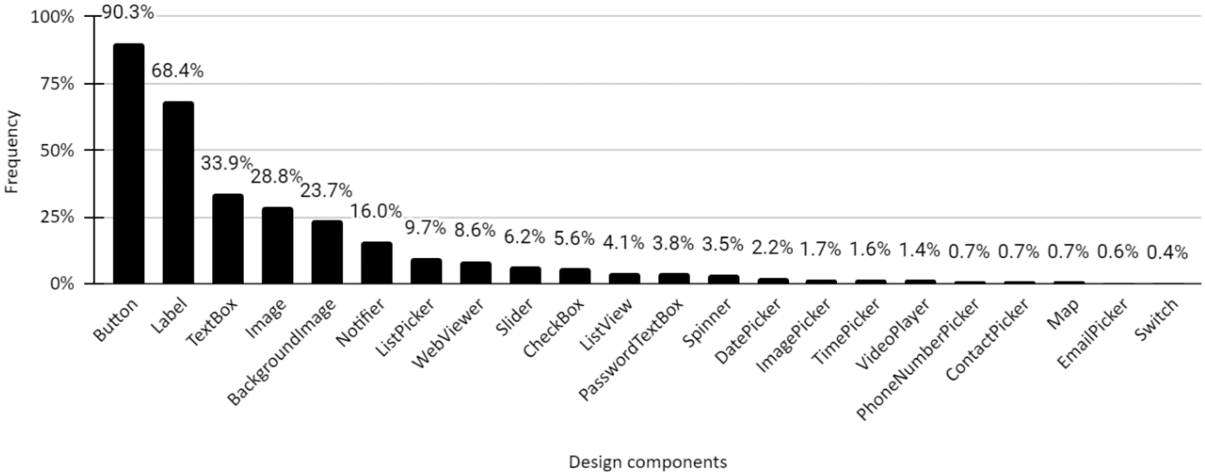
$$o_{fs,U}(a) = \frac{\frac{\sum_{f \in a} o_{f,U}}{m} + o_{cf,U}}{2} \tag{9}$$

4.3.1.2 Originality of user interface components

The conceptual assessment model of originality of user interface (UI) components refers to the visual design components of the app. The visual design components consist of components that are visible on the screen and enable specific actions to be carried out by the user (Table 9). The UI of apps differs from the UIs of other devices because of the limited screen space, interaction with these devices, and the context in which they are used (WASSERMAN, 2010), thus they are only comparable with the UI of other apps. In addition, although App Inventor also has hidden components, those are not considered part of UI because they are not visible to the user.

Most apps created with App Inventor include only a few UI components (Figure 15). An analysis of 102,567 apps from the App Inventor Gallery shows that the most commonly used UI components are *buttons* (90.3%), followed by *labels* (68.4%) and *text boxes* (33.9%). Among the less commonly used UI components are *email picker* (0.6%) and *switch* (0.4%) – a recent component included in App Inventor.

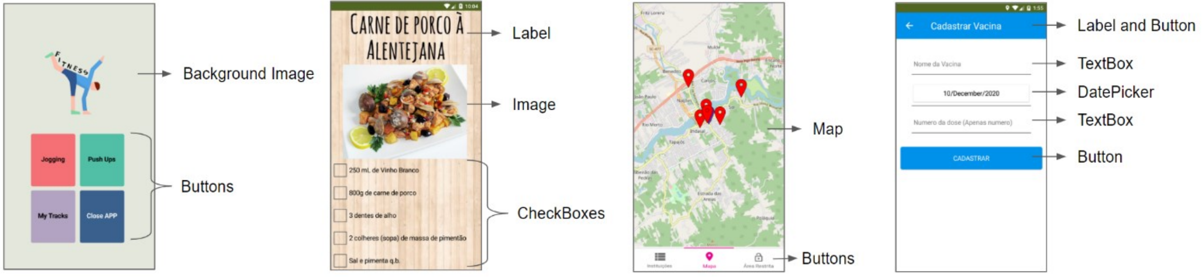
Figure 15 –UI components in mobile apps (n=102,567) from the App Inventor Gallery



Source: Elaborated by the author.

The prevalence of buttons is due to the traditional human-computer interaction in mobile apps in which almost all actions are triggered when the user clicks on a button on the interface (Figure 16). Labels are also useful for displaying information on different parts of the screen. Text boxes are used for text entries and are very common in login or forms-like screens. Images and background images are part of the visual language of an interface and aim to maximize usability and user experience.

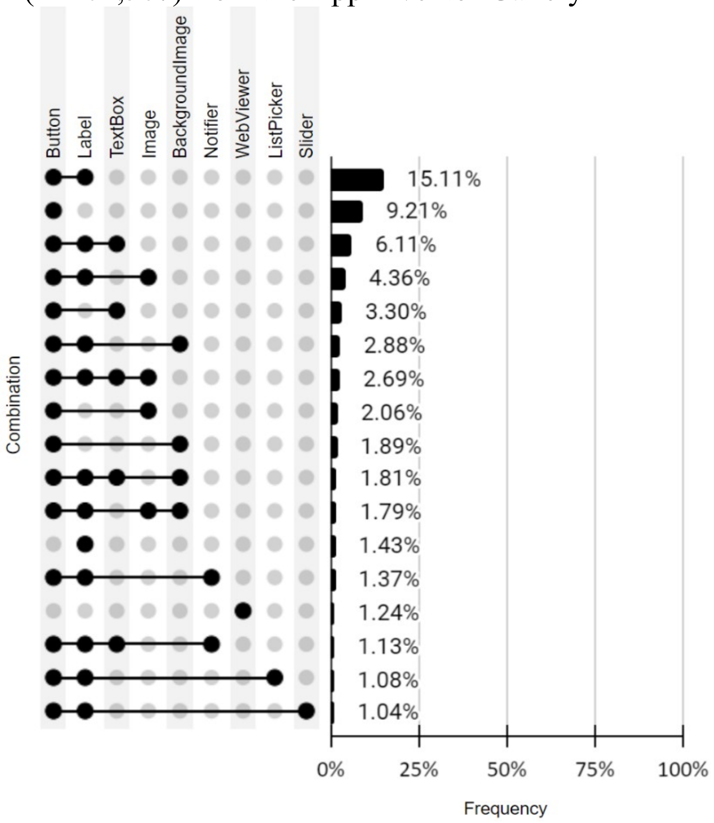
Figure 16 – Screens examples with design components



Source: Elaborated by the author based on projects of the App Inventor Gallery.

Few combinations of UI components are predominant, only 17 combinations appear in more than 1% of the projects (Figure 17). The predominant combinations (>1% of the projects) are composed of one to four UI components. Most combinations appear in a few projects, for example, the combination of *button* and *label* is the most common and appears in only 15.11% of the projects or in combination with other UI components.

Figure 17 – Combination of UI components that appear in more than 1% of mobile apps (n=102,567) from the App Inventor Gallery



Source: Elaborated by the author.

Originality of each UI component. The detection d of a UI component $c \in C$ in a mobile app a is 1 if the app contains the UI component, and 0 otherwise:

$$d_{c,a} = 1, if\ c \in a \tag{10}$$

$$d_{c,a} = 0, if\ c \notin a \tag{11}$$

The originality of a UI component c depends on the number of mobile apps in which c is detected in the reference universe $U = \{a_1, a_2, \dots, a_n\}$. A reference universe must contain $n > 0$ apps. The more the UI component is detected in the apps of the reference universe, the less original it is due to the sum of these detections.

$$o_{c,U} = \left(1 - \frac{1}{n}\sum_{a \in U} d_{c,a}\right) * 10 \tag{12}$$

Originality of the combination of UI components. The originality of combinations of UI components considers the set of combinations (without repetition) of UI components cc that appear in the reference universe.

$$cc = \{c \mid c\ appears\ sole\ or\ together\ with\ other(s)\ c\ and\ c \in C\} \tag{13}$$

The detection d of a combination of UI components in a mobile app a is 1 if the app contains the combination of the UI components, and 0 otherwise:

$$d_{cc,a} = 1, if\ cc \in a \tag{14}$$

$$d_{cc,a} = 0, if\ cc \notin a \tag{15}$$

The originality of a combination of UI components c also depends on the number of mobile apps in the reference universe $U = \{a_1, a_2, \dots, a_n\}$ in which cc is detected. The more the combination is detected in the apps of the reference universe, the less original it is. As in the originality of the combination of functionalities, the 'logarithm penalty' is also used here because combinations are rarer to detect than a single UI component (Figure 17).

$$o_{cc,U} = 10 - \left(log_n \sum_{a \in U} d_{cc,a} \right) * 10 \tag{16}$$

Originality of UI components of the mobile app. The overall originality grade of UI components cs is calculated from the average of the originality of the m extracted UI components plus its combinations divided by 2.

$$o_{cs,U}(a) = \frac{\frac{\sum_{c \in a} o_{c,U}}{m} + o_{cc,U}}{2} \tag{17}$$

4.3.1.3 Originality of topic

The conceptual assessment model of originality of topic refers to the most related topic of the mobile app from a set T of 21 topics (as defined in Table 11). Each topic is focused on a specific type of app and if an app seems to have two or more topics, it is assigned with the most related one.

Textual content extraction. The topic is assigned based on the textual content of the mobile app. Considering the input of the assessment model in the AIA file, the textual content that can be extracted from the App Inventor project AIA file structure (Figure 5) includes properties from the app's screens, as well as, user interface, media, maps, social and connectivity properties (Table 13). The content in textual blocks is also extracted from the BKY files for each screen.

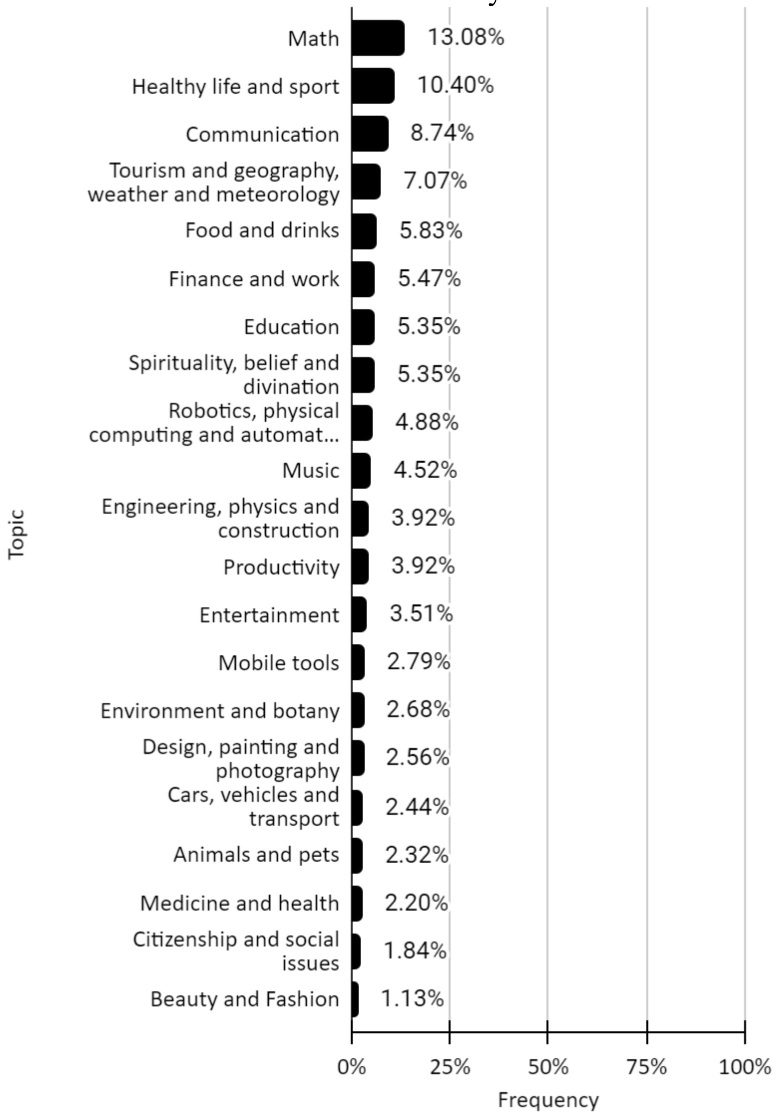
Table 13 – Textual content of mobile apps

File	**Type**	**Category**	**Component / Block**	**Property**
SCM	Screen	Screen	Screen	Title
				AboutScreen
	Designer	User Interface	Button	Text
			Checkbox	Text
			DatePicker	Text
			Label	Text
			ListPicker	Text and ElementsFromString
			PasswordTextBox	Text and Hint
			Spinner	ElementsFromString and Prompt
			Switch	Text
			TextBox	Text and Hint
			TimePicker	Text
		Media	ImagePicker	Text
		Maps	Marker	Description
		Social	ContactPicker	Text
			EmailPicker	Text
			PhoneNumberPicker	Text
			Texting	Message
		Connectivity	ActivityStarter	ResultName
BKY	Blocks	Text	String	Text

Source: Elaborated by the author.

An analysis of 1,682 mobile apps in Portuguese from the App Inventor Gallery shows that most apps in Portuguese are related to math, for example, providing information or performing math calculations, such as algebraic, trigonometric, and geometric calculations, or are related to healthy life or sport, including training, diet, nutrition, stress management, physical conditioning, and weight indicators. Among the less commonly identified topics are beauty and fashion; citizenship and social issues (Figure 18).

Figure 18 – Topics frequency in mobile apps in Portuguese (n=1,682) from the App Inventor Gallery



Source: Elaborated by the author.

Assignment of a topic. In a previous step, a machine learning model using the Complement Naive Bayes (CNB) classifier (RENNIE, SHIH, *et al.*, 2003), which is suited for imbalanced data sets, was trained and tested using five folds. The input for the model was the textual content of 1,682 App Inventor projects in Portuguese and the topic assigned to these apps by the author and reviewed by the supervisor. The average accuracy of 77.76, precision of 0.80, recall of 0.78, and f1 score of 0.77 were obtained in cross-validation. Compared to other models, this model presented the best results and was therefore adopted.

To assign a topic t to a mobile app a, the trained CNB classifier considers that the textual content of the app is in Portuguese. Considering that the raw textual content of an app may contain punctuation, numbers, or escape characters, such as \n, \t, etc. those are removed and not considered. In addition, all words are converted into lowercase lexical tokens and represented as bag-of-words b considering only words of the pre-trained machine learning model. Considering the bag-of-words representation of an app a_b, the topic is obtained using the class prior and parameter estimation output from the trained CNB model.

$$t_a = CNB(a_b) \tag{18}$$

The detection of a topic $t \in T$ (Table 11) of a mobile app a is 1 if the CNB assigned the topic t to the app a, and 0 otherwise:

$$d_{t,a} = 1, if\ t_a = t \tag{19}$$

$$d_{t,a} = 0, if\ t_a \neq t \tag{20}$$

Originality of a topic of a mobile app. The originality of a topic t depends on the number of apps in which t is detected in the reference universe $U = \{a_1, a_2, \dots, a_n\}$. A reference universe must contain $n > 0$ apps. The more the topic is detected in the apps of the reference universe, the less original it is. The originality of a topic in an app a is inversely proportional to the topic frequency in the reference universe U.

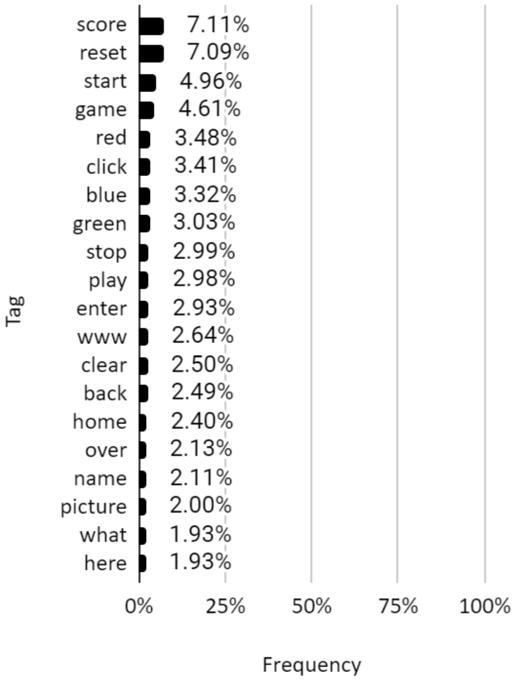
$$o_{t,U}(a) = \left(1 - \frac{1}{n}\sum_{a \in U} d_{t,a}\right) * 10 \tag{21}$$

4.3.1.4 Originality of tags

The conceptual assessment model of the originality of tags refers to the originality of keywords extracted from the textual content of a mobile app (Table 13). Tag extraction is language-independent, thus, the language of the textual content of the app can be in any language.

An analysis of tags extracted from 99,411 mobile apps from the App Inventor Gallery shows that the most common tags are in English because the majority of apps are in English. There are many apps created with App Inventor that are games, and this is reflected in the most common tags such as ‘score’, ‘reset’, ‘start’, and ‘game’ (Figure 19).

Figure 19 – 20 most frequency extracted keywords in mobile apps (n= 99,411) from the App Inventor Gallery



Source: Elaborated by the author.

Extraction of tags. The extraction of a tag k in a mobile app a is based on non-deterministic text statistical features as proposed by the YAKE model (CAMPOS, MANGARAVITE, *et al.*, 2020) with a maximum of 10 tags. As in the originality of the topic, the raw textual content of an app is pre-processed and words are converted into lowercase lexical tokens. In addition, a set of stop words for apps, such as 'button', 'screen', etc. is removed from the textual content resulting in a preprocessed textual content a_p of the app

$$k_a = YAKE(a_p) \tag{22}$$

The detection d of a tag k in an app a is 1 if the YAKE model extracted the tag k from the app a, and 0 otherwise:

$$d_{k,a} = 1, if\ k \in k_a \tag{23}$$

$$d_{k,a} = 0, if\ k \notin k_a \tag{24}$$

Originality of a tag. The originality of a tag k depends on the number of mobile apps in which k is extracted in the reference universe $U = \{a_1, a_2, \ldots, a_n\}$. A reference universe must contain $n > 0$ apps. The more the tag is extracted in the apps of the reference universe, the less original it is. Consider Uh as the number of the apps that contain the most frequent tag extracted from the reference universe U. Here, the ‘logarithm penalty’ is also used because most of the tags are extremely rare to detect, i.e., they are extracted in less than 1% of the apps (Figure 19).

$$o_{k,U} = \left(1 - log_{Uh} \sum_{a \in U} d_{k,a}\right) * 10 \tag{25}$$

Originality of tags. The overall originality grade of tags (ks) is calculated from the average of the originality of the z extracted tags of the mobile app.

$$o_{ks,U}(a) = \frac{\sum_{k \in a} o_{k,U}}{z} \tag{26}$$

4.3.2 Conceptual assessment model of flexibility

The measurement of flexibility refers to assessing the diversity of the categories of aspects incorporated within the mobile app. The aspects are components, programming, and functionalities. The categories of components and programming follow the categorization of App Inventor as presented in section 4.2.2.1. The categories of functionalities are related to each of the functionalities extracted from the app according to the rule-based system as presented in 4.2.2.1. Each rule for extracting a functionality is considered a category of functionality.

4.3.2.1 Flexibility of components

The conceptual assessment model of the flexibility of components refers to the number of different categories of components out of all 12 possible categories of components (see Table 1) used in the mobile app. The categories range from more generic components of the user interface, layout, and media, to very specific ones, such as LEGO® MINDSTORMS®.

Flexibility of a category of a component. The flexibility considers the set of components of the category (cs) that can be extracted from a mobile app. The flexibility of a category is 1 if the app contains any component c of that category and 0 otherwise.

$$f_{cs,a} = 1, if\ c \in a\ \wedge\ c \in cs \tag{27}$$

$$f_{cs,a} = 0, if\ c \notin a\ \wedge\ c \in cs \tag{28}$$

Flexibility of components. The flexibility of components (css) of a mobile app is the average of all flexibility of the total categories of components. Overall, there are 12 different categories of components (Table 1).

$$f_{css}(a) = \left(\frac{1}{12} \sum_{cs \in CS} f_{cs,a}\right) * 10 \tag{29}$$

4.3.2.2 Flexibility of programming

The conceptual assessment model of the flexibility of programming refers to the number of different categories of programming blocks used in the mobile app. The categories of programming blocks can be part of the built-in blocks, component blocks, screen blocks, helpers, or extensions (Table 2). Built-in blocks are available for use in any mobile app and refer to overall programming concepts, such as variables, conditionals, loops, procedures, logical and math operators, etc. Component blocks include events, set, get, call methods, and component object blocks that are available for specific design components added to the app.

Flexibility of a programming block category. The flexibility of a programming block category considers the set of programming blocks for the category (PS) that can be extracted from an app. The flexibility of a programming block category is 1 if the app contains any programming blocks (b) that are from the category and 0 otherwise.

$$f_{ps,a} = 1, if\ b \in a\ \land\ b \in ps \tag{30}$$

$$f_{ps,a} = 0, if\ b \notin a\ \land\ b \in ps \tag{31}$$

Flexibility of programming blocks. The flexibility of programming blocks (pss) of a mobile app is the average of the total flexibility of programming block categories. Overall, there are 24 different categories of programming.

$$f_{pss}(a) = \left(\frac{1}{24} \sum_{ps \in PS} f_{ps,a}\right) * 10 \tag{32}$$

4.3.2.3 Flexibility of functionalities

The conceptual assessment model of the flexibility of functionalities refers to the number of different functionalities that are detected in the app. Here the functionalities are detected through a rule-base system (as presented in Table 10).

Flexibility of functionalities. The flexibility of functionalities considers the set of all different functionalities that can be extracted from an app. Overall, there are 42 different functionalities (Table 10) and the detection of functionality in an app ($d_{f,a}$) is 1 if the app contains the functionality and 0 otherwise. The flexibility of functionalities (fs) of an app is the average of all detected functionalities.

$$f_{fs}(a) = \left(\frac{1}{42} \sum_{f \in F} d_{f,a} \right) * 10 \tag{33}$$

4.3.3 Conceptual assessment model of fluency

Fluency, when understood as the ability to generate a large number of ideas as opposed to generating no ideas, is a crucial component of creativity. Considering that the measurement is upon a single artifact, the conceptual model of fluency encompasses the number of components and programming blocks used when compared to an empty mobile app. A higher number of components reflects a greater potential for creative apps. Programming blocks are the fundamental building blocks that enable functionality in the app, and a larger number of blocks suggests a more extensive use of App Inventor's capabilities and a higher level of fluency, with the assumption that this helps to build more creative apps.

4.3.3.1 Fluency of components

The conceptual model of fluency of components refers to the sum of the overall number of components used in the mobile app. An analysis of 99,993 unique apps from the App Inventor Gallery shows that half the apps have 12 components or more (Table 14). The first quartile (Q1) indicates that 25% of the apps have 6 or fewer components and the third quartile (Q3) indicates that 25% of the apps have 24 or more components.

Fluency of a component. The fluency l of a component c refers to the overall count that the component appears in the app a.

$$l_{c,a} = |\{c \mid c \in a\}| \tag{34}$$

Fluency of components. The fluency of all components refers to the frequency distribution of the overall count of all the components that appear in the mobile app.

$$l_{cs}(a) = P\left(\sum_{c \in a} l_{c,a} \right) / 10 \tag{35}$$

Table 14 – Distribution of components in mobile apps (n= 99,993) from the App Inventor Gallery

Percentage	Number of components
5%	2
10%	3
15%	4
20%	5
25% (Q1)	6
30%	7
35%	8
40%	9
45%	10
50% (Median)	12
55%	13
60%	15
65%	17
70%	20
75% (Q3)	24
80%	30
85%	39
90%	54
95%	89
100%	3244

Source: Elaborated by the author.

P indicates the percentage in which the value of the overall count of components is equal to or above the number of components found in the reference universe. For example, if 12 components are detected in an app, the fluency of components is 5.0, since at least 50% of the apps have 12 or more components (Table 14). In the same sense, if 14 components are detected in an app, the fluency of components is 5.5, since at least 55% of the apps have 14 or more components. Considering that the number of components for the percentage of 100% is inadequate to use for grade calculation, the cut-off value has been decided in agreement with creativity and computing researchers to be 115, i.e., if 115+ components are detected in an app, the fluency of components is 10.

4.3.3.2 Fluency of programming

The conceptual model of fluency of programming refers to the overall number of programming blocks used in the mobile app. An analysis of 99,993 unique apps from the App Inventor Gallery shows that at least half the apps have at least 39 programming blocks (Table 15). The first quartile (Q1) indicates that 25% of the apps have 11 or fewer programming blocks and the third quartile (Q3) indicates that 25% of the apps have 104 or more programming blocks.

Table 15 – Distribution of programming blocks in mobile apps (n= 99,993) from the App Inventor Gallery

Percentage	Number of programming blocks
5%	0
10%	2
15%	5
20%	8
25% (Q1)	11
30%	15
35%	21
40%	27
45%	33
50% (Median)	39
55%	48
60%	57
65%	69
70%	84
75% (Q3)	104
80%	134
85%	178
90%	259
95%	462
100%	52621

Source: Elaborated by the author.

Fluency of a programming block. The fluency l of a programming block b refers to the overall count that the block appears in the app a.

$$l_{b,a} = |\{b \mid b \in a\}| \tag{36}$$

Fluency of programming blocks. The fluency of all programming blocks refers to the frequency distribution of the overall count of all the programming blocks that appear in the app.

$$l_{bs}(a) = P\left(\sum_{b\, \in\, a} l_{b,a}\right)/10 \tag{37}$$

P indicates the percentage of values below or equal to which a certain percentage of the components is found. For example, if 39 programming blocks are detected in a mobile app, the fluency of programming blocks is 5.0, since at least 50% of the apps have 39 or more programming blocks (Table 15). In the same sense, if 45 components are detected in an app, the fluency of programming blocks is 5.5, since at least 55% of the apps have 45 or more components. Considering that the number of programming blocks for the percentage of 100%

is inadequate to use for grade calculation, the cut-off value has been decided in agreement with creativity and computing researchers to be 650, i.e., if 650+ programming blocks are detected in an app, the fluency of programming blocks is 10.

4.3.4 Creativity final grade

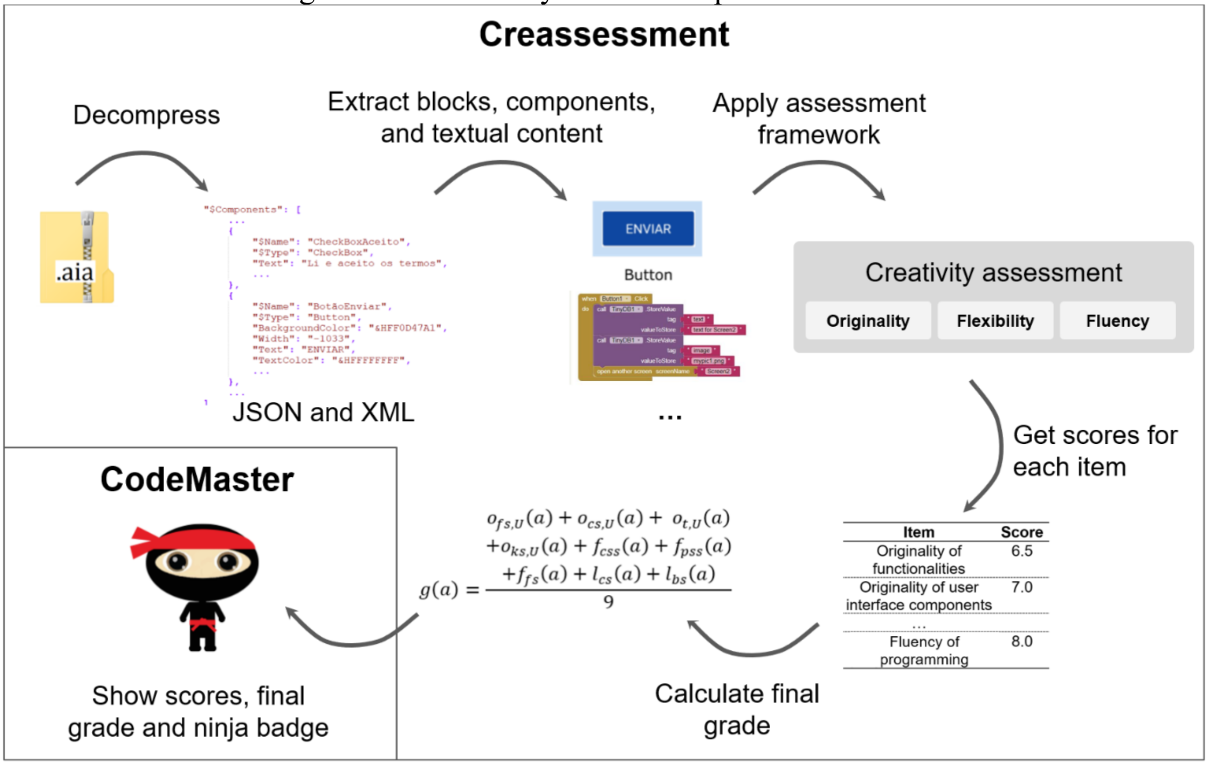
The final grade, denoted as g, for a mobile app, is determined by calculating the mean of all items within the assessment framework. While this approach may not necessarily be the optimal choice, it was selected for the sake of simplicity and ease of implementation. By averaging the scores across various assessment items, the final grade attempts to provide a comprehensive and overall assessment of the creativity of the app. However, it is important to acknowledge that this method does not account for potential variations in the significance or weighting of different assessment criteria. Depending on the specific context and desired outcomes, alternative grading approaches, such as weighted grading or prioritizing specific criteria, could offer a more nuanced and tailored evaluation of the creative attributes of apps. Nevertheless, the utilization of a mean-based grading system offers a straightforward and accessible method for summarizing the creativity of apps, facilitating ease of interpretation and communication of the final grade. Considering a [0, 10] scale, the grade is rounded to the nearest integer or the nearest 0.5.

$$g(a) = \frac{\begin{array}{c} o_{fs,U}(a) + o_{cs,U}(a) + o_{t,U}(a) \\ + o_{ks,U}(a) + f_{css}(a) + f_{pss}(a) \\ + f_{fs}(a) + l_{cs}(a) + l_{bs}(a) \end{array}}{9} \tag{38}$$

4.4 TECHNICAL IMPLEMENTATION OF THE ASSESSMENT MODEL

The conceptual assessment model was implemented by developing the Creassessment (Creativity Assessment) software module. Creassessment automatically parses, analyzes, and grades App Inventor projects through static code analysis (ALVES, 2023). The module is integrated into the CodeMaster tool with a graphical user interface (Figure 20).

Figure 20 – Creativity assessment process overview



```
"$Components": [
    ...
    {
        "$Name": "CheckBoxAceito",
        "$Type": "CheckBox",
        "Text": "Li e aceito os termos",
        ...
    },
    {
        "$Name": "BotãoEnviar",
        "$Type": "Button",
        "BackgroundColor": "&HFF0D47A1",
        "Width": "-1033",
        "Text": "ENVIAR",
        "TextColor": "&HFFFFFFFF",
        ...
    },
    ...
]
```

$$g(a) = \frac{o_{fs,U}(a) + o_{cs,U}(a) + o_{t,U}(a) + o_{ks,U}(a) + f_{css}(a) + f_{pss}(a) + f_{fs}(a) + l_{cs}(a) + l_{bs}(a)}{9}$$

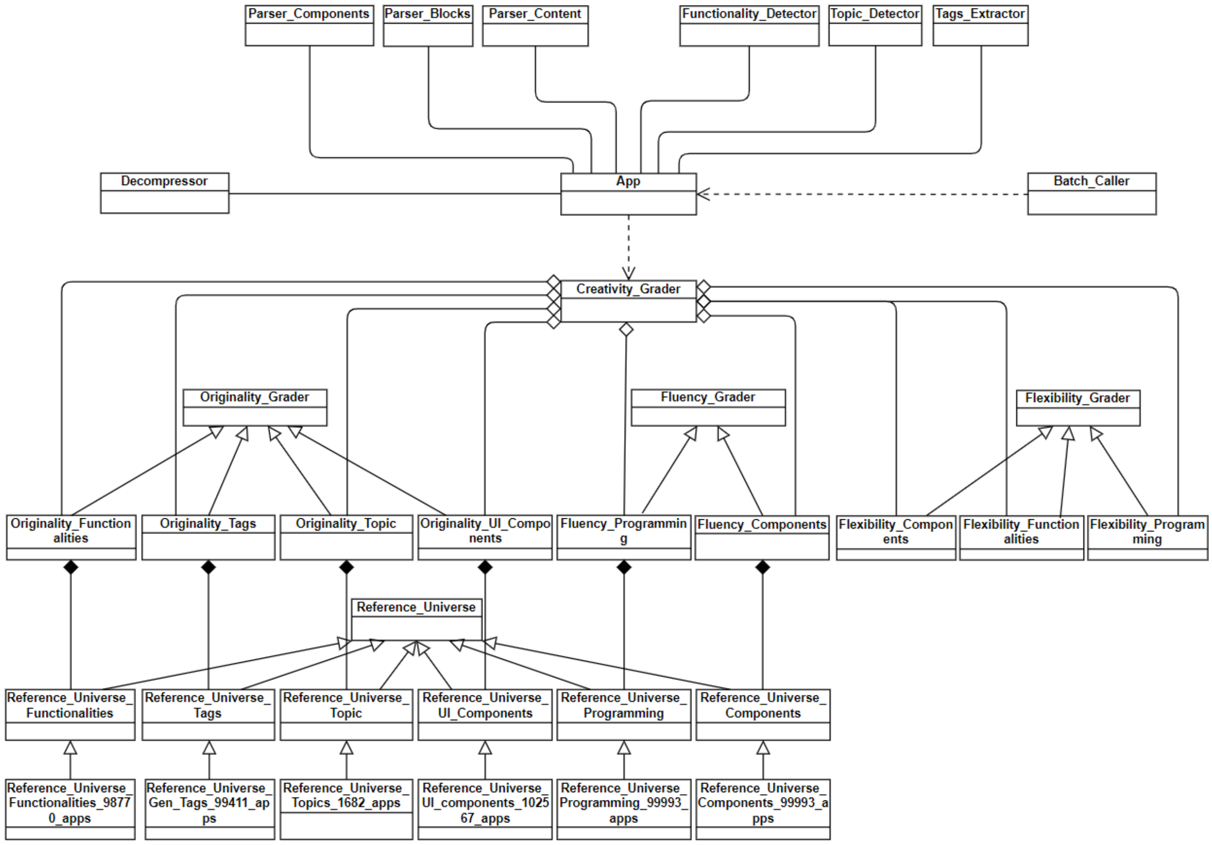
Item	Score
Originality of functionalities	6.5
Originality of user interface components	7.0
...	
Fluency of programming	8.0

Source: Elaborated by the author.

4.4.1 Creassessment module

Creassessment was designed to be extensible, as it can receive different reference universes or new layers of creativity assessment, allowing new dimensions to be added or change the existing ones in a low-effort way. Creassessment can also be used to extract data from mobile apps, for example, all the textual content or programming blocks used in the app.

Figure 21 – Creassessment: simplified class diagram


Parser_Components
Parser_Blocks
Parser_Content
Functionality_Detector
Topic_Detector
Tags_Extractor
Decompressor
App
Batch_Caller
Creativity_Grader
Originality_Grader
Fluency_Grader
Flexibility_Grader
Originality_Function alities
Originality_Tags
Originality_Topic
Originality_UI_Compo nents
Fluency_Programmin g
Fluency_Components
Flexibility_Compon ents
Flexibility_Function alities
Flexibility_Program ming
Reference_Universe
Reference_Universe_ Functionalities
Reference_Universe_ Tags
Reference_Universe_ Topic
Reference_Universe_ UI_Components
Reference_Universe_ Programming
Reference_Universe_ Components
Reference_Universe_ Functionalities_9877 0_apps
Reference_Universe_ Gen_Tags_99411_ap ps
Reference_Universe_ Topics_1682_apps
Reference_Universe_ UI_components_1025 67_apps
Reference_Universe_ Programming_99993_ apps
Reference_Universe_ Components_99993_a pps

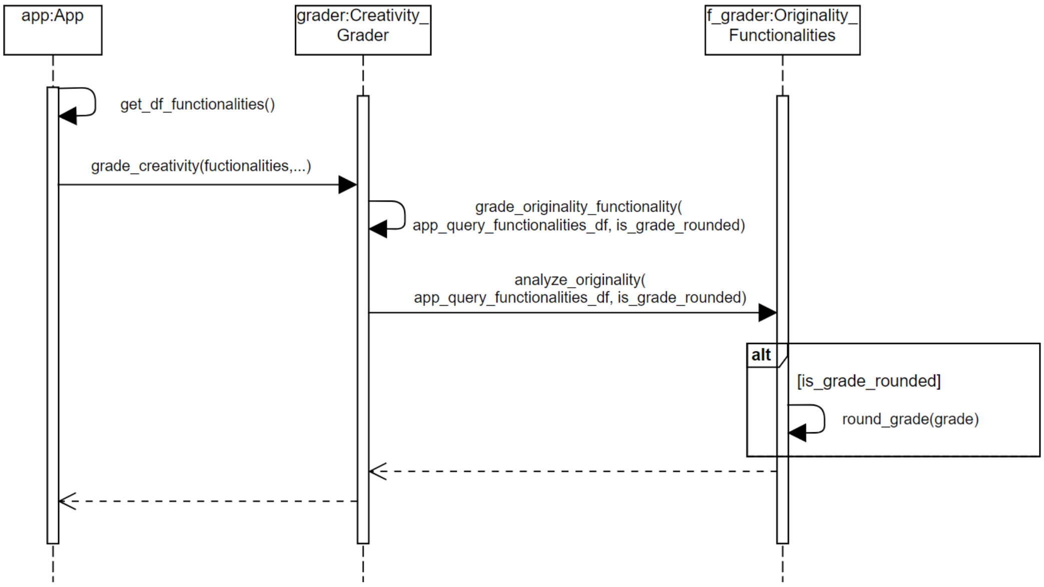
Source: Elaborated by the author.

The App class receives the project and coordinates the execution of the analysis, using decompressors, parsers, detectors, extractors, and the `Creativity_Grader` class (Figure 21). The `Decompressor` class is used for decompressing AIA files (App Inventor project files), composed of SCM files (JSON files) and BKY files (XML files). For each different type of raw data that can be extracted from the mobile app, there is a parser. `Parser_Components` can analyze JSON file content and identify which and how many components are used in the mobile app, `Parser_Blocks` can analyze the XML file content and identify which and how many programming blocks are used in the mobile app, and `Parser_Content` can extract all textual content from a mobile app.

The `Creativity_Grader` class contains a grader for each of the nine assessment criteria. The assessment criteria share similar behavior for the dimensions of originality, fluency, and flexibility; thus, they are implemented as subclasses of `Originality_Grader`, `Fluency_Grader`, and `Flexibility_Grader`. The criteria related to the graders for originality and fluency depend on a reference universe, thus they are composed of its related reference universe, for example, `Originality_Functionalities` is composed of

`Reference_Universe_Functionalities`. To improve performance, the reference universes can be specialized considering a fixed number of mobile apps, such as `Reference_Universe_Functionalities_98770_apps`. The `Batch_Caller` class is used for assessing or extracting data from many mobile apps at once.

Figure 22 – Simplified sequence diagram for grade_creativity method considering functionality only


app:App
grader:Creativity_ Grader
f_grader:Originality_ Functionalities
get_df_functionalities()
grade_creativity(fuctionalities,...)
grade_originality_functionality(app_query_functionalities_df, is_grade_rounded)
analyze_originality(app_query_functionalities_df, is_grade_rounded)
alt
[is_grade_rounded]
round_grade(grade)

Source: Elaborated by the author.

When an App object is created all its raw data is extracted. Raw data refers to the data that can be extracted directly from XML and JSON files using `Parser_Components`, `Parser_Blocks`, and `Parser_Content`. The method `grade_creativity` from the App class gets all the necessary information from the mobile app and uses a grader object from `Creativity_Grader` to perform the assessment for each criterion, such as functionality (Figure 22). The assessment criterion requires parsed data for detecting or extracting more information, for example, `Functionality_Detector` requires data on the components, blocks, and textual content extracted from the mobile app to identify which functionalities are in the mobile app (Table 16).

The module has been implemented in Python using the libraries from the Python Standard Library, including `math`, `re`, `os`, `zipfile`, `pathlib`, `path`, `json`, and `xml`. In addition, Python packages, including `pandas` 1.3.5, `nltk` 3.7, `yake` 0.4.8, and `sklearn` 1.0.2 were used. The software module has also been packaged as a pip module (ALVES, 2023).

Table 16 – Extraction and scoring techniques adopted

Dimension	Item	Extraction/identification technique	Scoring technique
Originality	Functionalities	Rule-based extraction (ALVES and GRESSE VON WANGENHEIM, 2023) from XML and JSON	Single and combined frequency
	UI components	Extraction from JSON	Single and combined frequency
	Topic	Natural language processing using Machine Learning classification model Complement Naive Bayes (RENNIE, SHIH, *et al.*, 2003) on text extracted from XML and JSON	Single frequency
	Tags	Natural language processing using the Yake keyword extractor (CAMPOS, MANGARAVITE, *et al.*, 2020) from text extracted from XML and JSON	Single frequency
Fluency	Components	Counting the number and relative percentage of components from JSON	Single frequency
	Programming	Counting the number and relative percentage of programming blocks from XML	Single frequency
Flexibility	Components	Counting the number of different categories of components from JSON	Count
	Programming	Counting the number of different categories of programming blocks from XML	Count
	Functionalities	Counting the number of different categories of functionalities extracted from XML and JSON using rule-based definitions	Count

Source: Elaborated by the author.

4.4.2 Integration into CodeMaster

Aiming at using the Creassessment module in the context of computing education by teaching mobile app development with App Inventor, the model is integrated into the CodeMaster tool with a graphical user interface. The functional requirements refer to the new functionalities integrated into CodeMaster (Table 17).

Table 17 – Functional requirements

FQ	Requirement	Description	Input	Output
#1	Assess the degree of creativity of a mobile app created with App Inventor	The tool must assess the degree of creativity and its dimensions of originality, flexibility, and fluency of the App Inventor project	AIA file, with a maximum size of 15 MB	JSON containing grades from 0 to 10 for each criterion representing the degree of creativity of the mobile app and the final grade for creativity
#2	Maintain a record of the scores	The tool must keep a record of an assessment made by a user in its database	JSON creativity grades	The system stores in its database the assessment of the project submitted by a user
#2	Present the assessment	The tool must present a user interface with detailed information on the assessment grades of the App Inventor project	JSON creativity grades	Graphical user interface displaying assessment grades for each criterion, the final grade, and the competency level.

Source: Elaborated by the author.

The non-functional requirements refer to the maintenance and technologies involved and were defined based on the non-functional requirements of CodeMaster 2.0 (SCHMITT, 2022) to maintain homogeneity with the existing system (Table 18).

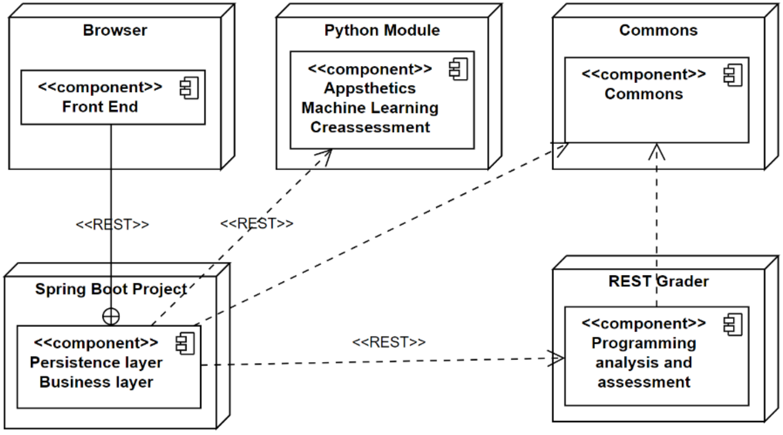
Table 18 – Non-functional requirements

NF	Requirement	Description
#1	Java 8.0 Programming Language	The inclusion of the new assessment model in CodeMaster must be implemented with the Java 8.0 programming language.
#2	Framework Angular 12 for front-end	The front-end of the tool must be developed in Typescript, using the Angular framework.
#3	Framework Spring Boot for back-end	The system rules engine, including data persistence, business rules, communication with the frontend, and with RESTGrader must be developed using Spring Boot framework.
#4	Python 3.7.2 Programming Language	The inclusion of the new assessment model in CodeMaster must be carried out with Python 3.7.2. The connection should be through a mini-server.
#5	REST Communication	Each part of the system (front, back, and assessment) must communicate through REST APIs.
#6	CnE visual identity	The user interface must have visual identity standards defined by the Computing at School Initiative.
#7	Web System	The tool must be accessed via a web browser with an internet connection.
#8	MySQL 5.5 and TomCat 8 for database	The database must use MySQL 5.5 and TomCat 8.

Source: Elaborated by the author.

The Creassessment module is included in the Python module, which contains a Python mini-server using the socket package to make a direct connection with the Spring Boot Project. The browser module consists of the user's Internet browser, where the graphical interface is displayed. The Creassessment unzips the file, analyzes it, assesses the degree of creativity, and returns the score obtained as a JSON to the Spring Boot Project module. The Spring Boot module saves the results to the database and the Front End displays the results on the user's Internet browser (Figure 23).

Figure 23 – CodeMaster architecture



Source: Elaborated by the author.

The server listens on port 9999 and establishes a connection upon receiving a request from the Spring Boot Project with an AIA file (Figure 24).

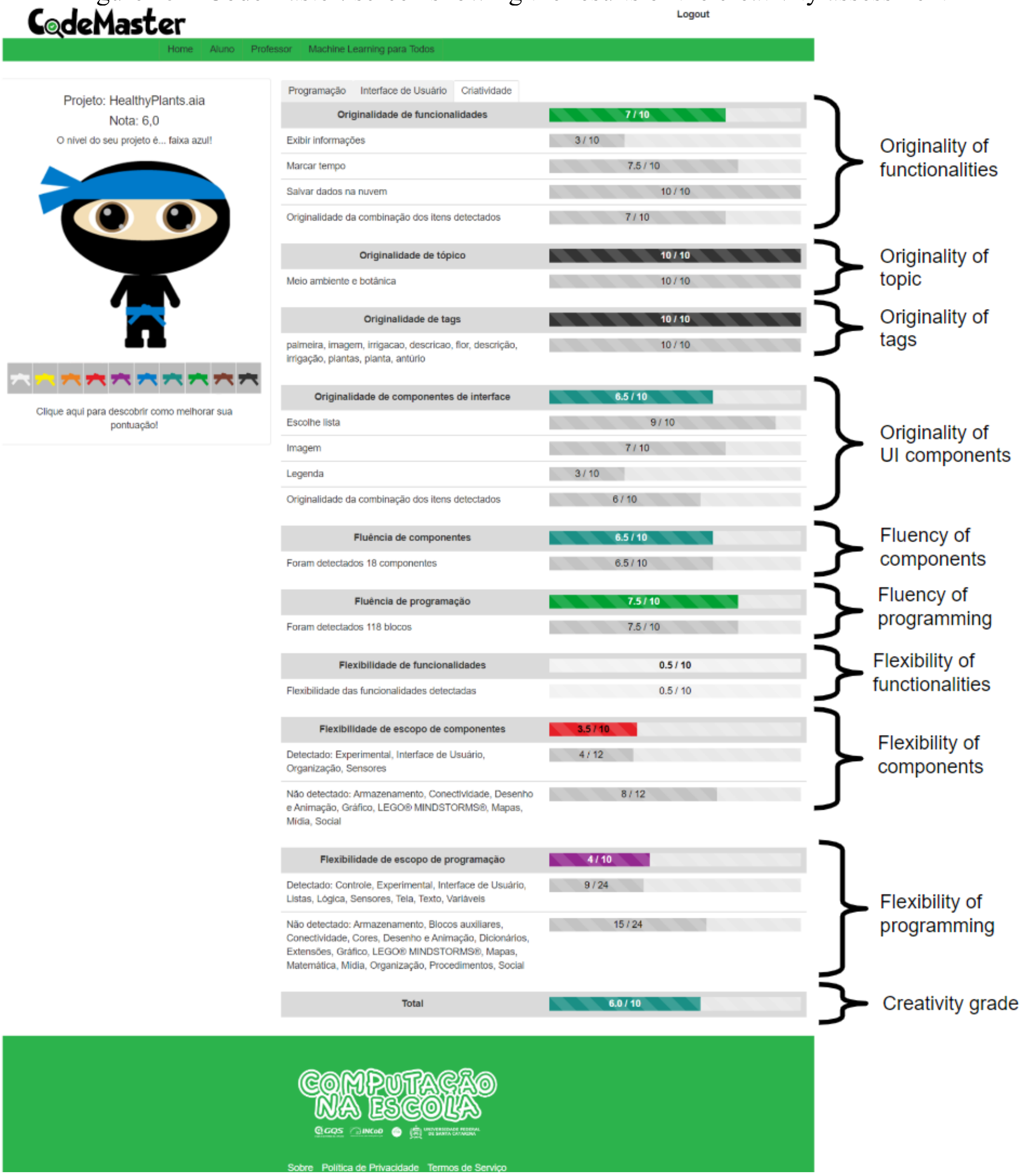
Figure 24 – Creativity assessment route

```
@app.route("/creassessment", methods=['POST'])
def creassesment() -> json:
    aia_file = request.files['file']
    app = App(aia_file)
    grader = Creativity_Grader()
    response = app.grade_creativity_to_json_wrapper(grader, is_grade_rounded=True)
    return response
```

Source: Elaborated by the author.

The CodeMaster interface is available in Brazilian Portuguese. The results for the creativity scores are shown in the *Criatividade* (Creativity) tab. Each creativity score for originality, fluency, and flexibility is shown separately, as well as the overall creativity grade (Figure 25).

Figure 25 – CodeMaster: screen showing the results of the creativity assessment


CodeMaster
Logout
Home
Aluno
Professor
Machine Learning para Todos
Originality of functionalities
Originality of topic
Originality of tags
Originality of UI components
Fluency of components
Fluency of programming
Flexibility of functionalities
Flexibility of components
Flexibility of programming
Creativity grade
COMPUTAÇÃO NA ESCOLA

Projeto: HealthyPlants.aia

Nota: 6,0

O nível do seu projeto é... faixa azul!

Clique aqui para descobrir como melhorar sua pontuação!

Programação | Interface de Usuário | Criatividade

Originalidade de funcionalidades	**7 / 10**
Exibir informações	3 / 10
Marcar tempo	7.5 / 10
Salvar dados na nuvem	10 / 10
Originalidade da combinação dos itens detectados	7 / 10
Originalidade de tópico	**10 / 10**
Meio ambiente e botânica	10 / 10
Originalidade de tags	**10 / 10**
palmeira, imagem, irrigacao, descricao, flor, descrição, irrigação, plantas, planta, antúrio	10 / 10
Originalidade de componentes de interface	**6.5 / 10**
Escolhe lista	9 / 10
Imagem	7 / 10
Legenda	3 / 10
Originalidade da combinação dos itens detectados	6 / 10
Fluência de componentes	**6.5 / 10**
Foram detectados 18 componentes	6.5 / 10
Fluência de programação	**7.5 / 10**
Foram detectados 118 blocos	7.5 / 10
Flexibilidade de funcionalidades	**0.5 / 10**
Flexibilidade das funcionalidades detectadas	0.5 / 10
Flexibilidade de escopo de componentes	**3.5 / 10**
Detectado: Experimental, Interface de Usuário, Organização, Sensores	4 / 12
Não detectado: Armazenamento, Conectividade, Desenho e Animação, Gráfico, LEGO® MINDSTORMS®, Mapas, Mídia, Social	8 / 12
Flexibilidade de escopo de programação	**4 / 10**
Detectado: Controle, Experimental, Interface de Usuário, Listas, Lógica, Sensores, Tela, Texto, Variáveis	9 / 24
Não detectado: Armazenamento, Blocos auxiliares, Conectividade, Cores, Desenho e Animação, Dicionários, Extensões, Gráfico, LEGO® MINDSTORMS®, Mapas, Matemática, Mídia, Organização, Procedimentos, Social	15 / 24
Total	**6.0 / 10**

Sobre Política de Privacidade Termos de Serviço

Source: Elaborated by the author.

The module integration is available online as part of the CodeMaster tool (CNE, 2023).

5 EVALUATION OF THE MODEL

This chapter presents the evaluation of the Creassessment model through a case study. The objective of the study is to analyze if the Creassessment model can differentiate mobile apps considered creative and not creative by human raters. As part of the evaluation, the reliability, validity, and quality of the Creassessment model are analyzed.

Research question: Can Creassessment differentiate between creative mobile apps vs. not creative mobile apps according to human raters in a consistent, reliable, and valid way? This research question is divided into the following analysis questions.

AQ1. Can Creassessment differentiate creative mobile apps positively when compared to non-creative mobile apps according to human raters?

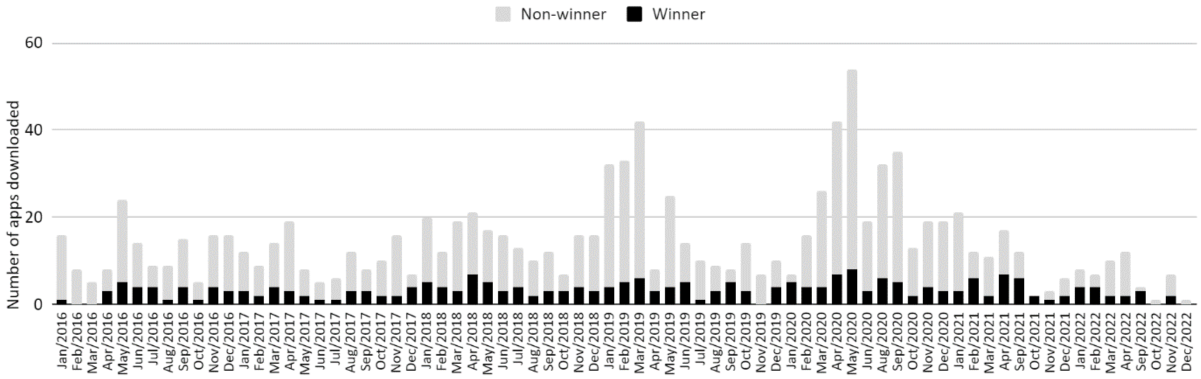
AQ2. Is there evidence of reliability and validity in Creassessment?

AQ3. Is there evidence of quality in Creassessment?

Data collection. In this study, a dataset that consists of mobile apps from the App of the Month contest from 2016 through 2022 made available by the App Inventor Team is used. App of the Month was a monthly program in which app creators could submit their mobile apps to participate in the monthly contest until December 2022. The mobile apps could be submitted by inventors of all ages and in any language. The mobile apps were required to have been made in App Inventor and be functional. The mobile apps were reviewed by members of the MIT App Inventor team considering design (the mobile app is the most aesthetically pleasing), innovation (the mobile app uses App Inventor technology in the most interesting/unique way), and creativity (the mobile app that best uses creative elements such as art, color, sound, or movement) (MIT APP INVENTOR, 2023).

During the 6-year period, there were 1,923 submissions to the contest from all over the world, of which 1,494 submissions were unique (not duplicated) and provided a link to the App Inventor Gallery. Using a Python script for automatically downloading the mobile apps, the source-code from 1,078 mobile apps was obtained (Figure 26). Some mobile apps could not be downloaded because the link provided was outdated and the mobile app was no longer available in the App Inventor Gallery. By performing a comparison of the link of the 1,078 mobile apps provided by the App Inventor team with all the links available on the App Inventor App of the Month Winners website (MIT APP INVENTOR, 2023), 246 (23%) mobile apps were identified as winners and 832 (77%) as non-winners of the App of the Month contest.

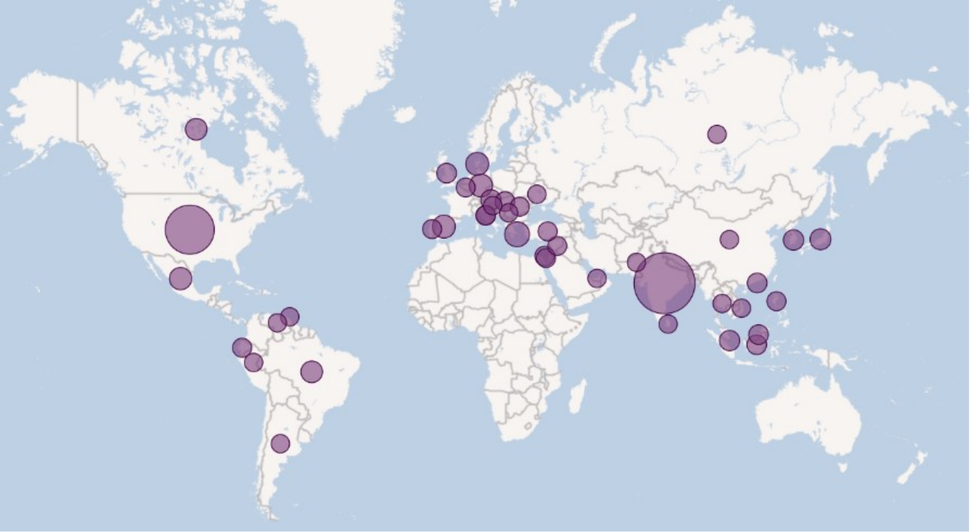
Figure 26 – Distribution of the mobile apps downloaded (n=1,078) across the years for each group (winners/non-winners)



Source: Elaborated by the author.

Additional background information about the creators of the mobile apps is available only for some mobile apps in the winners' group, no additional information is provided for mobile apps in the non-winners group. The creators of mobile apps that won the contest come from all over the world and represent countries from four continents, namely Asia, Europe, North America, and South America (Figure 27). There is a predominance of mobile apps from the United States and India. No specific African or Oceanian countries were represented in the provided data for the winners' group.

Figure 27 – Region of mobile app creators that won the App of the Month contest



Source: Elaborated by the author.

5.1 CAN CREASSESSMENT DIFFERENTIATE CREATIVE MOBILE APPS POSITIVELY WHEN COMPARED TO NON-CREATIVE MOBILE APPS ACCORDING TO HUMAN RATERS?

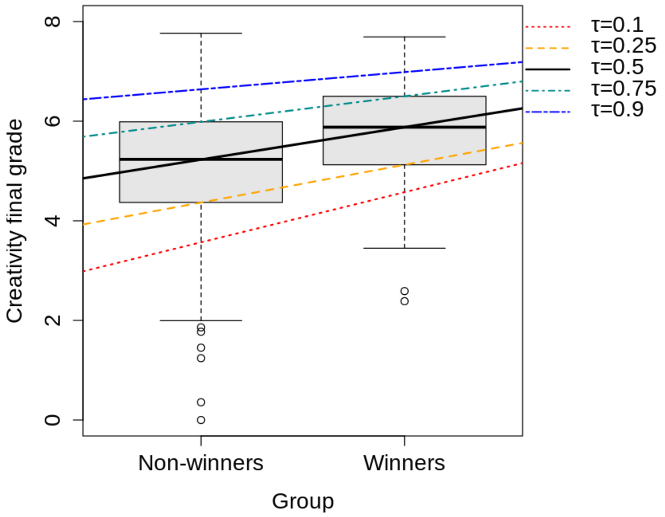
To answer this question, a series of statistical tests were performed to examine if there were significant differences between the winners (n=246) and the non-winners (n=832) group regarding the final grade and each variable of the Creassessment model. An exception is the variable “originality of topics” of the model because most of the mobile apps are in English and currently the model can identify topics only for mobile apps in Brazilian Portuguese. To run the analyses, R language was used (version 4.2.3).

5.1.1 Is the creativity grade generated by the Creassessment model higher for creative mobile apps when compared to non-creative mobile apps according to human raters?

Analyzing the creativity grades generated by the Creassessment model, the mean of the creativity grade for winners ($M = 6.18$) is higher than for non-winners ($M = 5.61$) as well as the median for winners ($Mdn = 6.27$) is higher than for non-winners ($Mdn = 5.69$). This is a first indication that the creativity grade can positively differentiate winners and non-winners.

In order to verify if the creativity grade is higher for winners than non-winners with a 95% confidence level, quantile regression was performed to explore the relationship between the groups (winners x non-winners) and specific percentiles (quantiles) of the creativity scores. Quantile regression is indicated when the means do not have a normal distribution for two groups of continuous measures, such as the creativity scores, as well as to estimate the conditional median and other quantiles. This quantile regression analysis uses a predictor variable (creativity grade scores) to predict an outcome variable (winner/non-winners) at different quantiles of the outcome variable

Figure 28 – Boxplot for creativity grade for each group (winners x non-winners)



Source: Elaborated by the author.

Superimposed on the plot (Figure 28) are five estimated quantile regression lines corresponding to the quantiles $\{0.1, 0.25, 0.5, 0.75, 0.9\}$. The median τ = 0.5 is indicated by the darker solid line. The plot reveals the tendency of the creativity grade to be greater for winners in all quartiles.

Table 19 – Results of quantile regression for the creativity grade

Quantile	**Intercept**	**Coefficient**	**S.E.**	***t***	***p***
0.10	3.57	1.01	0.09	11.45	<.001
0.25	4.37	0.75	0.12	6.23	<.001
0.50	5.24	0.64	0.10	6.18	<.001
0.75	5.99	0.51	0.10	5.18	<.001
0.90	6.64	0.35	0.08	4.14	<.001

Source: Elaborated by the author.

Results from the quantile regression indicated a p-value < 0.001 for all quartiles, indicating that there is statistically significant evidence at α = 0.05. This provides compelling evidence to support the claim that there is a significant positive difference between the creativity grades of winners and non-winners across all quartiles (Table 19). Specifically, the creativity grade scores of winners are consistently higher than those of non-winners.

Furthermore, when comparing the quartiles, it can be observed that the difference in creativity grade scores is more pronounced in the first quartile ($\beta = 0.75$) compared to the third quartile ($\beta = 0.51$). This finding suggests that the creativity grade is particularly effective at differentiating between winners and non-winners among mobile apps with lower grades. This observation may be attributed to the inherent characteristics of creative mobile apps as they become more complex. As the complexity of mobile apps increases, it becomes progressively more challenging for human raters to discern which mobile apps could be considered creative. This difficulty in differentiation could be due to the fact that creative mobile apps tend to possess higher levels of complexity overall.

The results of the analysis provide strong evidence to support the claim that the creativity grade generated by the Creassessment model is higher for creative mobile apps compared to non-creative mobile apps, according to human raters. Overall, the analysis demonstrates a positive association between the creativity grade generated by the Creassessment model and the perceived creativity of the mobile apps, supporting the notion that the model captures the creative aspects of the evaluated mobile apps through the creativity grade.

5.1.2 Is the score of each item generated by the Creassessment model higher for creative mobile apps when compared to non-creative mobile apps according to human raters?

Observing descriptive statistics for the winners and non-winners, the mean of the winners (W) group is higher than the mean of the non-winners (NW) group for all items. This is a first indication that all items of the model may also differentiate winners positively from non-winners (Table 20). This also occurs with the median, indicating that at least 50% of the winners have a score above the non-winners in all items (Figure 29).

Table 20 – Descriptive statistics (W=winners, NW=non-winners)

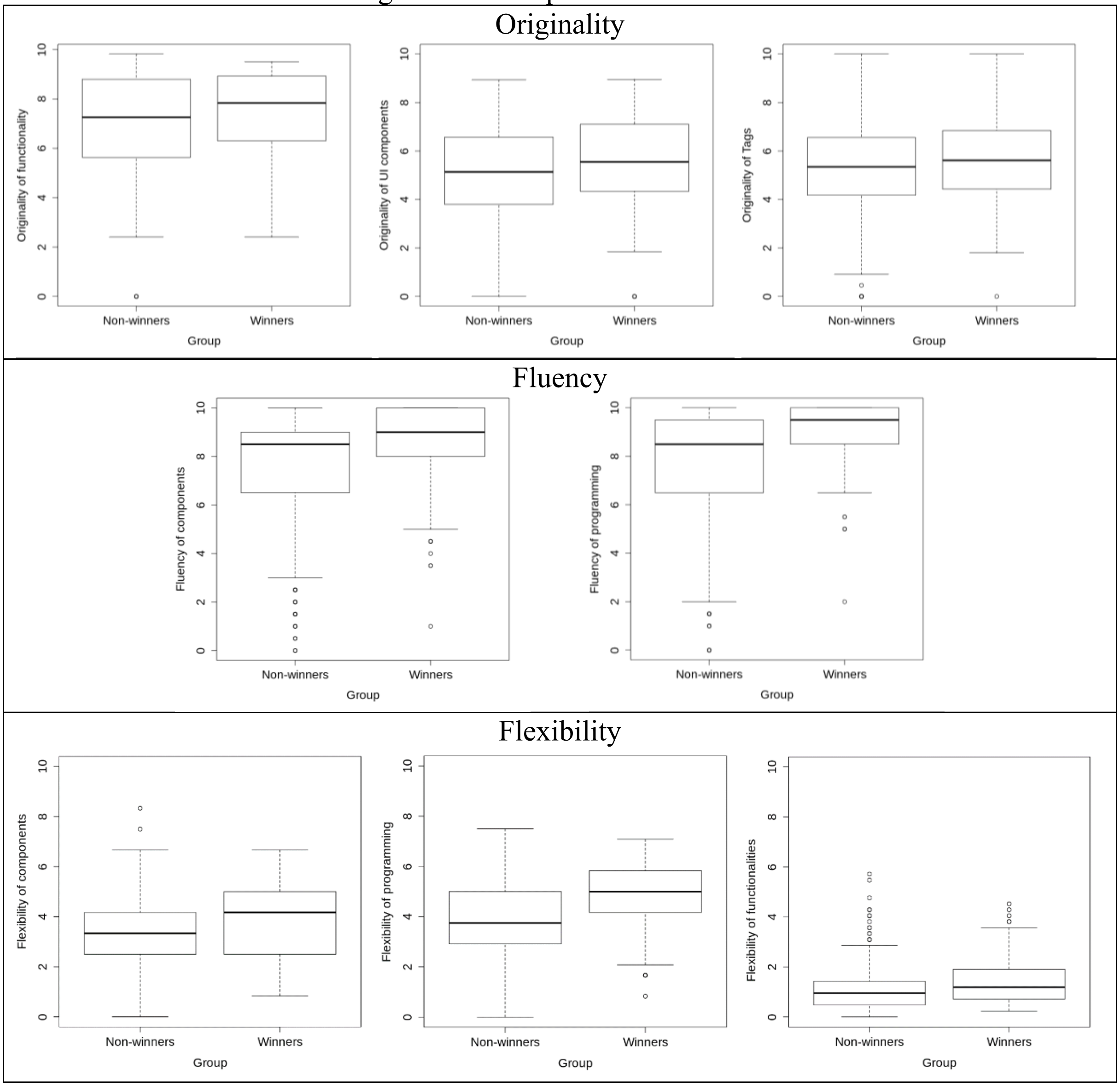
Dimension	Item	Group	M*	SD	Mdn*	Min	Max	Range
Originality	Functionalities	W	**7.37**	1.79	**7.84**	2.41	9.51	7.10
		NW	6.91	2.04	7.26	0.00	9.82	9.82
	UI components	W	**5.61**	1.97	**5.55**	0.00	8.96	8.96
		NW	5.17	1.95	5.14	0.00	8.94	8.94
	Tags	W	**5.62**	1.65	**5.61**	0.00	10.0	10.0
		NW	5.38	1.82	5.35	0.00	10.0	10.0
Fluency	Components	W	**8.66**	1.52	**9.00**	1.00	10.0	9.00
		NW	7.63	2.18	8.50	0.00	10.0	10.0
	Programming	W	**9.14**	1.17	**9.50**	2.00	10.0	8.00
		NW	7.67	2.31	8.50	0.00	10.0	10.00
Flexibility	Components	W	**3.78**	1.27	**4.17**	0.83	6.67	5.83
		NW	3.40	1.32	3.33	0.00	8.33	8.33
	Programming	W	**4.73**	1.24	**5.00**	0.83	7.08	6.25
		NW	3.84	1.48	3.75	0.00	7.50	7.50
	Functionalities	W	**1.39**	0.84	**1.19**	0.24	4.52	4.29
		NW	1.11	0.74	0.95	0.00	5.71	5.71

* The highest mean (M) and median (Mdn) values of each item are marked in bold.

Source: Elaborated by the author.

Considering originality and fluency dimensions, the mean is above 5.0 for all items in both groups, on the other hand, the mean for all items from the flexibility dimension is below 5.0, especially the flexibility of functionalities which has the lowest mean and median. This can be due to the fact that flexibility scores take into account how many different categories of aspects are used. Yet, typically, a mobile app does not need all categories of components, programming blocks, and functionalities, as defined in the conceptual model, to be considered creative by human raters.

Figure 29 – Boxplots for each item



Source: Elaborated by the author.

All items present a standard deviation below the mean, and most of the items for the winners have a standard deviation smaller than the non-winners, indicating that the winners' scores are more clustered around the mean. This also happens with the range values, which are smaller for the winners, because the minimum score is higher for winners than non-winners for most items (Figure 29). The tags item demonstrates the highest range among both groups, indicating that the mobile apps of both groups have received a minimum of zero points and a maximum of 10 points. This may be because the tags item is the only one that is assessed by extracting potential tags from the textual content of the mobile app, thus, any word can be a tag, so the range is broader for this item than for the other items.

The Shapiro-Wilk test was conducted on all items to test for normality, and the H0 was rejected for all items (p-value < 0.05) in a least one group (winner/non-winner). Considering that the scores are not normally distributed, the one-sample Wilcoxon signed rank test was used as a non-parametric alternative. The Wilcoxon test was conducted to examine if the one-sided (one-tail) difference between winners and non-winners is statistically significant across the eight items of the model, i.e., the scores for winners are greater than the scores for non-winners for each item with a 95% confidence level.

H_0: The median difference between winners and non-winners for an item score is zero

H_1: The median difference between winners and non-winners for an item score is positive

Table 21 – Results from the Wilcoxon test

Dimension	Item	W	p-value (95%)	Calculated effect size
Originality	Functionalities	115325	0.00123 **	0.092
	UI components	116540	0.00046 ***	0.101
	Tags	110439	0.02946 *	0.058
Fluency	Components	133688	0.00000 ***	0.224
	Programming	144432	0.00000 ***	0.303
Flexibility	Components	119599	0.00002 ***	0.125
	Programming	139108	0.00000 ***	0.262
	Functionalities	124022	0.00000 ***	0.155
<.05 *, <.01 **, <.001 ***				

Source: Elaborated by the author.

The results for all variables are significant, indicating that there is statistically significant evidence at α = 0.05, to show that the median difference is positive (i.e., that winners' scores are greater than non-winners). All the items of fluency and flexibility, as well as, UI components from originality, show strong evidence with a p-value < 0.001. The originality of functionality and tags show less strong evidence, however still significant at 0.01 and 0.05 respectively (Table 21). The calculated effect size is small (<.30) for most items. This can be due to the fact that the mobile apps that did not win the contest also have some degree of creativity, which diminishes the differences between them and the winning apps, resulting in a small effect size. In this context, a different set of apps that are not submitted to the contest may show an even greater effect size.

The analysis provides evidence that each of the items can score creative mobile apps more positively than non-creative apps according to human raters. Descriptive statistics revealed that the mean and median scores for all items were higher in the winners group compared to the non-winners group. The Wilcoxon test, which accounted for the non-normal distribution of scores, confirmed the statistically significant median differences between winners and non-winners for all items. This indicates that winners consistently scored higher across all dimensions, including originality, fluency, and flexibility.

5.2 IS THERE EVIDENCE OF RELIABILITY AND VALIDITY IN CREASSESSMENT?

Analyzing the reliability and validity of a model is crucial to assessing its trustworthiness and effectiveness in measuring and assessing creativity in mobile apps. Reliability analysis enables to determine the consistency and stability of the model's measurements, ensuring that it produces reliable and dependable results. Validity analysis examines whether the model accurately measures what it intends to measure, in this case, the creative aspects of mobile apps. The reliability and validity analyses are based on the scores of all mobile apps (n=1,078) pooled in a single sample.

Reliability was analyzed using the ω (omega coefficient). Omega takes into account factorial loads, which makes the calculations more stable, with a higher level of reliability regardless of the number of items in the instrument (as opposed to Cronbach's alpha) (FLORA, 2020; HAYES and COUTTS, 2020). According to the literature, $\omega > 0.7$ is the cut-off value, in which a value between 0.7 and 0.8 is acceptable, 0.8 to 0.9 is good, and above 0.9 is considered excellent. As a result, a good value of $\omega = 0.86$ was obtained.

Table 22 – Omega coefficient when excluding items

Dimension	Item	Omega if the item is dropped
Originality	Functionalities	0.84
	UI components	0.84
	Tags	0.89
Fluency	Components	0.84
	Programming	0.83
Flexibility	Components	0.82
	Programming	0.81
	Functionalities	0.82

Source: Elaborated by the author.

When analyzing whether reliability increases by eliminating an item, all items except for the originality of tags decrease the omega if they are eliminated (Table 22). One reason for the increase of the omega value if the originality of tags is eliminated can be due to the fact that this item does not correlate well with other items because the set of tags extracted from a mobile app can be virtually any set of words. In general, the results provide an indication of the internal consistency of the model.

To analyze convergent validity, the correlations between the eight variables of the model using Pearson correlation coefficients were explored. The Pearson correlation coefficient, often denoted as "r", measures the strength and direction of the linear relationship between two continuous variables. It can take values ranging from -1 to 1, and these values have specific meanings: when r is close to +1, it indicates a strong positive correlation, when r is close to 0, there is no linear correlation between the variables, and when r is close to -1, it indicates a strong negative correlation. In the context of this analysis, it is expected that the items measuring a single dimension show positive correlations greater than or equal to 0.30. Correlations between 0.3 to 0.5 are considered weak linear correlations, between 0.5 and 0.7 are considered moderate correlations, between 0.7 and 0.9 are considered strong correlations, and above 0.9 are considered very strong (DEVELLIS, 2017).

Table 23 – Correlation matrix

Dimension	**Item**	**Originality**			**Fluency**		**Flexibility**		
		Functionalities	UI components	Tags	Components	Programming	Components	Programming	Functionalities
Originality	Functionalities	1.00							
	UI components	**0.47**	1.00						
	Tags	0.00	0.18	1.00					
Fluency	Components	0.25	**0.40**	**0.30**	1.00				
	Programming	**0.33**	**0.33**	0.20	**0.72**	1.00			
Flexibility	Components	**0.69**	**0.44**	0.01	**0.40**	**0.46**	1.00		
	Programming	**0.60**	**0.45**	0.05	**0.48**	**0.73**	**0.78**	1.00	
	Functionalities	**0.71**	**0.53**	0.00	**0.39**	**0.45**	**0.81**	**0.73**	1.00

* Correlations above 0.29 are marked in bold.

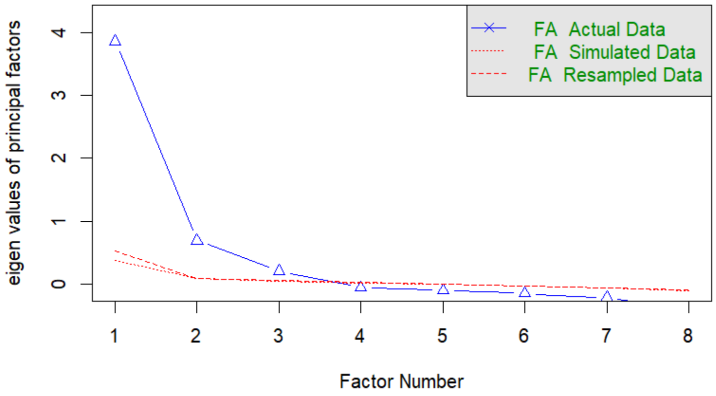
Source: Elaborated by the author.

Flexibility items show the strongest correlations for a dimension, all items are correlated $r > 0.7$ with the highest correlation between functionalities and components. The fluency items are also highly correlated internally with a strong correlation $r > 0.7$. On the other hand, the only pair of items that correlate internally in the originality dimension are UI components and functionalities with a weak correlation of $r > 0.3$.

Considering the correlation of the items across other dimensions, all items from flexibility and fluency correlate to a least a weak level of $r > 0.3$. This can indicate that the more components and programming blocks are used, the broader the components, programming, and functionalities. Items from the originality dimension also correlate well with items from flexibility and fluency. For fluency, this can indicate that as more programming blocks and components are used in a mobile app, chances are that it also scores higher in originality items. These results are also in accordance with findings reported in the literature on idea generation, in which individuals who generate highly original ideas are also likely to be highly fluent in idea generation (SILVIA, 2008). The only exception is the tags item, which only correlates weakly with the components item from fluency, indicating that the more components are used, the more diverse the textual content of a mobile app and, thus, more original tags can be extracted from it. On the other hand, the use of different components, programming blocks, or functionalities, i.e., more flexibility, does not implicate a more original set of tags (Table 23).

To identify the number of underlying factors that influence the variables of the Creassessment model, the possibility of performing an exploratory factor analysis was checked using the Kaiser-Meyer-Olkin (KMO) index. The KMO index measures the sampling adequacy with values between 0 and 1. A value near 1.0 supports a factor analysis and anything less than 0.5 is not likely suitable for useful factor analysis (BROWN, 2006). Here, a KMO index of 0.77 was obtained, demonstrating that factor analysis is suitable in this case.

Figure 30 – Parallel Analysis Scree Plots



Source: Elaborated by the author.

The exploratory analysis of the correlation matrix compared with parallel random matrices indicates the existence of three factors in the sample, which are represented by the triangles in blue above the red dotted line (Figure 30). To identify if the model fits the data well, an exploratory factor analysis was performed, in which statistics are not very sensitive to the sample size. The values to evaluate the quality of the model's fit are the root mean square error of approximation (RMSEA), comparative fit index (CFI), and the Tucker–Lewis index (TLI). The fit is considered adequate when RMSEA < 0.05, TLI > 0.90, and CFI > 0.90 (BROWN, 2006).

Table 24 – Results for exploratory factor analysis fit

Number of factors	RMSEA	TLI	CFI
1	0.2158	0.7847	0.8462
2	0.1453	0.9025	0.9547
3	0.0983	0.9553	0.9888

Source: Elaborated by the author.

Here, the fit indices suggest that one factor does not fit the data particularly well. A higher RMSEA value indicates a relatively poor fit, as values closer to zero are desirable, both TLI and CFI are below 0.9, suggesting a suboptimal fit for one factor. The fit indices for two factors indicate a better fit compared to one factor since the RMSEA value is lower. In addition, both TLI and CFI values are above 0.9, suggesting a reasonably good fit for two factors. Three factors show the best fit among the number of factors. The RMSEA value is the lowest but still above 0.05, indicating a relatively good fit. Both TLI and CFI values are above 0.9, with CFI approaching 1, indicating a very good fit for three factors (Table 24).

Based on the results provided, an exploratory factor analysis with two and three factors was conducted to explore the underlying dimensions represented by a set of observed variables (Table 25). The factor loading reflects the strength of the relationship between the item and the latent factor it is intended to measure. To determine a cut-off value for a factor loading that indicates that an item has loaded well on a factor no universally accepted standard exists. However, a commonly used conservative rule of thumb is that values greater than 0.50 are considered necessary for practical significance (HAIR, ANDERSON, *et al.*, 2009). This means that an item with a factor loading of 0.50 or higher is considered to be well-suited to measuring the underlying factor.

Table 25 – Results for exploratory factor analysis with three factors

Dimension	Item	F1 *	F2 *	F3 *
Originality	Functionalities	**0.9316**	-0.0752	-0.0450
	UI components	**0.5270**	-0.1280	0.2539
	Tags	-0.1071	-0.1298	0.4185
Fluency	Components	0.0825	-0.0756	**0.8894**
	Programming	-0.0396	0.4706	**0.6499**
Flexibility	Components	**0.7196**	0.2771	-0.0303
	Programming	0.3364	**0.7047**	0.0973
	Functionalities	**0.9164**	0.0490	0.0725

*Values greater than 0.50 are marked in bold

Source: Elaborated by the author.

The first factor (F1) seems to merge the dimensions of originality and flexibility. The originality of functionalities stands out with a high positive loading of 0.9316. This indicates a strong association between this item and the underlying dimension represented by Factor 1. The originality of UI components also shows a moderate positive loading of 0.5270, suggesting a relatively weaker relationship with Factor 1. Conversely, the originality of tags has a negative loading of -0.1071, indicating that it is negatively associated with Factor 1. The flexibility of components and functionalities also exhibit a high positive loading of 0.7196 and 0.9164 respectively. These high factor loadings may be related to the data structure, especially since the flexibility of components and the originality of UI components and functionalities correlate well.

Regarding the second factor (F2), the flexibility of programming exhibits a high positive loading of 0.7047. Interestingly, the fluency of programming also exhibits a moderate positive loading of 0.4706, implying a significant association with this factor. This may be due to the fact these two variables are strongly correlated (Table 23). On the other hand, the originality of UI components and tags both have negative loadings of -0.1280 and -0.1298, respectively, suggesting a weak relationship with factor 2.

The third factor (F3) seems to be more related to fluency since both items from fluency demonstrate a high positive loading while all other items show a relatively low positive or negative loading, except for the originality of tags, which exhibit a positive loading in F3.

Considering that TLI and CFI values are above 0.9 for two factors (Table 24), an exploratory factor analysis for two factors was also performed (Table 26).

Table 26 – Results for exploratory factor analysis with two factors

Dimension	Item	F1 *	F2 *
Originality	Functionalities	**0.9440**	-0.1351
	UI components	**0.5091**	0.0888
	Tags	-0.1905	0.3068
Fluency	Components	0.0258	**0.7399**
	Programming	-0.0043	**0.9939**
Flexibility	Components	**0.8829**	0.0419
	Programming	**0.5666**	0.4625
	Functionalities	**0.9479**	0.0540

*Values greater than 0.50 are marked in bold

Source: Elaborated by the author.

Regarding the first factor (F1), in terms of the originality dimension, functionalities show a high positive loading of 0.9440, indicating a strong association between the originality of functionalities and Factor 1. The originality of UI components also exhibits a positive loading of 0.5091, suggesting a relatively weaker relationship with F1. On the other hand, the originality of tags shows a negative loading of -0.1905, indicating a negative association with F1. Regarding the flexibility dimension, all three items show positive loadings in F1. Specifically, the flexibility of components has a high loading of 0.8829, programming has a moderate loading of 0.5666, and functionalities demonstrates a high loading of 0.9479. These loadings suggest a strong association between these items and F1.

Factor 2 groups items from the fluency dimension. In this factor, fluency of components shows a strong positive loading of 0.7399, and fluency of programming exhibits the strongest factor loading of 0.9939. This indicates a strong relationship between fluency items and F2.

In summary, based on the factor loadings, F1 appears to be primarily related to the originality and flexibility dimensions. F2, on the other hand, represents the fluency dimension. Only the originality of tags has not presented a strong positive loading in any of the factors.

The results of the analysis provide insights into the reliability and validity of the Creassessment model. Reliability analysis, assessed using the omega coefficient, revealed good overall reliability. In terms of convergent validity, the correlation matrix indicated that flexibility and fluency items demonstrated strong internal correlations, while the originality dimension exhibited weaker internal correlations. Furthermore, correlations between dimensions indicated that flexibility and fluency correlated with each other and with items from the originality dimension. Exploratory factor analysis revealed the presence of strong factor loading for two and three underlying factors in the model, indicating a reasonably good fit, especially for three factors.

5.3 IS THERE EVIDENCE OF QUALITY IN CREASSESSMENT?

To analyze the quality of items in the Creassessment model, an analysis adopting Item Response Theory (IRT) was performed. IRT provides a rigorous methodology for assessing the quality of a model by analyzing how well individual items within the model differentiate between different levels of the measured construct. Here, the IRT Graded Response Model (GRM) (SAMEJIMA, 1969) available in the mirt R package was used. The GRM is a suitable IRT model for ordered categorical data, which can include responses on a continuous scale like 0-10. The model allows for the estimation of item discrimination parameters and item threshold parameters for each response category. In this case, response categories can be defined by dividing the 0-10 scale into a set of ordered categories and treating each category as a separate response option.

To fit the GRM model, the original scores on a continuous scale of 0 to 10 of all mobile apps (n=1,078) were recoded into five categories to facilitate the analysis (Table 27). The recoding involved grouping the responses into distinct intervals. The five categories were defined as follows: [0, 2), [2, 4), [4, 6), [6, 8), and [8, 10]. By dividing the continuous scale into these discrete intervals, the responses were transformed into ordinal categories that better align with the assumptions and requirements of GRM. This recoding allows for a more manageable analysis of the data, where the focus is on the relationship between the latent dimension and the ordered categories of responses within each interval.

Table 27 – Frequencies of items with ordered categories

Dimension	Item	[0, 2)	[2, 4)	[4, 6)	[6, 8)	[8, 10]
Originality	Functionalities	0.56%	7.70%	**20.87%**	**32.28%**	**38.59%**
	UI components	8.63%	**19.11%**	**36.64%**	**24.12%**	11.50%
	Tags	2.69%	**18.37%**	**40.17%**	**31.73%**	7.05%
Fluency	Components	2.23%	3.90%	9.37%	**20.50%**	**64.01%**
	Programming	1.86%	5.01%	9.00%	**17.44%**	**66.70%**
Flexibility	Components	13.54%	**45.55%**	**39.05%**	1.76%	0.09%
	Programming	9.46%	**36.36%**	**46.10%**	8.07%	NA
	Functionalities	**88.13%**	10.85%	1.02%	NA	NA

Source: Elaborated by the author.

In general, for the categorical data, there is a trend of higher frequencies in the middle response categories. This suggests that the model tends to avoid the lowest and highest ends of the scale grade and is more inclined toward the middle range. Specifically, the originality of UI components and tags show a concentration of frequencies in the [4, 6) category. On the other hand, fluency of components and programming exhibit a broader distribution, with frequencies peaking in the [8, 10] category. In terms of distribution, the flexibility of components stands out with a substantial frequency in the [2, 4) category, indicating a concentration of scores on this specific range. The flexibility of programming demonstrates a considerable frequency in the [4, 6) category and no data points for the [8, 10] category. Lastly, the flexibility of functionalities reveals a dominant frequency in the [0, 2) category and no data points for [6, 8) and [8, 10] categories. Using these recoded scores, the data was fitted to the GRM model.

Table 28 – GRM IRT estimated parameters

Dimension	Item	a	b1	b2	b3	b4
Originality	Functionalities	**1.87**	-3.57	-1.84	-0.74	0.39
	UI components	**1.24**	-2.33	-1.03	0.56	2.05
	Tags	0.18	-20.54	-7.52	2.69	14.81
Fluency	Components	**1.55**	-3.08	-2.26	-1.44	-0.51
	Programming	**2.38**	-2.61	-1.79	-1.19	-0.53
Flexibility	Components	**2.78**	-1.29	0.25	2.55	3.97
	Programming	**3.93**	-1.41	-0.14	1.56	NA
	Functionalities	**6.96**	1.24	2.49	NA	NA

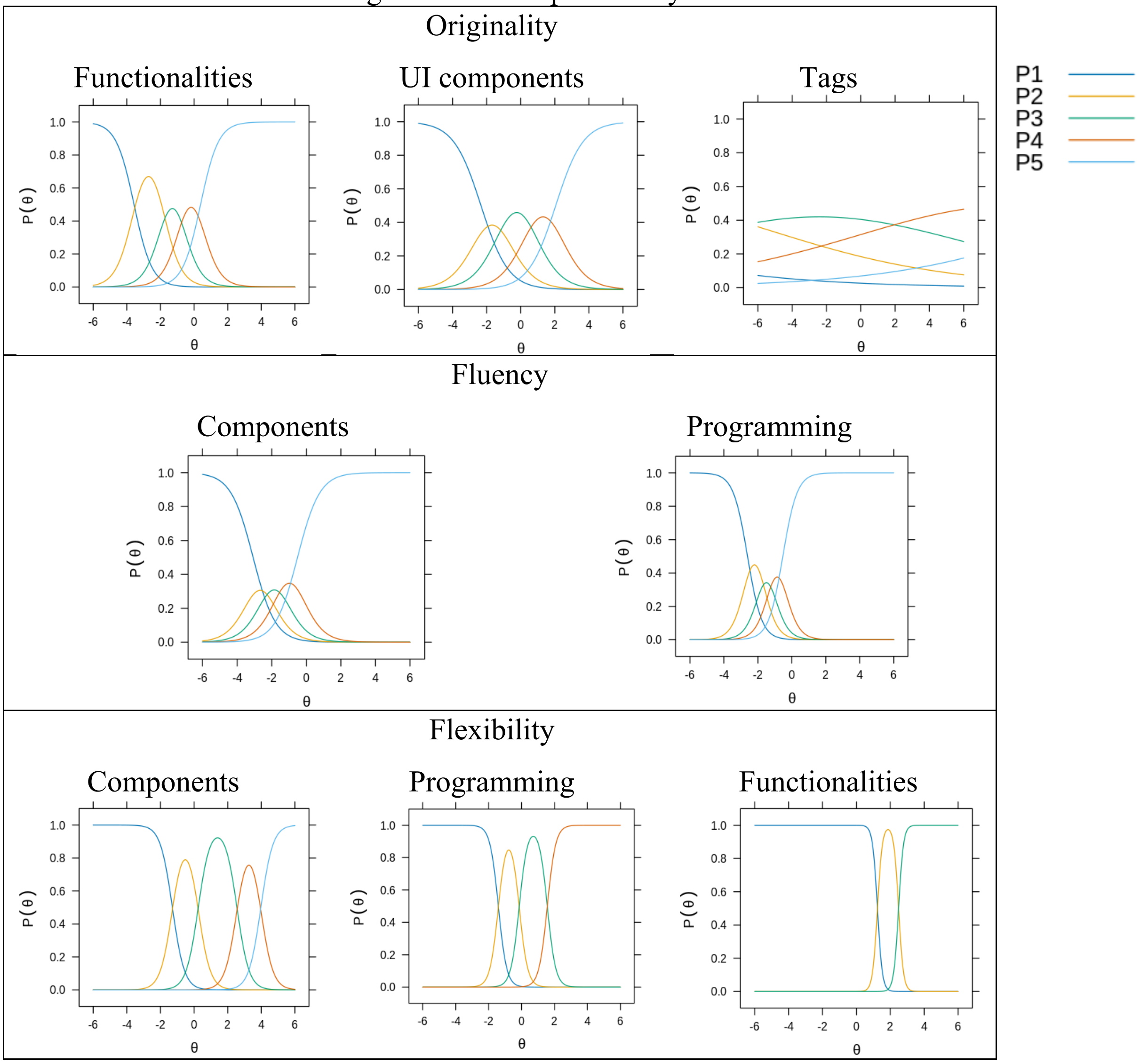
Source: Elaborated by the author.

Regarding the desired values in the parameter estimation process, values above 1 for slope parameter (a) are considered good. For the difficulty parameters (b1- b4), values typically within the range of [-5, 5] are expected, although this is not an absolute requirement. In addition, a good spacing between the values of the difficulty parameters is also desired, ensuring that all response categories have a significant probability. Based on the item calibration results, where

$\mu = 0$ and $\sigma = 1$, it can be observed that the majority of items calibrated well, with slope parameter (a) values above 1, indicating good discriminatory power (Table 28). On the other hand, the originality of tags exhibits a notably low slope parameter (a), suggesting limited discriminatory power. This item may not effectively differentiate different levels of the latent trait and the results may not provide relevant information.

The difficulty parameters (b1, b2, b3, b4) provide information about the item's position along the latent trait continuum. Positive values indicate more challenging items, while negative values indicate easier items. The magnitude of the difficulty parameters indicates the level of difficulty or ease of the item relative to the latent trait distribution. Items that do not have data points in higher categories do not have parameter estimates for b3 or b4, such as flexibility of programming, which has no data points for the [8, 10] category, and flexibility of functionalities which has no data points for [6, 8) and [8, 10] categories.

Figure 31 – Item probability functions



The probability (P) curves refer to the probability of getting a score in each of the five intervals:
P1: [0, 2); P2: [2, 4); P3: [4, 6); P4: [6, 8), P5: [8, 10]

Source: Elaborated by the author.

All items from originality and flexibility, with the exception of flexibility of functionalities, start with a negative difficulty parameter, indicating relative easiness for mobile apps that are not necessarily considered creative, but demonstrate a transition to positive difficulty parameters, indicating increasing difficulty in higher trait levels. On the other hand, both fluency items (programming and components) have a negative value for all *b* parameters, suggesting they are easier items that are likely to be present in mobile apps that are not necessarily considered creative (non-winners). This can also be observed in the item probability functions, in which the probability curves of fluency items are dislocated to the left side (Figure 31).

Based on the analysis using IRT and GRM, the quality of items in the Creassessment model was analyzed. The findings indicate that the majority of items calibrated well, with slope parameters (a) above 1, suggesting good discriminatory power. Overall, the analysis provides evidence of quality in the Creassessment model, as it demonstrates the model's capability to differentiate between different levels of the construct and provides insights into the difficulty levels of individual items. However, further investigation and refinement may be needed, particularly regarding the tags item, to enhance the model's overall quality.

5.4 DISCUSSION

The results of the evaluation obtained indicate that the Creassessment model can differentiate creative mobile apps positively when compared to non-creative mobile apps according to human raters.

In terms of discriminating winners vs. non-winners according to human raters, the results of statistical tests of the model final creativity grade, as well as, each item indicated that there is statistically significant evidence at $\alpha = 0.05$ to show that the difference between winners and non-winners is positive. Additionally, the descriptive statistics for individual items of the model indicate that winners consistently outperform non-winners in terms of mean and median scores. The Wilcoxon test, conducted due to non-normality, confirms the statistically significant median differences favoring winners across all items of the model. Overall, the findings suggest that creativity grade and the items of the model effectively differentiate between winners and non-winners, highlighting the significance of these factors in distinguishing the two groups.

The results of the statistical analysis of the Creassessment also show evidence to consider the reliability, construct validity, and quality as an acceptable model for measuring the creativity of mobile apps. The reliability of the model was assessed using the omega coefficient, which accounts for factorial loads and provides a stable measure of reliability. The obtained omega value of 0.86 indicated good reliability. Eliminating items, except for the originality of tags, decreased the omega coefficient. Correlation analysis revealed strong internal correlations within the flexibility and fluency dimensions, while the originality dimension showed weaker internal correlations. Items from flexibility and fluency also demonstrated correlations with each other, suggesting a relationship between the use of components and programming blocks and higher originality and fluency scores. The KMO index of 0.77 indicated suitability for factor analysis, which identified three factors. Factor 1 merged originality and flexibility, Factor 2 represented flexibility of programming, and Factor 3 was primarily related to fluency. Factor loadings showed significant associations between items and their underlying factors.

The quality of items in the Creassessment model was analyzed using the GRM model and the original continuous responses on a scale of 0 to 10 were recoded into five categories to facilitate the analysis. The categorical data showed a trend of higher frequencies in the middle response categories. Some items, such as the originality of UI components and tags, showed concentrated frequencies in the middle category, while others, such as fluency of components and programming, had broader distributions with frequencies peaking at the higher end. The item calibration results indicated good discriminatory power for most items, except for the tags item. Difficulty parameters showed the relative position of items along the latent trait continuum, with negative values indicating easier items and positive values indicating more challenging items. Items from originality and flexibility showed increasing difficulty with higher trait levels, while fluency items presented relatively easier difficulty parameters.

The results thus provide a first indication that the Creassessment model can be used to measure the creativity of mobile apps created with App Inventor considering originality, fluency, and flexibility. Here originality is assessed with respect to a reference universe. Using a reference universe as a basis to measure creativity in mobile apps can be a viable approach. By establishing a reference universe, which represents a collection of existing mobile apps originality can be evaluated based on their deviation from this established norm. Furthermore, using a reference universe provides a standardized and objective framework for measuring creativity, enabling meaningful comparisons and facilitating the interpretation of creativity measurement. The evaluation results show that the model can positively differentiate creative from non-creative mobile apps according to human raters in a reliable and valid way.

5.4.1 Threats to validity

The results of the empirical study are subject to several threats to validity. Thus, potential threats were identified and mitigation strategies were applied to minimize their impact on results.

Conclusion validity. Threats to conclusion validity revolve around factors that impede the accurate determination of relationships between the treatment and the outcome. These results are subjected to fishing for a specific result, the reliability of the model, and the reliability of human raters’ creativity assessment. To maintain impartiality and avoid actively seeking a particular outcome, independent data from the App Inventor App of the Month contest provided by the App Inventor Team was used. All mobile apps that had a workable link were downloaded and those that had won the contest were put in the winners group and those that did not win were put in the non-winners group. No intervention was made to manipulate the data to comply with the principles of avoiding searching for a specific result.

Concerned about the reliability of the model a reliability analysis was performed, which showed good results overall. However, for the originality of tags, a recurring pattern of insufficient evidence of reliability was noticed. One main factor contributing to this lack of reliability for the tags is the variability since any word in the mobile app can be a tag. This inherent characteristic, however, should not necessarily be viewed as negative but prompts the need for careful interpretation and consideration of multiple factors.

There is a risk that the reliability of human raters’ creativity assessment, i.e., the multiple raters assessing the creativity of mobile apps in the App Inventor of the Month contest did not apply the same criteria or that their criteria changed between different occasions/years. Even though the implementation should hence be as standard as possible over different subjects and occasions, it is assumed that if one mobile app won the contest it has passed the subjective minimum criteria to be considered creative and to be put in the winners' group. In addition, the validity of this dataset as the ground truth for creativity should be interpreted carefully as the criteria included design (visual aesthetics), innovation, and creativity (art use). In this sense, the human raters may measure not only creativity but also something more.

Construct validity. To mitigate design-related threats, a systematic methodology for the study using the Goal/Question/Metric approach was defined and documented (BASILI; CALDIERA; ROMBACH, 1994). In terms of the definition of the assessment, a definition of the conceptual model based on the overall definition of creativity was provided. Considering that the objective of the model is to assess the creativity of a single artifact, the creativity classic dimensions, referring to originality as the rarity of ideas, fluency as the number of ideas, and flexibility as the range of ideas, were adapted to reflect these dimensions on aspects of a single artifact (mobile app). Particularly for flexibility and fluency, approximations were made strongly based on frequency and counts. For flexibility, the use of different aspects of functionalities and programming blocks might not necessarily equate to flexibility in the classical sense but only in the context of a single artifact. For fluency, absolute code size does not necessarily equate to code fluency overall, which can be more related to the sophistication of using code constructs, i.e., in a clever way. Nevertheless, these approximations provide a first step in providing a conceptual model to assess the creativity of mobile apps.

External validity. The possibility of generalizing the results is related to the sample size and the diversity of the data used for the assessment. To reduce these threats, real mobile apps submitted to the App of the Month contest were used. The mobile apps in the dataset come from people from four continents with diverse backgrounds and in the App Inventor community (Figure 27). The dataset also comprises a 6-year period from 2016 through 2022 allowing it to represent a diverse population.

6 LIMITATIONS

The use of automated models, such as Creassessment, for measuring creativity also raises important considerations, such as faking behavior, capturing the nuanced aspects of creativity that require human judgment, and what are the future implications/uses of a mobile app. In addition, it is important to note that the Creassessment model does not consider the usefulness of a mobile app, an essential aspect of creativity.

Faking. Considering that the Creassessment model has well-defined measures for each item, there is still the question about the degree to which a mobile app creativity measurement is susceptible to faking. Faking poses significant challenges and ethical concerns. Research has shown that individuals can intentionally manipulate their responses to create a false impression of higher creativity levels, undermining the validity and reliability of the measurement (KYLLONEN, WALTERS, and KAUFMAN, 2005). Overall, while automated models for measuring creativity offer efficiency and objectivity, researchers must remain aware of the potential for faking behavior, and models such as Creassessment should be complemented with human analysis to ensure a comprehensive and accurate assessment.

Nuances. Automated models may also have limitations in capturing the nuanced aspects of creativity and may struggle to identify certain creative elements that require human judgment. While the Creassessment model has been introduced as a valuable tool for investigating creativity, it is not inherently superior in terms of reliability or validity when compared to subjective human ratings. Nonetheless, the findings indicate that the Creassessment model demonstrates statistical significance in effectively distinguishing winners from non-winners in the App of the Month contest. This suggests that the automated approach presents a viable alternative to human ratings, as it exhibits reliability and validity in this specific context. Other proposed automated approaches, such as SemDis (BEATY and JOHNSON, 2020) and AuDrA (PATTERSON, BARBOT, *et al.*, 2022) have also indicated that automated measures, such as semantic distance and deep learning models, are suited to capture novelty ratings.

Usefulness. The definition of creativity typically also includes usefulness, i.e., in this context, if the mobile app is useful, which is not currently captured by the Creassessment model. The model only addresses the dimensions of originality, fluency, and flexibility, as a proxy for creativity. Despite this, researchers have relied on metrics based on these dimensions, especially in divergent thinking tests, not accounting for the usefulness of an idea. This may be due to the fact that human raters do not all agree on what idea (and mobile app in the context of this research) is useful.

In addition, the burden of human rating in the assessment of creativity can be substantial, particularly due to the subjective nature of the task and the potential for biases and inconsistencies. Human raters often face challenges in providing consistent and reliable evaluations, which can be influenced by personal preferences, expertise, fatigue, and time constraints. Additionally, the process of manually reviewing and rating a large volume of creative works can be time-consuming and resource-intensive. Thus, while recognizing the limitations, the use of automated models in creativity assessment can enhance efficiency, objectivity, and scalability in the evaluation process as well as identify patterns and nuances in creative works that may be overlooked or undervalued by human raters.

Positive creativity. Even though the results for the Creassessment model are mostly positive, its use does not inherently enhance positive creativity since it does not take into account what are the benefits and positive uses or potential harms of a mobile app. In this sense, a mobile app can achieve a high score for creativity in the model but cause harm to others. Under these circumstances, human analysis is needed to identify the positive or negative creativity of the mobile apps assessed.

7 CONTRIBUTIONS

7.1 ANSWERING THE RESEARCH QUESTION

Based on the evaluation results it is thus possible to respond to the research question: it is possible to automatically assess the creativity of mobile apps as learning outcomes in computing education in a reliable and valid manner with the proposed assessment framework. The framework provides a comprehensive approach to assess the creativity of mobile apps developed as learning outcomes in computing education by considering dimensions of originality, flexibility, and fluency. Through statistical analyses and the comparison of mobile app aspects with a reference universe of existing mobile apps, the framework demonstrates its capability to differentiate creative mobile apps positively when compared to non-creative mobile apps according to human raters. Furthermore, the statistical analyses confirm the framework's reliability, construct validity, and overall quality. By answering the research question, the proposed assessment framework offers a standardized and objective basis for measuring creativity, providing valuable insights for researchers, educators, and stakeholders in the field of computing education. It allows for the reliable assessment of creative mobile app development, ensuring the cultivation and recognition of creativity as a vital learning outcome in the context of computing education.

7.2 COMPARING THE MODEL WITH EXISTING MODELS

Regarding the existing models in the literature, the key difference of this work lies in the type of projects being considered. The existing models primarily focus on well-defined problems (MANSKE and HOPPE, 2014; GAL, HERSHKOVITZ, *et al.*, 2017; KERSHAW, CLIFFORD, *et al.*, 2022; LUO, LU, and WANG, 2020; KOVALKOV, SEGAL, and GAL, 2020), or provide a manual assessment for open-ended free choice projects (GROVER, BASU, and SCHANK, 2018; BASU, 2019; GROENEVELD, MARTIN, *et al.*, 2022; ROMERO, LEPAGE, and LILLE, 2017). However, Creassessment incorporates open-ended free-choice projects specifically created with App Inventor and offers the assessment in an automated way.

Compared with the existing approaches proposed specifically for mobile apps created with App Inventor (MUSTAFARAJ, TURBAK, and SVANBERG, 2017; TURBAK, MUSTAFARAJ, *et al.*, 2017; BASU, 2019; GROVER, BASU, and SCHANK, 2018), the present research expands the assessment criteria beyond just originality. While the previous approaches primarily focused on evaluating the uniqueness and correctness of the mobile apps, the proposed framework takes into account additional dimensions of creativity, specifically fluency and flexibility. This aligns with the established definition of measuring creativity in the literature, providing a more comprehensive and nuanced evaluation of the creative aspects of mobile apps. By incorporating these three dimensions, the new model offers a more targeted and structured approach to understanding and assessing creativity in mobile apps. This advancement fills a crucial gap in the existing literature by providing a comprehensive approach to assessing the creativity of mobile apps developed with App Inventor, thereby contributing to the advancement of the field of computing education and fostering the development of innovative and creative mobile app development skills among students.

Results of statistical analysis, such as quantile regression, factor analysis, and item response theory, show that the model enables a thorough examination of the relationships, patterns, and interactions between the dimensions of originality, fluency, and flexibility. These statistical analyses provide quantitative measures and insights into the extent to which each dimension contributes to overall project creativity, allowing for a comprehensive understanding of the underlying factors that drive creative outcomes. With its emphasis on rigorous statistical analyses, this new model offers researchers and practitioners a powerful tool to delve deeply into the intricate dynamics of creativity and to gain valuable empirical evidence to inform future educational interventions, instructional design, and practices on the assessment of creativity of apps.

Furthermore, an important advancement in the present research is the automated nature of the assessment. Unlike the previous approaches that may involve manual or subjective evaluation processes (BASU, 2019; GROVER, BASU, and SCHANK, 2018), the proposed framework enables automated assessment of the creativity of mobile apps. This automation not only increases efficiency but also reduces potential biases and inconsistencies that may arise from subjective judgments. By leveraging automated assessment techniques, the present research contributes to the development of a more objective and reliable method for assessing the creativity of mobile apps created with App Inventor.

7.3 SCIENTIFIC PUBLICATIONS

As part of the Ph.D. Program in Computer Science at the Federal University of Santa Catarina (PPGCC/UFSC), candidates are required to publish and report their scientific contributions in accordance with the Brazilian official system for classifying scientific production. This expectation stems from the recognition that disseminating research findings is essential for advancing knowledge in the computing field as well as serving as a means of documenting and communicating the originality, significance, and rigor of the research.

During this Ph.D., partial results have been published as journal articles, conference papers, book chapters, and technical reports. Table 29 presents the scientific publications produced during the period of this research with their respective Qualis. Qualis refers to the Brazilian official system for classifying scientific production, which is updated every four years. The up-to-date Qualis is provided by CAPES based on the quadrennium 2017-2020, in which grades (so-called "strata") are on a 1–9 scale (A1, the highest; A2; A3; A4; B1; B2; B3; B4; C — not listed) (CAPES; MEC, 2023). There is no Qualis for book chapters or technical reports.

Table 29 – Scientific publications

#	Reference	Qualis 2017-2020
	Peer-Reviewed Journal Articles	
1	**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), 2021.	A1
2	**ALVES, N. da C.**; GRESSE von WANGENHEIM, C.; HAUCK, J. C. R., BORGATTO, A. F. A Detailed Item Response Theory Analysis of Algorithms and Programming Concepts in App Inventor Projects. Brazilian Journal of Informatics in Education (RBIE) - Special Edition Awarded Articles in the Brazilian Symposium on Computing Education, 29, 2021.	A4
3	LEHMKUHL, G.; GRESSE von WANGENHEIM, C.; MARTINS-PACHECO, L. H.; BORGATTO, A. F. **ALVES, N. da C.** SCORE – A Model for the Self-Assessment of Creativity Skills in the Context of Computing Education in K-12. Informatics in Education, 20(2), 2021.	A1
4	**ALVES, N. da C.**; GRESSE von WANGENHEIM, C.; HAUCK, J. C. R. Approaches to Assess Computational Thinking Competences Based on Code Analysis in K-12 Education: A Systematic Mapping Study. Informatics in Education, 18(1), 17-39, 2019.	A1
	Peer-Reviewed Conference/Workshop Papers	
5	**ALVES. N. da C.**, GRESSE von WANGENHEIM, C. A Large-Scale Analysis of the Functionalities of Apps Created with App Inventor (*in Brazilian Portuguese: Uma Análise em Larga-Escala das Funcionalidades de Aplicativos Criados com App Inventor*) In: Proc. of the 3rd Brazilian Symposium on Computing Education (EDUCOMP23), online, Brasil, 2023.	B3
6	**ALVES. N. da C.**, GRESSE von WANGENHEIM, C. Does teaching design thinking help in the development of original apps in the context of teaching computing? (*in Brazilian Portuguese: O ensino de design thinking ajuda no desenvolvimento de aplicativos originais no contexto do ensino de computação*?) In: Proc. of the 33rd Brazilian Symposium of Informatics in Education (SBIE22), Manaus/Brazil, 2022.	A3
7	**ALVES. N. da C.**; KREUCH, L.; GRESSE von WANGENHEIM, C. Analyzing Structural Similarity of User Interface Layouts of Android Apps using Deep Learning. In: Proc. of the 21st	A3

#	**Reference**	**Qualis 2017-2020**
	Brazilian Symposium on Human Factors in Computing Systems (IHC22), Diamantina/Brazil, 2022.	
8	**ALVES, N. da C.**; GRESSE von WANGENHEIM, C.; MARTINS-PACHECO, L. H.; BORGATTO, A. F. Are computational artifacts considered creative? (*in Brazilian Portuguese: Artefatos computacionais são considerados criativos*?) In: Proc. of the 2nd Brazilian Symposium on Computing Education (EDUCOMP22), Feira de Santana, Bahia/Brazil, 2022.	B3
9	**ALVES, N. da C.**; GRESSE von WANGENHEIM, C.; HAUCK, J. C. R.; BORGATTO, A. F. An Item Response Theory Analysis of Algorithms and Programming Concepts in App Inventor Projects. In: Proc. of the 1st Brazilian Symposium on Computing Education (EDUCOMP21), Jataí, Goiás/Brazil, 2021.	B3
10	**ALVES, N. da C.**, GRESSE von WANGENHEIM, C., MARTINS-PACHECO, L. H., BORGATTO, A. F. Is there agreement and reliability in the assessment of creativity of tangible results of computing education in K-12? (*in Brazilian Portuguese: Existem concordância e confiabilidade na avaliação da criatividade de resultados tangíveis da aprendizagem de computação na Educação Básica?*) In: Proc. of the 1st Brazilian Symposium on Computing Education (EDUCOMP21), Jataí, Goiás/Brazil, 2021.	B3
11	**ALVES, N. da C**.; ALBERTO, M.; GRESSE von WANGENHEIM, C. Automated Analysis of Originality of Android Apps in Educational Context: A Literature Mapping (*in Brazilian Portuguese: Análise Automatizada da Originalidade de Aplicativos Android no Contexto Educacional: Um Mapeamento da Literatura*). In: Proc. of the 29th Brazilian Workshop on Computing Education (WEI21), online, 2021.	A4
12	DE SOUZA, A., **ALVES, N. da C.**, GRESSE von WANGENHEIM, C., KREUCH, L. Automated Analysis of the Originality of User Interface Design in Educational Context: A Literature Mapping (*in Brazilian Portuguese: Análise Automatizada da Originalidade de Design de Interfaces de Usuário no Contexto Educacional: Um Mapeamento da Literatura)*. In: Proc. of the 32nd Brazilian Symposium of Informatics in Education (SBIE21), online, 2021.	A3
13	FERREIRA, M. N. F.; GRESSE von WANGENHEIM, C.; **ALVES, N. da C.** Development of an online course to teach User Interface Design in K-12 Education (*in Brazilian Portuguese: Desenvolvimento de um Curso on-line para Ensinar Design de Interface de Usuário na Educação Básica*) In: Proc. of the 20th Brazilian Symposium on Human Factors in Computing Systems (IHC21) - Workshop on Human-Computer Interaction Education, online, Brazil, 2021.	A3
14	**ALVES, N. da C.**, GRESSE von WANGENHEIM, C., HAUCK, J. C. R., BORGATTO, A. F., ANDRADE, D. F. An Item Response Theory Analysis of the Sequencing of Algorithms & Programming Concepts. In: Proc. of the 4th International Conference on Computational Thinking (CTE20), Hong Kong, 2020.	B2
15	**ALVES, N. da C.**, GRESSE von WANGENHEIM, C., HAUCK, J. C. R., BORGATTO, A. F. A Large-scale Evaluation of a Rubric for the Automatic Assessment of Algorithms and Programming Concepts. In: Proc. of the 51st ACM Technical Symposium on Computer Science Education (SIGCSE20), Portland/USA, 2020.	A2
16	SOLECKI, I.; PORTO, J. A.; **ALVES, N. da C.**, GRESSE von WANGENHEIM, C., HAUCK, J. C. R., BORGATTO, A. F. Automated Assessment of the Visual Design of Android Apps Developed with App Inventor. In: Proc. of the 51st ACM Technical Symposium on Computer Science Education (SIGCSE20), Portland, USA, 2020.	A2
17	**ALVES, N. da C.**, GRESSE von WANGENHEIM, C., ALBERTO, M., MARTINS-PACHECO, L. H. A Proposal for Assessing Product Originality in Teaching Algorithms and Programming in K-12 Education (*in Brazilian Portuguese: Uma Proposta de Avaliação da Originalidade do Produto no Ensino de Algoritmos e Programação na Educação Básica*). In: Proc. of the 31st Brazilian Symposium of Informatics in Education (SBIE20), online, 2020.	A3
18	**ALVES, N. da C.**, GRESSE von WANGENHEIM, C., HAUCK, J. C. R. Teaching Programming to Novices: A Large-scale Analysis of App Inventor Projects. In: Proc. of the 15th Latin-American Conference on Learning Technologies, Loja/Ecuador, 2020.	B3
19	**ALVES, N. da C.**, SOLECKI, I., GRESSE VON WANGENHEIM, C., BORGATTO, A. F. HAUCK, J. C. R., FERREIRA, M. N. F. Analysis of the Difficulty Level of User Interface Design Concepts using Item Response Theory (*in Brazilian Portuguese: Análise do Nível de Dificuldade*	A3

#	Reference	Qualis 2017-2020
	dos Conceitos de Design de Interface de Usuário usando a Teoria de Resposta ao Item) In: Proc. of the 31st Brazilian Symposium of Informatics in Education (SBIE20), online, 2020.	
20	**ALVES, N. da C.**, KRETZER, F., GRESSE von WANGENHEIM, C., FERREIRA, M. N. F.; HAUCK, J. C. R. Continued Training of in-Service K-12 Teachers for Teaching Algorithms and Programming (*in Brazilian Portuguese: Formação Continuada de Professores da Educação Básica para o Ensino de Algoritmos e Programação*). In: Proc. of the 31st Brazilian Symposium of Informatics in Education (SBIE20), online, 2020.	A3
21	**ALVES, N. da C.**, GRESSE von WANGENHEIM, C., HAUCK, J. C. R., BORGATTO, A. F., ANDRADE, D. F. An Analysis of Pedagogical Sequencing in the Teaching of Computing in K-12 Education (*in Brazilian Portuguese: Uma Análise do Sequenciamento Pedagógico no Ensino de Computação na Educação Básica*). Proc. of the 30th Brazilian Symposium of Informatics in Education (SBIE20), Brasília, DF/Brazil, 2020.	A3
22	MARTINS-PACHECO, L. H.; GRESSE von WANGENHEIM, C.; **ALVES, N. da C.** Polemics about Computational Thinking: Digital Competence in Digital Zeitgeist – Continued Search for Answers. In: Proc. of the 12th International Conference on Computer Supported Education (CSEDU20), Prague/Czech Republic, 2020.	A3
23	**ALVES, N. da C.**; GREESSE von WANGENHEIM, C.; HAUCK, J. C. R.; BORGATTO, A. F.; ANDRADE, D. F. CodeMaster: An Assessment Model of Computational Thinking in K-12 Education through Visual Programming Language Code Analysis (*in Brazilian Portuguese: CodeMaster: Um Modelo de Avaliação do Pensamento Computacional na Educação Básica através da Análise de Código de Linguagem de Programação Visual*). In: Proc. of the 10th Meeting of the Brazilian Association of Educational Assessment (ABAVE19). São Paulo, SP/Brazil, 2019.	C
24	**ALVES, N. da C.;** GREESSE von WANGENHEIM, C., HAUCK, J. C. R., BORGATTO, A. F., ANDRADE, D. F. An Analysis of the Guidelines for Teaching Computational Thinking proposed by SBC in K-12 Education (*in Brazilian Portuguese: Uma Análise das Diretrizes para Ensino de Pensamento Computacional propostas pela SBC na Educação Básica*). In: Proc. of the 10th Meeting of the Brazilian Association of Educational Assessment (ABAVE19). São Paulo, SP/Brazil, 2019.	C
25	SOLECKI, I., PORTO, J. A., JUSTEN, K. A., **ALVES, N. da C.**, GRESSE von WANGENHEIM, C., BORGATTO, A. F., HAUCK, J. C. R. CodeMaster UI Design – App Inventor: An UI Design Evaluation Rubric for Android Applications Developed with App Inventor (*in Brazilian Portuguese: CodeMaster UI Design – App Inventor: Uma Rubrica de Avaliação do Design de Interface de Aplicativos Android desenvolvidos com App Inventor*). In: Proc. of the 17th Brazilian Symposium on Human Factors in Computer Systems (IHC19), Vitória, ES/Brazil, 2019.	A3
26	MARTINS-PACHECO, L. H.; GRESSE von WANGENHEIM, C.; **ALVES, N. da C.** Assessment of Computational Thinking in K-12 Context: Educational Practices, Limits and Possibilities – A Systematic Mapping Study. In: Proc. of the 11th International Conference on Computer Supported Education (CSEDU19), Heraklion, Greece, 2019.	A3
	Book chapters	
27	GRESSE von WANGENHEIM, C.; **ALVES, N. da C.**; FORTUNA FERREIRA, M. N.; HAUCK, J. C. Creating Mobile Applications with App Inventor Adopting Computational Action. In Teaching Coding in K-12 Schools: Research and Application (pp. 305-318). Cham: Springer International Publishing. 2023.	NA
28	**ALVES, N. da C.**; GRESSE von WANGENHEIM, C.; HAUCK, J. C. R.; BORGATTO, A. F. Automating the Assessment of Algorithms and Programming Concepts in App Inventor Projects in Middle School. Handbook of Research on Tools for Teaching Computational Thinking in P-12 Education, eds. M. Kalogiannakis and S. Papadakis, IGI Global, 2020.	NA
29	MARTINS-PACHECO L. H.; **ALVES N. da C.**; GRESSE von WANGENHEIM C. Educational Practices in Computational Thinking: Assessment, Pedagogical Aspects, Limits, and Possibilities: A Systematic Mapping Study. In: Lane H.C., Zvacek S., Uhomoibhi J. (eds) Computer Supported Education. CSEDU 2019. Communications in Computer and Information Science, vol 1220. Springer. 2020.	NA
	Technical reports	

#	**Reference**	**Qualis 2017-2020**
30	ROSA, M. V. F.; GRESSE von WANGENHEIM; **ALVES, N. da C.** Development of a Originality Assessment Model for Mobile Applications Using Machine Learning Techniques (*in Brazilian Portuguese: Desenvolvimento de um Modelo de Avaliação da Originalidade de Aplicativos Móveis Usando Técnicas de Machine Learning*). INCoD-GQS.064.2021P, 2021.	NA
31	**ALVES, N. da C.**; GRESSE von WANGENHEIM; SILVA DE MEDEIROS, G. A.; HAUCK, J. C. R. Computational Thinking Skills in K-12 Education according to the National Common Curricular Base (BNCC) and SBC (*in Brazilian Portuguese: Habilidades do Pensamento Computacional na Educação Básica conforme Base Nacional Comum Curricular (BNCC) e SBC*). INCoD/GQS.02.2019.P, 2019.	NA

Source: Elaborated by the author.

8 CONCLUSION

This work presented a framework and an automated model for assessing the creativity of mobile apps. In general, the results of the evaluation show that the Creassessment model represents an instrument with acceptable reliability and validity that can be used for the assessment of the creativity of mobile apps created with App Inventor as part of computing education. To achieve this, a systematic process was followed, involving the collection of independent data. The projects from the MIT App Inventor App of the Month contest were selected to represent a diverse range of domains and complexity levels, as well as, the independent assessment of these projects by human experts from the App Inventor Foundation. These projects were also assessed regarding their creativity using the Creassessment. The results from the quantile regression and the Wilcoxon test comparing both assessments (human vs. automated via Creassessment) for each variable of the model demonstrated that there is a significant positive difference between the Creassessment creativity grades of winners and non-winners, i.e., the scores of winners are consistently higher than those of non-winners. Regarding reliability, a good value of the omega coefficient (ω=0.86) was obtained, and positive Pearson correlations for all items demonstrated convergent validity. Exploratory factor analysis revealed the presence of strong factor loading for two and three underlying factors in the model, indicating a reasonably good fit for these factors. The IRT analysis also indicated that the model has good discriminatory power.

Supported through the online tool CodeMaster, the automated support helps to ensure consistency and accuracy of assessment results as well as to eliminate bias. Furthermore, it can also reduce the teachers' workload and leave them free to spend more time on other activities with students as well as to conduct complementary assessments on factors that are not easily automated.

The model can be utilized by a wide range of individuals and institutions, by incorporating Creassessment in their teaching practices to evaluate and nurture students' creative abilities. Additionally, researchers and scholars can utilize the model to conduct rigorous studies and investigations on creativity. Furthermore, the implementation of Creassessment enhances the overall understanding of creativity in computing education. By assessing creativity, the model contributes to the accumulation of knowledge and the advancement of creative practices.

One potential avenue for future work is to extend the Creassessment model to identify and assess the originality of topics in other languages, e.g., English. Another direction is to perform longitudinal studies on creativity development to investigate the developmental trajectories of creativity using the Creassessment model. By tracking individuals' creative growth over time, researchers could examine how different factors, such as education, experiences, and interventions, influence the development of creativity. Considering also the existing creativity measurements, a potential future research endeavor could involve comparing the results obtained through the Creassessment model with other established measurements of creativity, such as the Torrance Tests of Creative Thinking (TORRANCE, 2008) and the Creative Solution Diagnosis Scale (CROPLEY and KAUFMAN, 2012). In addition, considering the person strand, personality traits could be studied using the Big Five model, as well as, the potential for creativity in domains using the Kaufman Domains of Creativity Scale (KAUFMAN, 2012). By conducting comparative studies, researchers could gain insights into the convergent and divergent validity of the Creassessment model and its compatibility with existing creativity assessment tools. This comparative analysis would allow for a deeper understanding of the strengths and limitations of each measurement approach, highlighting their unique contributions to assessing creativity. Additionally, investigating the relationships between the outcomes of different measurements could provide a more comprehensive and nuanced understanding of individuals' creative capabilities and provide valuable insights for educators, researchers, and practitioners in designing effective strategies to nurture and harness creativity.

REFERENCES

ALVES, N. D. C. **Creassessment - A Python Package for Assessing the Creativity of App Inventor Projects**. 2023. <https://pypi.org/project/creassessment/>.

ALVES, N. D. C. et al. **Formação Continuada de Professores da Educação Básica para o Ensino de Algoritmos e Programação**. Anais do Simpósio Brasileiro de Informática na Educação. Natal/RN: SBC. 2020.

ALVES, N. D. C. et al. **Uma Proposta de Avaliação da Originalidade do Produto no Ensino de Algoritmos e Programação na Educação Básica**. Anais do Simpósio Brasileiro de Informática na Educação. Natal/RN: SBC. 2020.

ALVES, N. D. C. et al. **Existem concordância e confiabilidade na avaliação da criatividade de resultados tangíveis da aprendizagem de computação na Educação Básica?** Anais do Simpósio Brasileiro de Educação em Computação. Jataí/GO: SBC. 2021c.

ALVES, N. D. C. et al. **Artefatos computacionais são considerados criativos?** Anais do Simpósio Brasileiro de Educação em Computação. Feira de Santana/BA: SBC. 2022. p. 01-09.

ALVES, N. D. C.; ALBERTO, M.; GRESSE VON WANGENHEIM, C. **Análise Automatizada da Originalidade de Aplicativos Android no Contexto Educacional:** Um Mapeamento da Literatura. Anais do 29º WEI – Workshop sobre Educação em Computação. Florianópolis/SC: SBC. 2021b.

ALVES, N. D. C.; GRESSE VON WANGENHEIM, C. **O ensino de design thinking ajuda no desenvolvimento de aplicativos originais no contexto do ensino de computação?** Anais do XXXIII Simpósio Brasileiro de Informática na Educação. Manaus/AM: SBC. 2022. p. 1268-1280.

ALVES, N. D. C.; GRESSE VON WANGENHEIM, C. **Uma Análise em Larga-Escala das Funcionalidades de Aplicativos criados com App Inventor**. Anais do Simpósio Brasileiro de Educação em Computação. Recife/PE: SBC. 2023.

ALVES, N. D. C.; GRESSE VON WANGENHEIM, C.; HAUCK, J. C. R. . **Teaching Programming to Novices:** A Large-scale Analysis of App Inventor Projects. 15th Latin-American Conference on Learning Technologies. Loja/Ecuador: IEEE. 2020.

ALVES, N. D. C.; GRESSE VON WANGENHEIM, C.; MARTINS-PACHECO, L. H. Assessing Product Creativity in Computing Education: A Systematic Mapping Study. **Informatics in Education**, v. 20, n. 1, p. 19-45, 2021a.

ALVES, N. D. C.; KREUCH, L.; GRESSE VON WANGENHEIM, C. **Analyzing Structural Similarity of User Interface Layouts of Android Apps using Deep Learning**. 21st Brazilian Symposium on Human Factors in Computing Systems. Diamantina/MG: SBC. 2022.

AMABILE, T. M. **The context of creativity**. Boulder, CO: Westview, 1996.

AMERSHI, S. et al. **Software engineering for machine learning:** A case study. International Conference on Software Engineering: Software Engineering in Practice. Montreal, QC, Canada: IEEE. 2019. p. 291-300.

ARMONI, M. Computer science, computational thinking, programming, coding: the anomalies of transitivity in K-12 computer science education. **ACM Inroads**, v. 7, n. 4, p. 24–27, 2016.

BAER, J. **Domain specificity of creativity**. San Diego: Academic Press, 2015.

BAER, J.; KAUFMAN, J. C. Bridging generality and specificity: The amusement park theoretical (APT) model of creativity. **Roeper Review**, v. 27, p. 158–163, 2005.

BAER, J.; KAUFMAN, J. C.; GENTILE, C. A. Extension of the consensual assessment technique to nonparallel creative products. **Creativity Research Journal**, v. 16, p. 113-117, 2004.

BARBOT, B. The Dynamics of Creative Ideation: Introducing a New Assessment Paradigm. **Frontiers in Psychology**, v. 9, 2018.

BASILI, V. R.; CALDIERA, G.; ROMBACH, H. D. Goal Question Metric Paradigm. In: MARCINIAK, J. J. **Encyclopedia of Software Engineering**. John Wiley & Sons, 1994.

BASU, S. **Using Rubrics Integrating Design and Coding to Assess Middle School Students' Open-ended Block-based Programming Projects**. Proc. of the 50th ACM Technical Symposium on Computer Science Education. Minneapolis: ACM. 2019. p. 1211–1217.

BATTELLE FOR KIDS. 21st Century Learning for Early Childhood Framework. **P21**, 2019. <http://static.battelleforkids.org/documents/p21/P21EarlyChildhoodFramework.pdf>.

BATTELLE FOR KIDS. **Framework for 21st Century Learning Definitions**. Battelle for Kids. Hilliard, Ohio, p. 9. 2019.

BEATY, R. E.; JOHNSON, D. R. Automating creativity assessment with SemDis: An open platform for computing semantic distance. **Behavior Research Methods**, 53, 2020. 757–780.

BEGHETTO, R. A. Creativity in the classroom. In: KAUFMAN, J. C.; STERNBERG, R. J. **Cambridge handbook of creativity**. New York: Cambridge University Press, 2010.

BEGHETTO, R. A.; KAUFMAN, J. C. Toward a broader conception of creativity: A case for mini-c creativity. **Psychology of Aesthetics, Creativity, and the Arts**, v. 1, p. 73–79, 2007.

BEKETAYEV, K.; RUNCO, M. A. Scoring divergent thinking tests by computer with a semantics-based algorithm. **Europe's Journal of Psychology**, v. 12, p. 210–220, 2016.

BENNETT, V. E.; KOH, K. H.; REPENNING, A. **CS education re-kindles creativity in public schools**. Proc. of the 16th Annual Joint Conference on Innovation and Technology in Computer Science Education. Darmstadt, Germany: Association for Computing Machinery. 2011. p. 183-187.

BENNETT, V. E.; KOH, K. H.; REPENNING, A. **Computing creativity:** divergence in computational thinking. Proc. of the 44th ACM Technical Symposium on Computer Science Education. Denver: ACM. 2013. p. 359–364.

BIALIK, M. et al. **Evolving Assessments for a 21st Century Education**. Center for Curriculum Redesign. 2016.

BRENNER, W.; UBERNICKEL, F. **Design thinking for innovation research and practice**. New York: Springer, 2016.

BROWN, T. Design thinking. **Harvard Business Review**, v. 86, n. 5, p. 84–92, 2008.

BROWN, T. A. **Confirmatory factor analysis for applied research**. New York: Press, 2006.

CAMERON, W. B. **Informal sociology A casual introduction to sociological thinking**. New York: Random House, 1963.

CAMPOS, R. et al. YAKE! Keyword Extraction from Single Documents using Multiple Local Features. **Information Sciences Journal**, v. 509, p. 257-289, 2020.

CAPES; MEC. **Plataforma Sucupira**. CAPES Foundation - Ministry of Education - Brazil. Brasília. 2023.

CLEMENTS, D. H. Teaching creativity with computers. **Educ Psychol Rev**, v. 7, p. 141–161, 1995.

CLEMENTS, D. H.; GULLO, D. F. Effects of computer programming on young children's cognition. **Journal of Educational Psychology**, v. 76, n. 6, p. 1051–1058, 1984.

CNE. **CodeMaster**, 2023. <http://apps.computacaonaescola.ufsc.br/codemaster/>.

COMER, D. E. et al. Computing as a discipline. **Communications of the ACM**, v. 32, n. 1, p. 9–23, 1989.

COUGER, J. D.; DANGATE, G. Measurement of creativity of I.S. products. **Creativity and Innovation Management**, v. 5, n. 4, p. 262–272, 1996.

CROPLEY, D. H.; KAUFMAN, J. C. Measuring functional creativity: Non-expert raters and the Creative Solution Diagnosis Scale. **The Journal of Creative Behavior**, v. 46, n. 2, p. 119–137, 2012.

CSTA. **K-12 Computer Science Framework**, 2016. <https://k12cs.org/>.

CSTA. K-12 computer science standards. **CS Standards**, 2017. <www.csteachers.org/standards>.

DE SOUZA, A. et al. **Análise Automatizada da Originalidade de Design de Interfaces de Usuário no Contexto Educacional:** Um Mapeamento da Literatura. Anais do Simpósio Brasileiro de Informática na Educação. Online: SBC. 2021. p. 1140-1151.

DEAN, D. L. et al. Identifying Good Ideas: Constructs and Scales for Idea Evaluation. **Journal of Association for Information Systems**, v. 7, n. 10, p. 646-699, 2006.

DEVELLIS, R. F. **Scale development:** theory and applications. 4th. ed. Thousand Oaks: SAGE, 2017.

DOUSAY, T. A. Designing for Creativity in Interdisciplinary Learning Experiences. In: PERSICHITTE, K.; SUPARMAN, A.; SPECTOR, M. **Educational Technology to Improve Quality and Access on a Global Scale**. International Publishing: Springer, Cham, 2018.

DUMAS, D. Relational reasoning and divergent thinking: An examination of the threshold hypothesis with quantile regression. **Contemporary Educational Psychology**, n. 53, p. 1-14, 2018.

DUMAS, D.; ORGANISCIAK, P.; DOHERTY, M. Measuring divergent thinking originality with human raters and text-mining models: A psychometric comparison of methods. **Psychology of Aesthetics, Creativity, and the Arts**, v. 15, n. 4, p. 645–663, 2021.

ENGELMAN, S. et al. **Creativity in Authentic STEAM Education with EarSketch**. Proc. of the Technical Symposium on Computer Science Education. Seattle: ACM. 2017. p. 183–188.

FALKNER, K.; SHEARD, J. Pedagogic Approaches. In: FINCHER, S. A. .; ROBINS, A. V. **The Cambridge Handbook of Computing Education Research**. Cambridge: University Press, 2019.

FERREIRA, M. N. F. et al. Learning user interface design and the development of mobile applications in middle school. **ACM Interactions**, v. 26, n. 4, p. 66–69, 2019.

FINCHER, S. A. .; ROBINS, A. V. **The Cambridge handbook of computing education research**. Cambridge: University Press, 2019.

FLORA, D. B. Your Coefficient Alpha Is Probably Wrong, but Which Coefficient Omega Is Right? A Tutorial on Using R to Obtain Better Reliability Estimates. **Advances in Methods and Practices in Psychological Science**, v. 3, n. 4, p. 484–501, 2020.

GAL, L. et al. **Suggesting a log-based creativity measurement for online programming learning environment**. Proc. of the 4th Conference on Learning at Scale. Cambridge, Massachusetts, USA: ACM. 2017. p. 273-277.

GÉRON, A. **Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow**. O'Reilly Media, Inc, 2019.

GLASS, R. L. **Software creativity**. Upper Saddle River: Prentice-Hall, Inc., 1995.

GLăVEANU, V. P. Rewriting the language of creativity: The Five A's framework. **Review of general psychology**, v. 17, n. 1, p. 69-81, 2013.

GLORFELD, L. W. An improvement on Horn’s parallel analysis methodology for selecting the correct number of factors to retain. **Educational and Psychological Measurement**, v. 55, n. 3, p. 377-393, 1995.

GRESSE VON WANGENHEIM, C. et al. CodeMaster – Automatic Assessment and Grading of App Inventor and Snap! Programs. **Informatics in Education**, v. 17, n. 1, p. 117-150, 2018.

GRESSE VON WANGENHEIM, C. et al. Creating Mobile Applications with App Inventor Adopting Computational Action. In: KEANE, T.; FLUCK, A. E. **Teaching Coding in K-12 Schools:** Research and Application. Cham: Springer International Publishing, 2023. p. 305-318.

GROENEVELD, W. et al. **Are Undergraduate Creative Coders Clean Coders? A Correlation Study**. Proceedings of the 53rd ACM Technical Symposium on Computer Science Education. Providence, RI, USA: ACM. 2022. p. 314–320.

GROVER, S.; BASU, S.; SCHANK, P. **What we can learn about student learning from open-ended programming projects in middle school computer science**. Proc. of the 49th Technical Symposium on Computer Science Education. Baltimore: ACM. 2018. p. 999-1004.

GROVER, S.; PEA, R. Computational Thinking in K–12: A review of the state of the field. **Educational Researcher**, v. 42, n. 1, p. 38–43, 2013.

GU, M.; TONG, X. Towards Hypotheses on Creativity in Software Development. In: BOMARIUS, F.; IIDA, H. **Product Focused Software Process Improvement. Lecture Notes in Computer Science**. Berlin: Springer, 2004. p. 47-61.

GUILFORD, J. P. Creativity. **American Psychologist**, v. 5, 1950.

HAIR, J. F. et al. **Multivariate data analysis**. 7th. ed. Prentice-Hall, 2009.

HAUCK, J. C. R. et al. Jovens tutores de programação: um relato de experiência. **Revista Eletônica de Extensão UFSC**, Florianópolis, v. 15, n. 29, p. 94-108, 2018.

HAYES, A. F.; COUTTS, J. J. Use omega rather than Cronbach's alpha for estimating reliability. But…. **Communication Methods and Measures**, v. 14, n. 1, p. 1-24, 2020.

HEDRICK, T. E.; BICKMAN, L.; ROG, D. J. **Applied research design:** A practical guide. Sage Publications, 1993.

HENRIKSEN, D.; MISHRA, P.; MEHTA, R. Novel, Effective, Whole: Toward a NEW Framework for Evaluations of Creative Products. **Journal of Technology and Teacher Education**, v. 23, n. 3, p. 455-478, 2019.

HERSHKOVITZ, A. et al. Creativity in the acquisition of computational thinking. **Interactive Learning Environments**, v. 27, n. 5-6, p. 628-644, 2019.

HOLINGER, M. et al. Taking a prospective look at creativity domains. In: KAUFMAN, J.; GLăVEANU, V.; BAER, J. **The Cambridge Handbook of Creativity across Domains**. Cambridge: Cambridge University Press, 2017.

INEP; MEC. **Sinopse Estatística da Educação Superior 2018**. Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira. Brasília. 2019.

INEP; MEC. **Sinopse Estatística da Educação Superior 2019**. Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira. Brasília. 2020.

INEP; MEC. **Sinopse Estatística da Educação Superior 2020**. Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira. Brasília. 2021.

INEP; MEC. **Sinopse Estatística da Educação Superior 2021**. Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira. Brasília. 2022.

ISRAEL-FISHELSON, R. et al. **Computational Thinking and Creativity:** A Test for Interdependency. Proc. of the 4th International Conference on Computational Thinking Education. Hong Kong: The Education University of Hong Kong. 2020.

ISRAEL-FISHELSON, R. et al. A log-based analysis of the associations between creativity and computational thinking. **Journal of Educational Computing Research**, v. 59, n. 5, p. 926-959, 2021.

ISRAEL-FISHELSON, R. et al. The associations between computational thinking and creativity: The role of personal characteristics. **Journal of Educational Computing Research**, v. 58, n. 8, p. 1415-1447, 2021.

ISTE. International Society for Technology in Education Standards, 2020. <http://www.iste.org/>.

JACKSON, P. W.; MESSICK, S. The Person, The Product, and the Response: Conceptual Problems in the Assessment of Creativity. **ETS Research Bulletin Series**, v. 2, 1964.

JANSSON, D. G.; SMITH, S. M. Design fixation. **Design Studies**, v. 12, n. 1, p. 3-11, 1991.

KAUFMAN, J. C. Counting the muses: Development of the Kaufman Domains of Creativity Scale (K-DOCS). **Psychology of Aesthetics, Creativity, and the Arts**, v. 6, n. 4, p. 298–308, 2012.

KAUFMAN, J. C.; BAER, J. Beyond new and appropriate Who decides what is creative? **Creativity Research Journal**, v. 24, n. 1, p. 83-91, 2012.

KAUFMAN, J. C.; BEGHETTO, R. A. Beyond Big and Little: The Four C Model of Creativity. **Review of General Psychology**, v. 13, n. 1, p. 1-12, 2009.

KAUFMAN, J. C.; PLUCKER, J. A.; BAER, J. **Essentials of creativity assessment**. John Wiley & Sons, 2008.

KERSHAW, T. C. et al. An initial examination of computer programs as creative works. **Psychology of Aesthetics, Creativity, and the Arts**, 2022.

KHAWAS, P. et al. **Unmixing Remixes:** The How and Why of Not Starting Projects from Scratch. Proc. of the IEEE Symposium on Visual Languages and Human-Centric Computing. Memphis, TN, USA: IEEE. 2019. p. 169-173.

KIESLER, N. **Reviewing Constructivist Theories to Help Foster Creativity in Programming Education**. IEEE Frontiers in Education Conference (FIE). Uppsala, Sweden: IEEE. 2022. p. 1-5.

KOENKER, R. **Quantile regression**. NY: Cambridge University Press, 2005.

KOH, K. H.; BENNETT, V.; REPENNING, A. **Computing indicators of creativity**. Proc. of the 8th Conference on Creativity and Cognition. Atlanta: ACM. 2011. p. 357–358.

KOVALKOV, A. et al. Automatic Creativity Measurement in Scratch Programs Across Modalities. **IEEE Transactions on Learning Technologies**, v. 14, n. 6, p. 740-753, 2021.

KOVALKOV, A.; SEGAL, A.; GAL, K. **Inferring Creativity in Visual Programming Environments**. Proceedings of the Seventh ACM Conference on Learning @ Scale. New York, NY, USA: ACM. 2020. p. 269–272.

KREUCH, L. **Desenvolvimento de um Modelo de Avaliação da Originalidade do Esqueleto de Design de Interface de Aplicativos Android**. Universidade Federal de Santa Catarina. Undergraduate thesis. Florianópolis. 2022.

KYLLONEN, P.; WALTERS, A. M.; KAUFMAN, J. C. Noncognitive constructs and their assessment in graduate education- A review. **Educational Assessment**, v. 10, n. 3, p. 153-184, 2005.

LARMAN, C.; BASILI, V. Iterative and incremental developments: a brief history. **IEEE Computer**, v. 36, p. 47–56, 2003.

LEE, I. et al. Computational thinking for youth in practice. **ACM Inroads**, v. 2, n. 1, p. 32–37, 2011.

LUO, J.; LU, F.; WANG, T. **A Multi-Dimensional Assessment Model and Its Application in E-learning Courses of Computer Science**. Proceedings of the 21st Annual Conference on Information Technology Education. Virtual Event, USA: ACM. 2020. p. 187-193.

LYE, S. Y.; KOH, J. H. L. Review on teaching and learning of computational thinking through programming: What is next for K-12? **Computers in Human Behavior**, v. 41, p. 51–61, 2014.

LYTLE, N. et al. **Use, Modify, Create:** Comparing Computational Thinking Lesson Progressions for STEM Classes. Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '19). New York, NY, USA: ACM. 2019.

MANSKE, S.; HOPPE, H. U. **Automated indicators to assess the creativity of solutions to programming exercises**. Proc. of the 14th Int. Conference on Advanced Learning Technologies. Athens: IEEE. 2014. p. 497-501.

MATHIJSSEN, P. AIA file structure. **App Inventor Community**, 2019. <https://community.appinventor.mit.edu/t/aia-file-structure/219>.

MEC. **Base Nacional Comum Curricular**. Ministry of Education - Brazil. Brasília. 2018.

MILLER, S. R.; BAILEY, B. P.; KIRLIK, A. Exploring the Utility of Bayesian Truth Serum for Assessing Design Knowledge. **Human–Computer Interaction**, v. 29, n. 5-6, 2014.

MISHRA, P.; HENRIKSEN, D. A NEW Approach to Defining and Measuring Creativity: Rethinking Technology & Creativity in the 21st Century. **TECHTRENDS TECH TRENDS**, v. 57, p. 10-13, 2013.

MISHRA, P.; YADAV, A. Of art and algorithms: Rethinking technology and creativity in the 21st century. **TechTrends** , v. 57, n. 3, p. 10–14, 2013.

MISLEVY, R. J.; ALMOND, R. G.; LUKAS, J. F. A brief introduction to evidence-centered design. **ETS Research Report Series**, v. 1, p. i-29, 2003.

MISLEVY, R. J.; HAERTEL, G. D. Implications of evidence-centered design for educational testing. **Educational measurement: issues and practice**, v. 25, n. 4, p. 6-20, 2006.

MIT. MIT App Inventor. **MIT App Inventor**, 2012. <https://appinventor.mit.edu/>.

MIT APP INVENTOR. Tutorials – Space Invaders. **App Inventor**, 2022a. <https://appinventor.mit.edu/explore/ai2/space-invaders>.

MIT APP INVENTOR. Beginner Tutorials – Hello Codi. **App Inventor**, 2022b. <http://appinventor.mit.edu/explore/ai2/hello-codi.html>.

MIT APP INVENTOR. Beginner Tutorials. **App Inventor**, 2022c. <https://appinventor.mit.edu/explore/ai2/beginner-videos>.

MIT APP INVENTOR. Tutorials – Golf. **App Inventor**, 2022d. <https://appinventor.mit.edu/explore/ai2/minigolf>.

MIT APP INVENTOR. Tutorials – Mole Mash. **App Inventor**, 2022e. <https://appinventor.mit.edu/explore/ai2/molemash>.

MIT APP INVENTOR. App of the Month Winners. **MIT App Inventor**, 2023. <https://appinventor.mit.edu/explore/app-month-gallery>.

MIT APP INVENTOR FOUNDATION. About Us. 2023. <https://www.appinventorfoundation.org/about>.

MORENO-LEÓN, J.; ROBLES, G. **Dr. Scratch:** a web tool to automatically evaluate Scratch projects. Proc. of the 10th Workshop in Primary and Secondary Computing Education. London, UK: ACM. 2015. p. 132–133.

MUSTAFARAJ, E.; TURBAK, F.; SVANBERG, M. **Identifying Original Projects in App Inventor**. Proc. of the 30th Int. Florida Artificial Intelligence Research Society Conference. Florida: Association for the Advancement of Artificial Intelligence. 2017. p. 567-572.

MYSZKOWSKI, N.; STORME, M. Judge response theory? A call to upgrade our psychometrical account of creativity judgments. **Psychology of Aesthetics, Creativity, and the Arts**, v. 13, n. 2, p. 167-175, 2019.

P21. 21st century skills, 2020. <http://www.p21.org/>.

PATTERSON, J. D. et al. AuDrA: An automated drawing assessment platform for evaluating creativity. **PsyArXiv**, 2022.

PATTON, E. W.; TISSENBAUM, M.; HARUNANI, F. MIT App Inventor: Objects, Design, and Development. In: KONG, S. C.; ABELSON, H. **Computational Thinking Education**. Nova Iorque/EUA: Springer Nature, 2019.

PEREZ-POCH, A. et al. On the influence of creativity in basic programming learning at a first-year Engineering course. **International Journal of Engineering Education**, v. 32, n. 5B, p. 2302-2309.

PETERSEN, K. et al. **Systematic mapping studies in software engineering**. 12th international conference on Evaluation and Assessment in Software Engineering. Swindon, UK: BCS Learning & Development Ltd. 2008. p. 68-77.

PETERSEN, K.; VAKKALANKA, S.; KUZNIARZ, L. Guidelines for conducting systematic mapping studies in software engineering: An update. **Information and Software Technology**, v. 64, p. 1-18, 2015.

PISA - OECD. **PISA 2021 Creative Thinking Framework (Third Draft)**. Programme for International Student Assessment. 2021.

PLUCKER, J. A.; BEGHETTO, R. A. Why creativity is domain general, why it looks domain specific, and why the distinction does not matter. In: STERNBERG, R. J.; GRIGORENKO, E. L.; SINGER, J. L. **Creativity:** From Potential to Realization. American Psychological Association, 2004. p. 153–167.

PLUCKER, J.; BEGHETTO, R. A.; DOW, G. Why isn't creativity more important to educational psychologists? Potential, pitfalls, and future directions in creativity research. **Educational Psychologist**, v. 39, p. 83–96, 2004.

RENNIE, J. D. et al. **Tackling the poor assumptions of naive Bayes text classifiers**. Proceedings of the 20th international conference on machine learning (ICML-03). Washington/DC: ICML. 2003. p. 616-623.

RENZULLI, J. The Malleability of Creativity: A Career in Helping Students Discover and Nurture Their Creativity. In: STERNBERG, R.; KAUFMAN, J. **The Nature of Human Creativity**. Cambridge: Cambridge University Press, 2018. p. 209-223.

RHODES, M. An Analysis of Creativity. **The Phi Delta Kappan**, v. 42, n. 7, p. 305-310, 1961.

RICONSCENTE, M. M.; MISLEVY, R. J.; HAMEL, L. **An introduction to PADI task templates**. SRI International. Menlo Park, Califórnia, p. 62. 2005.

RITCHIE, G. **Assessing Creativity**. Proceedings of the AISB symposium on AI and creativity in arts and science. York: The Society for the Study of Artificial Intelligence and Simulation of Behaviour. 2001. p. 3–11.

ROBERTSON, J. Requirements analysts must also be inventors. **IEEE Software**, v. 22, n. 1, p. 48-50, 2005.

RODE, J. A. et al. **From computational thinking to computational making**. Proc. of the Int. Joint Conference on Pervasive and Ubiquitous Computing. Osaka: ACM. 2015. p. 239-250.

ROMERO, M.; LEPAGE, A.; LILLE, B. Computational thinking development through creative programming in higher education. **International Journal of Educational Technology in Higher Education**, v. 14, n. 1, 2017.

RUNCO, M. A.; ALBERT, R. S. Creativity research: A historical view. In: KAUFMAN, J. C.; STERNBERG, R. J. **The Cambridge handbook of creativity**. Cambridge: Cambridge University Press, 2010. p. 3–19.

SAMEJIMA, F. A. Estimation of latent ability using a response pattern of graded scores. **Psychometric Monograph**, v. 17, 1969.

SAUNDERS, M.; LEWIS, P.; THORNHILL, A. **Research Methods for Business Students**. 8. ed. New York: Pearson, 2019.

SBC. **Diretrizes para ensino de Computação na Educação Básica**. Sociedade Brasileira de Computação. RS. 2018.

SCHMITT, T. M. **Refatoração da Ferramenta CodeMaster**. Federal University of Santa Catarina. Undergraduate thesis. Florianópolis. 2022.

SEERATAN, K.; MISLEVY, R. **Design patterns for assessing internal knowledge representations**. SRI International. Menlo Park, Califórnia. 2008.

SHELL, D. F. et al. **Improving learning of computational thinking using computational creativity exercises in a college CSI computer science course for engineers**. Proc. of the Frontiers in Education Conference. Madrid: IEEE. 2014. p. 1-7.

SHUTE, V. J.; SUN, C.; ASBELL-CLARKE, J. Demystifying computational thinking. **Educational Research Review**, v. 22, p. 142-158, 2017.

SILVIA, P. J. Creativity and intelligence revisited: A latent variable analysis of Wallach and Kogan (1965). **Creativity Research Journal**, v. 20, n. 1, p. 34–39, 2008.

SJøBERG, D. I. K. et al. Building theories in software engineering. In: SHULL, F.; SINGER, J.; SJøBERG, D. I. K. **Guide to Advanced Empirical Software Engineering**. London: Springer, v. 1, 2008. p. 312–336.

SOUZA, A. S. D. **Uma Abordagem para Avaliação de Originalidade de Design de Interface de Usuário de Aplicativos Móveis utilizando Técnicas de Inteligência Artificial para a Educação Básica**. Universidade Federal de Santa Catarina. Master thesis. Florianópolis. 2022.

STERNBERG, R. J. Teaching for creativity: The sounds of silence. **Psychology of Aesthetics, Creativity, and the Arts**, v. 9, n. 2, p. 115–117, 2015.

TISSENBAUM, M.; SHELDON, J.; ABELSON, H. From Computational Thinking to Computational Action. **Comm. of the ACM**, v. 62, n. 3, p. 34-36, 2019.

TORRANCE, E. P. Predicting the creativity of elementary school children (1958-80) — and the teacher who "made a difference". **Gifted Child Quarterly**, v. 25, n. 2, p. 55–62, 1981.

TORRANCE, E. P. **The Torrance Tests of Creative Thinking Norms— Technical Manual Figural (Streamlined) Forms A & B**. Bensenville: IL: Scholastic Testing Service, 2008.

TURBAK, F. et al. **Work in progress - Identifying and analyzing original projects in an open-ended blocks programming environment**. Proc. of the 23rd Int. Conference on Distributed Multimedia Systems, Visual Languages and Sentient Systems. Florida: Association for the Advancement of Artificial. 2017. p. 115-117.

UNAHALEKHAKA, A.; BERS, M. U. Evaluating young children's creative coding: rubric development and testing for ScratchJr projects. **Education and Information Technologies**, v. 27, n. 5, p. 6577–6597, 2022.

UYSAL, M. P. Improving first computer programming experiences: The case of adapting a web-supported and well-structured problem-solving method to a traditional course. **Contemporary Educational Technology**, v. 5, n. 3, p. 198-217, 2014.

VAN LAAR, E. et al. Determinants of 21st-Century Skills and 21st-Century Digital Skills for Workers: A Systematic Literature Review. **SAGE Open**, v. January-March, p. 1–14, 2020.

VOOGT, J.; ROBLIN, N. P. A comparative analysis of international frameworks for 21st century competences: Implications for national curriculum policies. **Journal of Curriculum Studies,** v. 44, n. 3, p. 299–321, 2012.

WALIA, C. A Dynamic Definition of Creativity. **Creativity Research Journal**, v. 31, n. 3, p. 237-247, 2019.

WARD, W. C.; WARREN, P. A. A field study of nonverbal creativity. **ETS Research Bulletin Series**, 1971.

WASSERMAN, A. I. **Software engineering issues for mobile application development**. Proceedings of the FSE/SDP workshop on Future of software engineering research. New York: ACM Press. 2010. p. 397-400.

WING, J. Computational thinking. **Comm. of the ACM**, v. 49, n. 3, p. 33–36, 2006.

YADAV, A.; COOPER, S. Fostering Creativity Through Computing. **Comm. of the ACM**, v. 60, n. 2, p. 31-33, 2017.

YANG, J. . R. K. Y. . E. B. **A Study of Effectiveness and Problem Solving on Security Concepts with Model-Eliciting Activities**. IEEE Frontiers in Education Conference (FIE). Uppsala, Sweden: IEEE. 2022. p. 1-9.

YIN, R. K. **Case study research - design and method**. 6th. ed. Thousand Oaks: Sage, 2017.

ZHONG, B. et al. An Exploration of Three-Dimensional Integrated Assessment for Computational Thinking. **Journal of Educational Computing Research**, v. 53, n. 4, p. 562-590, 2015.