UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
SISTEMAS DE INFORMAÇÃO

Matheus Alberto Pereira

DESENVOLVIMENTO DE UM MODELO DE AVALIAÇÃO DA ORIGINALIDADE DA FUNCIONALIDADE DE APPS DESENVOLVIDOS COM APP INVENTOR NA EDUCAÇÃO BÁSICA

Florianópolis
2022

Matheus Alberto Pereira

DESENVOLVIMENTO DE UM MODELO DE AVALIAÇÃO DA ORIGINALIDADE DA FUNCIONALIDADE DE APPS DESENVOLVIDOS COM APP INVENTOR NA EDUCAÇÃO BÁSICA

Trabalho de conclusão de curso apresentado como parte dos requisitos para obtenção do grau de Bacharel em Sistemas de informação.
Orientador(a): Prof. Dr. rer. nat. Christiane A. Gresse von Wangenheim, PMP
Coorientador(a): Prof. Nathalia Cruz Alves

Florianópolis
2022

Matheus Alberto Pereira

DESENVOLVIMENTO DE UM MODELO DE AVALIAÇÃO DA ORIGINALIDADE DA FUNCIONALIDADE DE APPS DESENVOLVIDOS COM APP INVENTOR NA EDUCAÇÃO BÁSICA

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de "Bacharelado" e aprovado em sua forma final pelo Curso de Sistemas de Informação

Florianópolis, 18 de Março de 2022.

_____
Prof. Álvaro Junio Pereira Franco
Coordenador do Curso

Banca Examinadora:

_____
Prof. Dr. rer. nat. Christiane A. Gresse von Wangenheim, PMP
Orientador(a)
Universidade Federal de Santa Catarina

_____
Prof. Nathalia da Cruz Alves
Coorientador(a)
Universidade Federal de Santa Catarina

_____
Prof. Dr. Mateus Grellert da Silva
Avaliador(a)
Universidade Federal de Santa Catarina

RESUMO

Com o avanço da tecnologia no século XXI, é necessária uma adaptação da população ao novo modo de se viver, seja no trabalho ou na sua vida pessoal e convivência em sociedade. Isto faz com que seja necessário o desenvolvimento de novas habilidades do século XXI. Dentre estas novas habilidades destaca-se a criatividade, cuja expressividade é amplamente relacionada com outras competências. A principal característica da criatividade é a novidade, que é a capacidade de criar algo novo ou encontrar novas soluções inovadoras. Uma das formas encontradas para estimular a criatividade é ensiná-la por meio da computação nas escolas, por exemplo, desenvolvendo apps com funcionalidades originais. Levando em consideração a importância de feedback ao aluno referente ao processo de aprendizagem, e mesmo que para a avaliação de aplicativos já existem propostas para analisar a originalidade em apps com App Inventor, esses focam somente na questão da similaridade do código e não da funcionalidade. Assim, o objetivo deste trabalho é criar um modelo de avaliação de *novelty*, focando na originalidade de funcionalidade, de projetos desenvolvidos com o App Inventor utilizando técnicas de Inteligência artificial, para ser utilizado no ensino de computação na educação básica. O modelo desenvolvido é integrado à ferramenta CodeMaster, para possibilitar a avaliar a originalidade dos apps desenvolvidos como resultados do processo de aprendizagem online. Espera-se contribuir para melhorar então a aprendizagem da habilidade de criatividade fornecendo feedback instrucional aos alunos e professores no processo de aprendizagem.

Palavras-chave: Criatividade, Novidade, Originalidade, Ensino de computação, App Inventor, Inteligência Artificial.

ABSTRACT

With the advancement of technology in the 21st century, it is necessary for the population to adapt to the new way of living, whether at work or in their personal life and living in society. This makes it necessary to develop new 21st century skills. Among these new skills, creativity stands out, whose expressiveness is largely related to other skills. The main characteristic of creativity is novelty, which is the ability to create something new or find new innovative solutions. One of the ways found to stimulate creativity is to teach it through computing in schools, for example, developing apps with original features. Considering the importance of feedback to the student regarding the learning process, and even though for the evaluation of applications there are already proposals to analyze the originality in apps with App Inventor, they focus only on the issue of code similarity and not functionality. Thus, the objective of this work is to create a novelty evaluation model, focusing on the originality of functionality, of projects developed with App Inventor using artificial intelligence techniques, to be used in computing teaching in basic education. The model developed is integrated with the CodeMaster tool, to make it possible to evaluate the originality of the apps developed as a result of the online learning process. It is expected to contribute to improve the learning of creativity skills by providing instructional feedback to students and teachers in the learning process.

Keywords: Creativity, Novelty, Originality, Teach computing, App Inventor, Artificial Intelligence.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

App - Aplicativo

GUI - *Graphical User Interface*

SUMÁRIO

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

A revolução da comunicação está transformando a natureza de como o trabalho é realizado e o significado das relações sociais (Binkley et al., 2010), com isso, novas habilidades se fazem necessárias para o desenvolvimento do cidadão, conhecidas como habilidades do século XXI. Habilidades como pensamento crítico, resolução de problemas e tarefas analíticas, devem ser ensinadas desde a infância. Também essencial para atender às novas demandas, é aplicar de forma criativa o conhecimento do conteúdo em situações novas (Binkley et al., 2010).

Dentre estas habilidades destaca-se a criatividade (P21, 2015), cuja expressividade é amplamente relacionada com outras competências, como o aprendizado independente e a solução inovadora de problemas. Criatividade é a capacidade de criar algo novo e útil. Assim, um dos fatores da criatividade é a novidade. Novidade refere-se à "extensão da novidade do produto", ou seja, qual o nível de novidade este produto apresenta. Ela pode abranger muitas características, incluindo originalidade e relação de paradigma (Dean et al., 2006). A originalidade se refere à “qualidade ou estado de ser original” (Merriam-Webster, 2020), a um produto pouco comum ou totalmente novo.

Uma das formas de estimular a originalidade é por meio do ensino da computação. O ensino da computação na educação básica visa popularizar essa competência, proporcionando dessa forma um conhecimento básico de computação para toda a população, pois a criatividade é uma das habilidades cruciais no século XXI. Além de simplesmente ensinar aos alunos como se tornar consumidores de tecnologia da informação, o ensino de computação pode formar futuros criadores inovadores que podem projetar sistemas de computador para melhorar a qualidade de vida das pessoas (Solecki et al., 2020). Com isso, a computação pode ser um meio eficaz para o desenvolvimento da criatividade.

Uma alternativa para ensinar computação por meio de desenvolvimento de apps na educação básica é o uso de ambientes de programação visual baseados em blocos, como o App Inventor. O App inventor é um ambiente online, que conta com mais de 400.000 usuários ativos mensalmente em 195 países (MIT, 2020). De acordo com o MIT (MIT, 2020), a ferramenta conta com um ambiente visual amigável e sua estrutura de programação

simplificada, facilitando a aprendizagem e a criação de apps para pessoas sem nenhuma experiência na área de computação, pois possui um ambiente didático.

Um aplicativo móvel pode ser considerado original tanto quando implementa uma solução totalmente inédita para um problema, quanto aprimorar ou otimizar uma solução já existente. Portanto, para incentivar a criatividade por meio do ensino de educação computacional de uma maneira fluida, o ciclo de aprendizagem "*Use-Modify-Create*" (Lee et al., 2011) é normalmente usado. O ciclo *"Use-Modify-Create"* é composto por três estágios: *"use"*, em que os alunos são introduzidos a algoritmos e tópicos de programação desenvolvendo um aplicativo seguindo um tutorial, *"Modify"*, em que os alunos modificarão o aplicativo criado usando o tutorial, podendo mudar características do aplicativo e adicionar novas funcionalidades, motivando-os a aprender novos conceitos de programação, e por fim, o estágio *"create"*, em que os alunos são incentivados a criar novos aplicativos visando a solução de um problema que eles identificaram.

Neste estágio pode-se adotar também uma estratégia educacional de ação computacional. Ação computacional, propõe que, enquanto aprendem, os jovens também devam ter oportunidades de criar algo que tenha impacto direto em suas vidas e comunidades (Tissenbaum et al., 2019). Ao ensinar computação, capacita o jovem e garante sua autonomia, o que faz desenvolver sua criatividade por meio da criação de apps de sua autoria (Araújo et al., 2020).

Assim, para acompanhar o desenvolvimento da criatividade no processo educacional, é importante o fornecimento de *feedback* ao aluno (Hattie & Timperley, 2007), o que pode ser feito por meio de avaliações de desempenho com base nos próprios artefatos criados pelo aluno durante o processo de aprendizagem.

Como esse processo de avaliação tipicamente requer um esforço considerável do instrutor, o julgamento humano para originalidade se torna alvo de influências e preferências subjetivas e variáveis entre cada pessoa (Zen; Vanderdonckt, 2016) (Gresse von Wangenheim et al., 2018b). Assim, uma alternativa pode ser automatizar o processo de avaliação da aprendizagem. Por isso, se vê a necessidade de criar um modelo de avaliação da originalidade baseado nos apps criados pelos alunos no contexto do ensino de computação na educação básica para fornecer o *feedback ao* aluno e auxiliar os educadores.

A avaliação da originalidade de apps desenvolvidos como resultados do processo de aprendizagem pode ser feita com base em diversas dimensões. De acordo com Garrett (2010), do ponto de vista da experiência do usuário, essas dimensões podem ser divididas em:

- estratégia, que por sua vez define o objetivo do aplicativo,
- escopo, que inclui as suas funcionalidades,
- *design* de interface,
- código.

Dentro dessas dimensões podemos destacar as *Funcionalidades*. *Funcionalidades* (ou *features*) são os recursos fornecidos por um aplicativo, tais como reproduzir áudio ou vídeo, acesso à câmera, localização, *bluetooth*, etc.

Elas afetam diretamente a utilidade e a usabilidade do aplicativo. A implementação de recursos ou funcionalidades totalmente novas pode ser um fator crucial para considerar um aplicativo inovador. Por isso, é relevante avaliar a originalidade de um aplicativo por suas funcionalidades, fazendo-os se destacarem de soluções já existentes.

Existem soluções como o modelo proposto por Mustafaraj et al. (2017) e Turbak et al. (2017) e Svanberg (2017) que avaliam a originalidade de um aplicativo do App Inventor adotando outras dimensões como a análise de blocos de código usando *machine learning*. Porém, essas abordagens se limitam apenas à análise da similaridade do código, o que não aborda a inovação de forma mais abrangente de um produto. Por isso, neste trabalho visa-se desenvolver um modelo de avaliação de originalidade focado em funcionalidades, a fim de avaliar a novidade de um aplicativo.

1.2 OBJETIVOS

Objetivo geral

O objetivo geral deste trabalho é desenvolver um modelo de avaliação da originalidade de *features* de apps desenvolvidos com App Inventor dentro do contexto da aprendizagem de criatividade no ensino de computação na educação básica no nível de crie referente ao ciclo de "*Use-Modify-Create*". Visa-se usar técnicas de Inteligência Artificial para desenvolver o modelo de avaliação.

Objetivos específicos

Os objetivos específicos do presente trabalho são:

Objetivo I: Síntese da fundamentação teórica em relação ao conceito de originalidade e App Inventor, feature *extraction e medidas de similaridade.*

Objetivo II: Análise do estado da arte em relação a modelos de análise e avaliação de originalidade de apps desenvolvidos com App Inventor no contexto da educação básica.

Objetivo III: Desenvolvimento do modelo de avaliação da originalidade em relação às funcionalidades.

Objetivo IV: Integração do modelo na ferramenta CodeMaster.

1.3 METODOLOGIA

Nessa pesquisa é usada uma abordagem multi-método. A metodologia de pesquisa utilizada neste trabalho é dividida nas seguintes etapas:

Etapa 1 – Fundamentação teórica

Atividade focada em estudar, analisar e sintetizar os conceitos principais e a teoria referente aos temas a serem abordados neste trabalho. Como resultado é apresentada a fundamentação teórica utilizando a metodologia de revisão narrativa (Cordeiro et al., 2007), e são realizadas as seguintes atividades:

A1.1 - Análise teórica sobre o ensino de computação na Educação básica;

A1.2 - Análise teórica sobre originalidade;

A1.3 - Análise teórica de características do App Inventor;

A1.4 - Análise teórica sobre *Feature extraction*.

Etapa 2 – Estado da arte

Nesta etapa é realizado um mapeamento sistemático da literatura seguindo o processo proposto por Petersen et al. (2015) para identificar e analisar modelos de análise automatizado da estética visual de interfaces de usuário de apps atualmente sendo utilizados. Esta etapa é dividida nas seguintes atividades:

A2.1 – Definição do protocolo de mapeamento

A2.3 – Execução da busca e seleção de artigos relevantes;

A2.4 – Extração e análise de informações relevantes.

Etapa 3 – Desenvolvimento

Nesta etapa é desenvolvido um modelo para avaliação da originalidade da funcionalidade de apps desenvolvidos com App Inventor adotando técnicas de Inteligência Artificial. Esta etapa é dividida nas seguintes atividades que foram realizadas por meio de diversas iterações:

A3.1 – Análise de requisitos;

A3.3 – Preparação de conjunto de dados;

A3.4 – Definição e implementação da extração de features;

A3.5 - Definição e implementação da inferência de funcionalidades baseada em regras

A3.5 – Definição e implementação de medida de similaridade

A3.6 – Definição e implementação de cálculo da nota de originalidade

Etapa 4 – Integração

Nesta etapa é integrada essa funcionalidade no sistema CodeMaster (Gresse von Wangenheim et al., 2018a) seguindo um processo de engenharia de software proposto por Pressman (2016). Esta etapa é dividida nas seguintes atividades:

A4.1 – Análise de requisitos;

A4.2 – Modelagem da arquitetura do sistema;

A4.3 – Modelagem detalhada e implementação;

A4.4 – Testes do sistema.

1.4 ESTRUTURA DO DOCUMENTO

No capítulo 2 é introduzida a fundamentação teórica dos conceitos necessários que sustentam a proposta deste trabalho. No capítulo 3 é o levantamento do estado da arte, contendo a revisão sistemática da literatura envolvendo modelos de avaliação de originalidades de aplicativos no contexto do App Inventor. No capítulo 4 é apresentada uma proposta de solução para o problema abordado e o desenvolvimento dessa solução, no capítulo 5 é feita a integração

do modelo de avaliação a plataforma App Inventor e no capítulo 6 são apresentadas as considerações finais.

2 FUNDAMENTAÇÃO TEÓRICA

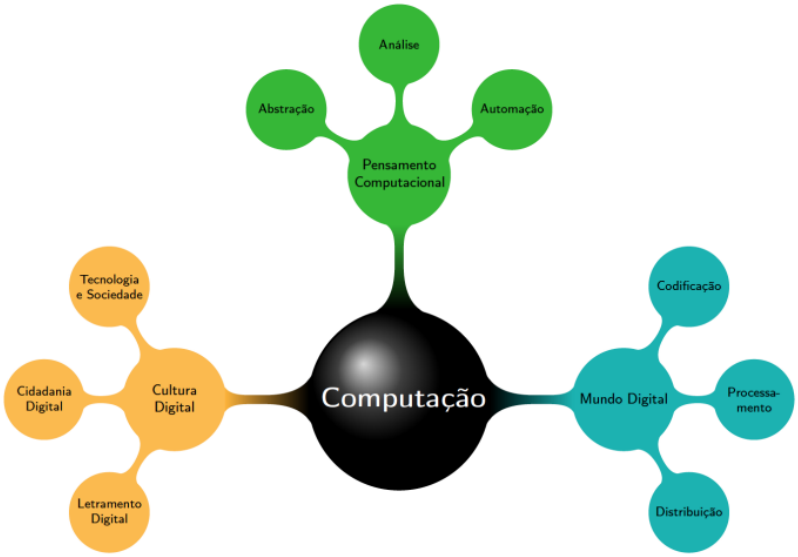
Neste capítulo são apresentados conceitos essenciais ao trabalho desenvolvido. Iniciando-se com uma análise teórica sobre novidade e originalidade, seguido por uma análise teórica do App Inventor e, por fim, uma análise teórica sobre técnicas da inteligência artificial, com foco em *feature extraction,* modelos baseados em regras, e medidas de similaridade.

2.1 ENSINO DE COMPUTAÇÃO NA EDUCAÇÃO BÁSICA

Computação é a ciência que estuda o desenvolvimento de linguagens e técnicas para descrever processos existentes e também métodos de resolução e análise de problemas, gerando novos processos (SBC, 2020). Os conhecimentos da área de computação podem ser divididos em três eixos (Figura 1):

- Pensamento computacional: refere-se à capacidade de compreender e solucionar problemas através da construção de algoritmos.
- Mundo Digital: refere-se aos veículos de comunicação e interação baseados em tecnologia digital, com o ensino a distância.
- Cultura Digital: refere-se a padrões de comportamento e novos questionamentos morais e éticos na sociedade que surgiram em decorrência do Mundo Digital.

Figura 1- Diretrizes e subdiretrizes para ensino de Computação na educação básica.



Fonte: SBC (2020).

Durante a educação básica, os alunos devem ser introduzidos a conceitos relacionados às estruturas abstratas necessárias à resolução de problemas para que o aluno tome consciência do processo de resolução de problemas, e compreenda a importância de ser capaz de descrever a solução em forma de algoritmo (SBC, 2020). Ao dominar estes conceitos, espera-se que os estudantes sejam capazes de selecionar e utilizar modelos e representações adequadas para descrever informações e processos, bem como dominem as principais técnicas para construir soluções algorítmicas.

Uma forma de aprender esses conceitos é por meio da criação de aplicativos (Araújo et al., 2020). Na educação básica, ambientes de programação baseados em blocos como o App Inventor dão a oportunidade para os alunos exercitarem e aprenderem conceitos de computação. Por lidarem com conceitos de programação de uma forma mais simples e indireta, facilita o jovem a aprender conceitos de computação.

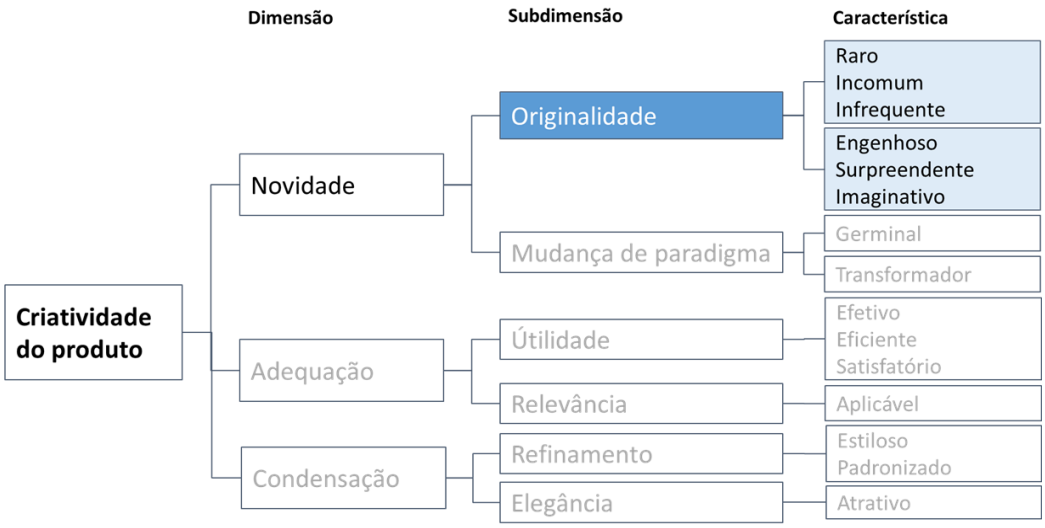
2.2 ORIGINALIDADE/NOVIDADE

Criatividade é a capacidade de uma pessoa ser criativa, ou seja, capaz de criar, inventar ou fazer inovações, de ser original. A **novidade** é uma das características essenciais da criatividade e refere-se a algo com uma qualidade nova ou incomum (Michaelis, 2020). Ela

pode abranger muitas outras características, as principais sendo originalidade e relação de paradigma (Dean et al., 2006).

A **originalidade** é normalmente representada como algo novo ou surpreendente que não existia antes (Runco et. al., 2012). Enquanto a relação de paradigma é definida como o grau em que uma idéia se relaciona com o paradigma atualmente prevalecente ou está modificando seu paradigma (Dean et al., 2006). Para ser considerado criativo, um produto deve exibir algum nível de novidade e algum grau de adequação e o usuário deve experimentar algum nível de apreciação ao interagir com ele (Horn & Salvendy, 2006).

Figura 2 - Dimensões e subdimensões da criatividade



Fonte: Alves et al. (2020).

No contexto da educação básica, nesse trabalho estamos focando somente na originalidade de uma forma menos extrema como mostra a Figura 2, pois não se espera que alunos de forma geral criem soluções e produtos "revolucionários", mas sim aprendam o conceito de novidade e possam exercê-lo ao longo de sua vida (Alves et al., 2020).

Uma das formas de incentivar a criatividade e, por consequência, a novidade, é por meio do ensino da computação na educação básica. Como normalmente os alunos não têm experiência prévia em programação, se adota um método de ensino de progressão, para que possam ir de um nível básico e progredirem para atividades mais avançadas. Por isso, o ciclo "*Use-Modify-Create*" (Lee et al., 2011) é geralmente proposto como base para a progressão da aprendizagem. Este modelo é composto por três estágios:

- *Use*: Durante a primeira fase, os alunos são introduzidos a criar um aplicativo tutorial base, para desenvolver conceitos básicos de programação como, variáveis, métodos, etc., eles seguindo um tutorial predefinido. Como esta etapa tem como objetivo introduzir noções de programação, a criatividade não é desenvolvida, pois o resultado esperado dessa atividade é o mesmo aplicativo para todos os alunos.
- *Modify*: Nesta fase os alunos aprendem a modificar o aplicativo base, tanto na interface do usuário quanto na criação de novos recursos para o aplicativo e/ou *design* de interface. Nesta etapa a novidade é demonstrada por meio da diferença entre o aplicativo base e o modificado pelo aluno.
- *Create*: Na terceira fase os alunos são encorajados a criar aplicativos com o conhecimento obtido nas fases anteriores para a solução de um problema atual dentro de sua vivência. Esta etapa tem ênfase no desenvolvimento da novidade ao incentivar alunos a desenvolverem soluções originais para problemas. Tipicamente se adota a ação computacional nesta etapa usando também *design thinking* (Brown, 2008). No presente trabalho focamos na avaliação da originalidade no estágio *Create*.

2.2.1 ORIGINALIDADE

Originalidade é uma característica essencial para a novidade do produto e normalmente é definida como a singularidade de um produto, conforme percebida pelo consumidor, em relação às ofertas anteriores (Goldenberg et al., 1999), ou seja, um produto original é algo incomum, diferente de um já existente. Uma maneira de determinar a originalidade de um produto é comparar as características do novo produto com as existentes.

Um aplicativo móvel pode ser composto por diferentes dimensões, os quais, com base em Garrett (2010) do ponto de vista da experiência do usuário podem ser representados pela Figura 3. Em objetivo, haverá objetivos de um novo produto, no escopo podem ser incluídos novos recursos para os objetivos do produto, em *design* da GUI será criado um novo *design* da interface para acompanhar o objetivo e / ou escopo, e no código serão implementações de maneiras diferentes (Alves et al., 2020).

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

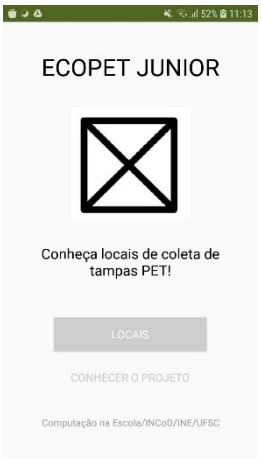
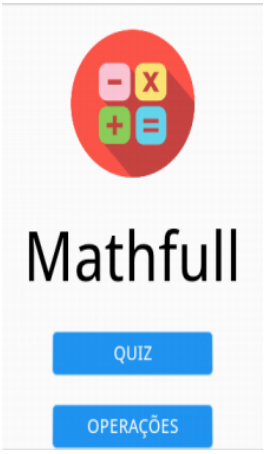
Fonte: Alves et al. (2020).

Dentro destas dimensões, uma forma de avaliar a originalidade de um aplicativo é por meio de suas *features*. *Features* ou funcionalidades são os recursos que o sistema de *software* oferece para atender aos objetivos do aplicativo, uma característica que desempenha determinada função (IEEE, 1990).

O estágio *Create* visa levar os alunos a desenvolver aplicativos novos que solucionem alguma questão na sua vida ou dentro de sua comunidade. Nessa fase espera-se o desenvolvimento de funcionalidades novas que desejarem para seus aplicativos, tornando seus aplicativos mais originais (Tabela 1).

Tabela 1- Exemplos de aplicações criadas em diferentes estágios do modelo use-Modify-Create.

Fase	*Use*	*Modify*	*Create*
Descri-ção	Aplicativo base	Alteração do aplicativo criando as funcionalidades “alterar locais” e “avalie o projeto”	Aplicativo com funcionalidades totalmente novas questões matemáticas
Features	Visualizar Mapa Marcar posição no mapa	Visualizar Mapa Marcar posição no mapa Salvar dados na nuvem	Reproduzir som Visualizar informações

	Salvar dados na nuvem	Mostrar geolocalização Visualizar informações	
Imagem do app			

Fonte: Elaborado pelo autor.

Dessa forma, a avaliação da originalidade é sempre feita em relação a um Universo de referência. Considerando o foco do presente trabalho no estágio *Create*, o universo de referência é composto de aplicativos criados com App Inventor em contextos educacionais semelhantes. Dessa forma a avaliação da originalidade é feito analisando o grau da similaridade de um novo aplicativo criado pelo aluno em relação a esse universo de referência.

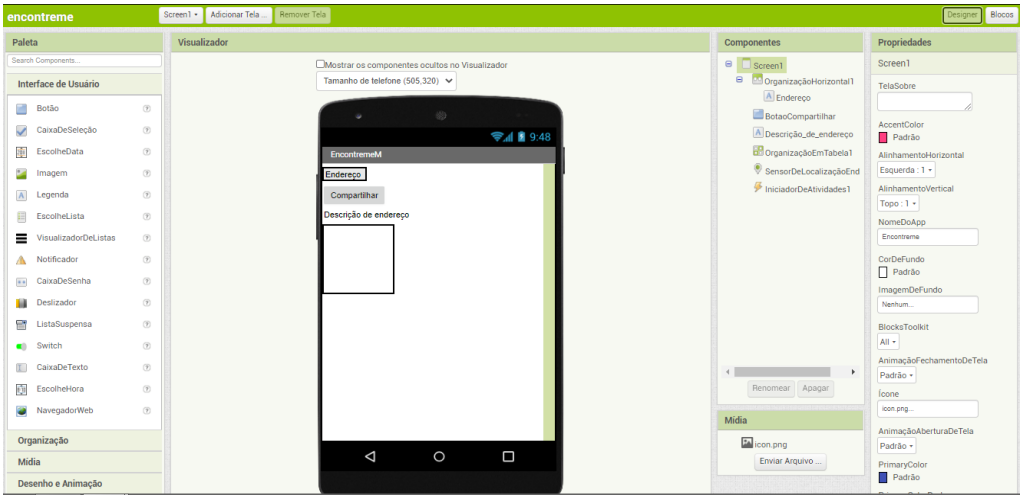
2.3 APP INVENTOR

O MIT App Inventor é um ambiente de programação visual intuitivo que permite a todos, até crianças, criar aplicativos totalmente funcionais para smartphones e tablets (MIT, 2020). É uma ferramenta baseada em blocos e, com isso, facilita a criação de aplicativos complexos e com menor esforço e tempo do que utilizando ambientes tradicionais de programação (MIT, 2020).

O App Inventor é dividido em dois módulos principais: Blocos e *design* modular. A área de *design* mostra ao usuário uma prévia do que será a tela do celular (Figura 4). Ao arrastar os componentes na tela, o usuário pode projetar o *design* de interface. A área de Blocos é semelhante à maioria das ferramentas de programação, mas apresenta os comandos de programação de forma visual e simplificada. Nesse módulo o usuário pode arrastar e soltar

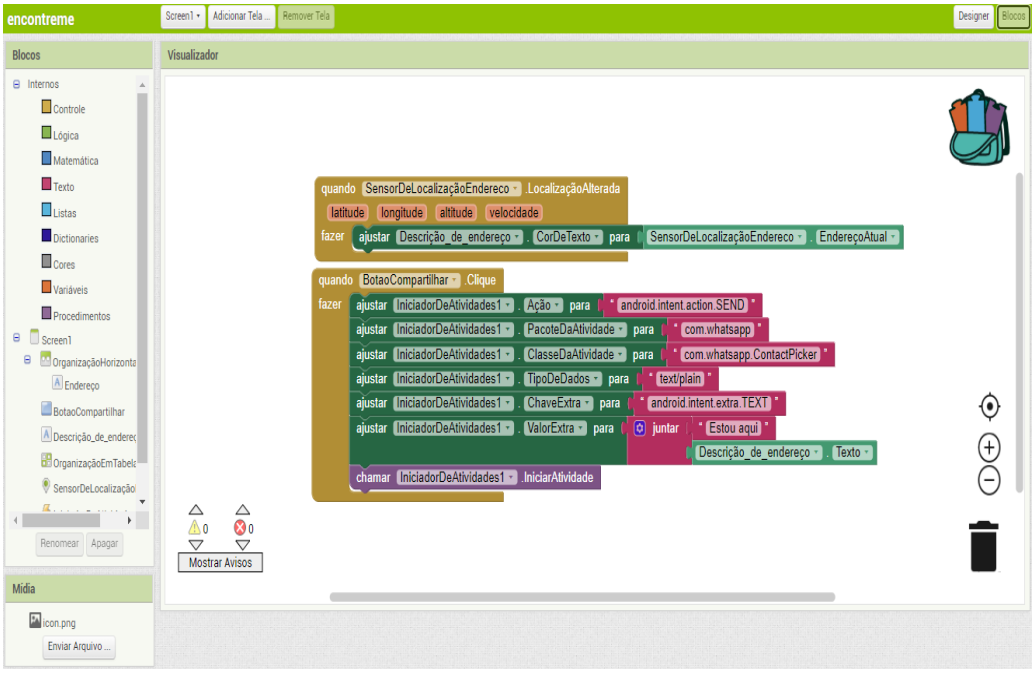
peças modulares representando os comandos de programação para programar a parte funcional do aplicativo (Figura 5) (Li, 2018).

Figura 4 - Área de design.



Fonte: Elaborado pelo autor.

Figura 5 - Área de blocos.



Fonte: Elaborado pelo autor.

Existem diversos componentes além dos comandos oferecidos pelo App Inventor (Tabela 2). Esses componentes podem ser alterados de acordo com a necessidade do usuário

para a resolução de um problema, dando oportunidade para a criação de soluções originais dentro do ambiente. Componentes como *Lego Mindstorms* e *Experimental* não serão abordados neste trabalho por conta de seu uso específico.

Tabela 2 - Componentes de aplicativos no App Inventor a partir de seus componentes de um ponto de vista de alto nível.

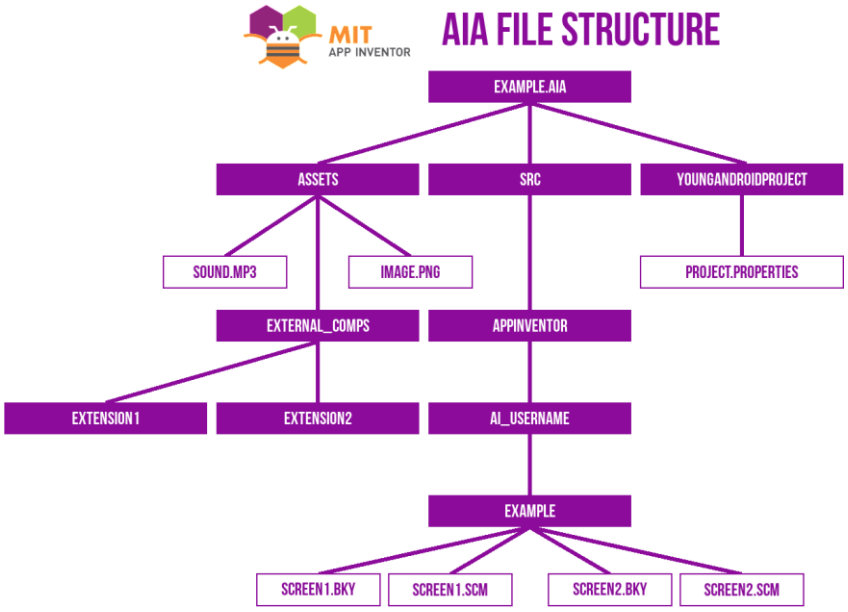
Categoria	**Descrição**	**Exemplos de componentes**
User Interface	Criando a parte visual do aplicativo. Todos os componentes visíveis do aplicativo estão neste grupo	Button, Checkbox, DatePicker, Image, Label, Notifier, etc.
Layout	Auxilia na organização dos componentes visíveis da categoria da interface do usuário	HorizontalArrangeme nt, TableArrangement, etc.
Media	Todos os componentes de mídia de um dispositivo que podem ser usados em aplicativos.	Camcorder, Camera, Player, ImagePicker, Sound, etc.
Drawing and Animation	Componentes que permitem ao usuário desenhar e visualizar animações.	Ball, Canvas, ImageSprite
Maps	Componentes de mapas que incluem navegação e marcadores de mapas.	Circle, Map, Marker, Polygon, etc.
Sensors	Componentes que obtêm informações dos sensores do dispositivo.	AccelerometerSensor, Clock, GyroscopeSensor, etc.
Social	Componentes que permitem que o aplicativo se comunique com outros aplicativos sociais.	ContactPicker, EmailPicker, Sharing, etc.
Storage	Componentes que permitem a criação de bancos de dados para armazenamento de dados.	File, FusiontablesControl, TinyDB, TinyWebDB
Connectivity	Componentes que permitem a conectividade do aplicativo com outros dispositivos	ActivityStarter, BluetoothClient, BluetoothServer, Web
Lego Mindstorms	Controle de robôs LEGO® MINDSTORMS® NXT usando Bluetooth.	NxtDirectCommands, NxtColorSensor, NxtLightSensor, etc.
Experimental	Componentes experimentais.	CloudDB, FirebaseDB
Extensions	Possibilidade de importar extensões com novos blocos.	-

Fonte: Elaborado pelo autor.

O App Inventor suporta a criação de arquivos .apk e permite exportar o projeto em arquivo .aia. Um arquivo .aia é um arquivo de projeto compactado (zip). Ele contém várias

pastas e arquivos, que incluem arquivos de código-fonte .BKY, SCM e YAIL e pode ser convertido em arquivos JSON (Figura 6).

Figura 6 - Estrutura de um arquivo .aia



Fonte: MIT(2021).

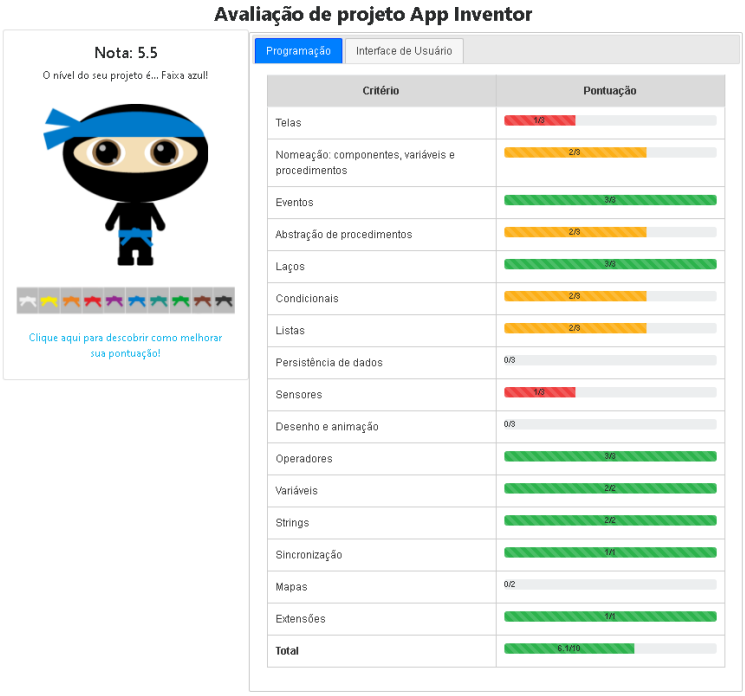
O App Inventor também possui uma galeria (https://gallery.appinventor.mit.edu) que suporta o compartilhamento dos apps entre usuários de todo o mundo. Diferente do que é tipicamente encontrado em galerias de aplicativos, nesta galeria há apenas a descrição do aplicativo, um botão para *download* e outro para "curtir" o aplicativo. Porém, não existe um padrão para a descrição dos apps e muitas vezes somente inclui o nome. Também não há uma seção dedicada a resenhas dos aplicativos. Além disso, o App Inventor não coleta dados demográficos de usuários além dos fornecidos em uma pesquisa opcional preenchida por apenas uma pequena porcentagem de usuários. Para a maioria dos autores dos aplicativos, não existem informações sobre sexo, idade, localização geográfica, histórico de programação, etc. Também não há informações em que contextos os aplicativos foram criados, por exemplo, se foi criado em colaboração com outras pessoas, ou como parte de uma aula ou outra atividade educacional.

2.4 CODEMASTER

O CodeMaster é um modelo para avaliação de competências de pensamento computacional relacionadas a algoritmos e programação, suportado por um sistema *web* para avaliar e atribuir uma nota automaticamente a aplicativos do App Inventor e também projetos das plataformas de criação de aplicativos BYOB e Snap! (Solecki et al., 2020), e sua atual versão gera um feedback instrucional sobre os elementos de programação (Figura 7) de *design* de interface (Figura 8).

O CodeMaster foca na avaliação de atividades educacionais de programação abertas e complexas em que não há uma resposta correta, em que, por exemplo, os alunos desenvolvem seus próprios aplicativos para resolver problemas da comunidade, que é o foco explorado no estágio "*create*".

Figura 7- CodeMaster aba Programação.



Fonte: Computação na escola (2021).

Figura 8- CodeMaster aba Interface do Usuário.

Fonte: Computação na escola (2021).

2.5 TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL

Nessa seção, são apresentadas técnicas de IA relevantes ao presente trabalho.

2.5.1 FEATURE EXTRACTION

Feature extraction é um processo de redução de dimensionalidade pelo qual um conjunto inicial de dados brutos é reduzido a grupos mais gerenciáveis para processamento (DeepAI, 2020). Um conjunto grande de dados geralmente contém um número muito grande de variáveis, elas exigem muitos recursos do computador para seu processamento, além de que muitas dessas variáveis podem ter valores redundantes. Assim, para otimizar o uso do processamento de um computador, é feita a extração de recursos. Dessa forma, métodos de extração geram *features* que contém os dados desejados de forma não redundante, reduzindo efetivamente a quantidade de dados que devem ser processados sem perder informações relevantes.

A extração de *features* pode gerar vetores binários ou de frequência (Mustafaraj et al., 2017). Vetores binários determinando a presença ou ausência de uma determinada palavra em uma determinada revisão, no contexto de um documento de texto. A presença é representada por 1 e a ausência por 0. Vetores de frequência identificam a frequência de uma palavra ao contar as palavras nos documentos em que aparecem.

2.5.2 MEDIDAS DE SIMILARIDADE

A medida de similaridade mede o quanto duas instâncias são parecidas: quanto mais parecidas, maior o valor. Quanto mais apps similares existirem no universo de referência, menos o app será considerado original Svanberg (2017), podendo assim ser usadas para mensurar originalidade. Portanto, o grau de originalidade se refere à quantidade de aplicativos cuja similaridade é medida dentro do intervalo [0,1] em relação a um universo específico (p.ex., apps na Galeria do App Inventor) esteja acima de um *threshold*.

São exemplos de medidas de similaridade:

- *Euclidean*: A similaridade euclidiana identifica parâmetros que estão próximos uns dos outros no sentido de distância euclidiana, o resultado euclidiano identifica parâmetros que estão próximos uns dos outros, fornecendo assim uma nova escolha de matriz de similaridade, ou seja, a distância euclidiana entre dois pontos é o comprimento do caminho que os conecta (Elmore et. al., 2001).
- *Jaccard*: A similaridade de *Jaccard* mede a similaridade entre conjuntos amostrais finitos e é definida como a cardinalidade da interseção dos conjuntos dividida pela cardinalidade da união dos conjuntos amostrais. Sendo a razão da proporção da cardinalidade dos itens co-classificados para a cardinalidade de todos os itens avaliados (Bag et. al., 2019).
- *Cosine*: A similaridade de cosseno é uma métrica usada para medir quão semelhantes dois elementos são, independentemente de seu tamanho. Matematicamente, o cosseno mede o ângulo entre dois vetores projetados em um espaço multidimensional (Xia et. al., 2015).

Com suas respectivas fórmulas definidas como:

$$Euclidean(p,q) = \sqrt{\sum_{i \in D}(q_i - p_i)^2}$$

$$Cityblock(p,q) = \sum_{i \in D} \mid q_i - p_i \mid$$

$$Jaccard^*(p,q) = 1 - \frac{\sum_{i \in D} min(q_i, p_i)}{\sum_{i \in D} max(q_i, p_i)}$$

$$Cosine(p,q) = 1 - \frac{\sum_{i \in D}(q_i * p_i)}{\sqrt{\sum_{i \in D} q_i^2}\sqrt{\sum_{i \in D} p_i^2}}$$

(2.1)

Onde p e q são vetores a serem medidas a distância e D são as dimensões (Svanberg, 2017). A partir da agrupação e extração destas funcionalidades, é necessário medir os resultados obtidos por ela. Por isso, se faz necessário cálculos para medir o desempenho de um modelo de extração de *features*.

$$\text{Completion Score} = \frac{\sum[(x*y_{basic\ template}) \cap (x*y_{submitted\ solution})]}{\sum(x*y_{basic\ template})}$$

$$\text{Creativity Score} = 1 - \frac{\sum[(x*y_{basic\ template}) \cap (x*y_{submitted\ solution})]}{\sum[(x*y_{basic\ template}) \cup (x*y_{submitted\ solution})]}$$

(2.2)

A fórmula proposta por Gopalan (2018) para calcular a pontuação de completude e criatividade de um determinado aplicativo do App Inventor no estágio *modify* - comparando o resultado criado pelo aluno com um modelo básico/tutorial, x representa os pesos que o usuário atribuiu e y representa a contagem de recursos na lista usada pelo autor. Assim, o numerador representa a soma da interseção das contagens de recursos multiplicados por seus pesos na solução básica e na solução apresentada. O denominador representa a soma de todas as contagens de recursos multiplicados por seus pesos no modelo básico. Elementos de recursos ponderados significam atribuir mais pesos a recursos e componentes importantes.

3 ESTADO DA ARTE

Essa seção é apresentado o levantamento do estado de arte em relação às seguintes perguntas relacionadas ao foco do presente trabalho:

1) Quais abordagens existem para automaticamente analisar a originalidade de aplicativos (com App Inventor) a partir de *features.*

2) Quais abordagens existem para automaticamente extrair as funcionalidades de aplicativos (com App Inventor).

O levantamento é feito por meio de mapeamentos sistemáticos da literatura seguindo o processo proposto por Petersen et al. (2015), para identificar e analisar e comparar as abordagens existentes.

3.1 ESTADO DA ARTE DE ABORDAGENS PARA ANÁLISE DE ORIGINALIDADE DE APPS

3.1.1 DEFINIÇÃO DO PROTOCOLO DE REVISÃO

O objetivo deste mapeamento é responder a seguinte pergunta de pesquisa: **Quais abordagens existem para automaticamente analisar a originalidade de aplicativos (com App Inventor) a partir de *features*?**

Esta pergunta de pesquisa é refinada nas seguintes questões de análise:

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

AQ2. Como os aplicativos são categorizados?

AQ3. Quais técnicas de algoritmos e programação são adotadas na análise?

AQ4. Como a qualidade das abordagens foi avaliada e quais as principais descobertas?

Critérios de inclusão/exclusão: Conforme o foco da pesquisa são definidos os seguintes critérios de inclusão e exclusão de artigos:

- São incluídos apenas artigos em inglês;
- São considerados somente artigos que apresentam abordagens para a extração automatizada de recursos com base no código de aplicativos para dispositivos móveis;

- São **excluídos artigos** que não sejam baseados em código, como descrição do aplicativo, revisões, dados de uso, etc, pelo fato que no contexto do App inventor não existem essas informações;
- São excluídos artigos que apresentam abordagens específicas para detecção de *malware*, detecção de plágio por estarem fora do contexto da criatividade.
- São excluídos artigos que usam como base descrições, avaliações de usuários, etc. Pois este tipo de informação não está disponível em relação a aplicativos AppInventor.
- São considerados apenas pesquisas publicadas desde 2010 levando em consideração o avanço recente especificamente em relação a aplicativos móveis.
- São considerados apenas artigos que apresentam informações substanciais, permitindo a extração de informações relevantes sobre as questões de análise. Portanto, artigos com resumo ou apenas uma página são excluídos.

Fontes. Foram pesquisados nos principais bancos de dados e bibliotecas digitais no campo da computação, incluindo as Bibliotecas Digitais ACM, a IEEE Xplore e Scopus com acesso por meio do Portal Capes. A pesquisa também foi feita dentro do site Google Scholar (Haddaway et. al., 2015, Piasecki et al. 2017), para complementar a busca.

Critérios de qualidade. São considerados apenas artigos que apresentam informações substanciais para se extrair informações referente às perguntas de análise.

Termos de busca. Com base na questão de pesquisa, várias pesquisas informais foram realizadas para calibrar o *string* de busca, identificando termos de pesquisa relevantes e seus sinônimos (Tabela 3). Foram utilizados sinônimos para minimizar o risco de omitir trabalhos relevantes.

Tabela 3 - Termos de busca.

Termo	**Sinônimos**	**Tradução (inglês)**
originalidade	originalidade, novidade	novelty, originality, similar, categorization
funcionalidade	semântica, funcionalidade	feature, semantics, functionality

Aplicativo móvel	App, iOS, Android, App Inventor	mobile application, app, iOS, Android, App Inventor

Fonte: Elaborado pelo autor.

String de busca. Após a definição dos termos e seus sinônimos, definiu-se o *string* de busca padrão a ser aplicado nas bases de dados:

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

Os *strings* de busca conforme a formatação de cada repositório estão apresentados na Tabela 4. Para o Google Scholar foi definido um *string* em especial para atender as limitações de opções de busca.

Tabela 4 - Strings de busca.

Repositório	***Search string***
ACM Digital Library	[[Publication Title: novelty] OR [Publication Title: originality] OR [Publication Title: similar] OR [Publication Title: categorization]] AND [[Publication Title: feature] OR [Publication Title: semantics] OR [Publication Title: functionality]] AND [[Publication Title: app] OR [Publication Title: ios] OR [Publication Title: "mobile application"] OR [Publication Title: android or] OR [Publication Title: "app inventor"]] AND [[Abstract: novelty] OR [Abstract: originality] OR [Abstract: similar] OR [Abstract: categorization]] AND [[Abstract: feature] OR [Abstract: semantics] OR [Abstract: functionality]] AND [[Abstract: app] OR [Abstract: ios] OR [Abstract: "mobile application"] OR [Abstract: android or] OR [Abstract: "app inventor"]] AND [Publication Date: (01/01/2010 TO 12/31/2020)]
IEEE Xplore Digital Library	(((novelty OR originality OR similar OR categorization) AND (feature OR semantics OR functionality) AND ("mobile application" OR app OR ios OR android OR "app inventor"))) Filters Applied: mobile computing smart phones Android (operating system) feature extraction learning (artificial intelligence) invasive software pattern classification application program interfaces 2010 - 2020
Scopus	TITLE-ABS (((novelty OR originality OR similar OR categorization) AND (feature OR functionality) AND ("mobile application" OR app OR ios OR android OR "app inventor"))) AND (LIMIT-TO (SUBJAREA , "COMP")) AND (LIMIT-TO (PUBYEAR , 2020) OR LIMIT-TO (PUBYEAR , 2019) OR LIMIT-TO (PUBYEAR , 2018) OR LIMIT-TO (PUBYEAR , 2017) OR LIMIT-TO (PUBYEAR , 2016) OR LIMIT-TO (PUBYEAR , 2015) OR LIMIT-TO (PUBYEAR , 2014) OR LIMIT-TO (PUBYEAR , 2013) OR LIMIT-TO (PUBYEAR , 2012) OR LIMIT-TO (PUBYEAR , 2011) OR LIMIT-TO (PUBYEAR , 2010)) AND (LIMIT-TO (PUBSTAGE , "final")) AND (LIMIT-TO (DOCTYPE , "ar"))
Google Scholar	identify original feature "in app inventor"

Fonte: Elaborado pelo autor.

3.1.2 EXECUÇÃO DA BUSCA

A busca foi realizada em Setembro de 2020 pelo autor do trabalho e revisada pelas (co-)orientadoras. A busca inicial resultou em 817 artigos (Tabela 5).

Tabela 5 - Número de artigos identificados por repositório e por fase de seleção.

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

Fonte: Elaborado pelo autor.

A partir do resultado inicial das buscas, foram selecionados artigos potencialmente relevantes de acordo com os critérios de inclusão e exclusão por meio de uma análise do título, resumo e palavra-chave de cada artigo, com a finalidade de confirmar a relevância dos trabalhos de acordo com os critérios de inclusão e exclusão. Foram analisados os primeiros 200 artigos encontrados em cada busca.

Em seguida, foram analisados os artigos potencialmente relevantes pelo artigo na íntegra. Como resultado final foram identificados 15 artigos relevantes (Tabela 6).

Artigos como os de Al-Subaihin et. al. (2019) ou Zhen et. al. (2019) foram excluídos por se tratarem de pesquisas que usam a descrição de um aplicativo como base para a extração de *features* ou são focados na detecção de *malwares*, isto é, não diretamente no nível de originalidade de um aplicativo.

Tabela 6 - Artigos relevantes.

Título do artigo	Referência
RomaDroid: A Robust and Efficient Technique for Detecting Android App Clones Using a Tree Structure and Components of Each App's Manifest File	(Kim et al., 2019)
Using feature vector representations to identify similar projects in app inventor	(Svanberg, 2017)

On automatically detecting similar Android apps	(Linares-Vásquez et al., 2016)
What's Inside My App?: Understanding Feature Redundancy in Mobile Apps	(Guo et al., 2018)
Robust App Clone Detection Based on Similarity of UI Structure	(Hu et al., 2020)
Robust Detection of Android UI Similarity	(Mao et. al., 2018)
SimiDroid: Identifying and Explaining Similarities in Android Apps	(Li et al., 2017)
AnDarwin: Scalable Detection of Android Application Clones Based on Semantics	(Crussell et al., 2015)
DroidCC: A Scalable Clone Detection Approach for Android Applications to Detect Similarity at Source Code Level	(Akram et al., 2018)
Appearance similarity evaluation for Android applications	(Zhu et al., 2015)
Recommending software features for mobile applications based on user interface comparison	(Chen et al., 2019)
Identifying original projects in App Inventor	(Mustafaraj et al., 2017)
Work in Progress: Identifying and Analyzing Original Projects in an Open-Ended Blocks Programming Environment	(Turbak et al., 2017)
An Assessment Tool to Analyze Code Written in App Inventor	(Gopalan, 2018)
An Automated Evaluation System for App Inventor Apps	(Li et al., 2018)

Fonte: Elaborado pelo autor.

Extração de dados

Os dados foram extraídos dos artigos de forma a responder às perguntas de análise conforme especificado na Tabela 7.

Tabela 7 - Especificação dos dados extraídos.

Pergunta de análise	**Item**	**Descrição**
AQ1. Quais abordagens existem para a avaliação da originalidade de aplicativos e quais suas características?	Referência	Indicando a referência do artigo
	Nome	Indicando o nome do artigo
	Breve descrição	Indicando uma breve descrição da abordagem
	Tipo de app (plataforma)	Android, IOS, App Inventor
	Contexto	Profissional (ex. clonagem de apps), Educacional
AQ2. Como os aplicativos são categorizados?	Referência	Indicando a referência do artigo
	Fator de "originalidade" avaliado	Indicando qual tipo de fator de originalidade será avaliado.
	Referente a qual dimensão	Código, GUI
	Tipo do feature	Indicando como é representada a funcionalidade alvo da abordagem
	Universo de referência	De onde são os apps cuja comparação é feita
	Escala de resposta	Indicando qual tipo de escala é alvo da abordagem
	Referência	Indicando a referência do artigo
	Feature "*extraction*"	Indicando qual a formatação de features é usada pela abordagem

AQ3. Quais técnicas são adotadas para a análise?	Análise de similaridade	Indicando qual metodologia foi usada pela abordagem para executar a análise
AQ4. Como a qualidade das abordagens foi avaliada e quais as principais descobertas?	Referência	Indicando a referência do artigo
	Fator(es) de qualidade avaliado(s)	Indicando qual o fator de qualidade usado para analisar a abordagem
	Tamanho da amostra	Indicando qual o tamanho total da amostra da abordagem
	Métodos de análise - ou junto com os fatores ou separado	Indicando qual método de análise foi utilizado para avaliar a abordagem
	Resultados	Indicando quais foram os resultados obtidos pela abordagem

Fonte: Elaborado pelo autor.

3.1.3 ANÁLISE DAS ABORDAGENS RELEVANTES

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

Foram encontrados 15 artigos (Tabela 8) que utilizam abordagens de agrupamento/*clustering* e classificação a fim de avaliar os aplicativos. As abordagens que utilizam técnicas de agrupamento/*clustering*, tipicamente, buscam agrupar em *clusters* distintos aplicativos originais (novos) e não originais (cópias, tutoriais, etc.). Observa-se também que a maioria dos artigos se trata de abordagens para a detecção de similaridade, com exceção de (Svanberg, 2017), que fez um estudo explorando e comparando diferentes técnicas de agrupamento a fim de encontrar a mais eficiente para a detecção de similaridade.

Tabela 8 - Visão geral das abordagens.

Ref	**Nome**	**Breve descrição**	**Tipo de app**	**Contexto**
(Kim et. al., 2019)	RomaDroid: A Robust and Efficient Technique for Detecting Android App Clones Using a Tree Structure and Components of Each App's Manifest File.	Abordagem usando a ferramenta RomaDroid voltada para detecção de clones para identificar o grau de similaridade entre aplicativos a partir de uma estrutura em árvore para extrair um arquivo manifest.XML, a fim de filtrar suas *tags*.	Android	App marketplace
(Svanberg, 2017)	Using feature vector representations to identify similar projects in app inventor	Estudo de técnicas de agrupamento para identificar qual mede melhor o grau de similaridade entre aplicativos criados por alunos e tutoriais.	Android/App Inventor	Pesquisa e Ensino Superior

		Também é identificado que tipo de feature tem o melhor desempenho (blocos versus componentes).		
(Linares-Vásquez et. al., 2016)	On automatically detecting similar Android apps	Classificação com agrupamento para detecção de aplicativos Android semelhantes em mercados de aplicativos gratuitos ou repositórios de código aberto. Abordagem para detectar automaticamente aplicativos semelhantes no Android (CLANdroid), contando com técnicas de recuperação de informações avançadas e cinco âncoras semânticas: identificadores, APIs do Android, intents, permissões e sensores.	Android	App marketplace
(Guo et. al., 2018)	What's Inside My App?: Understanding Feature Redundancy in Mobile Apps	Abordagem de agrupamento para identificar *features* em aplicativos	Android	App marketplace
(Hu et. al., 2020)	Robust App Clone Detection Based on Similarity of UI Structure	Abordagem de agrupamento para identificar o grau de similaridade entre aplicativos	Android	App marketplace
(Mao et. al., 2018)	Robust Detection of Android UI Similarity	Abordagem usando a ferramenta GEMINISCOPE, para detectar UIs semelhantes entre aplicativos Android. Analisa os layouts de IU de aplicativos Android, extrai os recursos fundamentais da aparência visual das IUs e avalia a similaridade entre as IUs dos aplicativos para detectar aplicativos Android maliciosos com base na semelhança de sua aparência de IU.	Android	App marketplace
(Li et. al., 2017)	SimiDroid: Identifying and Explaining Similarities in Android Apps	Abordagem usando a ferramenta SimiDroid, um framework para comparação de vários níveis de aplicativos Android. SimiDroid é construído com o objetivo de apoiar a compreensão de semelhanças / mudanças entre versões de aplicativos e entre aplicativos reempacotados que implementa esquemas de comparação de pares para dissecar as semelhanças e diferenças entre atualizações suspeitas de pares de aplicativos.	Android	App marketplace
(Crussell et., al, 2015)	AnDarwin: Scalable Detection of Android Application Clones Based on Semantics	Abordagem usando a ferramenta AnDarwin, que evita comparar aplicativos em pares, melhorando muito sua escalabilidade, analisa apenas o código do aplicativo e não depende de outras informações, como o mercado, assinatura ou descrição do aplicativo e pode detectar similaridade total e parcial do aplicativo além de poder detectar automaticamente o código da biblioteca e removê-lo da análise de similaridade.	Android	App marketplace
(Akram et. al., 2018)	DroidCC: A Scalable Clone Detection Approach for Android Applications to Detect Similarity at Source Code Level	Abordagem usando uma ferramenta DroidCC, uma abordagem de detecção de clones em aplicativos Android, que ajuda a detectar diferentes tipos de clones do código-fonte do APK. O DroidCC detecta clones tipo 1, tipo 2, tipo 3 e tipo 4 em aplicativos Android no nível do código-fonte. Ele também detecta fragmentos de código semelhantes, que foram injetados em muitos aplicativos, o que pode ser uma indicação de propagação de *malware*. além disso, pode detectar similaridade de nível total e parcial entre os aplicativos.	Android	App marketplace
(Zhu et. al., 2015)	Appearance similarity evaluation for Android applications	Abordagem projetada para extrair GUI de aplicativos semelhantes em marketplaces Android para detectar plágios ou *malwares* que se ocultam como um aplicativo legítimo pré-existente nos mercados Android por sua aparência. Usando principalmente a comparação da similaridade GUI entre aplicativos Android e escolha de alguns aplicativos com alta	Android	Pesquisa/educacional

		similaridade em sua aparência, extraindo alguns recursos dos aplicativos e calculando sua similaridade por vetores de recursos.		
(Chen et. al., 2019)	Recommending software features for mobile applications based on user interface comparison	Abordagem baseada em dados para recomendar recursos de software de aplicativos móveis com base na comparação da interface do usuário. Ele extrai informações de interfaces de usuário (UIs) semelhantes de repositórios públicos e, para calcular a similaridade de UI por meio das melhores correspondências de componentes de duas UIs, a similaridade de texto é usada para medir a similaridade de componentes de UI e um algoritmo genético é introduzido para melhorar a eficiência de comparação . Em seguida, usando um algoritmo para extrair recursos de UIs semelhantes com base em um conjunto de regras de identificação, que são agrupados com um algoritmo de similaridade de texto e finalmente recomendados.	Android	App marketplace
(Mustafaraj et. al., 2017)	Identifying original projects in App Inventor	Abordagem de classificação com agrupamento para classificar os projetos em original / não original com o objetivo de distinguir os projetos originais que eles criam dos não originais que surgem de atividades de aprendizagem como tutoriais e exercícios. dada uma coleção de todos os projetos do App Inventor de alunos em uma classe, essa abordagem é capaz de classificar automaticamente esses projetos como não originais (ou seja, com base em tutoriais globais ou exemplos locais) ou originais (ou seja, projetos que os alunos criaram por conta própria ou em pequenos grupos)	Android/App Inventor	Pesquisa/educacional K-12 education
(Turbak et. al., 2017)	Work in Progress: Identifying and Analyzing Original Projects in an Open-Ended Blocks Programming Environment	Abordagem de agrupamento para identificar grupos de usuários tendo a mesma disciplina	Android/App Inventor	Pesquisa/educacional
(Gopalan, 2018)	An Assessment Tool to Analyze Code Written in App Inventor	Abordagem que analisa o aplicativo submetido, extrai suas informações e o compara com o modelo básico, dando pontuação de conclusão e criatividade, com o objetivo de auxiliar professores a avaliar se um aluno concluiu o aplicativo e se um aluno fez algum trabalho criativo em a aplicação. As tarefas são alcançadas pela extração de recursos de códigos enviados pelos alunos e pela comparação dos envios com as soluções esperadas. Com os valores obtidos pela comparação das duas soluções, pode-se dizer se a inscrição enviada foi concluída ou não e, em caso afirmativo, se possui algum aspecto criativo.	Android/App Inventor	Pesquisa/educacional K-12 education
(Li et. al., 2018)	An Automated Evaluation System for App Inventor Apps	Abordagem com Método de pontuação automatizado baseado em TF-IDF e clustering para identificar o grau de similaridade entre aplicativos com app inventor no contexto de K-12. Usando a medida de cosine para calcular a similaridade, o projeto .aia é analisado e uma matriz de distância é construída. Em seguida, um algoritmo de agrupamento hierárquico é usado para classificar projetos semelhantes em uma classe de projetos. A pontuação da classe é usada para avaliar automaticamente a pontuação do novo projeto.	Android/App Inventor	Pesquisa/educacional K-12 education

Fonte: Elaborado pelo autor.

Figura 9 - Quantidade de publicações relevantes por ano.

Fonte: Elaborado pelo autor.

Observa-se que com a exceção de Crussell et. al. (2015), todos os artigos obtidos pela pesquisa existem há pelo menos 5 anos (Figura 9). Houve uma queda de artigos a partir de 2018, pelo foco maior de trabalhos da época em *malware detection*, divergindo do foco do presente trabalho.

Adicionalmente, podemos observar na Tabela 8 que todos os artigos são focados em aplicativos do tipo *Android*, podendo ter a pesquisa especializada para aplicativos da plataforma App Inventor. Observa-se também que o contexto destes artigos tem uma divisão equilibrada entre pesquisas de mercado e pesquisas em um contexto educacional.

Como os aplicativos são categorizados?

Para identificar como as abordagens categorizam os aplicativos, foram extraídas informações sobre o fator de originalidade avaliado, a dimensão (código ou GUI), tipo de *feature*, universo de referência e a escala de resposta (Tabela 9).

Tabela 9 - Categorização dos aplicativos.

Referência	**Fator de originalidade avaliado**	**Dimensão/fonte (dados brutos)**	**Ferramenta de extração**	**Tipo de *feature* (dados extraídos)**	**Universo de referência**	**Escala de resposta**
(Kim et. al., 2019)	Similaridade	APK para extrair AndroidManifest.xml (Documento XML)	Próprio: - RomaDroid	Estrutura da árvore e nodos pais de tags um arquivo manifest.XML	F-Droid	Numeral

(Svanberg, 2017)	Similaridade	AIA para extrair .bky (Documento XML) e .scm (JSON)	Próprio: - Baseado em TF-IDF	Frequências de blocos de código (BKY) e blocos de componentes de design (JSON) incluindo GUI (TF-IDF)	Apps tutoriais usados no curso, extraídos do site do App Inventor, App Inventor Maker Cards e exercícios em sala de aula do curso	Numeral
(Linares-Vásquez et. al., 2016)	Similaridade	APK para extrair permissões e JARS para extrair código-fonte, APIs, *intents*, Identificadores, e sensores	Externo: - apktool - dex2jar - 7zip - JAD Próprio: - CLANdroid	Pares de valores-chave de um arquivo strings.XML através de âncoras semânticas para elementos específicos do Android que podem ser usados para identificar aplicativos semelhantes (Identificadores de código-fonte, sensores, chamadas de API, intents e incorrer no Android)	Google Play	Ordinal, Escala Likert de 4 pontos (Completamente diferentes, na sua maior parte diferentes, na sua maior parte semelhantes e Completamente semelhantes)
(Guo et. al., 2018)	Similaridade	Recursos de IU presentes no pacote de instalação	Externo: - apktool - word2vec	Pares de valores-chave de um arquivo *strings*.XML	Google Play e CoolAPK3	Numeral
(Hu et. al., 2020)	Similaridade	APK para extrair AndroidManifest.xml (Documento XML)	Externo: - apktool - ‘‘AM’’ tool - UIAutomator (widget extractor) Próprio - Extrator de estrutura	Estrutura em geral da GUI, estrutura de texto da GUI, estrutura de widgets da GUI	AndrooZoo, Daniel et. al.(2014), Conjunto de dados Yunsheng,aplicativos obfuscados desenvolvidos pelo autor	Numeral
(Mao et. al., 2018)	Similaridade	APK para extrair XML Layout Files	Próprio: - GeminiScope	arquivos XML de *Layout* descrevendo o GUI	Google Play, anzhi e malgenomeproject	Numeral
(Li et. al., 2017)	Similaridade	APK para extrair AndroidManifest.xml (Documento XML), código-fonte e arquivos *resource*	Próprio: - SimiDroid (MPlugin, CPlugin e RPlugin)	Métodos, componentes e arquivos de resource	Google Play	Numeral
(Crussell et., al, 2015)	Similaridade	APK para extrair código fonte e bibliotecas	Próprio: - AnDarwin	Nodos de semantic vectors, Program Dependence Graphs	Google Play, SlideME, m360, Brothersoft, Android Online, 1Mobile, Gfan, Eoemarket, GoApk, Freeware Lovers,	Numeral

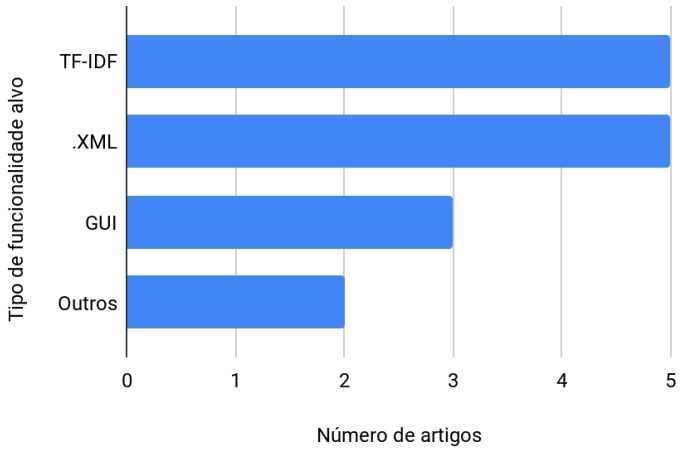
					AndAppStore, SoftPortal, Androidsoft, AppChina, ProAndroid, AndroidDownloadz, PocketGear	
(Akram et. al., 2018)	Similaridade	APK para extrair código-fonte	Externo: - dex2jar - JD-CORE Próprio: - DroidCC	Arquivos .APK	AppChina market	Ordinal (tipo-1, tipo-2, tipo-3)
(Zhu et. al., 2015)	Similaridade	APK para extrair arquivos XML relacionados a GUI e imagens da GUI (image resource, não screenshot)	Próprio - Zhu et al.	Elementos GUI em arquivos .XML	official Android market, eoemarket e Google Play	Numeral
(Chen et. al., 2019)	Similaridade	APK para extrair componentes da GUI (visíveis e invisíveis)	Próprio: - Chen et al.	Estrutura de texto de componentes da GUI	1Mobile e F-droid	Nominal/Clusters (grupos)
(Mustafaraj et. al., 2017)	Similaridade	AIA para extrair .bky (Documento XML) e .scm (JSON)	Próprio: - Baseado em TF-IDF	Frequências de blocos e componentes (TF-IDF)	pps tutoriais usados no curso, extraídos do site do App Inventor, App Inventor Maker Cards e exercícios em sala de aula do curso	Nominal (original, não original)
(Turbak et. al., 2017)	Similaridade	AIA para extrair .bky (Documento XML) e .scm (JSON)	Próprio: - Baseado em TF-IDF	Frequências de blocos e componentes (TF-IDF)	pps tutoriais usados no curso, extraídos do site do App Inventor, App Inventor Maker Cards e exercícios em sala de aula do curso	Nominal/Clusters (grupos)
(Gopalan, 2018)	Criatividade	AIA para extrair .bky (Documento XML) e .scm (JSON)	Próprio: - Baseado em TF-IDF	Frequências de blocos e componentes funcionais e de GUI (TF-IDF)	Tutoriais de aplicativos usados no curso	Numeral, ordinal (Imitation, Variation, Combination, Transformation. Original Creation)
(Li et. al., 2018)	Similaridade	AIA para extrair .bky (Documento XML) e .scm (JSON)	Próprio: - Baseado em TF-IDF	Frequências de blocos e componentes (TF-IDF)	NI	Numeral/Clusters (grupos)

Fonte: Elaborado pelo autor.

Observa-se que a maioria das abordagens caracterizam os aplicativos em relação a originalidade adotando um fator/métrica de similaridade (Tabela 9). Somente a abordagem proposta por Gopalan (2018) avalia o construto da criatividade após avaliar se a solução do aluno está correta como pré-requisito. Se a solução estiver correta, então é comparado se o aplicativo tem algo em comum com o *template* básico.

Todas as abordagens acima utilizam de componentes de código para a extração de *features*, não utilizando capturas de tela para avaliar a originalidade/similaridade da GUI de aplicativos (Tabela 9). As abordagens que avaliam componentes da GUI analisam exclusivamente arquivos de codificação da GUI (XML) para inferir se diferentes aplicativos utilizam os mesmos componentes.

Figura 10 - Tipo de *features* alvo extraídas pelos respectivos artigos encontrados



Fonte: Elaborado pelo autor.

Ao observar a Figura 10 destaca-se que os dois tipos de funcionalidades alvo mais usados são obtidos por meio do método TF-IDF e de extração de dados de um arquivo .XML. Sendo TF-IDF uma medida estatística que avalia a relevância de uma palavra para um documento em uma coleção de documentos e a extração de dados de arquivos .XML sendo elaboradas originalmente a partir de componentes que o autor da abordagem considera relevantes. Métodos que extraem componentes de GUI com base no código são usados com menos frequência. Crussell et al. (2015) e Akram et. al. (2018), usam métodos de extração de funcionalidades menos comuns, usando nodos semânticos e arquivos .APK, respectivamente.

Destaca-se também que a maioria das abordagens utiliza uma ferramenta própria para a extração de funcionalidades. Abordagens que usam ferramentas externas geralmente também fazem uso de uma ferramenta própria, com a exceção de Guo et. al. (2018), que é a única abordagem a usar exclusivamente uma ferramenta externa.

Observa-se também que, grande parte das abordagens encontradas é feita de forma numérica, como Hu et. al. (2020), que representa o nível de similaridade de um aplicativo por meio de um número dentro do intervalo entre 0 e 1. Apenas as abordagens de Akram et. al. (2018) e Linares-Vásquez et. al. (2016) usam uma escala ordinal para indicar o nível de similaridade. Somente a abordagem proposta por Gopalan (2018) avalia a originalidade focando no construto da criatividade incluindo uma análise se o aplicativo do aluno está correto e atende a alguns pré-requisitos. Caso positivo, então é realizada uma análise da originalidade comparando o aplicativo do aluno com um *template* básico com o objetivo de identificar se os dois têm algo em comum.

Os aplicativos usados para os testes normalmente são de repositórios populares como a Google Play e o F-droid. Porém, em abordagens com o foco na plataforma App Inventor, como Svanberg (2017), Mustafaraj et. al. (2017), Turbak et. al. (2017) e Gopalan (2018) utilizam de aplicativos usados em um curso aplicado como seu universo de referência. Somente Li et. al. (2018) não informa seu universo de referência dentre os artigos pesquisados.

Quais técnicas de algoritmos e programação são adotadas na análise?

Com o objetivo de identificar as técnicas adotadas na análise dos aplicativos, são extraídas informações em relação à forma que as *features* do aplicativo são extraídas. Além disso, são extraídas informações relacionadas ao tipo de análise realizado (Tabela 10).

Tabela 10 - Técnicas adotadas.

REF	**Feature "*extraction*"**	**Análise de similaridade**
(Kim et. al., 2019)	Criação de forma automatizada de uma string que representa a estrutura em árvore do arquivo manifest e reflete as tags intent-filter e suas tags de componente pai.	*Longest common subsequence algorithm* (LCS)
(Svanberg, 2017)	Uso de blocos e componentes de um projeto, aplicar TF-IDF e contar a frequência de cada feature,	Medidas de Euclides, *Cityblock*, Jaccard e Cosine

(Linares-Vásquez et. al., 2016)	Criação de forma automatizada de âncoras semânticas usadas para preencher uma matriz de termo de documento	Medida de Cosine
(Guo et. al., 2018)	Criação de forma automatizada de um feature vector contendo as palavras-chave de cada aplicativo	Cálculo de redundância elaborado pelo autor
(Hu et. al., 2020)	Criação de forma automatizada de um feature vector contendo os dados da interface (estrutura de texto e estrutura de widget)	Cálculo de similaridade entre vetores de features elaborado pelo autor
(Mao et. al., 2018)	Criação de forma automatizada de uma árvore de features contendo os dados da interface, gerando nós do layout como os recursos importantes influenciando a interface do usuário e representá-los em representações de recursos.	Método Húngaro (algoritmo K-M) usado para matriz de pontuação de similaridade entre os conjuntos de recursos
(Li et. al., 2017)	Criação de forma automatizada de um mapa com as chaves dos apps e valores com suas respectivas features para comparação.	Cálculo com base nas métricas de chave/valor baseada em mapas, calculando a semelhança pontuação dos dois aplicativos fornecidos (app1, app2), elaborado pelo autor
(Crussell et., al, 2015)	Criação de forma automatizada de um feature vector contendo as frequências dos comandos e componentes	Modelo LSH para agrupamento (clustering) em conjunto com medida de similaridade de Jaccard e algoritmo de MinHash
(Akram et. al., 2018)	Criação de forma automatizada de um mapa com os índices dos principais recursos do código-fonte dos arquivos APK extraídos para transformar a fonte em forma de representação para comparação de índice.	No processo de digitalização de semelhantes valores hash na tabela de indexação HBase, ele recupera todos os pedaços detectados do repositório de origem contra cada valor hash. MapReduce recupera todos os pedaços clonados e os armazena.
(Zhu et. al., 2015)	Criação de forma automatizada de um feature vector contendo os dados do efeito visual de uma tela (elementos de texto e imagem)	A pontuação de similaridade de dois aplicativos é fornecida com base na média das pontuações calculadas na última etapa. Todos esses recursos são comparados para obter uma pontuação de similaridade sc entre as telas. Além disso, calcula a pontuação de similaridade tanto do elemento de texto quanto do elemento de imagem, e obtém a pontuação de similaridade SCt e SCi respectivamente e usando sc = Sct! SCi como resultado final.
(Chen et. al., 2019)	Criação de forma automatizada de um feature vector contendo os dados de texto	*Latent Dirichlet Allocation* (LDA), algoritmo genético (GA) para calcular a similaridade entre UIs.
(Mustafa-raj et. al., 2017)	Criação de forma automatizada de um feature vector contendo as frequências dos comandos e componentes do App inventor	Modelo K-NN para agrupamento (agrupamento hierárquico) com base em semelhanças entre vetores de features usando a medida de similaridade de Jaccard
(Turbak et. al., 2017)	Criação de forma automatizada de um feature vector contendo as frequências dos comandos e componentes do App inventor	Algoritmo de cluster de Markov

(Gopalan, 2018)	Criação de um modelo que extrai informações de arquivos .aia convertendo-os em arquivos .xml e extraindo informações linha por linha de arquivos XML. Esses recursos linha por linha representam todos os recursos que podem ser vistos como blocos no AppInventor.	*Completion score*, que determina se a solução enviada possui recursos que atendem aos requisitos básicos do aplicativo, e *Creativity score* que analisa se uma feature do aplicativo do aluno está presente na solução "template", elaborado pelo autor
(Li et. al., 2018)	Criação de um modelo para analisar o código Aia e utilizar o algoritmo TF-IDF para analisar os principais componentes e blocos de código-chave de cada projeto, obtendo o valor TF-IDF para cada aresta que representa a importância de cada aresta. E então, cada aresta é considerada uma dimensão e seu valor TF-IDF é considerado o valor da dimensão.	A similaridade individual é calculada pela Medida de Cosine. Depois de calcular a similaridade e convertê-la em uma medida, ele analisa com sucesso o projeto .AIA e constrói uma matriz de distância. A seguir, é usado um algoritmo de agrupamento hierárquico para classificar trabalhos .AIA com similaridades em uma classe de trabalhos. A pontuação da classe é usada para avaliar automaticamente a pontuação do novo trabalho.

Fonte: Elaborado pelo autor.

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

Para a metodologia de avaliação, as abordagens usualmente usam medidas e algoritmos existentes. Por exemplo, Linares-Vásquez et. al. (2016) usa a medida de cosine, que mede a distância entre dois vetores de dimensão n, Mustafaraj et al. (2017) usam o índice de jaccard, que mede a similaridade entre conjuntos de amostras finitas e Mao et. al. (2018) que utiliza o algoritmo K-Means, que agrupa aplicativos similares dentro de um grupo mais próximo da média. Kim et al. (2019) utilizam o algoritmo LCS (*Longest Common Subsequence)*, que gera as diferenças (diff) entre dois elementos. Algumas abordagens também propõem medidas próprias como cálculo de redundância elaborado por Guo et. al. (2018), porém são menos frequentes.

Como a qualidade das abordagens foi avaliada e quais as principais descobertas?

Com o objetivo de analisar o desempenho destas abordagens, são extraídas informações referentes as avaliações das abordagens relatadas incluindo o(s) fator(es) de qualidade avaliados, tamanho da amostra, métodos de análise e os resultados obtidos pelos autores.

Tabela 11 - Avaliação das abordagens.

Referência	**Fator de qualidade avaliado**	**Tamanho da amostra usada na avaliação**	**Métodos de análise**	**Resultados**
(Kim et. al., 2019)	Acurácia, Precisão e *Recall*, limite, pontuação F1, Frequência de falso positivo	768 apps (148 app originais do repositório F-Droid, 620 apps clones)	-	Obtiveram pontuação acima de 90% de performance em detecção dependendo do número de componentes, em comparação a outro modelo de detecção de clones e em aplicativos originais ou ofuscados nos valores- Acurácia, Precisão e Lembrança, limite, pontuação F1, Frequência de falso positivo
(Svanberg, 2017)	Performance das métricas e da seleção de *features*	894 projetos do App Inventor criados por 16 alunos comparados com 169 tutoriais	-	Todos os métodos (Cosine, Jaccard, *CityBlock*, Euclidean) tiveram bom desempenho para achar similaridade. Os algoritmos Cosine e Jaccard apresentaram um resultado melhor do que os demais, representados por um gráfico com os respectivos valores aproximados de lembrança de cada métrica. Nos experimentos, o uso de blocos e blocos com componentes apresentou melhores resultados do que o uso de só componentes.
(Linares-Vásquez et. al., 2016)	Acurácia, categorização	14,450 aplicativos do Google Play	estatísticas descritivas, teste de Kruskal-Wallis, Teste U de Mann-Whitney and Cliff 's delta).	O CLAN droid é capaz de detectar aplicativos semelhantes, que pertencem a categorias diferentes, exceto ao usar o mecanismo de detecção baseado em sensores, fornecendo o maior número de aplicativos classificados como “altamente semelhante".Os resultados mostram que usar a semântica específica do Android âncoras são úteis para detectar aplicativos Android semelhantes em categorias diferentes. Também medindo o impacto de terceiros bibliotecas e código ofuscado ao identificar aplicativos semelhantes e os resultados sugerem que há uma diferença significativa na precisão quando bibliotecas de terceiros são excluídas.
(Guo et. al., 2018)	Acurácia, *Feature Overlapping*	4059 apps de repositórios Google Play e Cool APK	-	Depois de identificados os recursos nos aplicativos móveis, foi realizada uma pesquisa simples com os usuários, pedindo a cada usuário que indicasse manualmente quais funcionalidades eles acham desnecessárias para cada aplicativo. Foi solicitado que cada usuário fornecesse sua opinião sobre, no máximo, 10 aplicativos. Os resultados da pesquisa mostram a quantidade de funcionalidades listadas nas perguntas e o número de recursos indicados como desnecessários. No geral, de 1.336 recursos pesquisados, os usuários selecionaram cerca de 45% deles como desnecessários. Para características como “cronograma”, cerca de 65% das ocorrências são

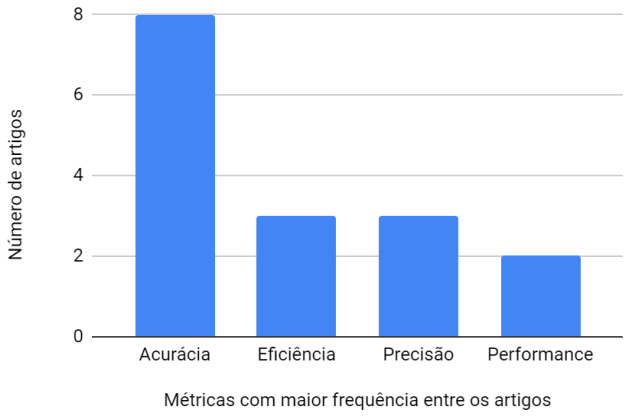
				consideradas desnecessárias pelos usuários, enquanto para "mapa / localização", apenas 16% são consideradas desnecessárias. Cerca de metade das funcionalidades estudadas na pesquisa de smartphones são considerados redundantes pelos usuários, principalmente aqueles que envolvem agendamento, gerenciamento de aplicativos, e-mail / mensagens e serviços de notícias.
(Hu et. al., 2020)	Acurácia, eficiência	dataset 1: 245 apps + 4,740 clones = 4985 dataset 2: 1.777 apps + 404.650 clones = 406427	-	Os resultados de uma experiência em um benchmark rotulado de 4.720 pares de aplicativos semelhantes mostram que a abordagem pode atingir uma precisão de 99,6%. Em comparação com outras abordagens existentes, a abordagem funciona na prática com alta eficácia, com 4.720 pares foram detectados com sucesso de 4.740 pares de clones e 3 pares eram falsos positivos, o que significa que a taxa de falsos positivos sendo FPR = 0,02% e taxa de falsos negativos FNR = 0,42%. Foi implementado um sistema de protótipo que foi aplicado a mais de 404.650 pares de aplicativos, e foram encontrados 1.037 pares de clones de aplicativos, a maioria deles sendo aplicativos "*piggybacking*" que introduziram cargas maliciosas.
(Mao et. al., 2018)	Eficiência	661 apps benignos e 120,518 malignos	-	Eficiência com mais de 90% dos pares de aplicativos benignos analisados não são similares, apontando que similaridade de layout pode ser um indicativo útil.
(Li et. al., 2017)	Eficiência	1000 pares (um maligno, um benigno) de apps	Teste de Mann-Whitney-Wilcoxon (MWW)	Obtiveram que a partir de comparações entre códigos se obtém maior similaridade, concluindo que a análise de similaridade explorada pelo SimiDroid está focada na comparação entre pares de aplicativos do tipo "*piggybacking*", obtendo detecção de similaridade com eficiência maior que 99%.
(Crussell et., al, 2015)	Performance, Acurácia	265,359 apps	-	Usando 75 threads, levou 4,47 dias para extrair vetores semânticos de todos os 265.359 aplicativos, levando apenas 109 segundos por thread para processar cada aplicativo no ambiente de pesquisa. Em seguida, foram comparados os aplicativos selecionados aleatoriamente de cada cluster com todos os outros aplicativos do cluster utilizando DNADroid, ferramenta utilizada para comparação, constatou-se que 96,28% dos clusters tinham pelo menos 70% dos valores de cobertura acima de 50%. Este limite é igual ao valor de similaridade de Jaccard usado pelo AnDarwin. Além disso, 95,50% dos clusters tinham 90% dos valores de cobertura acima do mesmo limite. Usando os critérios de 70%, apenas 3,72% dos clusters de detecção de similaridade de aplicativos completos do AnDarwin não foram verificados pelo DNADroid e, portanto, são considerados falsos positivos.
(Akram et. al., 2018)	Precisão e Lembrança	30 milhões	-	Para avaliar o recall e a precisão, foi realizada a avaliação manual de 100 resultados selecionados. Os resultados de lembrança e precisão são mostrados em suas respectivas tabelas e foram categorizados por seu tipo de clone. O DroidCC teve um desempenho

				muito bom na detecção de clones do tipo 1, em que ambos os arquivos são 100% iguais. Temos um valor de lembrança muito bom para clones do tipo 3 também. a tabela de recuperação mostra todas as outras medições de recuperação em relação aos tipos de clones. Os resultados nessas tabelas comprovam a capacidade de desempenho do DroidCC. Foi definido que o tipo 1 são clones exatos, o tipo 2 são variáveis renomeadas e o tipo 3 são clones de linhas modificadas (excluídas / adicionadas). Os tipos 1 e 2 têm 93% e 86%, respectivamente, e a detecção de clones do Tipo 3 foi detectada lembrança de 75%. A precisão do DroidCC foi mostrada e resumida por seu tipo de clone. Possui precisão de 96%, 89% e 81% para clones do tipo 1, tipo 2 e tipo 3, respectivamente.
(Zhu et. al., 2015)	-	1,000 apps respectivamente dos repositórios Android Market e eoemarket (total de 2000 apps)	-	Foi avaliado o design com 2 grupos de aplicativos: 1.000 aplicativos do *Android Market* e 1000 aplicativos do *Android Marketplace* de terceiros. Os resultados mostram que existem 25 pares de aplicativos com altos índices de similaridade e autores diferentes. Entre esses aplicativos selecionados, alguns deles adicionam ou modificam o SDK do anúncio em comparação com os aplicativos originais para obter a monetização e outros podem inserir algum código malicioso para cumprir atividades maliciosas. Para esses tipos de aplicativos, a abordagem pode mitigar o problema de privacidade e *malware* de forma eficaz.
(Chen et. al., 2019)	Precisão	10,477 apps dos repositórios 1Mobile e F-droid	-	Depois de pesquisar interfaces de usuário semelhantes para 30 interfaces de usuário fornecidas por aplicativos, foram obtidas informações de similaridade para interfaces de usuário semelhantes e armazenadas no banco de dados. Por meio do processamento dos dados da resposta ao questionário, é computado AP (*average precision*) para cada conjunto de dados e MAP (*mean average precision*) para todos os conjuntos de dados. Um gráfico mostra os resultados, com MAX representando o melhor valor (*ideal value*) de AP, que é obtido quando as cinco UIs semelhantes julgadas pelos desenvolvedores e nosso método são iguais. Em outras palavras, as cinco UIs semelhantes julgadas pelos desenvolvedores estão na posição 1-5 na lista de similaridade gerada por nosso método. Portanto, o valor máximo de AP é 0,457. A partir deste gráfico, é descoberto que o MAP é 0,418, que está muito próximo do valor ideal. Para os 30 conjuntos de dados, o AP de 20 conjuntos é maior do que o MAP médio. Existem apenas cinco conjuntos de dados cujo AP é inferior a 0,35. concluindo que o método proposto é eficaz para identificar IUs semelhantes em nível funcional. Além disso, é relatado que não há diferença significativa entre as pontuações dadas pelos desenvolvedores.
(Mustafaraj et. al., 2017)	Acurácia	902 projetos do App Inventor no	-	A métrica de distância Jaccard funcionou bem em projetos, e funcionou melhor quando o vetor de recurso do tipo de bloco usou contagens de frequência em vez de valores binários (onde 1 indica

		contexto de um curso		pelo menos um bloco de um determinado tipo), com acurácia de 89%.
(Turbak et. al., 2017)	-	Projetos criados por 6012 usuários	-	462 clusters com destaque para 11 clusters corretamente agrupados de uma turma
(Gopalan, 2018)	Acurácia	111 apps do App Inventor no contexto de um curso	-	A ferramenta pode detectar se um aplicativo foi concluído ou não, reconhecendo vários cenários como ambiguidade, blocos parcialmente e totalmente concluídos, blocos incompletos e correspondência de blocos com o modelo básico, detectar criatividade verificando duplicatas e verificando quaisquer componentes extras no modelo enviado solução. Ele pode detectar se um aplicativo foi concluído ou não com uma precisão média de 95,2%. Além disso, pode detectar se um aplicativo possui alguma criatividade ou personalização com uma precisão média de 81,80%.
(Li et. al., 2018)	Acurácia	778 arquivos .AIA do App Inventor no contexto de uma competição	-	Apenas 37% de todos os trabalhos avaliados têm pontuações idênticas entre a pontuação automatizada e a pontuação do juiz. 577 das 765 obras avaliadas pela ferramenta são suficientemente precisas com uma acurácia de 75,42%.

Fonte: Elaborado pelo autor.

Figura 11 - Métricas usadas com maior frequência na avaliação das abordagens.



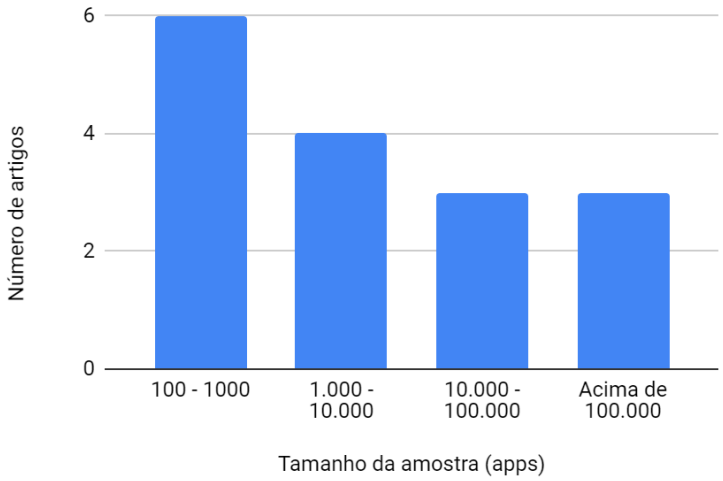
Fonte: Elaborado pelo autor.

A Figura 11 indica que grande parte das abordagens utiliza de acurácia para determinar o sucesso das abordagens, pois a maioria das pesquisas tem como objetivo classificar aplicativos como originais e não originais, por meio de um cálculo de similaridade, seguido por eficiência, que por sua vez determina se a abordagem utilizada foi capaz de classificar todos os aplicativos corretamente. Precisão, performance e desempenho são medidas com o objetivo de

medir a qualidade da abordagem de classificar um aplicativo e qual o desempenho da abordagem com uma grande amostra, respectivamente, com a mensuração dessas métricas divergindo entre cada autor.

Também há outras medidas usadas para avaliar a qualidade das abordagens como de Guo et. al.(2018), que além de acurácia mede *Feature Overlapping*, que são quaisquer características que compartilham dois ou mais vetores de aplicativos, ou Linares-Vásquez et. al.(2016) que mede a categorização de aplicativos em categorias elaboradas pelos autores. Zhu et. al.(2015) e Turbak et. al.(2017) são os únicos que não usam de medidas de avaliação para suas propostas (Tabela 11).

Figura 12 - Tamanho total da amostra usada na avaliação das abordagens.



Fonte: Elaborado pelo autor.

A maioria usou amostras relativamente pequenas de 100 - 1000 aplicativos. Observa-se também que em grande parte das abordagens o tamanho da amostra não passa de 10.000 aplicativos. As abordagens com maior tamanho de amostra são mais focadas em detectar aplicativos clones, ou seja, aplicativos totalmente similares, o que justifica seu tamanho (Figura 12).

3.1.4 DISCUSSÃO

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

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

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

Ameaças a validade. Ameaças à seleção de abordagens relevantes e extração de dados foram mitigadas fornecendo uma definição detalhada de critérios de inclusão/exclusão. É

definido e documentado um protocolo rígido para a seleção dos estudos, sendo discutido com os orientadores a seleção de artigos até que um consenso seja alcançado. Além disso, outro risco é a omissão de estudos relevantes. Para mitigar esse risco, a escolha das palavras-chave foi feita de forma a ser tão inclusiva quanto possível, considerando não apenas os conceitos principais, mas também seus sinônimos. O risco de excluir estudos primários relevantes foi mitigado pelo uso de múltiplas bases de dados que cobrem a maioria das publicações científicas nesta área.

3.2 ESTADO DA ARTE EM RELAÇÃO A ABORDAGENS EXISTENTES PARA AUTOMATICAMENTE EXTRAIR OS *FEATURES* DE APPS

3.2.1 DEFINIÇÃO DO PROTOCOLO DE REVISÃO

O objetivo deste mapeamento é responder a seguinte pergunta de pesquisa: **Quais abordagens existem para automaticamente extrair os *features* de apps (com App Inventor)?**

Esta pergunta de pesquisa é refinada nas seguintes questões de análise:

AQ1. Quais dados são usados de entrada para a extração?

AQ2. Como é feita a extração (qual o processo/*workflow*) e quais algoritmos/arquiteturas de ML/DL são usadas por etapa?

AQ3. Quais *features*/classes definem?

Critérios de inclusão/exclusão: Conforme o foco da pesquisa, são definidos os seguintes critérios de inclusão e exclusão de artigos:

- São incluídos apenas artefatos em inglês;
- São considerados somente artigos e artefatos que apresentam abordagens para a extração automatizada de recursos com base no código de aplicativos para dispositivos móveis;
- São **excluídos artigos** que não sejam baseados em código, como descrição do aplicativo, revisões, dados de uso etc, pelo fato que no contexto do App inventor não existem essas informações;
- São excluídos artigos que apresentam abordagens específicas para detecção de *malware*, detecção de plágio por estarem fora do contexto da criatividade

- Foram considerados apenas pesquisas publicadas desde 2010. Considerando apenas artigos que apresentam informações substanciais, permitindo a extração de informações relevantes sobre as questões de análise. Portanto, artigos com resumo ou apenas uma página são excluídos.

Fontes. Pesquisamos enfocando na biblioteca SCOPUS como esta base inclui os principais bancos de dados e bibliotecas digitais no campo da computação, incluindo publicações das bibliotecas ACM, IEEE e Springer, com acesso por meio do Portal Capes.

Critérios de qualidade. São considerados apenas artigos que apresentam informações substanciais para se extrair informações referente às perguntas de análise.

Termos de busca. Com base na questão de pesquisa, várias pesquisas informais foram realizadas para calibrar a sequência de pesquisa, identificando termos de pesquisa relevantes e seus sinônimos (Tabela 12). Foram utilizados sinônimos para minimizar o risco de omitir trabalhos relevantes.

Tabela 12 - Termos de busca.

Termo	**Sinônimos**	**Tradução (inglês)**
funcionalidade	Semântica, funcionalidade	feature, Semantics, functionality
Extração	Inferência, Mineração, recomendação, Recuperação	Extraction, Inference, Mining, recommendation, retrieval
Aplicativo móvel	App, iOS, Android, App Inventor	Mobile application, App, iOS, Android, App Inventor

Fonte: Elaborado pelo autor.

***String* de busca.** Após a definição dos termos e seus sinônimos, definiu-se o *string* de busca padrão a ser aplicado nas bases de dados:

```
TITLE-ABS-KEY( ( extract* OR detect* OR identify ) AND ( feature* ) AND ( app OR apps OR "android applications" ) ) AND ( LIMIT-TO ( PUBSTAGE , "final" ) ) AND ( LIMIT-TO ( SUBJAREA , "COMP" ) ) AND ( LIMIT-TO ( PUBYEAR , 2021 ) OR LIMIT-TO ( PUBYEAR , 2020 ) OR LIMIT-TO ( PUBYEAR , 2019 ) OR LIMIT-TO ( PUBYEAR , 2018 ) OR LIMIT-TO ( PUBYEAR , 2017 ) OR LIMIT-TO ( PUBYEAR , 2016 ) OR LIMIT-TO ( PUBYEAR , 2015 ) OR LIMIT-TO ( PUBYEAR , 2014 ) OR LIMIT-TO ( PUBYEAR , 2013 ) OR LIMIT-TO ( PUBYEAR
```

, 2012) OR LIMIT-TO (PUBYEAR , 2011) OR LIMIT-TO (PUBYEAR , 2010)) AND (LIMIT-TO (LANGUAGE , "English"))

Os *strings* de busca conforme a formatação de cada repositório estão apresentados na Tabela 13.

Tabela 13 - String de busca.

Repositório	***Search string***
Scopus	TITLE-ABS-KEY ((extract* OR detect* OR identify) AND (feature*) AND (app OR apps OR "android applications")) AND (LIMIT-TO (PUBSTAGE , "final")) AND (LIMIT-TO (SUBJAREA , "COMP")) AND (LIMIT-TO (PUBYEAR , 2021) OR LIMIT-TO (PUBYEAR , 2020) OR LIMIT-TO (PUBYEAR , 2019) OR LIMIT-TO (PUBYEAR , 2018) OR LIMIT-TO (PUBYEAR , 2017) OR LIMIT-TO (PUBYEAR , 2016) OR LIMIT-TO (PUBYEAR , 2015) OR LIMIT-TO (PUBYEAR , 2014) OR LIMIT-TO (PUBYEAR , 2013) OR LIMIT-TO (PUBYEAR , 2012) OR LIMIT-TO (PUBYEAR , 2011) OR LIMIT-TO (PUBYEAR , 2010)) AND (LIMIT-TO (LANGUAGE , "English"))

Fonte: Elaborado pelo autor.

3.2.2 EXECUÇÃO DA BUSCA E EXTRAÇÃO DOS DADOS

A pesquisa foi realizada em Fevereiro de 2020 pelo autor do artigo (Tabela 14). Na primeira fase de análise, títulos e resumos foram analisados. No segundo estágio, os materiais foram lidos por inteiro, para assegurar sua relevância com respeito aos critérios de inclusão/exclusão. Pela quantidade escassa de abordagens existentes para automaticamente extrair os *features* de aplicativos utilizando App Inventor, poucos artigos corresponderam aos critérios estabelecidos (Tabela 15).

Tabela 14 - Número de artigos identificados por repositório e por fase de seleção 2.

Fonte	**No. de resultados da busca**	**No. de resultados analisados**	**No. de documentos potencialmente relevantes**	**No. de documentos relevantes**
SCOPUS	1,799	1,799	8	4
Total				4

Tabela 15 - Artigos relevantes.

Referência	Título do artigo
(Svanberg, 2017)	Using feature vector representations to identify similar projects in app inventor
(Yuan et al., 2016)	Android Applications Categorization Using Bayesian Classification
(Kanda et al., 2013)	Semi-automatically Extracting Features from Source Code of Android Applications
(Wang et al., 2020)	Multi-classification of Android Applications Based on Convolutional Neural Networks

3.2.3 ANÁLISE DAS ABORDAGENS RELEVANTES

Quais dados são usados de entrada para a extração?

Em geral, as abordagens utilizam de permissões do usuário e componentes do aplicativo com entrada. Svanberg (2017) utiliza comandos de programação e componentes de *design* presentes no App Inventor; Yuan et al. (2016) utilizam além de permissões do usuário e de APIs, *strings* de componentes do código do aplicativo e sua descrição em um *marketPlace*. Kanda et al.(2013) baseia-se em chamadas de métodos em APIs para sua extração e Wang et al. (2020) utiliza de permissões e componentes analisando *tags* para seus dados de entrada. Observando que os aplicativos compartilhados pela *gallery* do App Inventor somente fornecem o nome do aplicativo, autor e código do aplicativo, percebe-se que não há dados como tipos de permissões ou *tags* disponíveis como entradas para este caso (Tabela 16).

Tabela 16 - Entradas utilizadas para extração de features.

Referência	Entrada	Saída
Using feature vector representations to identify similar projects in app inventor	Arquivo .aia	• Comandos de programação presentes no App Inventor • Componentes de design presentes no App Inventor
Android Applications Categorization Using Bayesian Classification	Arquivo .apk	• Permissão do usuário • Strings do aplicativo • API • Android Market data (Application's Description Extraction)

Semi-automatically Extracting Features from Source Code of Android Applications	Arquivo .apk	• chamada de métodos em API presentes no código do aplicativo
Multi-classification of Android Applications Based on Convolutional Neural Networks	Arquivo .apk	• Permissões analisando a tag <uses-permission> no arquivo de manifest. • Componentes analisando as tags <action> e <category> nas tags <intent-filter>. • Permissões e arquivos DEX para realizar avaliação de risco em arquivos APK.

Fonte: Elaborado pelo autor.

Como é feita a extração (qual o processo/*workflow*) e quais algoritmos/arquiteturas de ML/DL são usadas por etapa?

As abordagens utilizam diferentes métodos para o processo de extração. Svanberg (2017) usa do algoritmo TF-IDF para extrair as funcionalidades de um aplicativo e organizar essas informações extraídas em um *feature vector*. Yuan et. al. (2016) usam um processo em várias etapas:

- É usada a ferramenta APKtool para desmontar o aplicativo em um arquivo classes.dex e com arquivos de funcionalidades e coletar a descrição do aplicativo do Android Market
- Após usarem a ferramenta APKtool, extraem as permissões de API do aplicativo.
- Depois disso, os recursos são extraídos de *strings* analisando todos os arquivos de *layout* nos diretórios *resources*/*layout* e é usado a ferramenta stanfordnlp para remover palavras irrelevantes e terminar o trabalho de stemização das palavras.
- Como resultado da última etapa é criado *feature vectors* e aplicando a classificação Bayesiana para classificar todos os aplicativos em diferentes categorias.

Kanda et. al. (2013) traduz o aplicativo usado para extração em um conjunto de sequências de chamadas de API para em seguida usar o algoritmo LCS para extrair o valor de similaridade de uma funcionalidade de aplicativos, para esta ser armazenada em uma base de dados. Wang et. al.(2020) também utiliza a ferramenta APKtool para descompilação de um aplicativo e para cada amostra, seus recursos foram extraídos em um documento de texto como um arquivo de *features*, em que cada linha representa uma *feature*. Para isso usa uma rede

neural convolucional para classificação, convertendo este arquivo em uma matriz bidimensional (Tabela 17).

Tabela 17 - Processo e algoritmos de extração de features.

Título do artigo	**Processo de extração**
Using feature vector representations to identify similar projects in App Inventor	A partir de arquivos BKY e SCM é aplicado o algoritmo TF-IDF (term frequency over inverse document frequency) com feature scaling, valores binários e contagem. É gerada então uma tabela em que as colunas referem-se a tipos de comandos/componentes em App Inventor e as linhas referem-se à quantidade de comandos em um aplicativo.
Android Applications Categorization Using Bayesian Classification	A primeira etapa é a análise estática, onde é usado o APKtool para desmontar o aplicativo em um arquivo classes.dex e com arquivos de features e coletar a descrição do aplicativo do Android Market. Na segunda etapa, usando o código de Au et al. (2012) para obter as permissões do usuário, por meio da geração da chamada de gráfico para o código-fonte do Android, eles extraem os mapeamentos de permissão para cada API. Depois disso, eles extraem os recursos de strings analisando todos os arquivos de layout nos diretórios resources / layout * e usam stanfordnlp para remover palavras irrelevantes e terminar o trabalho de stemização das palavras. A última etapa cria vetores de recursos e aplica a classificação Bayesiana para classificar todos os aplicativos em diferentes categorias.
Semi-automatically Extracting Features from Source Code of Android Applications	Primeiro, cada aplicativo é traduzido em um conjunto de sequências de chamadas de API. Como os aplicativos Android são escritos em Java, é extraída uma sequência de chamadas da API Android de cada método do aplicativo. Em seguida, eles extraem sequências comuns de chamadas de API envolvidas em pelo menos dois aplicativos como candidatos para recursos em que LCS (s, t) é a subsequência comum mais longa de duas sequências s e t, excluindo sequências que consistem em apenas uma única chamada de API. Chamando o resultado definido por CommonAPI. Depois disso, é associado manualmente cada sequência S em CommonAPI com um nome de uma feature, e armazenando a associação em uma base de conhecimento.
Multi-classification of Android Applications Based on Convolutional Neural Networks	É usado o APKTool para descompilar o aplicativo Android para obter seu arquivo de manifest e features de permissão, de componentes e de ambiente são extraídas. Em seguida, é usado o módulo androrisk.py no androguard para analisar melhor o aplicativo e obter recursos relevantes do log de análise. Para cada amostra, seus recursos foram extraídos em um documento de texto como um arquivo de features, onde cada linha representa uma feature, usando uma rede neural convolucional para classificação, convertendo este arquivo em uma matriz bidimensional.

Fonte: Elaborado pelo autor.

Quais *features*/classes definem?

Yuan et. al. (2016) e Wang et. al. (2020) utilizam categorização para definir as funcionalidades extraídas, possivelmente de acordo com o contexto do universo de referência das abordagens. Svanberg (2017) avalia os aplicativos com suas funcionalidades extraídas como original e não original de acordo com critérios preestabelecidos. Kanda et. al. (2013) apresenta uma tabela com exemplos de funcionalidades extraídas de 5 diferentes aplicativos (Tabela 18).

Tabela 18 - Definição de features/classes.

Título do artigo	**Features - funcionalidade do app/classes**
Using feature vector representations to identify similar projects in app inventor	Avaliação do aplicativo como original/não original
Android Applications Categorization Using Bayesian Classification	Categorização de aplicativos (não é mostrado que tipo de categorização no artigo, possivelmente de acordo com o universo de referência)
Semi-automatically Extracting Features from Source Code of Android Applications	Apresenta somente 5 exemplos de features usadas pelos aplicativos dentro de uma base de conhecimento
Multi-classification of Android Applications Based on Convolutional Neural Networks	Categorização de aplicativos (não é mostrado que tipo de categorização no artigo, possivelmente de acordo com o universo de referência)

Fonte: Elaborado pelo autor.

3.2.4 DISCUSSÃO

Atualmente existem poucas propostas de abordagens com foco na extração de funcionalidades e no contexto deste trabalho somente Svanberg (2017) utiliza aplicativos do App Inventor, enquanto as outras abordagens focam em categorização (Yuan et al., 2016, Wang et al.,2020) e retenção de dados Kanda et al.(2013) dentro de um determinado mercado de aplicativos. Porém, uma desvantagem da abordagem proposta por Svanberg (2017), é que não são extraídas as *features*, porém aplicativos diretamente classificados em original ou não original sem detalhes de *feedback* para o avaliador e o aluno. Com isto, é demonstrado a necessidade de um suporte maior para extrair primeiramente as funcionalidades de aplicativos e a partir delas classificar/julgar o grau da originalidade dos aplicativos criados com App Inventor.

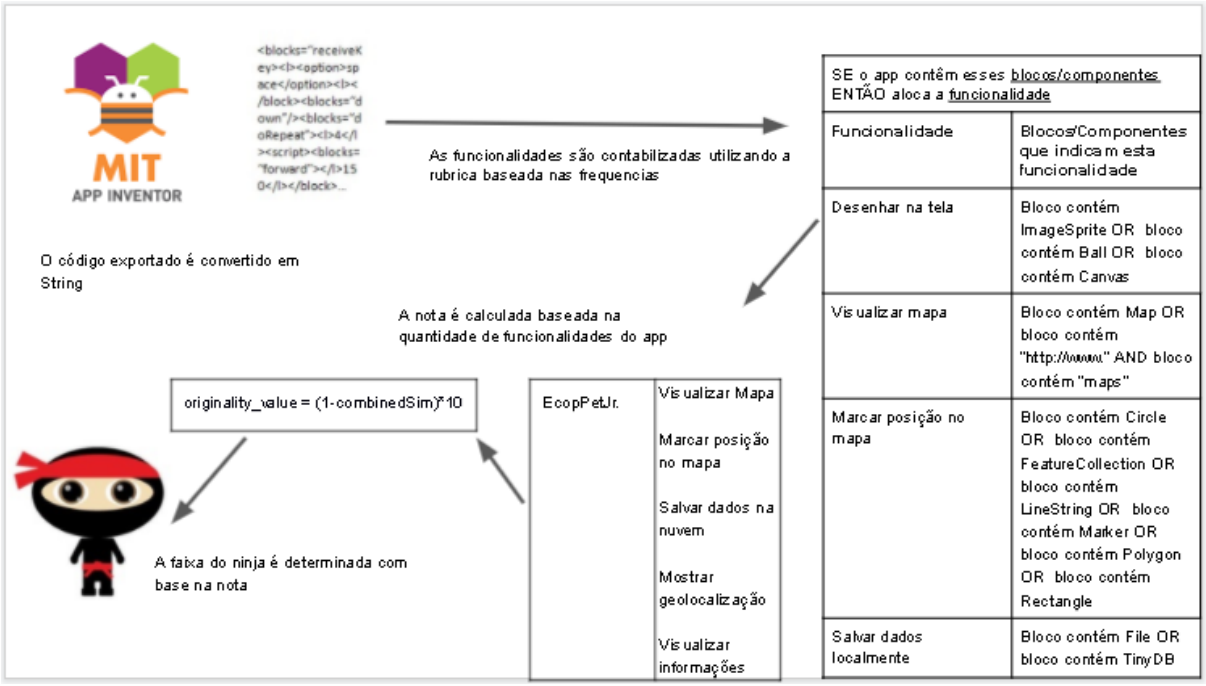
Ameaças a validade. A busca limitada ao SCOPUS pode ter causado a omissão de algum resultado relevante, para mitigar esse risco, foi construída cuidadosamente o *string* de busca para ser o mais inclusiva possível, considerando não apenas os conceitos principais, mas também seus sinônimos, levando em conta também a verificação por buscas de calibração em mais bases inclusive de forma ampla via google scholar, que demonstraram que este risco seja pequeno. Ameaças à seleção de abordagens relevantes e extração de dados foram mitigadas fornecendo uma definição detalhada de critérios de inclusão/exclusão. É definido e documentado um protocolo rígido para a seleção dos estudos, sendo discutido com os orientadores a seleção de artigos até que um consenso seja alcançado.

4 MODELO DE AVALIAÇÃO DA ORIGINALIDADE DA FUNCIONALIDADE DE APPS COM APP INVENTOR

Esta seção apresenta o modelo de avaliação da originalidade da funcionalidade de apps desenvolvidos com App Inventor voltado ao contexto da educação básica. São adotadas técnicas de IA para o desenvolvimento do presente modelo.

A partir da lista de funcionalidades definidas, são extraídas um conjunto de funcionalidades contidas dentro de determinados blocos de componentes de um aplicativo. A partir dos blocos de comando extraídos são inferidas a(s) funcionalidade(s) usando regras. O grau da similaridade da(s) funcionalidades de um novo aplicativo com as funcionalidades de todos os aplicativos dentro de um universo de referência é calculado utilizando uma medida da similaridade. Esse grau de similaridade é utilizado para o cálculo da nota de originalidade como demonstrado na Figura 13.

Figura 13 - Método de avaliação de originalidade.



Fonte: Elaborado pelo autor.

O universo de referência do modelo de avaliação é composto de 10.000 aplicativos criados com App Inventor. Esses aplicativos foram obtidos pela galeria do App Inventor (https://gallery.appinventor.mit.edu) e pela iniciativa Computação na Escola

(https://computacaonaescola.ufsc.br), com um foco educacional voltado à adoção da estratégia instrucional de ação computacional, foram excluídos aplicativos de jogos.

4.1 EXTRAÇÃO DE *FEATURES*

A(s) funcionalidade(s) de um aplicativo são identificadas automaticamente a partir do código do aplicativo (projeto.aia), inferindo a partir da presença de determinados componentes a presença das funcionalidades. Para automatizar a extração de componentes do código é utilizado o *parser* da ferramenta CodeMaster (Gresse von Wangenheim et al., 2018a). Nele, ao extrair os blocos de código de um arquivo .scm, são detectados se existem componentes contidos em um determinado aplicativo (Figura 14).

Figura 14 - Trecho de código de um arquivo .scm de um aplicativo do App Inventor.

Fonte: Elaborado pelo autor.

São considerados somente componentes do *core* do App Inventor. Componentes das categorias *Lego Mindstorms* e *Experimental* não são abordados neste trabalho por conta de seu uso específico.

4.2 INFERÊNCIA DE FUNCIONALIDADES

Com base nos componentes extraídos do código, são inferidas a(s) funcionalidade(s) do aplicativo. Para esse processo é adotada a técnica de sistemas baseado em regras (Tabela 19). As regras foram definidas manualmente pelo autor em conjunto com dois pesquisadores.

Tabela 19 - Definição das regras para a inferência de funcionalidades de aplicativos.

SE o app contêm esses blocos/componentes ENTÃO aloca a funcionalidade	
Funcionalidade	**Blocos/Componentes que indicam esta funcionalidade**
Acessar site	Bloco contém WebViewer OR bloco contém "http://www." OR bloco contém Web
Compartilhar artefato	Bloco contém Sharing OR bloco contém Texting OR bloco contém Twitter
Contar passos	Bloco contém Pedometer
Converter fala em texto	Bloco contém SpeechRecognizer
Converter texto em fala	Bloco contém TextToSpeech
Desenhar na tela < 2	Bloco contém ImageSprite OR bloco contém Ball OR bloco contém Canvas
Desenhar na tela >= 2	Bloco contém ImageSprite OR bloco contém Ball OR bloco contém Canvas AND contém ImageSprite OR bloco contém Ball OR bloco contém Canvas
Detectar aceleração	Bloco contém AccelerometerSensor
Escanear QR code	Bloco contém BarcodeScanner
Escolher imagem da galeria	Bloco contém ImagePicker
Fazer ligação	Bloco contém PhoneCall
Fazer login	Bloco contém File OR bloco contém TinyDB OR bloco contém FirebaseDB OR bloco contém TinyWebDB OR bloco contém CloudDB AND bloco contém PasswordTextBox
Fazer tradução	Bloco contém YandexTranslate
Gravar áudio	Bloco contém SoundRecorder
Gravar vídeo	Bloco contém Camcorder
Marcar posição no mapa	Bloco contém Circle OR bloco contém FeatureCollection OR bloco contém LineString OR bloco contém Marker OR bloco contém Polygon OR bloco contém Rectangle
Marcar tempo	Bloco contém Clock

Medir campo magnético	Bloco contém MagneticFieldSensor
Medir giroscópio	Bloco contém GyroscopeSensor
Medir nível de luz	Bloco contém LightSensor
Medir pressão do ar	Bloco contém Barometer
Medir proximidade	Bloco contém ProximitySensor
Medir temperatura	Bloco contém Thermometer
Medir umidade relativa do ar	Bloco contém Hygrometer
Mostrar geolocalização	Bloco contém LocationSensor
Mostrar orientação espacial do dispositivo	Bloco contém OrientationSensor
Reconhecer voz	Bloco contém SpeechRecognizer
Reproduzir som	Bloco contém Sound OR bloco contém Player
Reproduzir vídeo	Bloco contém VideoPlayer
Salvar dados localmente	Bloco contém File OR bloco contém TinyDB
Salvar dados na nuvem	Bloco contém FirebaseDB OR bloco contém TinyWebDB OR bloco contém CloudDB
Tirar foto	Bloco contém Camera
Usar api de outro app	Bloco contém ActivityStarter
Usar bluetooth	Bloco contém BluetoothClient OR bloco contém BluetoothServer
Visualizar informações	Bloco contém Label com caracteres >= 20 OR Bloco contém TextBox com caracteres >= 20
Visualizar lista de contatos	Bloco contém ContactPicker OR bloco contém EmailPicker OR bloco contém PhoneNumberPicker
Visualizar mapa	Bloco contém Map OR bloco contém “http://www.” AND bloco contém “maps”

Fonte: Elaborado pelo autor.

Exemplos de funcionalidade(s) extraídas de aplicativos são apresentadas na Tabela 20.

Tabela 20 - Exemplos de aplicativos com as suas funcionalidades extraídas.

Nome do app	**Breve descrição**	**Funcionalidade(s) extraída(s)**
Receitas	Aplicativo informando receitas culinárias	Visualizar informações
HelloPurr	Aplicativo que reproduz som de um gato ao clicar no botão	Reproduzir som
EcopPetJr.	Aplicativo para encontrar pontos de coleta de tampinhas recicláveis num mapa	Visualizar Mapa Marcar posição no mapa Salvar dados na nuvem Mostrar geolocalização Visualizar informações

Fonte: Elaborado pelo autor.

Toda a informação das telas de cada arquivo e seus respectivos componentes são guardados em uma lista. Ao percorrer essa lista, se a lista contém algum componente, é adicionado um marcador do determinado componente presente.

Para criação de um universo de referência, foram extraídos os componentes e inferidas a(s) funcionalidade(s) dos 10.000 aplicativos. Analisando a composição do universo de referência observa-se a distribuição de frequências de funcionalidades individuais conforme apresentado na Tabela 21.

Tabela 21 - Frequência de cada funcionalidade no universo de referência (com total de 10.000 aplicativos).

Funcionalidade	**Número de ocorrências**
Reproduzir som	3645
Visualizar informações	2386
Marcar tempo	1860
Converter texto em fala	1483
Desenhar na tela < 2	1403
Salvar dados localmente	1102
Detectar aceleração	968
Acessar site	778

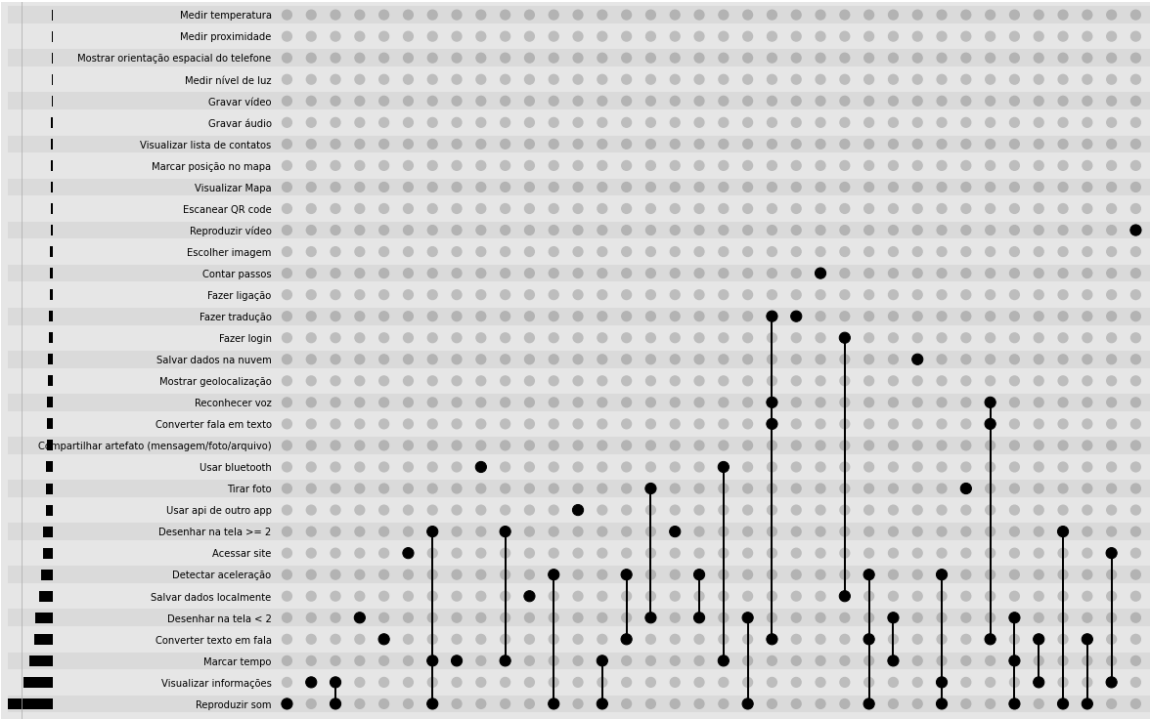
Desenhar na tela >= 2	749
Usar api de outro app	559
Tirar foto	557
Usar bluetooth	507
Compartilhar artefato	465
Reconhecer voz	429
Converter fala em texto	429
Mostrar geolocalização	391
Salvar dados na nuvem	368
Fazer login	273
Fazer tradução	272
Fazer ligação	211
Contar passos	192
Escolher imagem	186
Reproduzir vídeo	165
Escanear QR code	155
Visualizar Mapa	149
Marcar posição no mapa	126
Visualizar lista de contatos	125
Gravar áudio	97
Gravar vídeo	94
Medir nível de luz	79
Mostrar orientação espacial do dispositivo	72
Medir proximidade	40
Medir temperatura	16

Medir giroscópio	8
Medir pressão do ar	8
Medir umidade relativa do ar	2
Medir campo magnético	0

Fonte: Elaborado pelo autor.

A Figura 15 visualiza a distribuição de frequências em relação a combinações de funcionalidades presentes em conjunto nos aplicativos do universo de referência. Podemos perceber a ocorrência maior de aplicativos contendo apenas uma funcionalidade, e com apenas um aplicativo contendo mais de três funcionalidades.

Figura 15 - Extrato da distribuição das frequências de combinações de funcionalidades dos aplicativos no universo de referência.



Fonte: Elaborado pelo autor.

4.3 MEDIDA DE SIMILARIDADE

Para avaliar um aplicativo *query* (aplicativo que irá ser avaliado pelo modelo), a(s) respectivas funcionalidades do aplicativo são extraídas e comparadas com as funcionalidades extraídas de aplicativos do universo de referência (10.000 aplicativos). A comparação inclui quantas vezes cada funcionalidade aparece no aplicativo *query vs.* no universo de referência e qual(is) combinação(ões) ocorre(m) no aplicativo *query vs.* no universo de referência, utilizando de uma medida de similaridade. Foram experimentadas diferentes medidas de similaridade de *Cosine, Euclidean e Jaccard.* Além dessas medidas, foram criadas medidas específicas:

Individual occurrence: Percorre as linhas do conjunto de funcionalidades extraídas do universo de referência (10.000 aplicativos), adicionando a frequência de todas as funcionalidades contando o número de funcionalidades presentes neste aplicativo.

```
para app query em (universo de referência [10000])
    soma_frequência = 0
    soma_funcionalidades = 0

para funcionalidade em (total de funcionalidades [37])
    se contém funcionalidade
        soma_funcionalidades  =+ 1
        soma_frequência = soma_frequência + (frequência individual/universo de referência[10000])
Individual occurrence  = soma_frequência / soma_funcionalidades
```

Com:

- soma_frequência: soma das frequências de funcionalidades presentes no universo de referência.
- soma_funcionalidades: total de funcionalidades presentes no universo de referência.
- frequência individual: frequência para um recurso individual no universo de referência

Combined similarity: Percorre um aplicativo *query* para as funcionalidades presentes no universo de referência, adicionando as contagens de combinações de funcionalidades presentes no *query* e somando com os resultados obtidos em *Individual occurrence*:

soma_combinações = combinações de funcionalidades [universo de referência]
soma_ocorrência_individual = *Individual occurrence* [universo de referência]

Combined similarity = ((soma_combinações + soma_ocorrência_individual)/2)

Com:

- soma_combinações: soma das combinações de funcionalidades presentes nos aplicativos *query* no universo de referência.
- soma_ocorrência_individual: resultado de *Individual occurrence* presentes nos aplicativos *query* no universo de referência.

4.3.1 AVALIAÇÃO DO DESEMPENHO COMPARANDO AS MEDIDAS

Para analisar e comparar o desempenho das diferentes medidas de similaridade foi realizado um teste com 10 aplicativos. Esses aplicativos foram selecionados visando cobrir todo intervalo de grau de similaridade, incluindo aplicativos com funcionalidades raras até com aplicativos com funcionalidades muito comuns em relação ao universo de referência.

Para cada um dos aplicativos deste conjunto foi indicado um *ranking* de similaridade em relação ao universo de referência por um painel de especialistas (com 1° sendo o mais original e o 10° sendo o menos original). O painel foi composto pelo autor e dois pesquisadores nessa área de conhecimento.

Tabela 22 - Visão geral da comparação do grau de similaridade dos 10 aplicativos de referência usados para testes.

Aplicativo		**Funcionalidades extraídas**	***Ranking* de similaridade indicado por painel de especialistas**
Nome	**Breve descrição**		
EcopPetJr.	Aplicativo para encontrar pontos de coleta de tampinhas recicláveis num mapa	Visualizar Mapa Marcar posição no mapa Salvar dados na nuvem Mostrar geolocalização	6°

		Visualizar informações	
VamosNosOrganizar	App para anotações diárias	Salvar dados localmente Visualizar informações	8°
TeatroVirtiual	App que permite escanear um QR code	Escanear QR code	5°
Videorecorder	App para gravação e reprodução de vídeo	Salvar dados localmente Gravar vídeo Reproduzir vídeo	2°
BT-SMS	App para trocar mensagens e usar blueto-oth	Salvar dados localmente Marcar tempo Compartilhar artefato Visualizar lista de contatos Usar api de outro app Usar bluetooth	3°
SensorApp	App que possibilita utilizar vários sensores	Detectar aceleração Medir giroscópio Medir proximidade Medir pressão do ar Medir umidade relativa do ar Medir campo magnético Medir temperatura	1°

		Visualizar informações	
Paint	App para desenhar na tela	Desenhar na tela < 2 Compartilhar artefato Tirar foto	7°
Não late	Aplicativo que reproduz o som de um cachorro	Reproduzir som	9°
Receitas	Aplicativo informando receitas culinárias	Visualizar informações	10°
EducPortifolio	Aplicativo que permite a gravação de voz	Salvar dados localmente Reconhecer voz Converter fala em texto Usar api de outro app	4°

Fonte: Elaborado pelo autor.

Para analisar a correlação entre o *ranking* de similaridade automaticamente calculado e o *ranking* manualmente alocado pelos especialistas foi analisada a correlação entre os *rankings* utilizando a correlação de Spearman. O coeficiente de correlação de Spearman, (rho, também representado por rs) mede a força e a direção da associação entre duas variáveis classificadas (Laerd statistics, 2021). A correlação de Spearman varia de 1 a -1 de acordo com os valores comparados. Quando esses valores são perfeitamente relacionados, o coeficiente de correlação de Spearman se torna 1, ou seja, quanto mais a medida de similaridade se aproxime de 1, maior sua correlação. A relação é positiva quando as variáveis aumentam simultaneamente e negativa quando uma variável aumenta, a outra variável por consequência diminui.

Figura 16 - Comparação entre os valores das correlações entre o julgamento humano (médias manual) e cada uma das medidas de similaridade.

	Médias manual	Combined Similarity	Cosine	Euclidean	Jaccard
Médias manual	1.000	-0.807	0.721	-0.721	0.709
Combined Similarity	-0.807	1.000	-0.758	0.758	-0.826
Cosine	0.721	-0.758	1.000	-1.000	0.976
Euclidean	-0.721	0.758	-1.000	1.000	-0.976
Jaccard	0.709	-0.826	0.976	-0.976	1.000
Individual Occurence	-0.711	0.837	-0.942	0.942	-0.985

Fonte: Elaborado pelo autor.

Na Figura 16 a coluna “Médias manual” contém o julgamento humano comparada com as medidas de similaridade nas demais colunas, pode-se observar que as duas medidas de similaridade Cosine e *Combined similarity* obtiveram mais forte correlação positiva e negativa, respectivamente. Observando o valor em termos absolutos de correlação mais alto, Combined similarity (0.807) foi selecionada para o cálculo das notas.

4.4 CÁLCULO DA NOTA

Usando o grau de similaridade medido por *Combined similarity* como entrada é definido o cálculo da nota de originalidade. Observa-se que a originalidade representa o inverso da similaridade. A nota de originalidade é definida no intervalo de [0 - 10] conforme as notas tipicamente utilizadas na educação básica em escolas brasileiras. Dessa forma a nota 10 representa um aplicativo muito original e a nota 0 a um aplicativo nada original em relação a sua(s) funcionalidade(s).

A transformação do grau de similaridade de um aplicativo em uma nota de originalidade é feita por uma transformação usando uma função. A função foi definida com base numa alocação manual de notas de originalidade por especialistas utilizando os 10 aplicativos de testes e os valores de similaridade calculados automaticamente (Tabela 23).

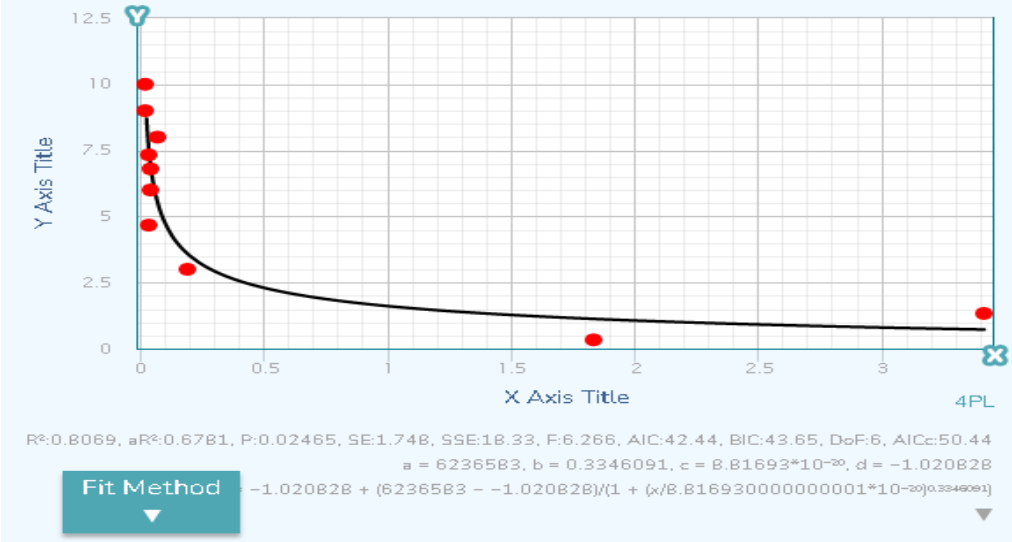
Tabela 23 - Valores de similaridade e notas de originalidade para o conjunto de testes.

App	**Valor de similaridade calculado automaticamente com combined similarity**	**Nota de originalidade manualmente alocado por especialistas**
SensorApp	0.02	10
Videorecorder	0.02	9
BT-SMS	0.04	6.8
EducPortifolio	0.03	7.33
TeatroVirtiual	0.04	6
EcopPetJr.	0.03	4.67
Paint	0.07	8
VamosNosOrganizar	0.19	3
Não late	3.41	1.33
Receitas	1.83	0.33

Fonte: Elaborado pelo autor.

Foi feita a alocação manual de notas e foram geradas curvas logísticas de quatro parâmetros (4PL), para transformar os valores das medidas de similaridade a notas. Utilizou-se a ferramenta do site MycurveFit (https://mycurvefit.com) para comparar alternativas de funções. Para os valores comparativos, serão usados os valores das medidas como X e as médias das notas manuais dos especialistas como Y (Figura 17), assim obtendo suas *Goodness Measures*, que são várias medidas fornecidas para quantificar quão bem o ajuste de curva modela os dados (Figura 18).

Figura 17 - Função de transformação com base na medida de Combined similarity.



Fonte: Elaborado pelo autor.

Figura 18 - Goodness Measures gerada pela medida de Combined similarity.

Goodness Measures What's this?		
R^2		0.8069
aR^2		0.6781
P		0.02465
SE		1.748
SSE		18.33
F		6.266
AIC		42.44
BIC		43.65
DoF		6
AICc		50.44
Coefficients		
a	6236583	± 30290000000000
b	0.3346091	± 2.803
c	$8.81693*10^{-20}$	± $1.282*10^{-12}$
d	−1.020828	± 18.6
Equation		
$y = -1.020828 + (6236583 - -1.020828)/(1 + (x/8.816930000000001*10^{-20})^{0.3346091})$		

Fonte: Elaborado pelo autor.

Dessa forma o valor de similaridade automaticamente calculado usando *Combined similarity* é transformado em uma nota de originalidade utilizando a seguinte fórmula:

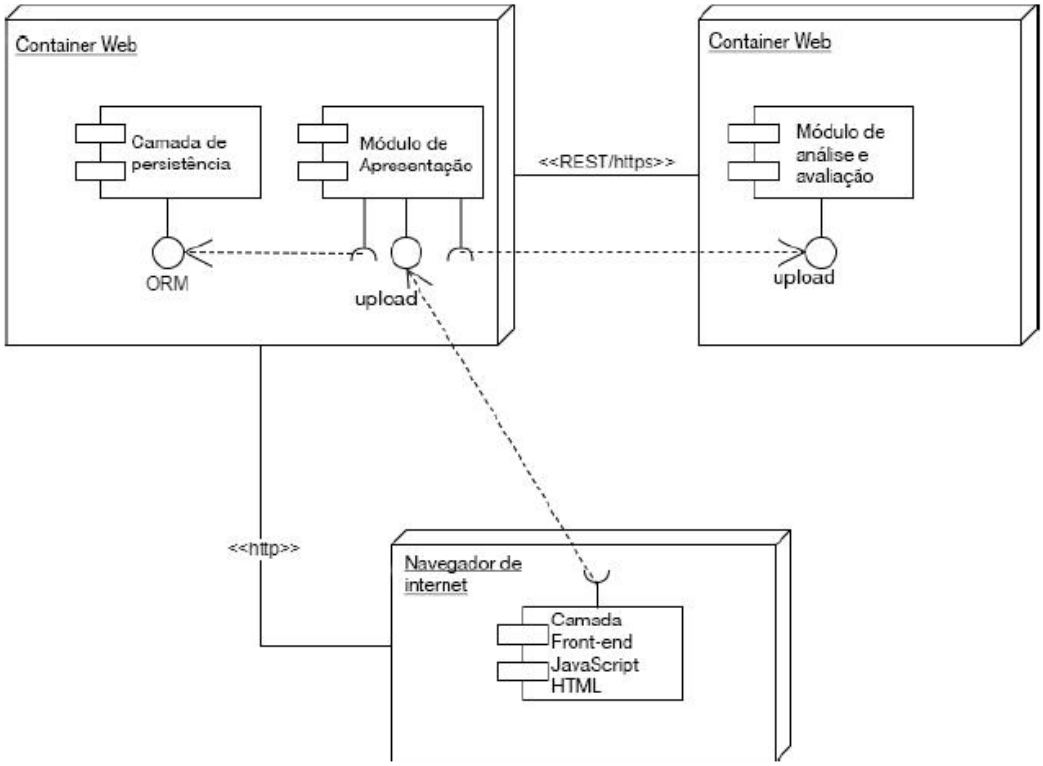
Nota de originalidade = -1.020828 + (6236583 + 1.020828) / (1 + (*Combined similarity*/8.816930000000001e-20) ^0.3346091)

(4.1)

5 INTEGRAÇÃO NO CODEMASTER

Para possibilitar o uso da avaliação de originalidade de funcionalidades de apps feitas com App Inventor, foi implementada a automação do processo de avaliação integrado à ferramenta CodeMaster. A ferramenta CodeMaster (Alves, 2019) é uma aplicação *web* gratuita que, em um contexto de aprendizagem baseada em problemas, permite avaliar automaticamente projetos desenvolvidos com App Inventor e Snap! em relação a conceitos de programação, *design* de interface de usuário e estética visual, num contexto educacional.

Figura 19 - Arquitetura atual do CodeMaster



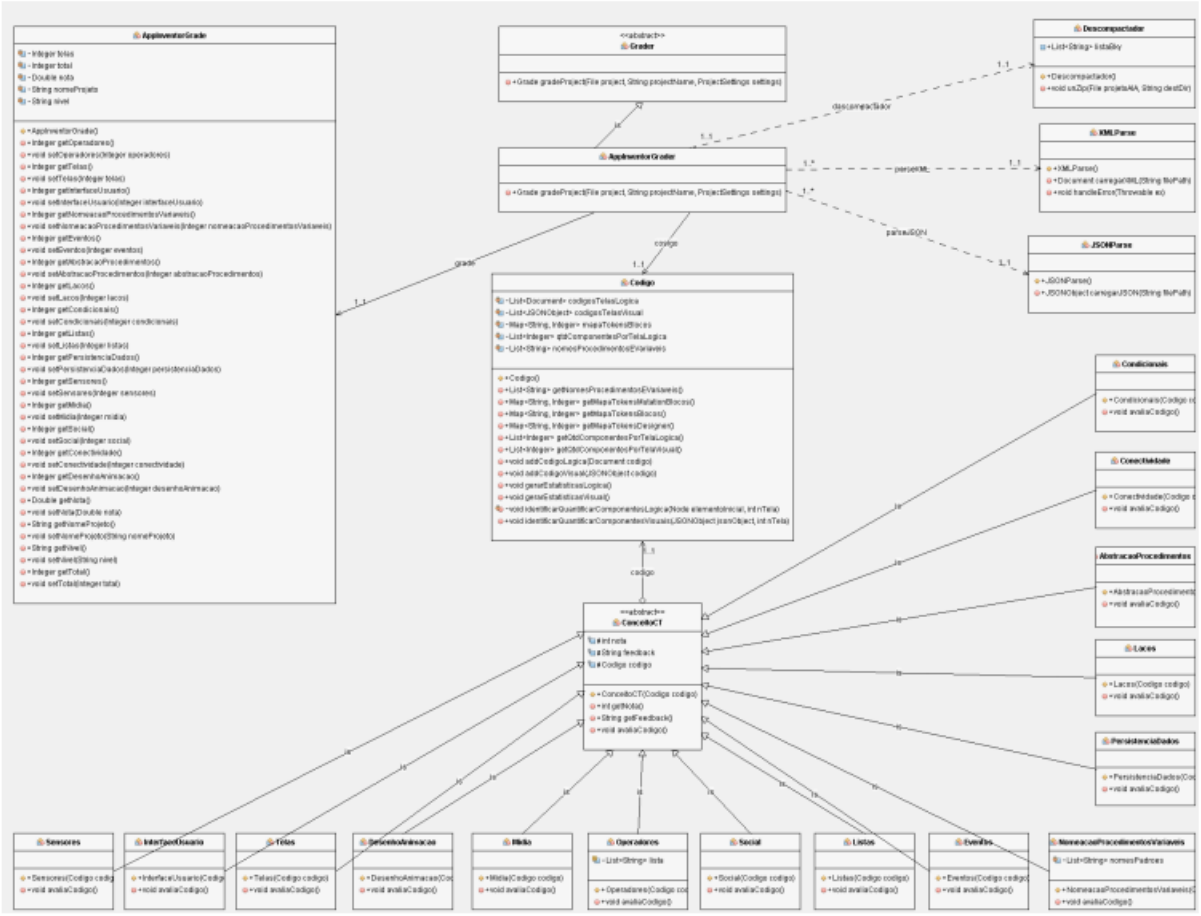
Fonte: Alves (2019).

No contexto do presente trabalho visa-se a inclusão de um novo fator de avaliação referente a avaliação de originalidade das funcionalidades de um aplicativo. Com isso, se manteve a arquitetura existente do CodeMaster V1.0 (Figura 19) (Alves, 2019).

Para adicionar um novo critério de avaliação no CodeMaster, foi criado uma nova classe chamada Originalidade que estende a classe ConceitoCT (Figura 20). Essa nova classe implementa o método avaliaCodigo. Por herança, essa nova classe receberá a instância de

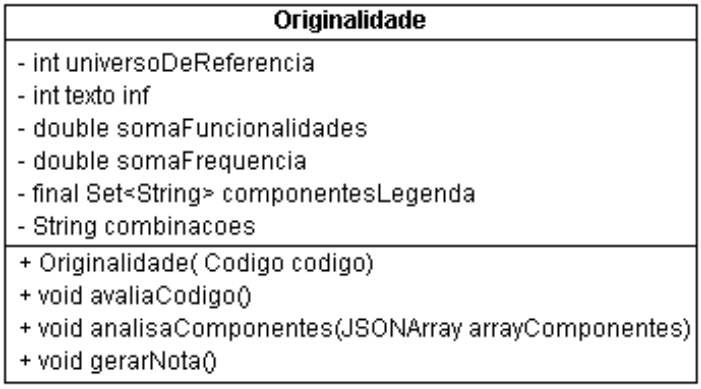
código, além dos dados referentes às demais avaliações já implementadas no CodeMaster (algoritmo e programação e *design* visual). A classe AppInventorGrader do CodeMaster v1.0 é o controle do analisador. ela deve obter a pontuação da nova classe, colocar a pontuação na instância de AppInventorGrader e somar na variável de pontuação total (Figura 21) (Demetrio, 2017).

Figura 20 - Diagrama de classes do analisador e avaliador do App Inventor.



Fonte: Demetrio (2017).

Figura 21 - Classe Originalidade.



Fonte: Elaborado pelo autor.

Seguindo o processo de avaliação conforme detalhada na Seção 4, a implementação inclui:

- Armazenamento de um universo de referência composto de informações da quantidade/frequência de funcionalidades individuais e das combinações de funcionalidades em relação aos 10.000 aplicativos, utilizando os atributos:
  - **int universoDeReferencia** para guardar o valor total dos aplicativos usados no universo de referência.
  - **double somaFuncionalidades** para guardar a soma de funcionalidades presentes dentro do aplicativo avaliado.
  - **double somaFrequencia** para guardar a frequência individuais em relação ao universo de referência.
  - **String combinacoes** para guardar as combinações de funcionalidades do aplicativo avaliado dentro da classe Originalidade.

Análise de um aplicativo através de seu código-fonte, por meio de análise estática de código reutilizando o *parser* do CodeMaster, para extrair os tipos de blocos utilizados no programa. As diferentes análises são obtidas através das funções já existentes no CodeMaster: analiseProgram para avaliação do algoritmo e programação e analiseVisual para avaliação do *design* visual. Voltado ao objetivo do presente trabalho foi acrescentado a função double **analiseOriginalidade** (ProjectSettings settings, Codigo codigo, AppInventorGradePrograma grade), que avalia a originalidade do aplicativo usando o método avaliaCodigo.

Figura 22 - Trecho de código com a análise de originalidade de um aplicativo em AppInventorGrader.

```
private double analiseOriginalidade (ProjectSettings settings, Codigo codigo, AppInventorGradePrograma grade){



    double totalOriginalidade = 0;

    if(settings.getOriginalidadeIsRelevant()) {
        ConceitoCT dados = new AvaliarOriginalidade(codigo);
        dados.avaliaCodigo();
        grade.setOriginalidade(dados.getNotaFinalOriginalidade());
        grade.setLista_funcionalidades_app(dados.getLista_funcionalidades_app());
        totalOriginalidade = dados.getNotaFinalOriginalidade();
        log.info("Sua nota em Persistencia de Dados foi: " + dados.getNotaFinalOriginalidade() + " | Feedback: " + dados
    }else {
        grade.setOriginalidade((double) 0);
    }

    //coloca total em grade
    grade.setTotalProgramacao((int) totalOriginalidade);
```

Fonte: Elaborado pelo autor.

O método avaliaCodigo é composto de condicionais extraindo as funcionalidades de um app novo e ao mesmo tempo usando as frequências das funcionalidades do universo de referência para calcular *Combined similarity* (Figura 22).

Figura 23 - Trecho de código com o cálculo de Combined similarity.

```
    private void gerarNota () {
        double ocorencia_individual = 0;
        ocorencia_individual = soma_frequencia/soma_funcionalidades;

        double combinacao_similaridade = (soma_funcionalidades + ocorencia_individual) / 2;

        this.notaFinalOriginalidade = ((-1.020828 + (6236583 - -1.020828))/(1 +
                (Math.pow(combinacao_similaridade/(Math.pow(8.816930000000001 *10, -20)),0.3346091))));
    }

}
```

Fonte: Elaborado pelo autor.

O cálculo da originalidade do novo app é feito com base no valor de *Combined similarity* usando a função gerarNota na classe Originalidade, conforme apresentado na seção 4.3.1 (Figura 23).

Como resultado para o usuário são apresentados conforme mostrado nas Figuras 24 e 25.

Figura 24 - Aba de originalidade no CodeMaster.

Fonte: Elaborado pelo autor.

Utilizando a classificação de raridade de funcionalidades e de combinações de funcionalidades, respectivamente, como apresentado nas Tabelas 24 e 25.

Tabela 24 - Categorias de classificação da raridade de funcionalidades.

Categorias de classificação da raridade de funcionalidades	**Frequência no universo de referência (n=10.000)**	**Cor do label**
Muito comum	frequência acima de 1000	#ee7f1d
Comum	frequência entre 500 a 1000	#92268e
Rara	frequência entre 100 a 500	#007aca
Muito rara	frequência menor que 100	#009e2c

Fonte: Elaborado pelo autor.

Tabela 25 - Categorias de classificação da raridade de combinação de funcionalidade.

Categorias de classificação da raridade de funcionalidades	**Frequência no universo de referência (n=10.000)**	**Cor do label**
Muito comum	frequência acima de 50	#ee7f1d

Comum	frequência 10 a 50	#92268e
Rara	frequência 1 a 10	#007aca
Muito rara	frequência igual a 0	#009e2c

Fonte: Elaborado pelo autor.

Figura 25 - Interface com informações do universo de referência.

Fonte: Elaborado pelo autor.

O código será disponibilizado em: https://codigos.ufsc.br/100000000394729/CodeMaster/-/tree/tcc/matheus.alberto.

6 CONCLUSÃO

Tendo em vista o desenvolvimento de um modelo de avaliação da originalidade da funcionalidade de aplicativos desenvolvidos com App Inventor na educação básica, foi feita a síntese em relação aos conceitos de originalidade, App Inventor, *feature extraction* e medidas de similaridade, a análise de modelos de avaliação de originalidade de aplicativos desenvolvidos com App Inventor no contexto da educação básica. Após isso, foi desenvolvido uma ferramenta de avaliação da originalidade em relação às funcionalidades presentes em um aplicativo *query* do App Inventor, e a integração deste modelo na ferramenta CodeMaster.

Como principal resultado, foi desenvolvida uma ferramenta que avalia a originalidade de aplicativos criados no App Inventor, extraindo cada funcionalidade presente nos componentes do aplicativo *query* e atribuindo uma nota para seu nível de originalidade. Com isso, espera-se contribuir para a avaliação do desenvolvimento da criatividade de alunos na educação básica no contexto de desenvolvimento de aplicativos.

Para trabalhos futuros é sugerida uma análise com maior ênfase em identificar outras possíveis funcionalidades contidas dentro de um código de um aplicativo no contexto do App Inventor, assim, podendo avaliar a originalidade de um aplicativo de forma mais ampla.

REFERÊNCIAS

Abraham, A., **Rule-Based Expert Systems, Handbook of Measuring System Design**. AISC (2005), Specification for Structural Steel Buildings, American Institute of Steel Construction, Inc.,Chicago, IL, USA, 2005.

Akram, J., Shi, Z., Mumtaz, M. and Luo, P. **DroidCC: A Scalable Clone Detection Approach for Android Applications to Detect Similarity at Source Code Level**. 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), 2018, pp. 100-105, doi: 10.1109/COMPSAC.2018.00021.

Al-Subaihin, A., Sarro, F., Black, S. et al. **Empirical comparison of text-based mobile apps similarity measurement techniques**. Empir Software Eng 24, 3290–3315 (2019).

Alves, N., C.**CodeMaster: Um Modelo de Avaliação do Pensamento Computacional na Educação Básica através da Análise de Código de Linguagem de Programação Visual**. Dissertação (Programa de Pós-Graduação em Ciência da Computação (PPGCC)) – Universidade Federal de Santa Catarina (2019).

Alves, N.C., Gresse von Wangenheim, C., Hauck, J.C.R. **Teaching Programming to Novices: A Large-scale Analysis of App Inventor Projects.** 2020 XV Conferencia Latinoamericana de Tecnologias de Aprendizaje (LACLO), Loja, Ecuador, 2020, pp. 1-10, doi: 10.1109/LACLO50806.2020.9381172.

Alves, N.C., Gresse von Wangenheim, C., Alberto, M., Martins,P., L., H. **Uma Proposta de Avaliação da Originalidade do Produto no Ensino de Algoritmos e Programação na Educação Básica**. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 31. , 2020, Online. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2020. p. 41-50.

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

Araújo, G., de Medeiros, S., Bergmann, J. C. F., & von Wangenheim, C. G. **Práticas pedagógicas com o desenvolvimento de aplicativos para dispositivos móveis por estudantes da Educação Básica**. TEXTURA-Revista de Educação e Letras. 2020, p.22-49.

Bag, S., Kumar, S. K., & Tiwari, M. K. **An efficient recommendation generation using relevant Jaccard similarity**. Information Science. 2019, 483, p.53-64.

Binkley, M., Erstad, O., Herman, J., Raizen, S., Ripley, M. and Rumble, M. **Defining 21st Century Skills.** Draft white paper 1. Melbourne: The University of Melbourne, Melbourne, Australia. 2010

Chen, X., Zou, Q., Fan, B., Zheng, Z., & Luo, X. . **Recommending software features for mobile applications based on user interface comparison**. Requirements Engineering (2019), 24(4), p.545-559.

Cordeiro A., Oliveira G.M., Rentería J.M., Guimarães C.A., Grupo de Estudo de RS do Rio de Janeiro. **Revisão sistemática: uma revisão narrativa.** Revista de Colégio Brasileiro de Cirurgiões, vol. 34, n. 6, 2007, p. 428-431.

Crussell, J., Gibler, C. and Chen, H., **AnDarwin: Scalable Detection of Android Application Clones Based on Semantics**. IEEE Transactions on Mobile Computing, vol. 14, no. 10, pp. 2007-2019, 1 Oct. 2015, doi: 10.1109/TMC.2014.2381212.

Daniel A., Michael S., Malte H.. **Drebin: Efficient and explainable detection of Android malware in your pocket**,' in Proc. 21th Annu. Netw. Distrib. Syst. Secur. Symp. (NDSS), 2014, pp. 23–26.

Dean, Douglas L. and Hender, Jill and Rodgers, Tom and Santanen, Eric. **Identifying Good Ideas: Constructs and Scales for Idea Evaluation (2006).** Journal of Association for Information Systems, vol. 7, n. 10, 2006, p. 646-699.

DeepAI, "What is Feature Extraction" disponível em: <https://deepai.org/machine-learning-glossary-and-terms/feature-extraction#:~:text=Feature%20extraction%20is%20the%20name,describing%20the%20original%20data%20set.>. Acesso em: Julho 2020.

Demetrio, M. F. **Desenvolvimento de um analisador e avaliador de código de App Inventor para ensino de computação**, 2017.

Dicionário Merriam-Webster, "Originality." disponível em: <https://www.merriam-webster.com/dictionary/originality>. Acesso em: Maio 2020.

Dicionário Online de Português, "Funcionalidade" disponível em: <https://www.dicio.com.br/funcionalidade/>. Acesso em: Julho 2020.

Elmore, K. L., & Richman, M. B. **Euclidean distance as a similarity metric for principal component analysis**. Monthly weather review, 129(3). 2001, p.540-549.

IEEE, "IEEE Standard Glossary of Software Engineering Terminology." disponível em: <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=159342>. Acesso em: Agosto 2020.

Garrett, J. **The Elements of User Experience: User-Centered Design for the Web and Beyond** (2nd. ed.). New Riders Publishing, USA. 2010

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

Goldenberg, J., Mazursky, D., Solomon, S. **Creativity templates: Towards identifying the fundamental schemes of quality advertisements.** Marketing Science, vol.18, n.3, 1999, p. 333-351

Gopalan, Rohit. "**An Assessment Tool to Analyze Code Written in App Inventor.**" 2018.

Gresse von Wangenheim, C., Hauck, J.C.R., Demétrio, M.F., Pelle, R., Alves, N. d. C., Barbosa, H., Azevedo, L. F. **CodeMaster - Automatic Assessment and Grading of App Inventor and Snap! Programs.** Informatics in Education, vol. 17, n. 1, 2018a, p. 117-150.

Gresse von Wangenheim,C., Araujo Porto, J.V., Hauck, J.C.R., Borgatto, A.F., **Do we agree on user interface aesthetics of Android apps?** arXiv:1812.09049[cs.SE], 2018b.

Guo, Y., Li, Y., Yang, Z., & Chen, X. . **What's inside my app? understanding feature redundancy in mobile apps**. In Proceedings of the 26th Conference on Program Comprehension, 2018, May (pp. 266-276).

Haddaway, N. R., Collins, A. M., Coughlin, D., & Kirk, S. **The role of Google Scholar in evi-dence reviews and its applicability to grey literature searching**. PLoS ONE,10(9), e0138237, 2015.

Hattie, J., Timperley, H. **The Power of Feedback. Review of Educational Research**, vol. 77, n. 1, 2007, p. 81-112.

Horn D., Salvendy G. **Product creativity: Conceptual model, measurement, and characteristics.** Theoretical Issues in Ergonomics Science, vol.7, n.4, 2006, p. 395-412

Hu, Y., Xu, G., Zhang, B., Lai, K., Xu, G. and Zhang, M. **Robust App Clone Detection Based on Similarity of UI Structure**.IEEE Access, vol. 8, pp. 77142-77155, 2020, doi: 10.1109/ACCESS.2020.2988400

Jackson, P. W., Messick, S. (1964). **The Person, The Product, and the Response**. ETS Research Bulletin Series, 2, i-27.

Johann, T., Stanik, C., Alireza, M. A. B., Maalej, W. **Safe: A simple approach for feature extraction from app descriptions and app reviews**, IEEE 25th International Requirements Engineering Conference (RE), pp. 21-30, 2017.

Laerd statistics, “Spearman's Rank-Order Correlation.” disponível em: <https://statistics.laerd.com/statistical-guides/spearmans-rank-order-correlation-statistical-guide.php>. Acesso em: Julho de 2021.

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J., Werner, L. **Computational thinking for Youth in Practice**, ACM Inroads, vol.2, n. 1. 2011

Li, Y., Pan, Y., Liu., W. and Zhang, X. **An Automated Evaluation System for App Inventor Apps**. 6th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech), Athens, Greece, 2018, pp. 230-235.

Li, L., Bissyandé, T., F. and Klein, J. **SimiDroid: Identifying and Explaining Similarities in Android Apps**. 2017 IEEE Trustcom/BigDataSE/ICESS, 2017, pp. 136-143, doi: 10.1109/Trustcom/BigDataSE/ICESS.2017.230.

Linares-Vásquez, M., Holtzhauer, A. and Poshyvanyk, D. **On automatically detecting similar Android apps**. 2016 IEEE 24th International Conference on Program Comprehension (ICPC), 2016, pp. 1-10, doi: 10.1109/ICPC.2016.7503721.

Kanda, T., Manabe, Y., Ishio, T., Matsushita, M., & Inoue, K. **Semi-automatically extracting features from source code of android applications**. IEICE TRANSACTIONS on Information and Systems (2013), 96(12), 2857-2859.

Kierski,M. **Machine learning development process - you’ve got it wrong**, 2017

Kim, B., Lim, K., Cho, S. and Park, M. **RomaDroid: A Robust and Efficient Technique for Detecting Android App Clones Using a Tree Structure and Components of Each App’s Manifest File**. IEEE Access, vol. 7, pp. 72182-72196, 2019.

Machine learning plus, Cosine Similarity – Understanding the math and how it works (with python codes), 2018. Disponível em: <https://www.machinelearningplus.com/nlp/cosine-similarity/>. Acesso em: Julho de 2021

Mao, J., Bian, J., Ma, H., Jia, Y., Liang, Z. and Jiang, X. **Robust Detection of Android UI Similarity**. 2018 IEEE International Conference on Communications (ICC), 2018, pp. 1-6, doi: 10.1109/ICC.2018.8422189.

MIT App Inventor, AIA file structure, 2019. Disponível em: <https://community.appinventor.mit.edu/t/aia-file-structure/219>. Acesso em: Julho de 2020

MIT App Inventor, About Us, 2012. Disponível em: <http://appinventor.mit.edu/about-us>. Acesso em: Maio de 2020

michaelis,“novidade”. Disponível em:

<https://michaelis.uol.com.br/moderno-portugues/busca/portugues-brasileiro/NOVIDADE/>

Mustafaraj, E., Turbak, F., Svanberg, M. **Identifying Original Projects in App Inventor.** In Proc. of the 30th Int. Florida Artificial Intelligence Research Society Conference, Marco Island, FL, USA, 2017, p. 567-572.

P21, Partnership for 21st century learning, P21 Framework Definitions, 2015.

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

Piasecki, J., Waligora, M. & Dranseika, V. **Google Search as an Additional Source in Systematic Reviews**. Sci Eng Ethics 24,2018, p. 809–810.

Pressman, R. **Engenharia de software: Uma abordagem profissional.** 8º edição. Porto Alegre: Bookman, 2016. p. 968.

Runco, M. A., Jaeger, G. J. **The standard definition of creativity.** Creativity Research Journal. vol. 24, n.1, 2012, p. 92-96.

Sociedade Brasileira de Computação, Diretrizes para ensino de Computação na Educação Básica. Disponível em:

<https://www.sbc.org.br/educacao/diretrizes-para-ensino-de-computacao-na-educacao-basica>. Acesso em: Julho de 2020

Solecki, I., Porto, J. A.; Alves, N. d. 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, Portland, OR, USA, 2020, p. 51–57.

Svanberg, M. "**Using feature vector representations to identify similar projects in app inventor**" 2017 IEEE Blocks and Beyond Workshop (B&B), Raleigh, NC, USA, 2017, p. 117-118.

Tissenbaum, M., Sheldon, J., & Abelson, H. **From Computational Thinking to Computational Action.** Communications of the ACM, vol. 62, n. 3, 2019, p. 34-36.

Towards data science, Text data representation with one-hot encoding, Tf-Idf, Count Vectors, Co-occurrence Vectors and Word2Vec. Disponível em:

<https://towardsdatascience.com/text-data-representation-with-one-hot-encoding-tf-idf-count-vectors-co-occurrence-vectors-and-f1bccbd98bef>. Acesso em: Julho de 2021

Turbak, F., Mustafaraj, F., Svanberg, M., Dawson, M. (2017). **Identifying and Analyzing Original Projects in an Open-Ended Blocks Programming Environment.** Proc. of the 23rd Int. Conference on Visual Languages and Sentient Systems, Pittsburgh, PA, USA, 2017.

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

Wang, Z., Li, G., & Chi, Y. **Multi-classification of Android Applications Based on Convolutional Neural Networks**. In Proceedings of the 4th International Conference on Computer Science and Application Engineering (2020, October, pp. 1-5).

Yuan, C., Wei, S., Wang, Y., You, T. and ZiLiang,S., G. **Android Applications Categorization Using Bayesian Classification**. 2016 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2016, p. 173-176.

Xia, P., Zhang, L., & Li, F. (2015). **Learning similarity with cosine similarity ensemble. Information Sciences**, 307, 2015, p.39-52.

Zen, M.; Vanderdonckt, J.; **Assessing User Interface Aesthetics Based on the Inter-Subjectivity of Judgement.** In: Proc. of the 30th International BCS Human Computer Interaction Conference: Fusion!, Poole, Reino Unido, 2016, artigo n. 25.

Zhu, J., Wu, Z., Guan, Z. and Chen, Z. **Appearance similarity evaluation for Android applications**. 2015 Seventh International Conference on Advanced Computational Intelligence (ICACI), 2015, p. 323-328.

Zhu, D., Ma, Y., Xi, T., and Zhang, Y.; **FSNet: Android Malware Detection with Only One Feature.** 2019 IEEE Symposium on Computers and Communications (ISCC), Barcelona, Espanha, 2019, p. 1-6.

APÊNDICE A

Cara a criação do modelo de avaliação da originalidade de aplicativos foi criada uma nova Classe Originalidade.java.

```
package br.ufsc.cne.codemaster.restgrader.appinventorgrader.conceitos;

import java.util.Arrays;
import java.util.HashSet;
import java.util.List;
import java.util.Set;

import org.json.simple.JSONArray;
import org.json.simple.JSONObject;

import br.ufsc.cne.codemaster.restgrader.appinventorgrader.codigo.Codigo;

public class Originalidade extends ConceitoCT {

int unviversoDeReferencia = 10000;
int textoInf = 0;
double somaFuncionalidades = 0;
double somaFrequencia = 0;
String combinacoes;
final Set<String> componentesLegenda = new HashSet<>(
                    Arrays.asList("Label", "TextBox")
                    );

public Originalidade(Codigo codigo) {
        super(codigo);
}

@Override
public void avaliaCodigo() {
```

```
        int qtdAnimacao = 0;
        List<String> marcadores_mapas = Arrays.asList(
                        "Circle", "FeatureCollection", "LineString", "Marker", "Polygon",
                        "Rectangle");

        List<String> marcadores_animacao = Arrays.asList(
                        "Canvas", "Ball", "ImageSprite");



        for(String marcadorAni : marcadores_animacao){
                if (codigo.getMapaTokensMutationBlocos().containsKey(marcadorAni)){
                        if
(codigo.getMapaTokensMutationBlocos().containsKey("ImageSprite")) {
                                qtdAnimacao++;
                                somaFuncionalidades++;
                                //System.out.println(qtdAnimacao);
                        } else
                                if
(codigo.getMapaTokensMutationBlocos().containsKey("Ball")) {
                                        qtdAnimacao++;
                                        somaFuncionalidades++;
                                        //System.out.println(qtdAnimacao);
                                } else
                                        if
(codigo.getMapaTokensMutationBlocos().containsKey("Canvas")) {
                                                qtdAnimacao++;
                                                somaFuncionalidades++;
                                                //System.out.println(qtdAnimacao);
                                        }
                }
        }

        //System.out.println(qtdAnimacao);
```

```
        if (qtdAnimacao > 0 && qtdAnimacao < 2){
                somaFrequencia = (somaFrequencia +  1403)/unviversoDeReferencia;
                lista_funcionalidades_app.add("desenharMenorQueDois");
                lista_raridades_app.add("<html><font color=\"#ee7f1d\">Muito
comum</font></html>");
                System.out.print("Funcionalidade: Desenhar na tela < 2 ");
        } else if(qtdAnimacao >= 2){
                somaFrequencia = (somaFrequencia +  749)/unviversoDeReferencia;
                lista_funcionalidades_app.add("desenharMaiorQueDois");
                lista_raridades_app.add("<html><font
color=\"#92268e\">Comum</font></html>");
                System.out.print("Funcionalidade: Desenhar na tela >= 2 ");
        }

        // Desenhar na tela

        // Visualizar Mapa
        if (codigo.getMapaTokensMutationBlocos().containsKey("Map") ||
codigo.getMapaTokensMutationBlocos().containsKey("http://www.") &&
codigo.getMapaTokensMutationBlocos().containsKey("maps")){
                somaFrequencia = (somaFrequencia +  149)/unviversoDeReferencia;
                somaFuncionalidades++;
                lista_funcionalidades_app.add("visualizarMapa");
                lista_raridades_app.add("<html><font
color=\"#007aca\">Rara</font></html>");
                System.out.print("Funcionalidade: Visualizar Mapa ");
        }
        // Visualizar Mapa

        // Marcar posição no mapa
        for(String marcadorMap : marcadores_mapas){
                if (codigo.getMapaTokensMutationBlocos().containsKey(marcadorMap)){
                        somaFrequencia = (somaFrequencia +  126)/unviversoDeReferencia;
                        somaFuncionalidades++;
                        lista_funcionalidades_app.add("marcarPosicaoNoMapa");
```

```
                    lista_raridades_app.add("<html><font
color=\"#007aca\">Rara</font></html>");
                    System.out.print("Funcionalidade: Marcar posicao no mapa ");
                    break;
                }
        }
        // Marcar posição no mapa

        // Salvar dados localmente
        if (codigo.getMapaTokensMutationBlocos().containsKey("File") ||
codigo.getMapaTokensMutationBlocos().containsKey("TinyDB")) {
                somaFrequencia = (somaFrequencia +  1102)/unviversoDeReferencia;
                somaFuncionalidades++;
                lista_funcionalidades_app.add("salvarDadosLocalmente");
                lista_raridades_app.add("<html><font color=\"#ee7f1d\">Muito
comum</font></html>");
                System.out.print("Funcionalidade: Salvar dados localmente ");
        }
        // Salvar dados localmente

        // Salvar dados na nuvem
        if (codigo.getMapaTokensMutationBlocos().containsKey("FirebaseDB") ||
codigo.getMapaTokensMutationBlocos().containsKey("TinyWebDB") ||
codigo.getMapaTokensMutationBlocos().containsKey("CloudDB")) {
                somaFrequencia = (somaFrequencia +  368)/unviversoDeReferencia;
                somaFuncionalidades++;
                lista_funcionalidades_app.add("salvarDadosNaNuvem");
                lista_raridades_app.add("<html><font
color=\"#007aca\">Rara</font></html>");
                System.out.print("Funcionalidade: Salvar dados na nuvem ");
        }
        // Salvar dados na nuvem

        // Persistencia de dados

        // Sensores
```

```
        if (codigo.getMapaTokensMutationBlocos().containsKey("AccelerometerSensor")){
                somaFrequencia = (somaFrequencia +  968)/unviversoDeReferencia;
                somaFuncionalidades++;
                lista_funcionalidades_app.add("detectarAceleracao");
                lista_raridades_app.add("<html><font
color=\"#92268e\">Comum</font></html>");
                System.out.print("Funcionalidade: Detectar aceleração ");
        }

        if (codigo.getMapaTokensMutationBlocos().containsKey("BarcodeScanner")){
                somaFrequencia = (somaFrequencia +  155)/unviversoDeReferencia;
                somaFuncionalidades++;
                lista_funcionalidades_app.add("escanearQrCode");
                lista_raridades_app.add("<html><font
color=\"#007aca\">Rara</font></html>");
                System.out.print("Funcionalidade: Escanear QR code ");
        }

        if (codigo.getMapaTokensMutationBlocos().containsKey("Clock")){
                somaFrequencia = (somaFrequencia +  1860)/unviversoDeReferencia;
                somaFuncionalidades++;
                lista_funcionalidades_app.add("marcarTempo");
                lista_raridades_app.add("<html><font color=\"#ee7f1d\">Muito
comum</font></html>");
                System.out.print("Funcionalidade: Marcar tempo ");
        }

        if (codigo.getMapaTokensMutationBlocos().containsKey("GyroscopeSensor")){
                somaFrequencia = (somaFrequencia +  8)/unviversoDeReferencia;
                somaFuncionalidades++;
                lista_funcionalidades_app.add("medirGiroscopio");
                lista_raridades_app.add("<html><font color=\"#009e2c\">Muito
rara</font></html>");
                System.out.print("Funcionalidade: Medir giroscópio ");
        }
```

```
        if (codigo.getMapaTokensMutationBlocos().containsKey("LocationSensor")){
                somaFrequencia = (somaFrequencia +  391)/unviversoDeReferencia;
                somaFuncionalidades++;
                lista_funcionalidades_app.add("mostrarGeolocalizacao");
                lista_raridades_app.add("<html><font
color=\"#007aca\">Rara</font></html>");
                System.out.print("Funcionalidade: mostrar geolocalização ");
        }

        if (codigo.getMapaTokensMutationBlocos().containsKey("OrientationSensor")){
                somaFrequencia = (somaFrequencia +  72)/unviversoDeReferencia;
                somaFuncionalidades++;
                lista_funcionalidades_app.add("mostrarOriEspacial");
                lista_raridades_app.add("<html><font color=\"#009e2c\">Muito
rara</font></html>");
                System.out.print("Funcionalidade: Mostrar orientação espacial do
dispositivo");
        }

        if (codigo.getMapaTokensMutationBlocos().containsKey("Pedometer")){
                somaFrequencia = (somaFrequencia +  192)/unviversoDeReferencia;
                somaFuncionalidades++;
                lista_funcionalidades_app.add("contarPassos");
                lista_raridades_app.add("<html><font
color=\"#007aca\">Rara</font></html>");
                System.out.print("Funcionalidade: Contar passos ");
        }

        if (codigo.getMapaTokensMutationBlocos().containsKey("ProximitySensor")){
                somaFrequencia = (somaFrequencia +  40)/unviversoDeReferencia;
                somaFuncionalidades++;
                lista_funcionalidades_app.add("medirProximidade");
                lista_raridades_app.add("<html><font color=\"#009e2c\">Muito
rara</font></html>");
```

```
                System.out.print("Funcionalidade: Medir proximidade ");
        }

        if (codigo.getMapaTokensMutationBlocos().containsKey("Barometer")){
                somaFrequencia = (somaFrequencia +  8)/unviversoDeReferencia;
                somaFuncionalidades++;
                lista_funcionalidades_app.add("medirPressaoDoAr");
                lista_raridades_app.add("<html><font color=\"#009e2c\">Muito
rara</font></html>");
                System.out.print("Funcionalidade: Medir pressão do ar ");
        }

        if (codigo.getMapaTokensMutationBlocos().containsKey("Hygrometer")){
                somaFrequencia = (somaFrequencia +  2)/unviversoDeReferencia;
                somaFuncionalidades++;
                lista_funcionalidades_app.add("medirUmidadeDoAr");
                lista_raridades_app.add("<html><font color=\"#009e2c\">Muito
rara</font></html>");
                System.out.print("Funcionalidade: Medir umidade relativa do ar ");
        }

        if (codigo.getMapaTokensMutationBlocos().containsKey("LightSensor")){
                somaFrequencia = (somaFrequencia +  79)/unviversoDeReferencia;
                somaFuncionalidades++;
                lista_funcionalidades_app.add("medirNivelDeLuz");
                lista_raridades_app.add("<html><font color=\"#009e2c\">Muito
rara</font></html>");
                System.out.print("Funcionalidade: Medir nível de luz ");
        }

        if (codigo.getMapaTokensMutationBlocos().containsKey("MagneticFieldSensor")){
                somaFrequencia = somaFrequencia + 0; //+  0/unviverso_de_referencia;
                somaFuncionalidades++;
                lista_funcionalidades_app.add("medirCampoMagnetico");
```

```
                lista_raridades_app.add("<html><font color=\"#009e2c\">Muito
rara</font></html>");
                System.out.print("Funcionalidade: Medir campo magnético ");
        }
        if (codigo.getMapaTokensMutationBlocos().containsKey("Thermometer")){
                somaFrequencia = (somaFrequencia +  16)/unviversoDeReferencia;
                somaFuncionalidades++;
                lista_funcionalidades_app.add("medirTemperatura");
                lista_raridades_app.add("<html><font color=\"#009e2c\">Muito
rara</font></html>");
                System.out.print("Funcionalidade: Medir temperatura ");
        }

        // Sensores



        // Acessar site
        if (codigo.getMapaTokensMutationBlocos().containsKey("http://www.") ||
codigo.getMapaTokensMutationBlocos().containsKey("WebViewer") ||
codigo.getMapaTokensMutationBlocos().containsKey("Web")){
                somaFrequencia = (somaFrequencia +  778)/unviversoDeReferencia;
                somaFuncionalidades++;
                lista_funcionalidades_app.add("acessarSite");
                lista_raridades_app.add("<html><font
color=\"#92268e\">Comum</font></html>");
                System.out.print("Funcionalidade: Acessar site ");
        }
        // Acessar site



        // Fazer ligação
        if (codigo.getMapaTokensMutationBlocos().containsKey("PhoneCall")){
                somaFrequencia = (somaFrequencia +  211)/unviversoDeReferencia;
                somaFuncionalidades++;
                lista_funcionalidades_app.add("fazerLigacao");
```

```
                lista_raridades_app.add("<html><font
color=\"#007aca\">Rara</font></html>");
                System.out.print("Funcionalidade: Fazer ligação ");
        }
        // fazer ligação



        // Compartilhar artefato (mensagem/foto/arquivo)
        if          (codigo.getMapaTokensMutationBlocos().containsKey("Sharing")          ||
codigo.getMapaTokensMutationBlocos().containsKey("Texting")                              ||
codigo.getMapaTokensMutationBlocos().containsKey("Twitter")){
                somaFrequencia = (somaFrequencia +  465)/unviversoDeReferencia;
                somaFuncionalidades++;
                lista_funcionalidades_app.add("compartilharArtefato");
                System.out.print("Funcionalidade: Compartilhar artefato");
        }
        // Compartilhar artefato (mensagem/foto/arquivo)


        // Visualizar lista de contatos
        if      (codigo.getMapaTokensMutationBlocos().containsKey("ContactPicker")      ||
codigo.getMapaTokensMutationBlocos().containsKey("EmailPicker")                         ||
codigo.getMapaTokensMutationBlocos().containsKey("PhoneNumberPicker")){
                somaFrequencia = (somaFrequencia +  125)/unviversoDeReferencia;
                somaFuncionalidades++;
                lista_funcionalidades_app.add("visualizarListaContatos");
                lista_raridades_app.add("<html><font
color=\"#007aca\">Rara</font></html>");
                System.out.print("Funcionalidade: Visualizar lista de contatos ");
        }
        // Visualizar lista de contatos


        // Escolher imagem
        if (codigo.getMapaTokensMutationBlocos().containsKey("ImagePicker")){
                somaFrequencia = (somaFrequencia +  186)/unviversoDeReferencia;
                somaFuncionalidades++;
                lista_funcionalidades_app.add("escolherImagem");
```

```
                lista_raridades_app.add("<html><font
color=\"#007aca\">Rara</font></html>");
                System.out.print("Funcionalidade: Escolher imagem da galeria");
        }
        // EscolherImagem


        // Tirar foto
        if (codigo.getMapaTokensMutationBlocos().containsKey("Camera")){
                somaFrequencia = (somaFrequencia +  557)/unviversoDeReferencia;
                somaFuncionalidades++;
                lista_funcionalidades_app.add("tirarFoto");
                lista_raridades_app.add("<html><font
color=\"#92268e\">Comum</font></html>");
                System.out.print("Funcionalidade: Tirar foto ");
        }
        // TirarFoto


        //Reproduzir som
        if          (codigo.getMapaTokensMutationBlocos().containsKey("Sound")          ||
codigo.getMapaTokensMutationBlocos().containsKey("Player")){
                somaFrequencia = (somaFrequencia +  3645)/unviversoDeReferencia;
                somaFuncionalidades++;
                lista_funcionalidades_app.add("reproduzirSom");
                lista_raridades_app.add("<html><font                color=\"#ee7f1d\">Muito
comum</font></html>");
                System.out.print("Funcionalidade: Reproduzir som ");
        }
        //Reproduzir som


        //Fazer tradução
        if (codigo.getMapaTokensMutationBlocos().containsKey("YandexTranslate")){
                somaFrequencia = (somaFrequencia +  272)/unviversoDeReferencia;
                somaFuncionalidades++;
                lista_funcionalidades_app.add("fazerTraducao");
```

```
                lista_raridades_app.add("<html><font
color=\"#007aca\">Rara</font></html>");
                System.out.print("Funcionalidade: Fazer tradução ");
        }
        //Fazer tradução


        //Reconhecer voz
        if (codigo.getMapaTokensMutationBlocos().containsKey("SpeechRecognizer")){
                somaFrequencia = (somaFrequencia +  429)/unviversoDeReferencia;
                somaFuncionalidades++;
                lista_funcionalidades_app.add("reconhecerVoz");
                lista_raridades_app.add("<html><font
color=\"#007aca\">Rara</font></html>");
                System.out.print("Funcionalidade: Reconhecer voz ");
        }
        //Reconhecer voz

        //Converter texto em fala
        if (codigo.getMapaTokensMutationBlocos().containsKey("TextToSpeech")){
                somaFrequencia = (somaFrequencia +  1483)/unviversoDeReferencia;
                somaFuncionalidades++;
                lista_funcionalidades_app.add("converterTextoEmFala");
                lista_raridades_app.add("<html><font                color=\"#ee7f1d\">Muito
comum</font></html>");
                System.out.print("Funcionalidade: Converter texto em fala ");
        }
        //Converter texto em fala

        //Converter fala em texto
        if (codigo.getMapaTokensMutationBlocos().containsKey("SpeechRecognizer")){
                somaFrequencia = (somaFrequencia +  429)/unviversoDeReferencia;
                somaFuncionalidades++;
                lista_funcionalidades_app.add("converterFalaEmTexto");
```

```
            lista_raridades_app.add("<html><font
color=\"#007aca\">Rara</font></html>");
            System.out.print("Funcionalidade: Converter fala em texto ");
        }
        //Converter fala em texto


        //Gravar áudio
        if (codigo.getMapaTokensMutationBlocos().containsKey("SoundRecorder")){
            somaFrequencia = (somaFrequencia +  97)/unviversoDeReferencia;
            somaFuncionalidades++;
            lista_funcionalidades_app.add("gravarAudio");
            lista_raridades_app.add("<html><font                color=\"#009e2c\">Muito
rara</font></html>");
            System.out.print("Funcionalidade: Gravar áudio ");
        }
        //Gravar áudio


        //Fazer login
        if          (codigo.getMapaTokensMutationBlocos().containsKey("File")          ||
codigo.getMapaTokensMutationBlocos().containsKey("TinyDB")                            ||
codigo.getMapaTokensMutationBlocos().containsKey("FirebaseDB")                        ||
codigo.getMapaTokensMutationBlocos().containsKey("TinyWebDB")                         ||
codigo.getMapaTokensMutationBlocos().containsKey("CloudDB")){
            somaFuncionalidades++;
            if
(codigo.getMapaTokensMutationBlocos().containsKey("PasswordTextBox")) {
                somaFrequencia = (somaFrequencia +  273)/unviversoDeReferencia;
                somaFuncionalidades++;
                lista_funcionalidades_app.add("fazerLogin");
                lista_raridades_app.add("<html><font
color=\"#007aca\">Rara</font></html>");
                System.out.print("Funcionalidade: Fazer login ");
            }
        }
        //Fazer login
```

```
        //Usar api de outro app
        if (codigo.getMapaTokensMutationBlocos().containsKey("ActivityStarter")){
                somaFrequencia = (somaFrequencia +  559)/unviversoDeReferencia;
                somaFuncionalidades++;
                lista_funcionalidades_app.add("usarApiDeOutroApp");
                lista_raridades_app.add("<html><font
color=\"#92268e\">Comum</font></html>");
                System.out.print("Funcionalidade: Usar api de outro app ");
        }
        //Usar api de outro app

        //Usar bluetooth
        if     (codigo.getMapaTokensMutationBlocos().containsKey("BluetoothClient")     ||
codigo.getMapaTokensMutationBlocos().containsKey("BluetoothServer")){
                somaFrequencia = (somaFrequencia +  507)/unviversoDeReferencia;
                somaFuncionalidades++;
                lista_funcionalidades_app.add("usarBluetooth");
                lista_raridades_app.add("<html><font
color=\"#92268e\">Comum</font></html>");
                System.out.print("Funcionalidade: Usar bluetooth ");
        }
        //Usar bluetooth

        //Gravar vídeo
        if (codigo.getMapaTokensMutationBlocos().containsKey("Camcorder")){
                somaFrequencia = (somaFrequencia +  94)/unviversoDeReferencia;
                somaFuncionalidades++;
                lista_funcionalidades_app.add("gravarVideo");
                lista_raridades_app.add("<html><font                color=\"#009e2c\">Muito
rara</font></html>");
                System.out.print("Funcionalidade: Gravar vídeo ");
        }
        //Gravar vídeo

        //Reproduzir vídeo
```

```
        if (codigo.getMapaTokensMutationBlocos().containsKey("VideoPlayer")){
                somaFrequencia = (somaFrequencia +  165)/unviversoDeReferencia;
                somaFuncionalidades++;
                lista_funcionalidades_app.add("reproduzirVideo");
                lista_raridades_app.add("<html><font
color=\"#007aca\">Rara</font></html>");
                System.out.print("Funcionalidade: Reproduzir vídeo ");
        }
        //Reproduzir vídeo

        // Visualizar informações

        List<JSONObject> telas = codigo.getcodigosTelasVisual();
        for (JSONObject tela : telas){
                analisaComponentes((JSONArray)                    ((JSONObject)
tela.get("Properties")).get("$Components"));

        }
        if(textoInf != 0) {
                somaFrequencia = (somaFrequencia + 2386)/unviversoDeReferencia;
                somaFuncionalidades++;
                lista_funcionalidades_app.add("visualizarInfo");
                lista_raridades_app.add("<html><font          color=\"#ee7f1d\">Muito
comum</font></html>");
                System.out.print("Funcionalidade: Visualizar informações ");

        }

        // Visualizar informações

        gerarNota();
}

// Visualizar informações
```

```
private void analisaComponentes(JSONArray arrayComponentes) {

	// Para cada elemento de "componentesLegenda"
	if(arrayComponentes != null){
		for (Object o : arrayComponentes)
		{
			JSONObject comp = (JSONObject) o;
			// Verifica se e botão ou legenda e contem texto
			if (componentesLegenda.contains(comp.get("$Type")))
			{
				JSONObject legenda = comp;
				//System.out.println(comp);
				if ((legenda.containsKey("Text"))) {
					String tamLegenda = (String) legenda.get("Text");
					//int tam = tamLegenda.length();
					//System.out.println(tamLegenda + " " + tam);
					if (tamLegenda.length() > 30) {
						textoInf++;
						break;
					}
				}
			}
		}
	}
}

// Visualizar informações

private void gerarNota () {
	double ocorencia_individual = 0;
	ocorencia_individual = somaFrequencia/somaFuncionalidades;

	System.out.println("NOTA FREQUENCIAS: " + ocorencia_individual * 100);
	int nivelCombinacao = (int) (ocorencia_individual * 100);
```

```
        if(nivelCombinacao == 0 || nivelCombinacao < 0) {
                resultadoDasCombinacoes = "<html><font color=\"#009e2c\">Muito
rara</font></html>";
        }else if(nivelCombinacao > 0 && nivelCombinacao <= 10) {
                resultadoDasCombinacoes = "<html><font
color=\"#007aca\">Rara</font></html>";
        }else if(nivelCombinacao > 10 && nivelCombinacao <= 20) {
                resultadoDasCombinacoes = "<html><font
color=\"#92268e\">Comun</font></html>";
        }else if(nivelCombinacao > 20) {
                resultadoDasCombinacoes = "<html><font color=\"#ee7f1d\">Muito
comun</font></html>";
        }
        double combinacao_similaridade = (somaFuncionalidades + ocorencia_individual) /
2;
        this.notaFinalOriginalidade = -1.020828 + (6236645 - -1.020828)
                        /(1 + Math.pow((combinacao_similaridade/8.816668000000002e-
20),0.3346091));
}

}
```

APÊNDICE B

Foi criada também uma nova aba para a avaliação de originalidade de um aplicativo na interface do CodeMaster.

```
<%@page import="com.google.gson.JsonArray"%>
<%@page import="com.google.gson.JsonParser"%>
<%@page import="br.ufsc.cne.codemaster.commons.AppInventorGradeDesign"%>
<%@page import="java.util.ResourceBundle"%>
<%@page import="java.util.Locale"%>
<%@page import="br.ufsc.cne.codemaster.commons.AppInventorGradePrograma"%>
<%@page import="br.ufsc.cne.codemaster.commons.CriterioDesignAI"%>
<%@page
import="br.ufsc.cne.codemaster.util.ResourceBundleCriteriosAppInventor"%>
<%@page import="com.google.gson.Gson"%>
<%@page import="java.text.NumberFormat"%>
<%@page import="java.util.*"%>
<%@page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@page errorPage="erro.jsp"%>

<!DOCTYPE html>
<html lang="pt-br" xml:lang=" pt-br">

<head>
<style>
.small-title {
font-size: 30px;
font-weight: 400;
display: block;
}
```

```
td, th {
border: 1px solid #cccccc;
text-align: left;
padding: 4px;
width: 50%
}

.col-centered {
float: none;
margin: 0 auto;
}

table {
width: 100%;
border: 1px solid #cccccc;
}

.collapsible {
background-color: #FFF;
color: #000;
cursor: pointer;
padding: 18px;
width: 100%;
border: 1px solid #cccccc;
text-align: left;
outline: none;
font-size: 16px;
}
```

```
.active, .collapsible:hover {
background-color: #2cb34c;
}

.content {
padding: 0 18px;
max-height: 0;
overflow: hidden;
transition: max-height 0.4s ease-out;
background-color: #f5fdf7;
}
</style>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
<meta http-equiv="Content-Language" content="pt-br">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css"
type="text/css">
<link rel="stylesheet" href="css/styles.css">
<link rel="stylesheet"
href="//code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">
<link rel="stylesheet" href="/resources/demos/style.css">
</head>

<%
String language;
String country;
if (request.getSession().getAttribute("language") == null ||
request.getSession().getAttribute("country") == null) {
```

```
Locale automaticLocale = request.getLocale();
language = automaticLocale.getLanguage();
country = automaticLocale.getCountry();
request.getSession().setAttribute("language", language);
request.getSession().setAttribute("country", country);
} else {
language = (String) request.getSession().getAttribute("language");
country = (String) request.getSession().getAttribute("country");
}
Locale locale = new Locale(language, country);
ResourceBundle messages = ResourceBundle.getBundle("AlunoResultadoAppInventorBundle", locale);
ResourceBundleCriteriosAppInventor bundleCriterios = new ResourceBundleCriteriosAppInventor(locale);
%>

<%
Gson gson = new Gson();
String msg = request.getAttribute("msg").toString();
JsonParser jsonParser = new JsonParser();
JsonArray arrayGrades = (JsonArray) jsonParser.parse(msg);
AppInventorGradePrograma gPrograma = gson.fromJson(arrayGrades.get(0), AppInventorGradePrograma.class);
AppInventorGradeDesign gDesign = gson.fromJson(arrayGrades.get(1), AppInventorGradeDesign.class);
ArrayList<Integer> aestheticScores = (ArrayList<Integer>) request.getAttribute("score");
ArrayList<String> uiImages = (ArrayList<String>) request.getAttribute("images");
int imageScorePair = 0;
NumberFormat nf = NumberFormat.getInstance();
nf.setMaximumFractionDigits(1);
nf.setMinimumFractionDigits(1);
```

```
String level = gPrograma.getNivel();
String displayLevel = "";
List<String> dadosDasFuncionalidades = new ArrayList<>();
switch (level) {
case "faixa branca" :
        displayLevel = messages.getString("FaixaBranca");
        break;
case "faixa amarela" :
        displayLevel = messages.getString("FaixaAmarela");
        break;
case "faixa vermelha" :
        displayLevel = messages.getString("FaixaVermelha");
        break;
case "faixa laranja" :
        displayLevel = messages.getString("FaixaLaranja");
        break;
case "faixa verde" :
        displayLevel = messages.getString("FaixaVerde");
        break;
case "faixa roxa" :
        displayLevel = messages.getString("FaixaRoxa");
        break;
case "faixa marrom" :
        displayLevel = messages.getString("FaixaMarrom");
        break;
case "faixa preta" :
        displayLevel = messages.getString("FaixaPreta");
        break;
case "faixa azul" :
```

```
            displayLevel = messages.getString("FaixaAzul");
            break;
    case "faixa turquesa" :
            displayLevel = messages.getString("FaixaTurquesa");
            break;
    }
    %>
    <%!public String getProgressBar(int value) {
            switch (value) {
                    case 0 :
                            return "bg-danger";
                    case 1 :
                            return "bg-danger";
                    case 2 :
                            return "bg-warning";
                    case 3 :
                            return "bg-success";
            }
            return "";
    }

    public String getProgressBar3Point(int value) {
            switch (value) {
                    case 0 :
                            return "bg-danger";
                    case 1 :
                            return "bg-warning";
                    case 2 :
                            return "bg-success";
```

```
        }
        return "";
}

public String getProgressBarDichotomous(int value) {
        switch (value) {
                case 0 :
                        return "bg-danger";
                case 1 :
                        return "bg-success";
        }
        return "";
}

public String corParaBarraDeDesign(int valor) {
        return getProgressBar3Point(valor);
}

public String corParaBarraDeNota(double nota) {
        if (nota < 3.0)
                return "bg-danger";
        if (nota < 6.0)
                return "bg-warning";
        return "bg-success";
}%>

<%!public class DadosParaBarra {
        int nota;
        String nomeDoConceito;
```

```
        String tipoDaBarra;
        double larguraDaBarra;
        String textoDaNota;
        int notaMáxima;

        public DadosParaBarra(String nomeDoConceito, int nota, int notaMáxima) {
                this.nota = nota;
                this.nomeDoConceito = nomeDoConceito;
                tipoDaBarra = corParaBarraDeDesign(nota);
                larguraDaBarra = nota >= 0 ? (100.0 / notaMáxima) * nota : 0.0;
                textoDaNota = (nota >= 0) ? nota + "/" + notaMáxima : "NA";
                this.notaMáxima = notaMáxima;
        }
}
public class ListaDeDadosParaBarras {
        List<DadosParaBarra> dadosDosConceitos = new LinkedList<>();

        public void adicionar(String nomeDoCriterio, int nota, int notaMáxima) {
                dadosDosConceitos.add(new DadosParaBarra(nomeDoCriterio, nota,
notaMáxima));
        }

        public double mediaDasNotas() {
                double somaNotas = 0;
                int somaMaximoNotas = 0;
                for (DadosParaBarra dado : dadosDosConceitos) {
                        if (dado.nota >= 0) {
                                somaNotas += dado.nota;
                                somaMaximoNotas += dado.notaMáxima;
                        }
```

```
            }
            if (somaMaximoNotas == 0)
                return 0.0;
            return Math.round((10 * somaNotas / somaMaximoNotas) * 100.0) / 100.0;
        }
}%>

<body>
<div class="py-3">
        <div class="container">
            <div class="row">
                <div class="col-md-9">
                    <img class="center-block img-fluid d-block"
                        src="resources/LogoCodeMaster.png" width="30%">
                </div>
                <div class="col-md-3 text-right align-self-end">
                    <a href="#" class="btn text-gray-dark px-0 disabled"></a>
                </div>
            </div>
        </div>
</div>
<div class="bg-success">
        <div class="container">
            <div class="row">
                <div class="col-centered">
                    <div class="btn-group">
                        <a href="index.jsp" class="btn btn-success text-gray-
dark"><%=messages.getString("HomeBtn")%></a>
                        <a href="aluno.jsp" class="btn btn-
success"><b><%=messages.getString("AlunoBtn")%></b></a>
```

```
                    <div class="dropdown">
                        <button class="btn btn-success dropdown-toggle text-gray-dark"
                                type="button" data-toggle="dropdown"><%=messages.getString("ProfessorBtn")%><span
                                class="caret"></span>
                        </button>
                        <ul class="dropdown-menu text-left px-2">
                            <li><a href="professor.jsp" class="text-gray-dark"><%=messages.getString("ProfessorDropAvaliar")%></a></li>
                            <li class="divider"></li>
                            <li><a href="professor_turmas.jsp" class="text-gray-dark"><%=messages.getString("ProfessorDropTurmas")%></a></li>
                        </ul>
                    </div>
                    <div class="dropdown">
                        <button class="btn btn-success dropdown-toggle" type="button"
                                data-toggle="dropdown">

    <b><%=messages.getString("AdminBtn")%></b><span class="caret"></span>
                        </button>
                        <ul class="dropdown-menu text-left px-2">
                            <li><a href="admin.jsp" class="text-gray-dark"><%=messages.getString("AdministradorDropCadastro")%></a></li>
                            <li class="divider"></li>
                            <li><a href="admin_estatic.jsp" class="text-gray-dark"><%=messages.getString("AdministradorDropEstatistica")%></a></li>
                        </ul>
                    </div>
                </div>
```

```
                    </div>
                </div>
            </div>
        </div>


<div class="py-4">
            <div class="container">
                <div class="row">
                    <div class="col-md-12 text-center">
                        <h1 class="small-title">

            <b><%=messages.getString("TituloAvaliacao")%></b>
                        </h1>
                    </div>
                </div>
                <div class="row">
                    <div class="col-md-4">
                        <div class="row">
                            <div class="card">
                                <div class="card-block">
                                    <h4 id="notaGeral" class="card-title
text-center"><%=messages.getString("Nota")%>

            <%=nf.format(gPrograma.getNotaFinal())%><br>
                                    </h4>
                                    <h4 id="notaOriginalidade"
class="card-title text-center"><%=messages.getString("Nota")%>

            <%=nf.format(gPrograma.getNotaFinalOriginalidade())%><br>
                                    </h4>
```

<%=messages.getString("Nivel")%><%=displayLevel%>!

<img class="center-block img-fluid d-block py-3 mx-auto"

src="resources/ninjas/<%=gPrograma.getNivel()%>.png"

width="75%"> <img

class="center-block img-fluid d-block py-3 mx-auto"

src="resources/rank.png"> <a class="nav-link text-center"

href="rubrica_appinventor.jsp"><%=messages.getString("MelhorarPontuacao")%>

<%=messages.getString("AbaProgramacao")%>

<%=messages.getString("AbaInterfaceUsuario")%>

<%=messages.getString("AbaOriginalidade")%>

<table class="table">

<thead class="">

<tr class="table-active">

```
                                                    <th class="text-
center"><%=messages.getString("ColunaConceito")%></th>
                                                    <th class="text-
center"><%=messages.getString("ColunaPontuacao")%></th>
                                                </tr>
                                            </thead>
                                            <tbody>
                                                <tr>

        <td><%=messages.getString("ConceitoTelas")%></td>
                                                    <td>
                                                        <div class="progress">
                                                            <div

        class="progress-bar progress-bar-striped
<%=getProgressBar(gPrograma.getTelas())%>"

        role="progressbar" aria-valuenow="<%=gPrograma.getTelas()%>"
                                                                aria-
valuemin="0" aria-valuemax="3"

        style="width: <%=(100.0 / 3) * gPrograma.getTelas()%>%">

        <span><strong class="text-gray-
dark"><%=gPrograma.getTelas()%>/3</strong></span>
                                                            </div>
                                                        </div>
                                                    </td>
                                                </tr>
                                                <tr>

        <td><%=messages.getString("ConceitoNomeacao")%></td>
                                                    <td>
```

```
                                                    <div class="progress">
                                                        <div
        class="progress-bar progress-bar-striped
<%=getProgressBar(gPrograma.getNomeacaoProcedimentosVariaveis())%>"
        role="progressbar"
                                                                aria-
valuenow="<%=gPrograma.getNomeacaoProcedimentosVariaveis()%>"
                                                                aria-
valuemin="0" aria-valuemax="3"
        style="width: <%=(100.0 / 3) *
gPrograma.getNomeacaoProcedimentosVariaveis()%>%">
        <span><strong class="text-gray-
dark"><%=gPrograma.getNomeacaoProcedimentosVariaveis()%>/3</strong></span>
                                                        </div>
                                                    </div>
                                                </td>
                                            </tr>
                                            <tr>
        <td><%=messages.getString("ConceitoEventos")%></td>
                                                <td>
                                                    <div class="progress">
                                                        <div
        class="progress-bar progress-bar-striped
<%=getProgressBar(gPrograma.getEventos())%>"
        role="progressbar"
                                                                aria-
valuenow="<%=gPrograma.getEventos()%>"
```

```
                                                                    aria-
valuemin="0" aria-valuemax="3"

        style="width: <%=(100.0 / 3) * gPrograma.getEventos()%>%">

        <span><strong                                               class="text-gray-
dark"><%=gPrograma.getEventos()%>/3</strong></span>

                                                                    </div>

                                                                </div>

                                                            </td>

                                                        </tr>

                                                        <tr>

        <td><%=messages.getString("ConceitoAbstracao")%></td>

                                                            <td>

                                                                <div class="progress">

                                                                    <div

        class="progress-bar                                         progress-bar-striped
<%=getProgressBar(gPrograma.getAbstracaoProcedimentos())%>"

        role="progressbar"

                                                                    aria-
valuenow="<%=gPrograma.getAbstracaoProcedimentos()%>"

                                                                    aria-
valuemin="0" aria-valuemax="3"

        style="width: <%=(100.0 / 3) * gPrograma.getAbstracaoProcedimentos()%>%">

        <span><strong                                               class="text-gray-
dark"><%=gPrograma.getAbstracaoProcedimentos()%>/3</strong></span>

                                                                    </div>

                                                                </div>

                                                            </td>
```

```
                                                    </tr>
                                                    <tr>
        <td><%=messages.getString("ConceitoLacos")%></td>
                                                        <td>
                                                            <div class="progress">
                                                                <div
        class="progress-bar progress-bar-striped
<%=getProgressBar(gPrograma.getLacos())%>"
        role="progressbar" aria-valuenow="<%=gPrograma.getLacos()%>"
                                                                aria-
valuemin="0" aria-valuemax="3"
        style="width: <%=(100.0 / 3) * gPrograma.getLacos()%>%">
        <span><strong class="text-gray-
dark"><%=gPrograma.getLacos()%>/3</strong></span>
                                                                </div>
                                                            </div>
                                                        </td>
                                                    </tr>
                                                    <tr>
        <td><%=messages.getString("ConceitoCondicionais")%></td>
                                                        <td>
                                                            <div class="progress">
                                                                <div
        class="progress-bar progress-bar-striped
<%=getProgressBar(gPrograma.getCondicionais())%>"
```

role="progressbar"

aria-valuenow="<%=gPrograma.getCondicionais()%>"

aria-valuemin="0" aria-valuemax="3"

style="width: <%=(100.0 / 3) * gPrograma.getCondicionais()%>%">

<**span**><**strong** class="text-gray-dark"><%=gPrograma.getCondicionais()%>/3</**strong**></**span**>

</**div**>

</**div**>

</**td**>

</**tr**>

<**tr**>

<**td**><%=messages.getString("ConceitoListas")%></**td**>

<**td**>

<**div** class="progress">

<**div**

class="progress-bar progress-bar-striped <%=getProgressBar(gPrograma.getListas())%>"

role="progressbar"

aria-valuenow="<%=gPrograma.getListas()%>" aria-valuemin="0"

aria-valuemax="3"

style="width: <%=(100.0 / 3) * gPrograma.getListas()%>%">

<**span**><**strong** class=*"text-gray-dark"*><%=gPrograma.getListas()%>/3</**strong**></**span**>

</**div**>

</**div**>

</**td**>

</**tr**>

<**tr**>

<**td**><%=messages.getString("ConceitoPersistencia")%></**td**>

<**td**>

<**div** class=*"progress"*>

<**div**

class=*"progress-bar progress-bar-striped* <%=getProgressBar(gPrograma.getPersistenciaDados())%>*"*

role=*"progressbar"*

aria-valuenow=*"*<%=gPrograma.getPersistenciaDados()%>*"*

aria-valuemin=*"0"* aria-valuemax=*"3"*

style=*"width:* <%=(100.0 / 3) * gPrograma.getPersistenciaDados()%>*%"*>

<**span**><**strong** class=*"text-gray-dark"*><%=gPrograma.getPersistenciaDados()%>/3</**strong**></**span**>

</**div**>

</**div**>

</**td**>

</**tr**>

<**tr**>

```
<td><%=messages.getString("ConceitoSensores")%></td>
                                                    <td>
                                                        <div class="progress">
                                                            <div
class="progress-bar progress-bar-striped
<%=getProgressBar(gPrograma.getSensores())%>"
role="progressbar"
                                                                 aria-
valuenow="<%=gPrograma.getSensores()%>"
                                                                 aria-
valuemin="0" aria-valuemax="3"
style="width: <%=(100.0 / 3) * gPrograma.getSensores()%>%">
<span><strong class="text-gray-
dark"><%=gPrograma.getSensores()%>/3</strong></span>
                                                            </div>
                                                        </div>
                                                    </td>
                                                </tr>
                                                <tr>
<td><%=messages.getString("ConceitoDesenhoAnimacao")%></td>
                                                    <td>
                                                        <div class="progress">
                                                            <div
class="progress-bar progress-bar-striped
<%=getProgressBar(gPrograma.getDesenhoAnimacao())%>"
role="progressbar"
```

```
                                                                    aria-
valuenow="<%=gPrograma.getDesenhoAnimacao()%>"
                                                                    aria-
valuemin="0" aria-valuemax="3"
        style="width: <%=(100.0 / 3) * gPrograma.getDesenhoAnimacao()%>%">
        <span><strong                                               class="text-gray-
dark"><%=gPrograma.getDesenhoAnimacao()%>/3</strong></span>
                                                                    </div>
                                                                </div>
                                                            </td>
                                                        </tr>
                                                        <tr>
        <td><%=messages.getString("ConceitoOperadores")%></td>
                                                            <td>
                                                                <div class="progress">
                                                                    <div
        class="progress-bar                                         progress-bar-striped
<%=getProgressBar(gPrograma.getOperadores())%>"
        role="progressbar"
                                                                    aria-
valuenow="<%=gPrograma.getOperadores()%>"
                                                                    aria-
valuemin="0" aria-valuemax="3"
        style="width: <%=(100.0 / 3) * gPrograma.getOperadores()%>%">
        <span><strong                                               class="text-gray-
dark"><%=gPrograma.getOperadores()%>/3</strong></span>
                                                                    </div>
```

```
                                        </div>
                                    </td>
                                </tr>
                                <tr>

    <td><%=messages.getString("ConceitoVariaveis")%></td>
                                    <td>
                                        <div class="progress">
                                            <div

    class="progress-bar progress-bar-striped
<%=getProgressBar3Point(gPrograma.getVariaveis())%>"

    role="progressbar"
                                                aria-
valuenow="<%=gPrograma.getVariaveis()%>"
                                                aria-
valuemin="0" aria-valuemax="2"

    style="width: <%=(100.0 / 2) * gPrograma.getVariaveis()%>%">

    <span><strong class="text-gray-
dark"><%=gPrograma.getVariaveis()%>/2</strong></span>
                                        </div>
                                    </td>
                                </tr>
                                <tr>

    <td><%=messages.getString("ConceitoStrings")%></td>
                                    <td>
                                        <div class="progress">
                                            <div
```

class="*progress-bar* *progress-bar-striped* <%=getProgressBar3Point(gPrograma.getStrings())%>"

role="*progressbar*"

aria-valuenow="<%=gPrograma.getStrings()%>"

aria-valuemin="*0*" aria-valuemax="*2*"

style="*width:* <%=(100.0 / 2) * gPrograma.getStrings()%>*%*">

<**span**><**strong** class="*text-gray-dark*"><%=gPrograma.getStrings()%>/2</**strong**></**span**>

</**div**>

</**div**>

</**td**>

</**tr**>

<**tr**>

<**td**><%=messages.getString("ConceitoSincronizacao")%></**td**>

<**td**>

<**div** class="*progress*">

<**div**

class="*progress-bar* *progress-bar-striped* <%=getProgressBarDichotomous(gPrograma.getSincronizacao())%>"

role="*progressbar*"

aria-valuenow="<%=gPrograma.getSincronizacao()%>"

aria-valuemin="*0*" aria-valuemax="*1*"

```
style="width: <%=(100.0 / 1) * gPrograma.getSincronizacao()%>%">

<span><strong class="text-gray-dark"><%=gPrograma.getSincronizacao()%>/1</strong></span>
</div>
</div>
</td>
</tr>
<tr>

<td><%=messages.getString("ConceitoMapas")%></td>
<td>
<div class="progress">
<div

class="progress-bar progress-bar-striped <%=getProgressBar3Point(gPrograma.getMapas())%>"

role="progressbar" aria-valuenow="<%=gPrograma.getMapas()%>"
aria-valuemin="0" aria-valuemax="2"

style="width: <%=(100.0 / 2) * gPrograma.getMapas()%>%">

<span><strong class="text-gray-dark"><%=gPrograma.getMapas()%>/2</strong></span>
</div>
</div>
</td>
</tr>
<tr>
```

<td><%=messages.getString("ConceitoExtensoes")%></td>

<td>

<div

class="progress-bar progress-bar-striped
<%=getProgressBarDichotomous(gPrograma.getExtensoes())%>"

role="progressbar"

aria-
valuenow="<%=gPrograma.getExtensoes()%>"

aria-
valuemin="0" aria-valuemax="1"

style="width: <%=(100.0 / 1) * gPrograma.getExtensoes()%>%">

<strong class="text-gray-
dark"><%=gPrograma.getExtensoes()%>/1

</td>

</tr>

<tr>

<td><%=messages.getString("Total")%></td>

<td>

<div

class="progress-bar progress-bar-striped
<%=corParaBarraDeNota(gPrograma.getNotaProgram())%>"

role="progressbar"

```
                                                                    aria-
valuenow="<%=gPrograma.getNotaProgram()%>"
                                                                    aria-
valuemin="0" aria-valuemax="41"

        style="width: <%=100.0 * gPrograma.getNotaProgram() / 10.0%>%">

        <span><strong                                               class="text-gray-
dark"><%=gPrograma.getNotaProgram()%>/10</strong></span>
                                                                    </div>
                                                                </div>
                                                            </td>
                                                        </tr>
                                                    </tbody>
                                                </table>
                                            </div>

                                            <div id="tabs-2">

                                                <%-- CRITeRIOS DE DESIGN ORGANIZADOS
POR CATEGORIA --%>

                                                <table>
                                                    <tr class="table-active">
                                                        <th                 class="text-
center"><%=messages.getString("ColunaCategoria")%></th>
                                                        <th                 class="text-
center"><%=messages.getString("ColunaPontuacao")%></th>
                                                    </tr>
                                                </table>

                                                <%
```

```
                                    // Cada categoria
                                    for (CriterioDesignAI.Categorias categoria :
CriterioDesignAI.listarCategorias()) {
                                        ListaDeDadosParaBarras lista = new
ListaDeDadosParaBarras();

                                        // Cada criterio na categoria
                                        for (CriterioDesignAI criterio :
categoria.criteriosDaCategoria())

        lista.adicionar(bundleCriterios.nomeDoCriterio(criterio),
gDesign.notaNoCriterio(criterio),

        CriterioDesignAI.MAIOR_NIVEL_DE_DESEMPENHO);

                                        double media = lista.mediaDasNotas();
                                    %>
                                    <table class="collapsible">
                                        <tr>

        <td><%=bundleCriterios.nomeDaCategoria(categoria)%></td>
                                            <td class="text-center">
                                                <%
                                                /*

TODO Criar custom tag para as barras

tipoDaBarra = corParaBarraDeNota(media); valor = media; máximo = 10;
larguraPorcentagem = 100.0 * (media / 10.0); legenda = media + "/10";

<barra .../>

*/
```

```
                                                %>
                                                <div class="progress">
                                                    <div
                                                        class="progress-
bar progress-bar-striped <%=corParaBarraDeNota(media)%>"
                                                        role="progressbar" aria-valuenow="<%=media%>"
                                                        aria-
valuemin="0" aria-valuemax="10"
                                                        style="width:
<%=100.0 * (media / 10.0)%>%">
                                                        <span><strong
class="text-gray-dark"><%=media%>/10</strong></span>
                                                    </div>
                                                </div>
                                            </td>
                                        </tr>
                                    </table>
                                    <div class="content">
                                        <table class="table">
                                            <thead class="">
                                                <tr class="table-active">
                                                    <th class="text-
center"><%=messages.getString("ColunaConceito")%></th>
                                                    <th class="text-
center"><%=messages.getString("ColunaPontuacao")%></th>
                                                </tr>
                                            </thead>
                                            <tbody>
                                                <%
                                                for (DadosParaBarra dados :
lista.dadosDosConceitos) {
```

```
                                                    %>
                                                    <tr>
        <td><%=dados.nomeDoConceito%></td>
                                                        <td>
                                                            <%
                                                            /*
tipoDaBarra = dados.tipoDaBarra; valor = dados.nota; máximo = dados.notaMáxima;
larguraPorcentagem = dados.larguraDaBarra; legenda = dados.textoDaNota;

<barra ...>

*/
                                                            %>
                                                            <div
class="progress">
                                                                <div
        class="progress-bar progress-bar-striped <%=dados.tipoDaBarra%>"
        role="progressbar" aria-valuenow="<%=dados.nota%>"
        aria-valuemin="0" aria-valuemax="<%=dados.notaMáxima%>"
        style="width: <%=dados.larguraDaBarra%>%">
        <span><strong                                       class="text-gray-
dark"><%=dados.textoDaNota%></strong></span>
                                                                </div>
                                                            </div>
                                                        </td>
                                                    </tr>
```

```
                                        <%
                                        }
                                        %>
                                    </tbody>
                                </table>
                            </div>
                            <%
                            }
                            %>

                            <table class="table">
                                <tr>

    <td><b><%=messages.getString("Total")%></b></td>
                                    <td class="text-center">
                                        <%
                                        /*
                                        tipoDaBarra = corParaBarraDeNota(gDesign.getNotaInterUsuario()); valor = gDesign.getNotaInterUsuario(); máximo = 10; larguraPorcentagem = (100.0 / 10) * gDesign.getNotaInterUsuario(); legenda = gDesign.getNotaInterUsuario() + "/10";
                                        <barra ...>
                                        */
                                        %>
                                        <div class="progress">
                                            <div
                                                class="progress-bar progress-bar-striped <%=corParaBarraDeNota(gDesign.getNotaInterUsuario())%>"

    role="progressbar"
                                                aria-valuenow="<%=gDesign.getNotaInterUsuario()%>"
```

```
                                                    aria-
valuemin="0" aria-valuemax="10"
                                                    style="width:
<%=100.0 * gDesign.getNotaInterUsuario() / 10.0%>%">
                                                    <span><strong
class="text-gray-dark"><%=gDesign.getNotaInterUsuario()%>/10</strong></span>
                                                        </div>
                                                    </div>
                                                </td>
                                            </tr>
                                        </table>

                                        <div id="aesthetic">
                                            <table id='aesthetic-table'>
                                                <thead class="">
                                                    <tr class="table-active">
                                                        <th class="text-
center"><%=messages.getString("AvaliacaoEstetica")%></th>
                                                        <th class="text-
center"><%=messages.getString("ColunaPontuacao")%></th>
                                                    </tr>
                                                </thead>
                                                <tbody>
                                                    <!-- TODO: replace scriplet
format code with JSTL format -->
                                                    <%
                                                    if (uiImages != null) {
                                                        while (imageScorePair <
uiImages.size()) {

        out.print("<tr>");
```

```
                                                                    out.print("<td>"
+ "<p class=\"text-center\">" + "<img id=\"\" align=\"center\" width=\"135\"
height=\"240\""
                                                                    +
"src=\" " + uiImages.get(imageScorePair) + "\"/>" + "</p>" +
"</td>");
                                                                    out.print("<td
class=\"text-center\">" + "<div class=\"progress\">" + "<div "
                                                                    +
"class=\"progress-bar progress-bar-striped" + aestheticScores.get(imageScorePair) + "\""
                                                                    +
"role=\"progressbar\"" + "aria-valuenow=\"" + aestheticScores.get(imageScorePair) + "\""
                                                                    + "aria-
valuemin=\"0\" aria-valuemax=\"10\"" + "style=\"width:"
                                                                    + (100.0 / 10) *
aestheticScores.get(imageScorePair) + "%\">"
                                                                    +
"<span><strong class=\"text-gray-dark\">" + aestheticScores.get(imageScorePair)
                                                                    +
"/10</strong></span>" + "</div>" + "</div>" + "</td>");

        imageScorePair++;

        out.print("</tr>");
                                                                }
                                                            }
                                                            out.print("<tr
id=\"filechooser\">");
                                                            out.print("<td>" + "<p
class=\"text-center\">" + "<div style=\"margin-top: 10px;\">"
                                                            + "<form
id=\"ui-selection-form\" class=\"well\"" + "enctype=\"multipart/form-data\"
method=\"post\">"
                                                            + "<div
class=\"evaluate-aesthetic\">" + "<input id=\"ui-files\" accept=\".jpg\" type=\"file\" />"
```

```
                                                                    +          "</div>
</form> </div> </p>" + "</td>");
                                                                    out.print("<td        class=\"text-
center\">" + "<span><strong class=\"text-gray-dark\">"
                                                                    +
messages.getString("MsgEscolherImg") + "</strong></span>" + "</td>");
                                                                    out.print("</tr>");
                                                                    %>
                                                                </tbody>
                                                            </table>
                                                        </div>
                                                    </div>

                                                    <%/*---------------------------------
                                                                Aba Originalidade
                                                            ----------------------------------------*/%>
                                                    <div id="tabs-3">
                                                    <%
                                                        // Cada funcionalidade
                                                        for(int        i        =        0;        i        <
gPrograma.getLista_funcionalidades_app().size(); i++){
                                                            dadosDasFuncionalidades.add(

        bundleCriterios.nomeDaFuncionalidade(gPrograma.getLista_funcionalidades_app()
.get(i)));
                                                        }
                                                        %>
                                                        <table class="table">
                                                            <thead class="">
                                                                <tr class="table-active">
                                                                    <th                class="text-
center"><%=messages.getString("ColunaFuncionalidade")%></th>
```

<th class="text-center"><%=messages.getString("ColunaOriginalidade")%></th>

</tr>

</thead>

<tbody>

<% for(int i = 0; i < dadosDasFuncionalidades.size(); i++) { %>

<tr>

<td><%=dadosDasFuncionalidades.get(i)%></td>

<td class="text-center"><%=gPrograma.getLista_raridades_app().get(i)%></td>

</tr>

<% } %>

</tbody>

</table>

<table class="table">

<thead class="">

<tr class="table-active">

<th class="text-center"><%=messages.getString("ColunaCombinacoes")%></th>

<th class="text-center"><%=gPrograma.getResultadoDasCombinacoes()%></th>

</tr>

</thead>

<tbody>

<tr>

<td>A avaliação é feita com base em um universo de referência de 10.000 aplicativos feitos com App Inventor</td>

<td class = "text-center">

```
                                                    <button
onclick="showAndHideUl()">Universo de referência</button>
                                                    </td>
                                                </tr>
                                            </tbody>
                                        </table>
                                    </div>

                                    <div id="info-Originalidade">

                                        <table class="table">
                                            <thead class="">
                                                <tr class="table-active">
                                                    <h1>Originalidade das
funcionalidades</h1>
                                                    <p>
                                                        <h7>Tamanho do
universo de referência: 10.000 aplicativos</h7>
                                                    </p>
                                                    <p>
                                                        <h7>Distribuição de
funcionalidades no universo de referência</h7>
                                                    </p>
                                                    <img class="center-block img-
fluid d-block py-3 mx-auto"
                                                        src="resources/distribuicao_funcionalidades_universo.png"
                                                        width="75%">
                                                    <p>
                                                        <h7>Distribuição das
combinações de funcionalidades mais frequentes no universo de referência</h7>
                                                    </p>
```

```
                                        <img class="center-block img-fluid d-block py-3 mx-auto"
                                        src="resources/combinacoes_trecho.PNG"
                                        width="75%">
                                    </tr>
                                </thead>
                                <tbody class = "text-center">
                                    <button class = "text-center" onclick="showAndHideUl()">Voltar</button>
                                </tbody>
                            </table>
                        </div>

                        <%/*---------------------------------
                                Aba Originalidade
                        ----------------------------------------*/%>

                    </div>
                </div>
            </div>
        </div>

        <div class="container py-3">
            <div class="row">
                <div class="col-md-12"></div>
            </div>
        </div>
```

```
<footer class="footer text-success bg-success">
    <div class="container">
        <div class="row">
            <div class="col-centered col-lg-9">
                <img class="center-block img-fluid"
src="resources/Logo1.png"
                    width="10%"> <img class="center-block img-fluid"
                    src="resources/Logo2.png" width="10%"> <img
                    class="center-block img-fluid"
src="resources/Logo3.png"
                    width="6%"> <img class="center-block img-fluid py-
2"
                    src="resources/Logo4.png" width="4%">
            </div>
            <div class="col-md-2 text-left">
                <img class="center-block img-fluid d-block my-4"
                    src="resources/LogoCnE.png">
            </div>
        </div>
    </div>
</footer>
<footer class="footer bg-success">
    <div class="container">
        <div class="row">
            <div class="col-centered">
                <a href="sobre.jsp" class="card-link text-
white"><%=messages.getString("SobreBtn")%>
                         <b>|</b></a> <a
href="politica_de_privacidade.jsp"
                    class="card-link text-
white"><%=messages.getString("PoliticaBtn")%>
```

```
                                           <b>|</b></a>  <a
href="termos_de_servico.jsp"
                                    class="card-link text-
white"><%=messages.getString("TermosBtn")%></a>
                        </div>
                </div>
        </div>
</footer>
<script src="https://code.jquery.com/jquery-3.1.1.slim.min.js"></script>
<script
        src="https://cdnjs.cloudflare.com/ajax/libs/tether/1.4.0/js/tether.min.js"></script>
<script  src="https://pingendo.com/assets/bootstrap/bootstrap-4.0.0-
alpha.6.min.js"></script>
<script src="https://code.jquery.com/jquery-1.12.4.js"></script>
<script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>


<script>
        $(function() {
                $("#tabs").tabs();
        });
</script>


<script>
        var coll = document.getElementsByClassName("collapsible");
        var i;

        for (i = 0; i < coll.length; i++) {
                coll[i].addEventListener("click", function() {
                        this.classList.toggle("active");
                        var content = this.nextElementSibling;
                        if (content.style.maxHeight) {
```

```
                content.style.maxHeight = null;
            } else {
                content.style.maxHeight = content.scrollHeight + "px";
            }
        });
    }
</script>

<script>
    const evaluateAesthetic = event => {
        const files = event.target.files
        const formData = new FormData()
        formData.append('file', files[0])
        fetch('http://127.0.0.1:9999/', {
            method: "POST",
            body: formData
        })
        .then(response => response.json())
        .then(data => {
            var roundedScore = Math.round(parseFloat(data)*10)
            var reader = new FileReader()
            var table = document.getElementById("aesthetic-table")
            var oldRow = document.getElementById("filechooser")
            console.log(oldRow)
            table.deleteRow(-1)
            var newRow = table.insertRow()
            var tdImage = newRow.insertCell(0)
            var tdScore = newRow.insertCell(1)
```

```
                reader.onload = function (event) {
                        var newParagraph = document.createElement('p')
                        newParagraph.className = "text-center"
                        var newImage = document.createElement('img')
                        newImage.setAttribute('width', 135)
                        newImage.setAttribute('height', 240)
                        newImage.src = event.target.result
                        newParagraph.appendChild(newImage)
                        tdImage.appendChild(newParagraph)
                }
                reader.readAsDataURL(files[0])

                tdScore.innerHTML = "<td class=\"text-center\">" + "<div
class=\"progress\">" + "<div "
                                + "class=\"progress-bar progress-bar-striped" +
roundedScore + "\""
                                + "role=\"progressbar\"" + "aria-valuenow=\"" +
roundedScore + "\""
                                + "aria-valuemin=\"0\" aria-valuemax=\"10\"" +
"style=\"width:"
                                + (100.0 / 10) * roundedScore + "%\">"
                                + "<span><strong class=\"text-gray-dark\">" +
roundedScore
                                + "/10</strong></span>" + "</div>" + "</div>" +
"</td>"

                table.appendChild(oldRow)

        })
        .catch(error => {
                console.error(error)
```

```
            })
        }

        document.querySelector("#ui-files").addEventListener("change", event => {
            evaluateAesthetic(event)
        })
</script>

<script>
function updateTable(tableId, fields, data) {
        var rows = '';
        $.each(data, function(index, item) {
            var row = '<tr>';
            $.each(fields, function(index, field) {
                row += '<td>' + item[field+''] + '</td>';
            });
            rows += row + '<tr>';
        });
        $('#' + tableId).html(rows);
}
</script>

<script>
        $(document).ready(function(){
           $("#info-Originalidade").toggle();
           showAndHideUl = function(){
              $("#tabs-3").toggle();
              $("#tabs-3").toggleClass("component");
              $("#info-Originalidade").toggle();
```

```
        }

        var mostrandoNotaGeral = true;
        $("#notaOriginalidade").toggle();
        hideNotaOriginalidade = function(){
            if(!mostrandoNotaGeral){
                $("#notaGeral").toggle();
            $("#notaGeral").toggleClass("component");
            $("#notaOriginalidade").toggle();
            }
            mostrandoNotaGeral = true;
        }

        hideNotaGeral = function(){
            if(mostrandoNotaGeral){
                $("#notaGeral").toggle();
            $("#notaGeral").toggleClass("component");
            $("#notaOriginalidade").toggle();
            }
            mostrandoNotaGeral = false;
        }
    });
</script>
</body>

</html>
```

APÊNDICE C

Artigo SBC.

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

Nathalia da Cruz Alves[1], Christiane Gresse von Wangenheim[1], Matheus Alberto[1], Lucia Helena Martins Pacheco[1]

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

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

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

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

1. Introdução

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

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

Criando soluções tecnológicas os alunos aprendem a expressar suas ideias e competências de forma criativa [Clements 1995; Yadav e Cooper 2017], por exemplo por meio do desenvolvimento de artefatos computacionais, como apps, usando ambientes de programação

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

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

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

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

2. Trabalhos relacionados

Resultados de um mapeamento sistemático da literatura demonstram que existem poucas abordagens para avaliar a originalidade de produtos criados pelos alunos dentro do contexto do ensino de computação [Alves et al. 2020]. As abordagens encontradas focam principalmente na análise do código-fonte. Outros aspectos do produto, como o design de interface, o conteúdo e as funcionalidades, tipicamente não são considerados. Dependendo do ambiente de programação adotado, a originalidade do produto é avaliada comparando o produto do aluno com produtos desenvolvidos por outros alunos em um mesmo contexto educacional [Mustafaraj et al. 2017; Turbak et al. 2017], comparando o artefato do aluno com alguma solução predefinida [Bennett et al., 2013] ou pelo professor pontuando o produto [Scaico et al. 2013] de acordo com uma rubrica [Basu 2019]. Mustafaraj et al. (2017) e Turbak et al. (2017) avaliam a originalidade usando modelos de classificação de aprendizado de máquina. Bennet et al. (2013) avaliam a divergência entre produtos por meio de padrões

de pensamento computacional definidos com fórmulas matemáticas. Basu (2019) avalia a novidade de forma limitada por meio de apenas um item em uma rubrica. O *feedback* é apresentado por meio de uma nota final ou uma classificação do produto como 'original' ou 'não original'.

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

3. Método de Pesquisa

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

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

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

Aplicação exemplo. Nesta etapa, o modelo é aplicado exemplarmente de forma exploratória por meio de quatro avaliações de apps desenvolvidos por alunos da Educação Básica em um contexto real de aprendizagem. Esta aplicação exemplo é realizada por professores de computação seguindo as rubricas definidas, e tem como objetivo obter evidências iniciais da aplicabilidade e corretude das rubricas Modifique e Crie dentro do contexto educacional conforme recomendado por Moskal (2000). Os resultados da aplicação são discutidos e melhorias são propostas.

4. Desenvolvimento do modelo de avaliação da originalidade

4.1. Análise de domínio de conteúdo

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

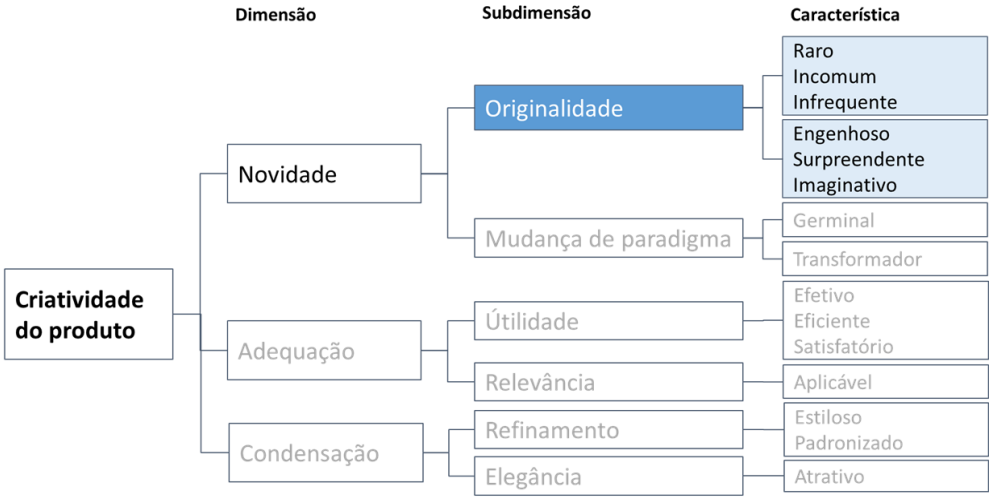


Figura 1. Características do produto criativo [Alves et al. 2020]

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

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

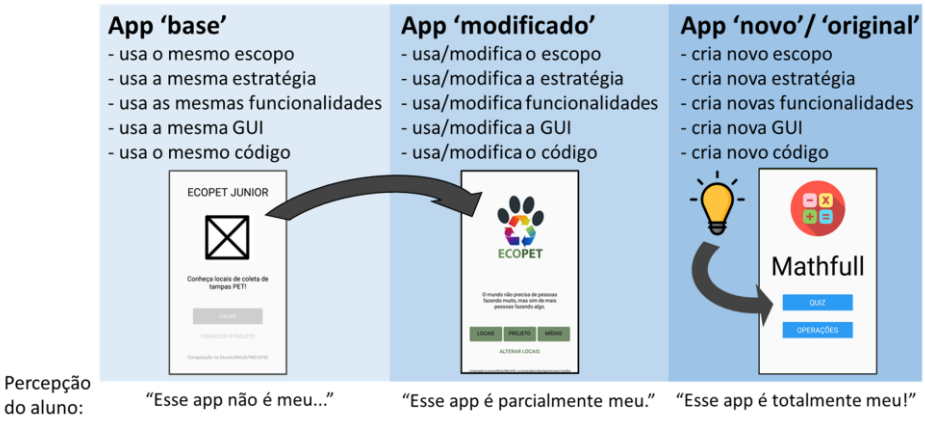


Figura 2. Abordagem UMC no contexto de desenvolvimento de apps

4.2. Definição do modelo de avaliação da originalidade de apps

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

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

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

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

5. Aplicação Exemplo

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

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

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

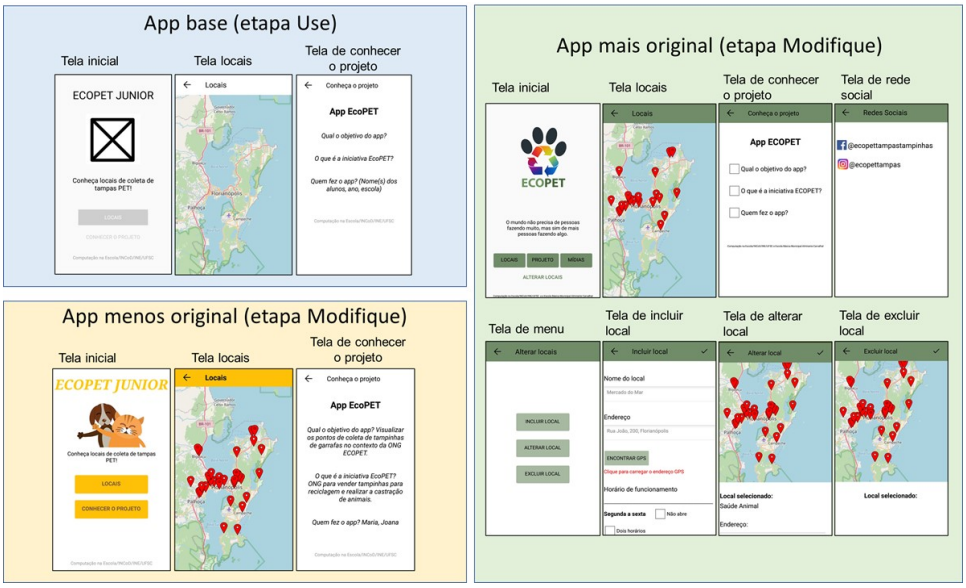


Figura 4. Ilustração dos apps das etapas Use e Modifique.

Os apps "novos" representam apps criados por alunos na etapa Crie. Novamente, com base em uma avaliação por especialistas foram selecionados dois apps: um menos original consistindo em um app com ideias comuns. O app menos original tem como objetivo ajudar

estudantes com dificuldades em matemática fornecendo um quiz sobre as 4 operações básicas da matemática e uma explicação sobre cada operação (Figura 5). O app mais original consiste em um app sobre a balneabilidade das praias, com escopo e objetivo mais incomum. As funcionalidades incluem descrição e balneabilidade de praias, mapa de todas as praias e suporte ao usuário (Figura 5).

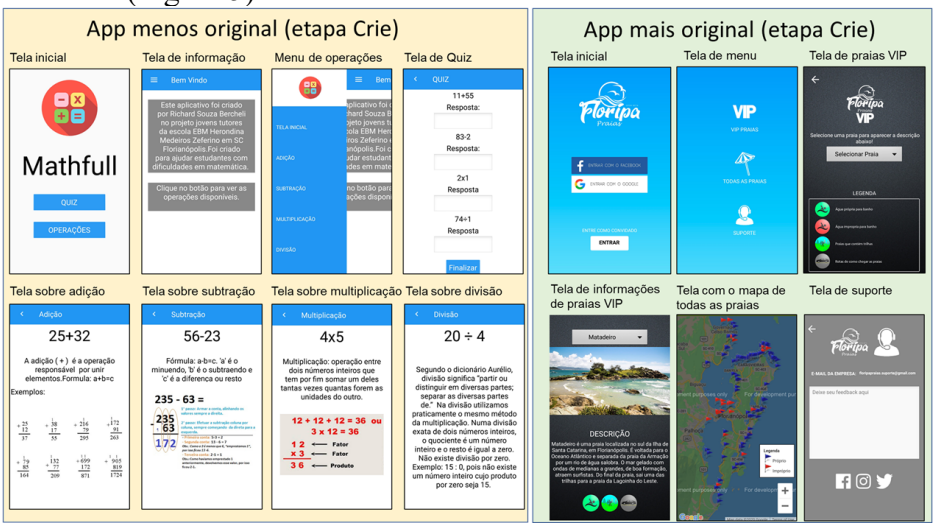


Figura 5. Ilustração dos apps da etapa Crie.

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

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

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

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

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

Rubrica			Avaliação do app 'Modificado' (em relação ao app 'Base')		Avaliação do app 'Novo' (em relação a apps da Galeria App inventor)	
Dimensão	**Elemento**	**Critério**	**menos**	**mais**	**menos**	**mais**

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

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

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

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

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

6. Conclusão

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

Agradecimentos

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

Referências

Allen, S., Knight, J. (2009). A Method for Collaboratively Developing and Validating a Rubric. International Journal for the Scholarship of Teaching and Learning, 3(2), 10.

Alves N. da C., Gresse von Wangenheim, C., Martins-Pacheco, L. H. (2020). Assessing Product Creativity in Computing Education: A Systematic Mapping Study. Informatics in Education. (*in-press*)

Basili, V. R., Caldiera, G., Rombach, H. D. (1994). The goal question metric approach. In Encyclopedia of Software Engineering, John Wiley.

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

Bennett, V. E., Koh, K. H., Repenning, A. (2013). Computing creativity: divergence in computational thinking. In Proc. of the 44th Technical Symposium on Computer Science Education, 359–364. ACM.

Besemer, S., Treffinger, D. J. (1981). Analysis of Creative Products: Review and Synthesis. Journal of Creative Behavior, 15(3), 158-178.

Cavallo, D., Singer, H., Gomes, A., Bittencourt, I., Silveira, I. (2016). Inovação e Criatividade na Educação Básica: Dos conceitos ao ecossistema. Revista Brasileira de Informática na Educação, 24(2).

Clements, D. H. (1995). Teaching creativity with computers. Educational Psychology Review, 7, 141–161.

Ferreira, M. N. F., Pinheiro, F. da C., Gresse von Wangenheim, C., Filho, R. M., Hauck, J. C. R. (2020). Ensinando design de interface de usuário de aplicativos móveis no ensino fundamental. Revista Brasileira de Informática na Educação, 28.

Gal, L., Hershkovitz, A., Morán, A. E., Guenaga, M., Garaizar, P. (2017). Suggesting a log-based creativity measurement for online programming learning environment. In Proc. of the 4th Conference on Learning at Scale, 273-277. ACM.

Garrett, J. (2010). The Elements of User Experience: User-Centered Design for the Web and Beyond, (2nd. ed.). New Riders Publishing, USA.

Hauck, J. C. R, Gresse von Wangenheim, C., Medeiros, G., Filho, R. M., Alves N., Laurentino, S., Santos, V. (2018). Jovens tutores de programação: um relato de experiência. Revista Eletrônica de Extensão UFSC, 15(29).

Jackson, P. W., Messick, S. (1964). The Person, The Product, and the Response. ETS Research Bulletin Series, 2, i-27.

Lye, S. Y., Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12?. Computers in Human Behavior, 41, 51–61.

Moskal, B. M. (2000). Scoring Rubrics: What, When and How?," Practical Assessment, Research, and Evaluation, 7(3).

Mustafaraj, E., Turbak, F., Svanberg, M. (2017). Identifying Original Projects in App Inventor. In Proc. of the 30th Int. Florida Artificial Intelligence Research Society Conference, 567-572.

Runco, M. A., Jaeger, G. J. (2012). The standard definition of creativity. Creativity Research Journal, 24(1), 92-96.

Scaico, P. D., Lima, A. A., Silva, J. B. B., Azevedo, S., Paiva, L. F., Raposo, E. H., Alencar, Y., Mendes, J. P., Scaico, A. (2013). Ensino de Programação no Ensino Médio: Uma Abordagem Orientada ao Design com a linguagem Scratch. Revista Brasileira de Informática na Educação, 21(2).

Turbak, F., Mustafaraj, E., Svanberg, M., Dawson, M. (2017). Identifying and Analyzing Original Projects in an Open-Ended Blocks Programming Environment. In Proc. of the 23rd Int. Conference on Visual Languages and Sentient Systems, Pittsburgh, PA, USA.

Voogt, J., Roblin, N. P. (2012). A comparative analysis of international frameworks for 21st century competences. Journal of Curriculum Studies, 44(3), 299–321

Walia, C. (2019). A Dynamic Definition of Creativity. Creativity Research Journal, 31(3), 237-247.

Yadav, A., Cooper, S. (2017). Fostering Creativity Through Computing. Comm. of the ACM, 60(2), 31-33.