

Uma Análise em Larga-Escala das Funcionalidades de Aplicativos criados com App Inventor

Nathalia da Cruz Alves
Departamento de Informática e Estatística
Universidade Federal de Santa Catarina
Florianópolis/Santa Catarina/Brasil
nathalia.alves@posgrad.ufsc.br

Christiane Gresse von Wangenheim
Departamento de Informática e Estatística
Universidade Federal de Santa Catarina
Florianópolis/Santa Catarina/Brasil
c.wangenheim@ufsc.br

RESUMO

O ensino de algoritmos e programação na Educação Básica se tornou essencial. Um dos ambientes de programação utilizados nesse nível de ensino é o App Inventor, que permite criar aplicativos móveis. Embora existam diversos estudos sobre o aprendizado de programação com o App Inventor, uma questão que permanece é quais funcionalidades são tipicamente implementadas. Assim, este artigo apresenta os resultados de uma análise estatística em larga escala baseada na extração automatizada de funcionalidades a partir do código de 100.000 projetos de aplicativos criados com o App Inventor. Os resultados demonstram que algumas funcionalidades são muito comuns e relacionadas a própria idiossincrasia de aplicativos. Por outro lado, apesar de existir suporte por parte do App Inventor, algumas funcionalidades são extremamente raras devido a um nicho de uso e falta de materiais didáticos. Esses resultados podem ajudar em uma melhor compreensão das funcionalidades atualmente presentes na maioria dos projetos, indicando oportunidades para o desenvolvimento de novos materiais e suporte na avaliação da aprendizagem, incluindo o desenvolvimento de criatividade.

CCS CONCEPTS

• **Social and professional topics** → Computing education.

PALAVRAS-CHAVE

App Inventor, Funcionalidade, Extração, Programação, Educação Básica

1 INTRODUÇÃO

A importância da computação hoje em dia é amplamente reconhecida. Em diversos países a computação já faz parte do currículo da Educação Básica. No Brasil, recentemente, o MEC homologou as Normas sobre Computação na Educação Básica – Complemento à BNCC [31] aprovadas pelo Conselho Nacional de Educação [1]. As normas abordam competências e premissas específicas da computação que as escolas devem iniciar a

implementação a partir de 2023, as quais incluem, no Ensino Médio [2]:

Aplicar boas práticas da Engenharia de Software, tanto para construir uma solução usando níveis de abstração diferentes partindo da definição dos requisitos, especificação, projeto e implementação (refinamento vertical), quanto para fazer a evolução do sistema partindo de um protótipo e inserindo gradualmente as **funcionalidades** desejadas até chegar ao sistema completo (refinamento horizontal). {grifo adicionado}

Existem diversas alternativas para implementar as normas da Computação, uma delas é usando ambientes de programação para a criação de aplicativos, como o App Inventor. O App Inventor [3] é um ambiente de programação baseado em blocos que permite projetar a interface de usuário e programar a parte funcional de aplicativos Android. É usado por uma ampla gama de pessoas de diversas idades e origens, com mais de 1 milhão de usuários ativos mensais de 195 países que criaram quase 30 milhões de aplicativos móveis [4]. Os projetos de aplicativos criados com App Inventor podem ser compartilhados por meio da Galeria AI [5], sob a licença *creative Commons*, com metadados de nome, descrição do aplicativo e créditos. Nenhuma informação biográfica dos autores é incluída. Atualmente, a Galeria AI possui mais de 243.000 projetos com diferentes funcionalidades.

As funcionalidades de um aplicativo permitem identificar o objetivo que o aplicativo atende em diferentes níveis de granularidade. A partir das funcionalidades identificadas, é possível inferir também as decisões do processo de criação do aplicativo, incluindo, por exemplo, se um aplicativo foi criado seguindo um tutorial ou com base em ideias próprias. Existem funcionalidades que são usadas com mais frequência e outras que quase nunca aparecem. A partir de uma análise estatística em larga escala é possível obter respostas para essas questões.

Existem diversos estudos que analisam grandes conjuntos de dados compostos por projetos criados com ambientes de programação visual tipicamente usados na Educação Básica [8]. Enfocando App Inventor, os estudos encontrados na literatura analisam a desde a complexidade programática e progressão da aprendizagem [6] a conceitos de algoritmos e programação relacionados ao pensamento computacional [8]. Existem também estudos enfocando a detecção de funcionalidades em lojas de aplicativos, como Google Play e Windows Phone Store [9]. No entanto, não foram encontrados estudos analisando as funcionalidades de aplicativos criados com App Inventor. Assim,

Fica permitido ao(s) autor(es) ou a terceiros a reprodução ou distribuição, em parte ou no todo, do material extraído dessa obra, de forma verbatim, adaptada ou remixada, bem como a criação ou produção a partir do conteúdo dessa obra, para fins não comerciais, desde que sejam atribuídos os devidos créditos à criação original, sob os termos da licença CC BY-NC 4.0.

EduComp'23, Abril 24-29, 2023, Recife, Pernambuco, Brasil (On-line)

© 2023 Copyright mantido pelo(s) autor(es). Direitos de publicação licenciados à Sociedade Brasileira de Computação (SBC).

este artigo preenche essa lacuna apresentando uma análise em larga-escala das funcionalidades presentes em aplicativos App Inventor. Com o objetivo de apresentar uma compreensão de quais funcionalidades as pessoas programam com o App Inventor, são analisados 100 mil aplicativos compartilhados via Galeria AI [5]. Para tanto, é realizada a detecção automática das funcionalidades usando um módulo da ferramenta CodeMaster [32] baseado em regras para obtenção dos dados de análise. Os dados obtidos são analisados usando estatística descritiva, buscando dar uma visão geral de como é o uso de funcionalidades em aplicativos da Galeria AI [5].

2 APP INVENTOR

O App Inventor é um ambiente de programação visual de código aberto usado para criar aplicativos móveis para dispositivos Android [3]. É um ambiente de programação executado em um navegador da web e utiliza um editor de blocos de arrastar e soltar. Foi originalmente fornecido pela Google e atualmente é administrado pelo *Massachusetts Institute of Technology*.

Tabela 1: Componentes do App Inventor

Categoria	Descrição	Exemplos de componentes
Interface	Componentes de design de interface.	Botão, Imagem, Deslizador, etc.
Layout	Componentes de organização da interface.	Organização Horizontal, em Tabela, etc.
Mídia	Componentes de interação de mídia.	Câmera, Reprodutor de som, etc.
Desenho e Animação	Componentes de criar desenhos e animações.	Canvas, Sprite, Bola
Mapas	Componentes de navegação por mapa e marcadores.	Mapa, Círculo, Marcador, Polígono, etc.
Sensores	Componentes que obtêm informações dos sensores do dispositivo.	Acelerômetro, GPS, Relógio, Giroscópio, etc.
Social	Componentes de comunicação.	Compartilhamento, Contatos, etc.
Armazenamento	Componentes de acesso a bases de dados.	Arquivo, TinyDB, etc.
Conectividade	Componentes de conectividade com outros dispositivos.	API, Bluetooth, Web, etc.
Lego Mindstorms	Componentes do LEGO® MINDSTORMS® NXT.	NxtDirectCommand, NxtColorSensor, NxtLightSensor, etc.
Experimental	Componentes experimentais.	CloudDB, FirebaseDB
Extensões	Componentes personalizados.	Componentes de extensões.

Um aplicativo móvel pode ser criado em duas etapas com o App Inventor. Primeiro, usando o editor de designer para configurar os componentes da interface do usuário (Tabela 1), como botões, rótulos, etc. O designer também permite que os componentes não visuais, como sensores, componentes sociais e de mídia, que acessam os recursos do telefone, sejam especificados. Os componentes do App Inventor são divididos em categorias (Tabela 1) que refletem em blocos de eventos, métodos e propriedades específicas para cada componente.

O comportamento do aplicativo é programado em um segundo momento, conectando blocos de programação visual no editor de blocos. Cada bloco corresponde a nós de uma árvore de sintaxe abstrata em linguagens de programação textuais. Os blocos podem representar conceitos clássicos de programação, como laços, funções, condicionais, etc., ou condições, eventos e ações para um componente do aplicativo.

Em relação ao contexto educacional e público-alvo, o ambiente é voltado a estudantes da Educação Básica, especialmente nos Anos Finais do Ensino Fundamental [34] ou Ensino Médio [35]. No entanto, existem relatos de uso do App Inventor no primeiro ano do Ensino Superior [17][18].

O código-fonte de um projeto do App Inventor é salvo automaticamente na nuvem, mas também pode ser exportado como um arquivo AIA. Um arquivo AIA é uma coleção de arquivos compactados que inclui um arquivo de propriedades do projeto, arquivos de mídia que o projeto usa e dois arquivos são gerados para cada tela no aplicativo, um arquivo BKY e um arquivo SCM. O arquivo BKY envolve uma estrutura XML incluindo todos os blocos de programação usados para implementar as funcionalidades do aplicativo. O arquivo SCM envolve uma estrutura JSON que contém todos os componentes visuais usados no aplicativo [3].

2.1 Funcionalidades

No desenvolvimento de aplicativos, as funcionalidades desejadas são levantadas, tipicamente, por meio de uma análise de requisitos [33]. Nesse contexto, o conjunto de requisitos funcionais definem as funcionalidades que um aplicativo deve conter, por outro lado, as funcionalidades que de fato foram implementadas são descritas como especificações [33]. As funcionalidades descritas em especificações podem ser, por exemplo, fazer login, tirar fotos, salvar dados, etc.

A intencionalidade predominante do uso do App Inventor dentro do contexto educacional é ensinar a criar algo no ambiente (saber fazer). Desta forma, o App Inventor é utilizado para ensinar os estudantes a criar uma ampla gama aplicativos com diversas funcionalidades [7][14]. Os aplicativos podem ser criados a partir de ideias próprias ou seguindo tutoriais e cursos, que são disponibilizados no próprio ambiente App Inventor, na internet ou ministrados em escolas e universidades [19][26]. Alguns dos tutoriais mais populares permitem criar aplicativos com as funcionalidades de mostrar informações, como o clássico “Olá, mundo!”, reproduzir sons e animações, além de funcionalidades relacionadas a jogos ou sensores.

Dependendo do contexto, os aplicativos podem ter funcionalidades muito parecidas ou até mesmo idênticas a de

tutoriais ou podem ser originais quando criados a partir de ideias próprias. Com base na análise do código do aplicativo, é possível identificar as funcionalidades implementadas e fazer uma análise em larga-escala de quais funcionalidades são mais frequentes nos aplicativos.

3 TRABALHOS RELACIONADOS

Diversos estudos já fizeram uma análise em larga escala sobre diferentes aspectos de aplicativos criados com App Inventor (Tabela 2), incluindo conceitos de abstração [10][16], complexidade programática [11][15] e pensamento computacional [6][8]. A quantidade de projetos utilizada nos trabalhos correlatos varia de cerca de 5 mil a 1,5 milhão de projetos analisados sob diferentes enfoques.

Tabela 2: Comparação entre trabalhos relacionados

Referência	Conjunto de dados (App Inventor)	Foco da análise
Alves et al. 2020 [15]	88.606 projetos da Galeria AI	Complexidade programática e de interface e comparação com Scratch
Alves et al. 2021 [8]	88.812 projetos da Galeria AI	Dificuldade de blocos de algoritmos e programação
Li et al. 2017 [10]	1.577.039 projetos de usuários aleatórios e prolíficos	Abstração e uso de procederes via blocos
Patton et al. [16]	1.576.135 projetos de usuários aleatórios e prolíficos	Uso de abstração via blocos e design de linguagem de programação informado por dados
Xie et al. 2015 [11]	5.228 projetos aleatórios de um corpus de 9,7 milhões de projetos	Complexidade programática via número de blocos únicos
Xie e Abelson 2016 [6]	Aproximadamente 120 mil projetos de uma amostra aleatória de 10.571 usuários que criaram pelo menos 20 projetos.	Pensamento computacional e progressão de habilidades via blocos

Existem ainda diversos outros estudos sobre diferentes aspectos do App Inventor, no entanto, esses estudos não analisam uma quantidade expressiva de dados em larga-escala. Park e Shin [12], por exemplo, analisaram o uso de blocos relacionados a conceitos do pensamento computacional em 379 projetos provenientes de tutoriais oficiais com projetos curtidors por mais de 10 usuários e aplicativos do mês de 2015 a 2019 da Galeria AI. Svanberg [17] analisou a similaridade com tutoriais em 894 projetos de alunos da disciplina CS0 do *Wellesley College*. Mustafaraj et al. [18] analisaram a similaridade de projetos com

tutoriais e exercícios em 902 projetos de 16 alunos da disciplina *CS117 Inventing Mobile Apps* do *Wellesley College*.

Embora existam diversos estudos em larga-escala na literatura, até o momento, não foram encontradas pesquisas com foco direto na análise das funcionalidades implementadas em aplicativos móveis criados com App Inventor.

4 MÉTODOS

Assim, o objetivo deste estudo é analisar quais funcionalidades estão presentes em projetos do App Inventor e em que grau. Desta forma, esta pesquisa consiste em um estudo de caso para compreender os fenômenos associados à presença ou não de funcionalidades no que tange ao uso do ambiente de programação App inventor para o desenvolvimento de aplicativos. Adotando a abordagem *Goal Question Metric* [13] é definida a seguinte pergunta de pesquisa:

Pergunta de pesquisa: Quais funcionalidades e em que grau estão presentes em projetos App Inventor?

O método de pesquisa é dividido em três etapas e a pergunta de pesquisa é dividida em duas perguntas de análise:

PA1. Qual a prevalência de funcionalidades em projetos App Inventor?

PA2. Qual a prevalência da combinação de funcionalidades em projetos App Inventor?

Definição do estudo. Durante esta fase, o objetivo principal do estudo é especificado. Isso inclui a definição de quais funcionalidades são extraídas com base em uma análise sistemática dos componentes do App Inventor e suas respectivas regras de extração automatizada via código [30].

Coleta de dados. A pesquisa é baseada em dados na forma de projetos App Inventor. Para otimizar o tamanho da amostra, foram baixados aplicativos publicamente disponíveis e acessíveis na web, em sua maioria (>99%) provenientes da Galeria AI [5]. Também foram inclusos aplicativos de páginas de aplicativos do mês [24] e de disciplinas de graduação ou pós-graduação de cursos de computação [25]. Como resultado, foi obtido o código-fonte de 100 mil projetos App Inventor. Os projetos são analisados e suas funcionalidades são extraídas com base nas regras de extração definidas (Tabela 3). A extração segue a seguinte definição: *se o aplicativo contém os blocos/componentes indicativos, então detecta a funcionalidade* (Tabela 3).

Tabela 3: Exemplos de regras de extração de funcionalidades

Funcionalidade	Regra de extração
Acessar website	Bloco contém WebViewer OU Bloco contém Web OU text Bloco contém "http://www."
Exibir informações	Bloco contém Label com caracteres >= 20 OU Bloco contém TextBox com caracteres >= 20
Fazer login	(Bloco contém File OU Bloco contém TinyDB OU Bloco contém FirebaseDB OU Bloco contém TinyWebDB OU Bloco contém CloudDB) E Bloco contém PasswordTextBox

As regras foram definidas de forma sistemática observando diferentes aplicativos da Galeria AI de forma a obter um conjunto representativo de funcionalidades. Considerando que a Galeria AI não permite o envio de aplicativos com extensões, que fazem o papel de bibliotecas externas no ambiente App Inventor, são extraídas somente as funcionalidades referentes a blocos do App Inventor para as quais não é necessário o uso de extensões. Ao total, foram definidas 42 funcionalidades. Por questões de espaço, a lista completa de extração está disponível no Apêndice ao final deste artigo.

Análise de dados. Os dados coletados são agrupados em uma única amostra para analisar as funcionalidades em larga-escala (ao invés de um aplicativo específico). Uma análise estatística é realizada para tentar esclarecer a decisão ou o conjunto de decisões que foram tomadas em relação à implementação de uma ou mais funcionalidades. Para analisar a prevalência de funcionalidades em projetos App Inventor é realizada uma análise estatística descritiva. A prevalência da combinação de funcionalidades é analisada elencando-se todas as combinações de funcionalidades existentes nos aplicativos do conjunto de dados.

5 RESULTADOS

Nas seções a seguir, são apresentados os resultados obtidos por meio da análise dos projetos do App Inventor para cada uma das perguntas de análise. Dentre 100 mil aplicativos, em 1,2 mil não foi detectada nenhuma funcionalidade conforme as regras de extração definidas (Tabela 5, Apêndice). A maioria dos aplicativos que não tiveram nenhuma funcionalidade detectada consiste em projetos sem funcionalidade programática, isto é, projetos compostos apenas por telas com componentes de design sem nenhum bloco de programação. Além disso, também se observou alguns projetos totalmente em branco, sem nenhum componente de design ou bloco de programação. Desta forma, nenhuma funcionalidade foi detectada nesses aplicativos. Assim, são

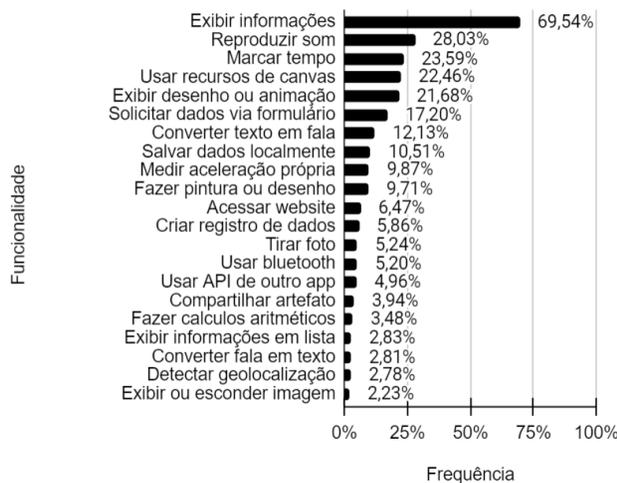
considerados na análise somente os 98.770 aplicativos nos quais foram detectadas pelo menos uma funcionalidade. Esses aplicativos foram criados por pessoas de vários lugares do mundo e incluem projetos em diversas línguas, como inglês, italiano, francês, espanhol, hindi, alemão, chinês, japonês, coreano e português.

5.1 Qual a prevalência de funcionalidades em projetos App Inventor?

A partir das funcionalidades extraídas, foram calculadas as frequências de cada uma das funcionalidades dentre os 98.770 aplicativos (Figura 1). Observa-se que a funcionalidade que mais aparece é a de exibir informações, cuja regra de extração é “Bloco contém *Label* com caracteres ≥ 20 OR bloco contém *TextBox* com caracteres ≥ 20 ” (Tabela 3), presente em mais de 69,54% dos projetos do App Inventor. Isso é esperado, pois compreende um design básico de interfaces de usuário com rótulos ou caixas de texto usados para mostrar algumas informações na tela ao usuário, inclusive informações de ajuda sobre uso do aplicativo (Figura 2). Além disso, é comum que materiais didáticos mais simples apresentem o ambiente de programação do App Inventor por meio de tutoriais que trocam informações exibidas na tela de forma a demonstrar ao aprendiz um aplicativo semelhante ao clássico “Olá Mundo”. Assim, infere-se que esta é a funcionalidade de maior interesse por parte do público-alvo do App Inventor, ainda que possa ser utilizada com diferentes objetivos em mente.

A segunda funcionalidade que mais aparece é a de reproduzir som em 28,03% dos aplicativos (Figura 1). Considerando a regra de extração definida, isso inclui aplicativos para ouvir músicas específicas a aplicativos com efeitos sonoros (Figura 3), como o tutorial Hello Codi [20], o qual ensina a criar um aplicativo que reproduz o som de uma abelha ao clicar em um botão. A funcionalidade de marcar tempo aparece em 23,59% dos aplicativos (Figura 1). Isso pode ser explicado pelo fato de que, tipicamente, a marcação de tempo é utilizada para sincronização

As 21 funcionalidades mais detectadas



As 21 funcionalidades menos detectadas

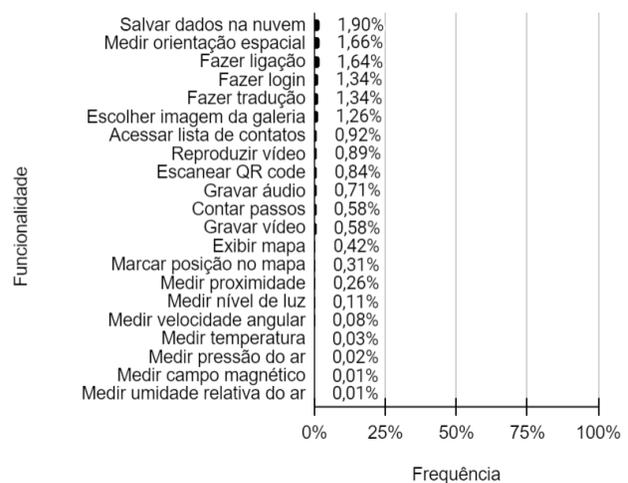


Figura 1: Frequência das 42 funcionalidades extraídas de 98.770 aplicativos

de efeitos na tela, eventos ou outras funcionalidades programadas dentro do projeto.



Figura 2: Exemplo de um projeto que possui a funcionalidade de visualizar informações

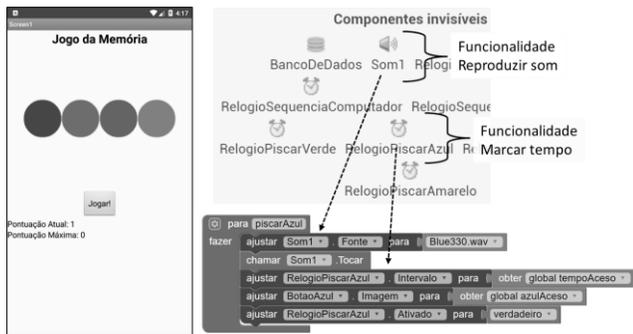


Figura 3: Exemplo de um projeto que possui a funcionalidade de reproduzir som e marcar tempo

A funcionalidade de usar recursos de canvas aparece em 22,46% dos aplicativos (Figura 1). Os recursos de canvas podem ser usados com diversos objetivos, incluindo fazer pinturas, desenhos, mostrar e editar imagens ou *sprites*, que são componentes do

canvas. A funcionalidade de fazer pintura ou desenho aparece em 9,71% dos projetos (Figura 1) e inclui fazer desenhos arrastando o dedo sobre a tela, usando botões ou arrastando *sprites* de um canvas. É possível também usar os *sprites* para mostrar animações programadas via blocos. Isso é demonstrado pela funcionalidade exibir desenho ou animação, que aparece em 21,68% dos projetos App Inventor (Figura 1). Em alguns projetos, essas animações podem fazer parte de personagens de jogos, como no popular tutorial para criar um jogo do tipo *Space Invaders* [21].

Algumas funcionalidades relacionadas a dados aparecem em vários projetos, por exemplo, solicitar dados via formulário, que aparece em 17,20% dos projetos (Figura 1) e se refere a aplicativos com no mínimo três campos de texto para inserção de dados (Figura 4). Outras funcionalidades relacionadas a dados incluem salvar dados localmente, que aparece em 10,51% dos aplicativos (Figura 1), salvar dados na nuvem, que é menos usada e aparece em 1,90% dos aplicativos, e criar registro de dados, que aparece em 5,86% dos aplicativos.

Algumas funcionalidades estão relacionadas a aspectos de usabilidade de aplicativos móveis. Por exemplo, a conversão de texto em fala é uma das funcionalidades mais usadas e aparece em 12,13% dos projetos (Figura 1). Uma das razões para esse uso expressivo é a própria existência do componente *TextToSpeech* do App Inventor e do tutorial *Talk to Me*, um dos tutoriais iniciantes mais populares do App Inventor, disponibilizado direto no próprio site ensinando a criar um aplicativo que fala [19]. Apesar de o componente *TextToSpeech* estar disponível somente nos idiomas inglês, alemão, espanhol, francês, italiano, isso não impede que seja usado também para textos em português. Projetos com essa funcionalidade ajudam na acessibilidade para pessoas cegas ou analfabetas, na qual parte ou todas as informações da tela são faladas para auxiliar no uso do aplicativo. Nesse contexto, a funcionalidade de medir aceleração própria pode ser usada com o mesmo objetivo de acessibilidade, permitindo que a interação com o aplicativo seja feita por meio de gestos, por exemplo, um aplicativo que fala o resultado do cálculo do IMC após sacudir o celular. A funcionalidade de converter fala em texto aparece em

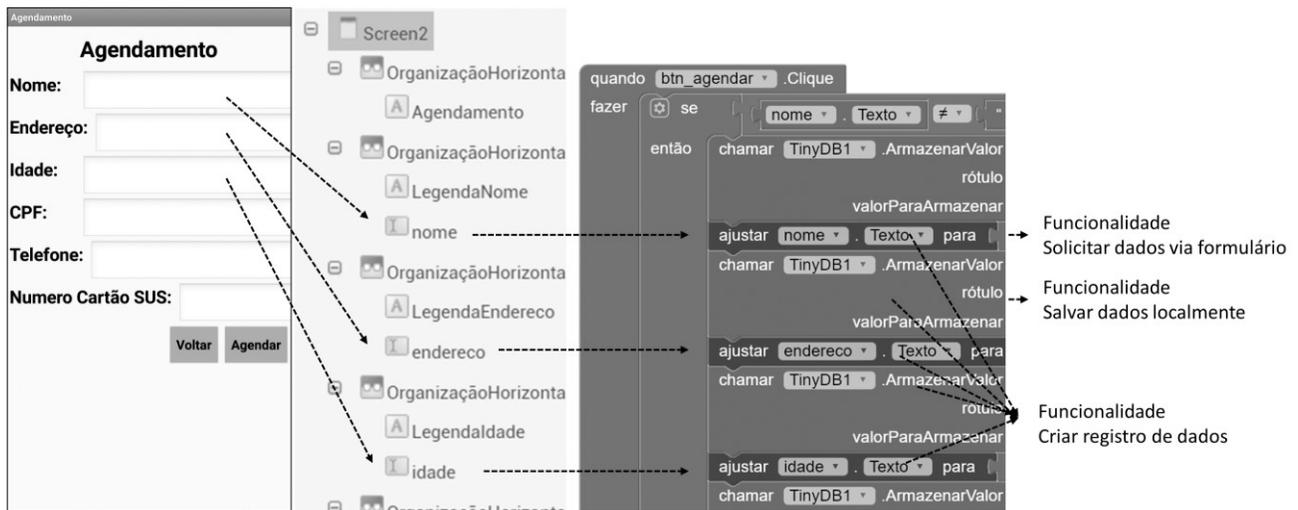


Figura 4: Exemplo de um projeto que possui funcionalidades relacionadas a dados

2,81% dos projetos (Figura 1) e permite que entradas de dados do tipo texto sejam feitas via comandos de voz. Também pode ser usada para facilitar a usabilidade de dispositivos móveis.

Algumas funcionalidades que incluem conexão do aplicativo com componentes externos do celular e estão dentre as 21 funcionalidades mais usadas são acessar website, usar bluetooth, usar API de outro aplicativo e compartilhar artefato (Figura 1). Essas funcionalidades tipicamente são utilizadas para acessar recursos específicos, por exemplo, acessar um site de uma loja, conectar-se com um dispositivo físico, acessar aplicativos de mensageria, como WhatsApp. Uma exceção é a funcionalidade de compartilhar artefato, que em versões mais recentes do Android, permite usar o próprio componente do sistema operacional para compartilhar qualquer coisa com qualquer aplicativo instalado no celular.

Entre as funcionalidades menos detectadas, em geral, observam-se funcionalidades mais limitadas que se referem a algo específico, como as funcionalidades de medição, por exemplo, medir proximidade, nível de luz, velocidade angular, temperatura, pressão do ar, campo magnético, umidade relativa do ar, etc. (Figura 1). A implementação dessas funcionalidades só é possível pois o App Inventor possui componentes específicos para isso que trocam mensagens com componentes de hardware do celular, um exemplo é o acelerômetro. Além disso, seu funcionamento depende de o celular ter suporte para a funcionalidade em questão, por exemplo, um celular sem barômetro não permitirá o uso da funcionalidade de medir pressão do ar pois depende exclusivamente desse componente, que está presente somente em alguns celulares.

5.2 Qual a prevalência da combinação de funcionalidades em projetos App Inventor?

A análise da combinação das funcionalidades considera a definição matemática de combinação sem repetição. Assim, são considerados todos os possíveis grupos de combinação de funcionalidades distintas que podem ser formados de 1 até 42 (total de funcionalidades extraídas). No total, foram detectadas 6.905 combinações (Tabela 4).

Tabela 4: Exemplos de regras de extração de funcionalidades

Frequência de combinação de funcionalidades nos projetos	Quantidade de combinações
Mais de 10%	1
Entre 5% e 10%	1
Entre 1% e 5%	16
Entre 0,5% e 1%	16
Entre 0,01% e 0,5%	73
Acima de 0% e até 0,01%	6.798
Total	6.905

Observa-se que poucas combinações são predominantes, sendo que apenas 18 combinações aparecem em mais de 1% dos projetos (Tabela 4). Além disso, as combinações predominantes (>1% dos projetos) são compostas por 1 a 5 funcionalidades (Figura 5). A maior parte das combinações aparece em poucos projetos, haja vista foram detectadas 4.432 combinações compostas desde 2

a 29 funcionalidades que aparecem somente em um projeto, isto é, 4.432 aplicativos com combinações únicas que só aparecem neles, resultando em uma baixa representatividade dentre os 98.770 aplicativos.

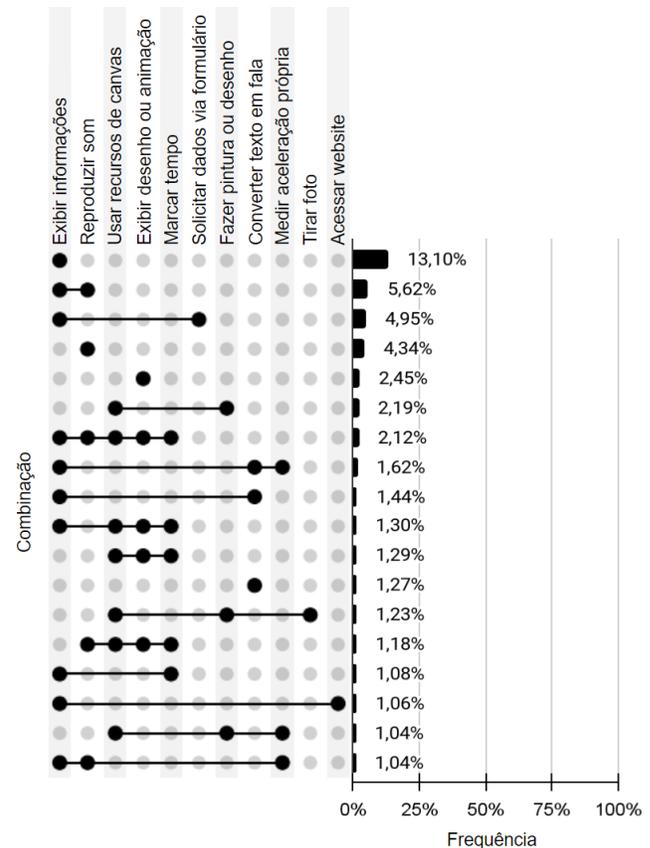


Figura 5: Frequência das combinações de funcionalidades que aparecem em mais de 1% dos 98.770 aplicativos

Combinações que aparecem em mais de 1% dos aplicativos compostas por uma funcionalidade. Essas incluem a combinação com maior frequência, que é composta apenas pela funcionalidade de exibir informações e aparece em 13,10% dos projetos (Figura 5). Esse resultado está consistente com a frequência dessa funcionalidade em si, que está presente em 69,54% dos projetos, número que a inclui sozinha ou em conjunto com outras funcionalidades (Figura 1). A combinação composta por projetos que possuem apenas reprodução de som é representada por 4,34% (Figura 5). Geralmente isso engloba projetos do tipo *soundboards* com um ou mais botões que tocam algum som, como o tutorial *Hello Codi* [20], ou projetos simples que usam sinais sonoros para indicar alguma ação dentro do aplicativo. Projetos com apenas a funcionalidade de exibir desenho ou animação representam 2,45% dos aplicativos (Figura 5) e estão relacionados a animações de *sprites* se movendo sobre um canvas, como o tutorial *Ball Bounce* [19] ou *sprites* sendo controlados pelo usuário, como em jogos. Projetos com a combinação composta apenas pela funcionalidade de converter

texto em fala aparecem em 1,27% dos aplicativos e, normalmente, são inspirados ou idênticos ao tutorial *Talk to Me* [19].

Combinações que aparecem em mais de 1% dos aplicativos compostas por duas funcionalidades. A combinação mais frequente de duas funcionalidades é a de exibir informações e reproduzir som, aparecendo em 5,62% dos projetos (Figura 5). Tipicamente, esses projetos são um pouco mais trabalhados do que apenas uma cópia dos tutoriais disponibilizados no site do App Inventor [19]. Isso inclui também a combinação composta pelas funcionalidades de exibir informações e solicitar dados via formulário representando 4,95% dos projetos (Figura 5). Essa combinação normalmente engloba projetos que fazem alguma ação a partir dos dados informados pelo usuário por meio de *TextBox*, por exemplo, calculando o IMC a partir do peso e altura informados. A combinação composta pela funcionalidade de usar recursos de canvas e fazer pintura ou desenho representa 2,19% dos projetos (Figura 5). Esses projetos são muito parecidos com o clássico software *Paint*, isto é, contam com uma área para desenho e, às vezes, alguns botões para selecionar cores, formas geométricas como quadrados, retângulos, etc., ou apagar o desenho. A combinação composta pelas funcionalidades de converter texto em fala e exibir informações são geralmente bastante parecidas ao tutorial *Talk to Me* [19] com a inclusão de mais informações na tela e representa 1,44% dos projetos. A combinação de exibir informações e marcar tempo representa 1,08% dos projetos e inclui aplicativos com cronômetros para diversas atividades, como a sincronização de eventos da interface, por exemplo, habilitando um botão após alguns segundos. A combinação de acessar website e exibir informações aparece em 1,06% dos projetos (Figura 5) e em geral incluem aplicativos bastante simples sem uso extensivo de blocos de programação mais complexos.

Combinações que aparecem em mais de 1% dos aplicativos compostas por três funcionalidades. Foram detectadas cinco combinações com cardinalidade três em mais de 1% dos aplicativos (Figura 5). Três delas incluem medir aceleração própria envolvendo uma ação do aplicativo a partir de uma mudança da orientação do dispositivo (Figura 5) para, por exemplo, exibir informações. A funcionalidade de usar recursos do canvas também aparece três vezes nessas combinações, sendo que a combinação das funcionalidades exibir desenho ou animação, usar recursos do canvas e marcar tempo aparece em 1,29% dos projetos (Figura 5) e, em geral, está bastante relacionada a jogos parecidos com o jogo desenvolvido no tutorial *Space Invaders* [21]. A combinação das funcionalidades de usar recursos de canvas, medir aceleração própria e fazer pintura ou desenho aparece em 1,04% dos aplicativos (Figura 5) e inclui projetos com uma área para desenho e ações por meio de gestos, por exemplo, apagamento do desenho ao chacoalhar o celular mudando sua aceleração própria detectada via acelerômetro.

Combinações que aparecem em mais de 1% dos aplicativos compostas por quatro funcionalidades. Apenas duas combinações são compostas por quatro funcionalidades e aparecem em de 1% dos aplicativos (Figura 5). Ambas as combinações incluem as funcionalidades de exibir desenho ou animação, usar recursos de canvas e marcar tempo. Essas três

funcionalidades juntas estão relacionadas a jogos, podendo ter mais informações na tela quando se inclui a funcionalidade de exibir informações, como em jogos do tipo *Mini Golf* [22], nos quais a informação refere-se a instruções do jogo. Alternativamente, podem possuir efeitos sonoros quando se inclui a funcionalidade de reproduzir som, e, nesse caso, em geral, essa combinação está relacionada a jogos do tipo *Mole Mash*, inclusive de acordo com o próprio tutorial disponível no site do App Inventor [23].

Combinações que aparecem em mais de 1% dos aplicativos compostas por cinco funcionalidades. Apenas a combinação formada pelas funcionalidades de exibir desenho ou animação, usar recursos de canvas, exibir informações, reproduzir som e marcar tempo aparece em mais de 1% dos projetos. Em geral, são jogos similares ou derivados do tutorial *Mole Mash* [23] com algumas modificações.

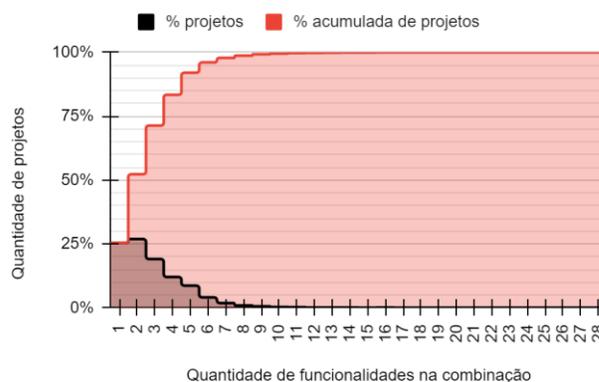


Figura 6: Frequência de projetos por cardinalidade da combinação de funcionalidades.

Observa-se que mais de 50% dos projetos tem até duas funcionalidades e mais de 75% tem até 4 funcionalidades (Figura 6). Menos de 1% dos projetos tem mais de 8 funcionalidades. Considerando que, usualmente, aplicativos têm algum objetivo específico, esse número pequeno de funcionalidades por aplicativo é esperado, haja vista um projeto com muitas funcionalidades pode não ter um foco claro. Cabe ressaltar que não foi identificado nenhum projeto com mais de 28 funcionalidades (Figura 6).

6 DICUSSÃO

Em geral, os resultados deste estudo indicam que existem diferentes processos de gênese dos projetos que são implementados. Um grupo de projetos se refere àqueles que são criados seguindo um tutorial. Nesse contexto, existem diversos tutoriais disponibilizados não só no próprio site do App Inventor, como também em outros sites educacionais [26] e no Youtube. No contexto educacional, esses tutoriais podem ser utilizados com o objetivo de criar um aplicativo idêntico ao tutorial, especialmente como uma forma de inicialização do aprendiz no ambiente App Inventor. Assim a predominância de certas funcionalidades, por exemplo, tocar som, pode ser explicada pela adoção frequente desse tipo de tutorial.

Durante o processo de criação do aplicativo, inclusive a partir de um tutorial, podem surgir ideias do que poderia ser feito diferente. Assim, os aprendizes podem fazer algumas modificações nas funcionalidades existentes ou adicionar novas funcionalidades, por exemplo, modificar o comportamento do *sprite* foguete do jogo *Space Invaders* [21] para fugir em vez de atacar a nave alienígena, ou colocar um efeito sonoro em um jogo *Mini Golf* [22] quando a bolinha colide com algum objeto. Desta forma, esses projetos mantêm um foco em funcionalidades frequentes originadas dos tutoriais, mas acrescentam, uma nova funcionalidade na combinação total de um aplicativo.

Por outro lado, observou-se 4.432 projetos originais que possuem combinações únicas, isto é, combinações que só aparecem neles dentro do universo de 98.770 aplicativos. Esse tipo de projeto, em geral, é caracterizado por construções de blocos de programação mais complexos, uma indicação de que são criados por aprendizes mais experientes em programação e conhecimento maior sobre o ambiente App Inventor. Embora algumas pesquisas apontem viés na percepção da criatividade na área da computação [27][28], é possível afirmar, com base nessas evidências, que o App Inventor pode estimular a criatividade e criação de artefatos computacionais originais.

Dentre as funcionalidades e combinações mais utilizadas, observou-se projetos provenientes ou inspirados em tutoriais, além de funcionalidades relevantes no contexto de aplicativos móveis, como a de exibir informações. O fato de que algumas funcionalidades pouco aparecem pode ser devido a vários fatores. Uma razão pode ser um nicho muito específico, por exemplo, medir umidade relativa do ar e exibir essa informação na tela. No entanto, pode ser também porque ainda não se pensou em ideias em que essa funcionalidade pode ser usada como intermediária para prover alguma informação mais trabalhada, por exemplo, para indicar a sensação térmica de um local. Outra razão pode ser porque algumas funcionalidades menos usadas dependem de que o celular possua componentes específicos, como o acelerômetro.

A influência dos tutoriais nos projetos da Galeria AI é inegável, especialmente em relação a jogos e suas funcionalidades. Considerando que a maioria das crianças e jovens gosta de jogos, muitos aprendizes têm interesse no desenvolvimento de jogos com suas próprias regras, pois traz consigo os elementos de diversão e personalização [29]. Nesse sentido, o ambiente App Inventor reconhece essa motivação intrínseca e possui diversos componentes que facilitam a criação de jogos simples a complexos de forma intuitiva. Esses componentes permitem que o aprendiz se preocupe com a programação do comportamento de personagens e abstraia questões de baixo nível, no entanto, sem esconder conceitos importantes.

As implicações desta pesquisa para a comunidade de educação em computação incluem a inferência de lacunas de materiais didáticos que poderiam ser desenvolvidos explorando as potencialidades do App Inventor em relação às funcionalidades e combinações que pouco aparecem. A quantidade expressiva de certas funcionalidades dá uma primeira indicação de que existe um suporte didático para isso, como os tutoriais, e, possivelmente, sejam mais fáceis de se implementar. Considerando que em um design instrucional o ideal é começar por construções mais

simples e ir aumentando a complexidade. Esses resultados também podem dar indicações da melhor forma de se apresentar esse ambiente para iniciantes, por exemplo, ensinando os aprendizes primeiro a implementar funcionalidades mais comuns e depois as mais raras, cujo nível de detalhamento ou disponibilidade de material didático seja menor.

Em relação à pesquisa, esses resultados podem ser usados para construção de modelos de avaliação ou triangulação com outros dados extraídos dos projetos. Por exemplo, a triangulação com os tipos de comandos de programação utilizados, permitindo um panorama da complexidade dos aplicativos considerando sua respectiva combinação de funcionalidades. Os resultados podem auxiliar no desenvolvimento de um detector de jogos de projetos App Inventor, especialmente, considerando que se identificou a partir desta pesquisa que a combinação de certas funcionalidades está relacionada a jogos e sua representatividade é expressiva. Além disso, esses resultados podem auxiliar na detecção da originalidade de aplicativos, por exemplo, detectar aplicativos baseados em tutoriais a partir das funcionalidades de tutoriais existentes, separando-os de aplicativos cuja combinação de funcionalidades é inédita ou rara.

Os resultados podem também ser utilizados para reforçar a criação de tutoriais e materiais didáticos voltados a funcionalidades desejadas para os alunos aprenderem, mas atualmente sub-representadas.

6.1 Limitações

Este estudo está sujeito a algumas limitações. Para minimizar seu impacto foram aplicadas estratégias de mitigação. Em relação aos riscos relacionados à generalização desses resultados foi extraída uma amostra de quase 50% dos aplicativos publicamente disponíveis da Galeria AI, incluindo aplicativos mais antigos, criados em 2015, a aplicativos mais novos, criados no corrente ano. Também foram incluídos aplicativos publicamente disponíveis na web, buscando dar um retrato atualizado do uso do ambiente. As funcionalidades extraídas foram definidas sistematicamente com base em regras de inferência a partir do código. Assim, funcionalidades que precisam de outros recursos para serem identificadas, como exibir informações por meio de imagens, não são consideradas no estudo, pois a regra de extração considera apenas o arquivo AIA e não as capturas de tela dos aplicativos. A definição de 42 regras foi construída de forma iterativa e incremental, buscando abranger o máximo de funcionalidades extraíveis via código. A detecção das funcionalidades foi realizada automaticamente usando uma ferramenta automatizada buscando demover erros na aplicação das regras. Este estudo foi realizado considerando apenas aplicativos desenvolvidos utilizando o ambiente App Inventor. No entanto, devido ao App Inventor ser diferente das tradicionais linguagens de programação e ter sido desenvolvido especificamente para um público-alvo mais jovem, incluindo crianças e adolescentes, considera-se que os resultados desta pesquisa são relevantes para a comunidade de computação na Educação Básica. Especialmente considerando que as linguagens clássicas de programação tipicamente usadas no Ensino Superior não são adequadas para a Educação Básica devido ao baixo nível de abstração.

7 CONSIDERAÇÕES FINAIS

Esta pesquisa apresenta os resultados de um estudo em larga escala sobre as funcionalidades de 100.000 projetos App Inventor, dentre os quais 98.770 apresentaram no mínimo uma funcionalidade. Os resultados demonstram que algumas funcionalidades aparecem substancialmente e, por outro lado, outras são extremamente raras e foram detectadas em apenas 0,01% dos aplicativos. Observou-se que a detecção de certas combinações de funcionalidades é idêntica ou inspirada em tutoriais e materiais didáticos publicamente disponíveis. Além disso, uma parte dos projetos é bastante original, contendo combinações únicas dentro do universo dos projetos analisados. Os resultados desta pesquisa podem ser usados como apoio no desenvolvimento de materiais didáticos adaptados às características específicas do App Inventor e preenchendo lacunas considerando os materiais atualmente existentes. Além disso, podem ser usados como apoio na tomada de decisão de sequenciamento de material didático, por exemplo, definindo uma trajetória de aprendizagem iniciando com o desenvolvimento de aplicativos com funcionalidades mais simples indo em direção a combinações mais complexas. Os resultados podem ser utilizados também como base para a avaliação de aplicativos em termos da sua originalidade, contribuindo para avaliação do desenvolvimento de criatividade no contexto do ensino de computação.

AGRADECIMENTOS

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

REFERÊNCIAS

- [1] Conselho Nacional de Educação, Ministério da Educação. 2022. Normas sobre Computação na Educação Básica – Complemento à BNCC. Disponível em: http://portal.mec.gov.br/index.php?option=com_docman&view=download&alias=182481-texto-referencia-normas-sobre-computacao-na-educacao-basica&category_slug=abril-2021-pdf&Itemid=30192. Acessado em 14 set. 2022.
- [2] CNE/MEC. Conselho Nacional de Educação, Ministério da Educação. 2022. Competências e premissas específicas da computação na BNCC. Disponível em: http://portal.mec.gov.br/index.php?option=com_docman&view=download&alias=233371-documentos-consolidados-comp-bncc-xlsx&category_slug=janeiro-2022-pdf&Itemid=30192. Acessado em 14 set. 2022.
- [3] MIT App Inventor. 2022. App Inventor. Disponível em: <http://ai2.appinventor.mit.edu>. Acessado em 14 set. 2022.
- [4] MIT App Inventor. 2022. App Inventor - About Us. Disponível em: <https://appinventor.mit.edu/about-us>. Acessado em 14 set. 2022.
- [5] MIT Galeria AI. 2022. MIT App Inventor Gallery. Disponível em: <https://gallery.appinventor.mit.edu/>. Acessado em 14 set. 2022.
- [6] B. Xie, H. Abelson. 2016. Skill progression in MIT App Inventor. In IEEE Symposium on Visual Languages and Human-Centric Computing, Cambridge, pp. 213–217. DOI: <http://doi.org/10.1109/VLHCC.2016.7739687>
- [7] S. C. Kong, H. Abelson. 2019. Computational thinking education. Springer Nature. DOI: [10.1007/978-981-13-6528-7](https://doi.org/10.1007/978-981-13-6528-7)
- [8] N. da C. Alves, C. Gresse von Wangenheim, J. C. R. Hauck, A. F. Borgatto. 2021. An Item Response Theory Analysis of Algorithms and Programming Concepts in App Inventor Projects. In: Anais do Simpósio Brasileiro de Educação em Computação, Jataí, Goiás, 2021. DOI: [10.5753/educomp.2021.14466](https://doi.org/10.5753/educomp.2021.14466)
- [9] W. Martin, F. Sarro, Y. Jia, Y. Zhang, & M. Harman. 2016. A survey of app store analysis for software engineering. IEEE transactions on software engineering, 43(9), 817–847. DOI: [10.1109/TSE.2016.2630689](https://doi.org/10.1109/TSE.2016.2630689)
- [10] I. Li, F. Turbak, & E. Mustafaraj. 2017. Calls of the wild: Exploring procedural abstraction in app inventor. 2017 IEEE Blocks and Beyond Workshop (B&B). DOI: [10.1109/blocks.2017.8120417](https://doi.org/10.1109/blocks.2017.8120417)
- [11] B. Xie, I. Shabir, & H. Abelson. 2015. Measuring the usability and capability of app inventor to create mobile applications. In Proceedings of the 3rd International Workshop on Programming for Mobile and Touch (pp. 1-8). DOI: [10.1145/2824823.2824824](https://doi.org/10.1145/2824823.2824824)
- [12] Y. Park, & Y. Shin. 2019. Comparing the effectiveness of scratch and app inventor with regard to learning computational thinking concepts. Electronics, 8(11), 1269. DOI: [10.3390/electronics8111269](https://doi.org/10.3390/electronics8111269)
- [13] V. R. Basili, G. Caldiera, & H. D. Rombach. 1994. The goal question metric approach. In Encyclopedia of Software Engineering, John Wiley & Sons.
- [14] MIT. 2022. App of the Month. Disponível em: <https://appinventor.mit.edu/explore/app-month-gallery>. Acessado em 03 out. 2022.
- [15] N. da C. Alves, C. G. von Wangenheim, C. G., & J. C. R. Hauck. 2020. Teaching Programming to Novices: A Large-scale Analysis of App Inventor Projects. In 2020 XV Conferencia Latinoamericana de Tecnologias de Aprendizaje (LACLO) (pp. 1-10). IEEE. DOI: [10.1109/LACLO50806.2020.9381172](https://doi.org/10.1109/LACLO50806.2020.9381172)
- [16] E. W. Patton, A. Seo, & F. Turbak. 2019. Enhancing Abstraction in App Inventor with Generic Event Handlers. In 2019 IEEE Blocks and Beyond Workshop (B&B) (pp. 63-71). IEEE. DOI: [10.1109/BB48857.2019.8941193](https://doi.org/10.1109/BB48857.2019.8941193)
- [17] M. Svanberg. 2017. Using feature vector representations to identify similar projects in App Inventor. In 2017 IEEE Blocks and Beyond Workshop (B&B) (pp. 117-118). IEEE. DOI: [10.1109/BLOCKS.2017.8120430](https://doi.org/10.1109/BLOCKS.2017.8120430)
- [18] E. Mustafaraj, F. Turbak, & M. Svanberg. 2017. Identifying original projects in App Inventor. In The Thirtieth International Flairs Conference. Disponível em: <http://cs.wellesley.edu/~tinkerblocks/flairs-2017.pdf>. Acessado em 19 set. 2022.
- [19] MIT App Inventor. 2022. Beginner Tutorials. Disponível em: <https://appinventor.mit.edu/explore/ai2/beginner-videos>. Acessado em 19 set. 2022.
- [20] MIT App Inventor. 2022. Beginner Tutorials – Hello Codi. Disponível em: <http://appinventor.mit.edu/explore/ai2/hello-codi.html>. Acessado em 19 set. 2022.
- [21] MIT App Inventor. 2022. Tutorials – Space Invaders. Disponível em: <https://appinventor.mit.edu/explore/ai2/space-invaders>. Acessado em 21 set. 2022.
- [22] MIT App Inventor. 2022. Tutorials – Golf. Disponível em: <https://appinventor.mit.edu/explore/ai2/minigolf>. Acessado em 21 set. 2022.
- [23] MIT App Inventor. 2022. Tutorials – Mole Mash. Disponível em: <https://appinventor.mit.edu/explore/ai2/molemash>. Acessado em 21 set. 2022.
- [24] MIT App Inventor. 2022. App of the Month Winners. Disponível em: <https://appinventor.mit.edu/explore/app-month-gallery>. Acessado em 26 set. 2022.
- [25] UFSC. Aplicativos de Engenharia de Usabilidade. Disponível em: <https://www.gqs.ufsc.br/usability-engineering/disciplina-ine5624-08208-engenharia-de-usabilidade-cco/>. Acessado em 26 set. 2022.
- [26] UFSC. Iniciativa Computação na Escola – Cursos. Disponível em: <https://cursos.computacaonaescola.ufsc.br/>. Acessado em 27 set. 2022.
- [27] N. da C. Alves, C. Gresse von Wangenheim, L. Martins-Pacheco, e A. Ferreti Borgatto. 2022. Artefatos computacionais são considerados criativos? In Anais do II Simpósio Brasileiro de Educação em Computação, online, Brasil. SBC, Porto Alegre, Brasil, 01-09. DOI: [10.5753/educomp.2022.19193](https://doi.org/10.5753/educomp.2022.19193).
- [28] C. Tang, J. Baer, & J. C. Kaufman. 2015. Implicit theories of creativity in computer science in the United States and China. The Journal of Creative Behavior, 49(2), 137–156. DOI: [10.1002/jobc.61](https://doi.org/10.1002/jobc.61)
- [29] Y. Kafai. 2001. The educational potential of electronic games: From games-to-teach to games-to-learn. Playing By the Rules, Cultural Policy Center, University of Chicago, Chicago, IL, 2001. [Google Scholar](https://scholar.google.com/citations?user=Kafai)
- [30] F. Hayes-Roth. 1985. Rule-based systems. *Communications of the ACM*, 28(9), 921–932. DOI: [10.1145/4284.4286](https://doi.org/10.1145/4284.4286)
- [31] MEC. 2022. DESPACHO DE 30 DE SETEMBRO DE 2022. Homologação das normas sobre Computação na Educação Básica - Complemento à Base Nacional Comum Curricular – BNCC. Disponível em: <https://www.in.gov.br/en/web/dou/-/despacho-de-30-de-setembro-de-2022-433295144>. Acessado em 03 out. 2022.
- [32] C. Gresse von Wangenheim, J. C. R. Hauck, M. F. Demetrio, R. Pelle, N. da C. Alves, H. Barbosa, L. F. Azevedo. 2018. CodeMaster – Automatic Assessment and Grading of App Inventor and Snap! Programs. *Informatics in Education*, 17(1), 117–150. DOI: [10.15388/infedu.2018.08](https://doi.org/10.15388/infedu.2018.08)
- [33] J. J. Garrett. 2011. The Elements of User Experience: User-Centered Design for the Web and Beyond. Berkeley: New Riders.
- [34] M. N. F. Ferreira, C. Gresse von Wangenheim, N. da C. Alves. 2021. Desenvolvimento de um Curso on-line para Ensinar Design de Interface de Usuário na Educação Básica. Anais do Workshop on Human-Computer Interaction Education (WEIHC), online, Brasil. DOI: [10.5753/ihc.2021.19589](https://doi.org/10.5753/ihc.2021.19589)
- [35] J. Gray, H. Abelson, D. Wolber, M. Friend. 2012. Teaching CS principles with App Inventor. In Proceedings of the 50th Annual Southeast Regional Conference (pp. 405–406). ACM. DOI: [10.1145/2184512.2184628](https://doi.org/10.1145/2184512.2184628)

8 APÊNDICE

A tabela 5 apresenta a lista completa de todas as funcionalidades e suas respectivas regras de extração.

Tabela 5: Funcionalidades extraídas

Funcionalidade	Regra de extração
Acessar lista de contatos	Bloco contém ContactPicker OU Bloco contém EmailPicker OU Bloco contém PhoneNumberPicker
Acessar website	Bloco contém WebViewer OU Bloco contém Web OU text Bloco contém "http://www."
Compartilhar artefato	Bloco contém Sharing OU Bloco contém Texting OU Bloco contém Twitter
Contar passos	Bloco contém Pedometer
Converter fala em texto	Bloco contém SpeechRecognizer
Converter texto em fala	Bloco contém TextToSpeech
Criar registro de dados	Bloco contém TextBox >= 3 E (Bloco contém File OU Bloco contém TinyDB OU Bloco contém FirebaseDB OU Bloco contém TinyWebDB OU Bloco contém CloudDB)
Detectar geolocalização	Bloco contém LocationSensor
Escanear QR code	Bloco contém BarcodeScanner
Escolher imagem da galeria	Bloco contém ImagePicker
Exibir desenho ou animação	Bloco contém ImageSprite OU Bloco contém Ball
Exibir informações	Bloco contém Label com caracteres >= 20 OU Bloco contém TextBox com caracteres >= 20
Exibir informações em lista	Bloco contém ListView
Exibir mapa	Bloco contém Map OU (Bloco contém "http://www." E Bloco contém "maps")
Exibir ou esconder imagem	Bloco contém Image_set_Visible
Fazer calculos aritméticos	Bloco contém math_add E Bloco contém math_subtract E Bloco contém math_multiply E Bloco contém math_division E Bloco contém math_number E Bloco contém Label E Bloco contém Button
Fazer ligação	Bloco contém PhoneCall
Fazer login	(Bloco contém File OU Bloco contém TinyDB OU Bloco contém FirebaseDB OU Bloco contém TinyWebDB OU Bloco contém CloudDB) E Bloco contém PasswordTextBox
Fazer pintura ou desenho	Bloco contém Canvas_Clear OU Bloco contém Canvas_DrawArc OU Bloco contém Canvas_DrawCircle OU Bloco contém Canvas_DrawLine OU Bloco contém Canvas_DrawPoint OU Bloco contém Canvas_DrawShape OU Bloco contém Canvas_DrawText OU Bloco contém Canvas_DrawTextAtAngle OU Bloco contém Canvas_SetBackgroundPixelColor
Fazer tradução	Bloco contém YandexTranslate
Gravar áudio	Bloco contém SoundRecorder
Gravar vídeo	Bloco contém Camcorder
Marcar posição no mapa	Bloco contém Circle OU Bloco contém FeatureCollection OU Bloco contém LineString OU Bloco contém Marker OU Bloco contém Polygon OU Bloco contém Rectangle
Marcar tempo	Bloco contém Clock
Medir aceleração própria	Bloco contém AccelerometerSensor
Medir campo magnético	Bloco contém MagneticFieldSensor
Medir nível de luz	Bloco contém LightSensor
Medir orientação espacial	Bloco contém OrientationSensor
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
Medir velocidade angular	Bloco contém GyroscopeSensor
Reproduzir som	Bloco contém Sound OU Bloco contém Player
Reproduzir vídeo	Bloco contém VideoPlayer
Salvar dados localmente	Bloco contém File OU Bloco contém TinyDB
Salvar dados na nuvem	Bloco contém FirebaseDB OU Bloco contém TinyWebDB OU Bloco contém CloudDB
Solicitar dados via formulário	Bloco contém TextBox >= 3
Tirar foto	Bloco contém Camera
Usar API de outro app	Bloco contém ActivityStarter
Usar bluetooth	Bloco contém BluetoothClient OU Bloco contém BluetoothServer
Usar recursos de canvas	Bloco contém Canvas