Revista Brasileira de Informática na Educação – RBIE
Brazilian Journal of Computers in Education
(ISSN online: 2317-6121; print: 1414-5685)
http://br-ie.org/pub/index.php/rbie

Submission: 20/04/2019; 1st round notif.: 08/09/2019; New version: 20/09/2019; 2nd round notif.: 28/11/2019
Camera ready: 16/12/2019; Edition review: 15/01/2020; Available online: 16/02/2020; Published16/02/2020

Estado da Prática do Design Visual de Aplicativos Móveis desenvolvidos com App Inventor

State of the Practice of the Visual Design of Mobile Applications developed with App Inventor

Igor da Silva Solecki
Universidade Federal de Santa Catarina
igor.solecki@posgrad.ufsc.br

Karla Aparecida Justen
Universidade Federal de Santa Catarina
karla.justen@grad.ufsc.br

João V. A. Porto
Universidade Federal de Santa Catarina
joao.porto@grad.ufsc.br

Christiane Gresse von Wangenheim
Universidade Federal de Santa Catarina
c.wangenheim@ufsc.br

Jean C. R. Hauck
Universidade Federal de Santa Catarina
jean.hauck@ufsc.br

Adriano F. Borgatto
Universidade Federal de Santa Catarina
adriano.borgatto@ufsc.br

Resumo

O App Inventor é um ambiente de programação baseado em blocos que permite a qualquer usuário final criar aplicativos móveis. O App Inventor é frequentemente usado para ensinar computação na educação básica por meio do desenvolvimento de aplicativos móveis. Embora a aprendizagem de computação com o App Inventor tenha sido investigada sob diferentes pontos de vista, uma questão que permanece é se o design visual da interface de usuário (IU) dos aplicativos desenvolvidos é adequado, sendo um dos principais fatores de sucesso de aplicativos móveis. Assim, este artigo analisa o design visual das interfaces de usuário de 88.861 aplicativos da galeria do App Inventor, quanto à conformidade com guias de estilo por meio da análise automatizada do código fonte, bem como sua estética com base em um survey com 95 participantes avaliando 110 IUs. Os resultados mostram que a maioria das IUs não está em conformidade com diretrizes de design e não possui estética visual. Assim, mostra-se importante abordar não apenas conceitos de programação, mas também o design de IU no ensino de computação por meio do desenvolvimento de aplicativos móveis. Os resultados dessa análise podem direcionar o desenvolvimento e a melhoria de unidades instrucionais e do ambiente do App Inventor, viabilizando assim o desenvolvimento de aplicativos com melhor design de IU e aumentando suas chances de ter sucesso.

Palavras-Chave: *Design Visual; Estética; Design de Interface do Usuário; Educação Básica; App Inventor; Programação Visual.*

Abstract

App Inventor is a blocks-based programming environment that allows any end-user to create mobile applications. It is frequently used in order to teach computing in K-12 through the development of mobile apps. Although learning computing with App Inventor has been investigated from different points of view, a question that remains is whether the visual design of the user interfaces (UI) of the applications developed with App Inventor is adequate, being one of the major success factors of a mobile application. Therefore, this article analyzes the visual design of the user interfaces of 88,861 apps from the App Inventor gallery in relation to its compliance with style guides through an automated source code analysis, as well as its aesthetics based on a survey with 95 participants assessing 110 UIs. The results show that most UIs do not comply to design guidelines and do not have visual aesthetics. Therefore, it seems important to cover not only programming concepts, but also UI design in computing education when developing mobile applications. The results of this analysis can drive the development and improvement of

Cite as: Solecki, I. S., Justen, K. A., Porto, J. V. A., Gresse von Wangenheim, C., Hauck, J. C. R. & Borgatto, A. F. (2020). State of the Art of the Visual Design of Mobile Applications Developed with App Inventor (Estado da Arte do Design Visual de Aplicativos Móveis desenvolvidos com App Inventor). Brazilian Journal of Computers in Education (Revista Brasileira de Informática na Educação - RBIE), 28, 30-47. DOI: 10.5753/RBIE.2020.28.0.30

instructional units and the App Inventor environment, thus contributing to the development of applications with better UI design and increasing their chances of having success.
Keywords: *Visual Design; Aesthetics; User Interface Design; K–12; App Inventor; Visual Programming.*

1 Introdução

A computação permeia diversas áreas de estudo, e entender seus conceitos tem se tornado cada vez mais importante para atuar na sociedade (Wilensky, Brady, & Horn, 2014). Isso tem motivado o ensino da computação a alunos da educação básica no mundo todo (Grover & Pea, 2013a), que tipicamente é realizado ensinando-se algoritmos e programação utilizando linguagens de programação baseadas em blocos (Grover & Pea, 2013) (CSTA, 2016). Nessas linguagens, criam-se programas arrastando e encaixando blocos visuais, o que diminui a carga cognitiva quando comparado ao uso de linguagens de programação baseadas em texto (Lye & Koh, 2014).

Uma das alternativas para o ensino de computação é o desenvolvimento de aplicativos móveis utilizando o App Inventor. O App Inventor (http://ai2.appinventor.mit.edu) é um ambiente de programação baseado em blocos usado para criar aplicativos móveis para dispositivos Android. Ele atualmente possui mais de 400.000 usuários ativos de várias partes do mundo (MIT, 2019). Os aplicativos do App Inventor consistem em componentes visuais (botões, imagens etc.) e não visuais (câmera, sensor de GPS etc.), além de um conjunto de blocos compondo o código que controla esses componentes. Assim, o processo de desenvolvimento de um aplicativo inclui tanto o design de interface de usuário quanto a programação da funcionalidade.

Como resultado do ensino de computação por meio do desenvolvimento de aplicativos móveis com App Inventor, tipicamente espera-se que os alunos desenvolvam aplicativos funcionais. Porém, além da corretude da funcionalidade, a usabilidade também é uma das principais qualidades de software que podem influenciar o sucesso de um aplicativo móvel (Lee, Moon, Kim, & Yi, 2015) (Kumar, Purani, & Viswanathan, 2018). A usabilidade depende, entre outras características, do design visual da interface de usuário (IU) (International Organization for Standardization [ISO], 2011) (Tractinsky, Katz, & Ikar, 2000). O design visual se organiza a partir de meta-princípios (consistência, hierarquia e personalidade), considerando os princípios de contraste, uniformidade, coerência e *layout*, e envolve o uso de elementos como cor, tipografia e imagens (Garrett, 2011; Schlatter & Levinson, 2013). O uso equilibrado desses elementos pode resultar em uma interface bonita e agradável (Rogers, Sharp, & Preece, 2011). Como resultado, a estética da IU é considerada um fator chave que afeta a usabilidade percebida e efetiva da IU, satisfação subjetiva, confiança, credibilidade e preferência (Zen & Vanderdonckt, 2016).

Atualmente, existem poucos guias de estilo voltados ao design de IU de aplicativos Android que tenham sido desenvolvidos e validados sistematicamente. Nesse contexto, destaca-se o guia de estilo Material Design (https://material.io/design). O Material Design sintetiza princípios clássicos de design para auxiliar no desenvolvimento de produtos de software com melhor usabilidade. Complementarmente, as diretrizes *Web Content Accessibility Guidelines* 2.0 (WCAG 2.0) (W3C, 2008) auxiliam no desenvolvimento de conteúdo *web* acessível para pessoas com deficiências, incluindo visão limitada, perda de audição, entre outros. Essas mesmas diretrizes aplicam-se também a conteúdo apresentado em dispositivos móveis (W3C, 2019).

Nesse contexto, surge o questionamento de até que ponto aplicativos desenvolvidos com o App Inventor possuem um bom design visual de IU, em conformidade com diretrizes de guias de estilo e com alto grau de estética visual.

Atualmente já existem pesquisas voltadas ao levantamento do estado da arte e prática de aplicativos desenvolvidos com App Inventor. Porém, essas pesquisas focam na análise de contribuições ao desenvolvimento de habilidades de programação. Okerlund e Turbak (2013)

analisaram características de 270.000 aplicativos criados por 40.000 usuários para entender quão efetivo o App Inventor é para a criação de programas e o aprendizado de programação. Xie, Shabir e Abelson (2015) analisaram 5.228 aplicativos, agrupando-os por funcionalidade, buscando determinar a usabilidade do ambiente do App Inventor em si (mas não dos aplicativos desenvolvidos). Xie e Abelson (2016) analisaram aplicativos de 10.571 usuários explorando a progressão do seu aprendizado. Alves, Gresse von Wangenheim, Hauck, e Borgatto (2019) analisaram o uso de blocos de programação com base em 88.606 aplicativos da galeria do App Inventor focando em conceitos de programação. No entanto, ainda não existem estudos avaliando sistematicamente o design visual de aplicativos criados com o App Inventor.

Assim, este artigo tem o objetivo de apresentar a análise do design visual de aplicativos móveis criados com o App Inventor, apresentando uma visão geral sobre o estado atual do design visual desses aplicativos. Na seção 2 é apresentado o ambiente App Inventor, detalhando especificamente os elementos de design de IU fornecidos. A pesquisa e coleta de dados é definida na seção 3. A seção 4 apresenta a análise dos dados para cada uma das perguntas de análise definidas. Na seção 5 são discutidas as implicações para o design de IUs com o App Inventor e para o ensino de computação, como também as ameaças à validade da presente pesquisa. A conclusão é apresentada na seção 6. Dessa forma, este trabalho contribui: (i) evidenciando a necessidade de unidades instrucionais que abordem também o design visual de IU de aplicativos móveis; (ii) apresentando direções concretas para o aprimoramento do ambiente App Inventor, para que haja maior suporte ao design visual de IU e ao ensino desse conhecimento como parte do ensino de computação.

2 App Inventor

2.1 O ambiente App Inventor

O App Inventor é um ambiente de programação baseado em blocos para criar aplicativos móveis para dispositivos Android. Ele pode ser acessado e utilizado online gratuitamente por meio de um navegador *web*. O App Inventor consiste em um ambiente de programação visual em que se arrastam e posicionam blocos e componentes para desenvolver o aplicativo.

Os aplicativos são criados em dois passos no App Inventor. Inicialmente, componentes da IU, como botões e imagens, são posicionados e configurados no *Component Designer* (Figura 1). Uma visão geral dos componentes de IU do App Inventor é apresentada na Tabela 1. A aparência desses componentes pode ser configurada alterando propriedades como tamanho, cor, fonte, alinhamento, entre outros. Também é possível adicionar componentes não visuais, como câmera, sensores (de localização, acelerômetro etc.), componentes para o armazenamento de dados (em bancos de dados *web* ou locais) e para a comunicação com outros aplicativos.

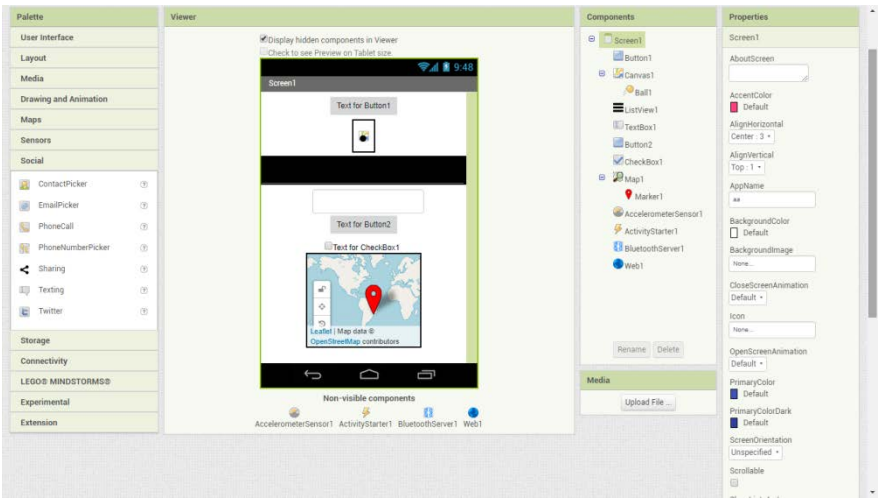


Figura 1: *Component Designer* do App Inventor.

Tabela 1: Componentes de IU do App Inventor.

Componente	Descrição
Botão	Componente capaz de detectar cliques. Quando clicado, dispara alguma ação.
CaixaDeSeleção	Componente capaz de detectar toques do usuário, alterando seu estado booleano em resposta.
EscolheData	Um botão de propósito especial. Quando clicado, exibe um diálogo para o usuário escolher uma data.
Imagem	Componente para exibir imagens.
Legenda	Componente para mostrar texto.
EscolheLista	Um botão de propósito especial. Quando clicado, exibe uma lista de textos e pede para o usuário escolher um elemento da lista.
VisualizadorDeListas	Exibe uma lista de textos como parte da tela.
Notificador	Exibe diálogos de alerta, mensagens e alertas temporários.
CaixaDeSenha	Uma caixa de texto que esconde o texto digitado nela.
Tela	Componente que contém todos os outros componentes.
Deslizador	Uma barra com um pino arrastável. À medida que o usuário arrasta o pino para os lados, a posição é notificada.
ListaSuspensa	Componente que exibe um *popup* com uma lista de elementos para o usuário selecionar um item.
CaixaDeTexto	Componente que permite que o usuário insira texto.
EscolheHora	Um botão de propósito especial. Quando clicado, exibe um diálogo para o usuário escolher uma hora.
NavegadorWeb	Componente para visualizar páginas *web*.

Num segundo passo, o funcionamento do aplicativo é programado conectando blocos visuais. Blocos de programação do tipo *built-in* representam conceitos tradicionais de programação (condicionais, laços, procedimentos etc.). Adicionalmente, existem blocos de programação para elementos do *Component Designer*, que representam eventos, estados e ações de componentes específicos (p. ex., botão pressionado, ativar temporizador) (Mustafaraj, Turbak, & Svanberg, 2017). O funcionamento do aplicativo é definido no Editor de Blocos (Figura 2).

O aplicativo pode ser testado à medida que é desenvolvido por meio do aplicativo App Inventor Companion. O App Inventor Companion executa o aplicativo em desenvolvimento em tempo real em um dispositivo móvel.

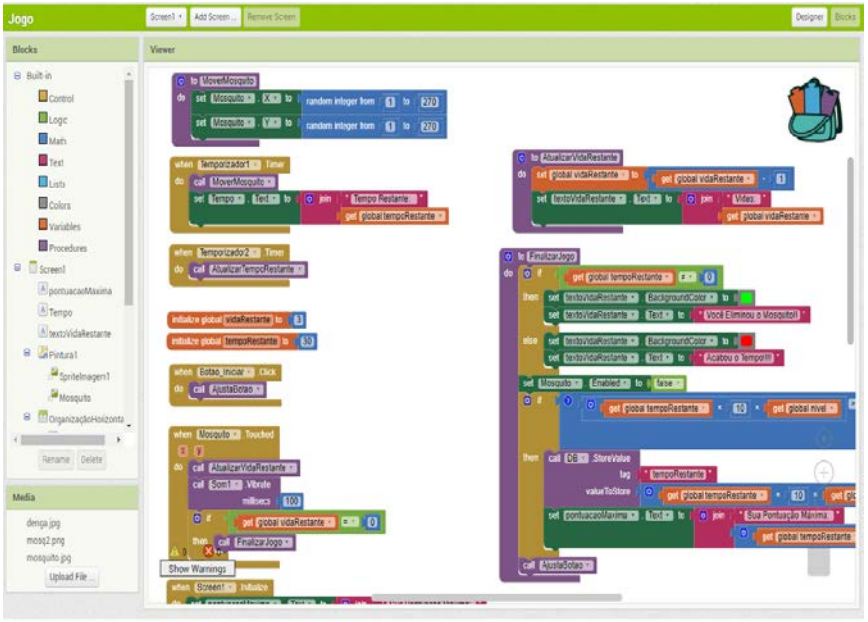


Figura 2: Editor de Blocos do App Inventor.

Os arquivos de código fonte de projetos do App Inventor são salvos automaticamente na nuvem, mas podem também ser exportados como arquivos ".aia". Um arquivo ".aia" é um conjunto de arquivos compactados que inclui um arquivo de propriedades do projeto, arquivos de mídia usados pelo aplicativo e, para cada tela do aplicativo, um arquivo ".bky" e um arquivo ".scm". O arquivo ".bky" contém uma estrutura XML com os blocos de programação usados na lógica do aplicativo, e o arquivo ".scm" contém uma estrutura JSON descrevendo os componentes usados (Mustafaraj et al., 2017).

O App Inventor é um software de código aberto disponível no GitHub (https://github.com), de forma que qualquer pessoa pode baixá-lo e configurá-lo em um servidor próprio (Wolber, Abelson, & Friedman, 2015).

2.2 Ensino de computação com App Inventor

O ensino de computação na educação básica deve abordar vários *conceitos e práticas*, que se referem a competências que alunos computacionalmente fluentes utilizam para se envolver plenamente com a computação. Conforme o *K-12 Computer Science Framework* (CSTA, 2016), essas competências em computação são organizadas em **conceitos fundamentais**, que representam áreas de conteúdo chave na computação, e **práticas fundamentais**, que representam ações que os alunos usam para se envolver com os conceitos de maneira rica e significativa, sendo definidos também **conceitos transversais** relacionados à computação (Figura 3).

Atualmente, a introdução do ensino de computação nas escolas é uma tendência mundial, apoiada por diversas iniciativas voltadas, principalmente, para o ensino da programação por meio de ambientes de programação baseados em blocos (Grover & Pea, 2013a) (Lye & Koh, 2014). O ambiente App Inventor foi criado nesse contexto com o objetivo de permitir a qualquer usuário final criar aplicativos móveis (Patton, Tissenbaum, & Harunani, 2019). Atualmente, mais de oito milhões de pessoas já utilizaram o App Inventor, criando ao todo 34 milhões de aplicativos, dos quais mais de 30.000 foram publicados no Google Play (https://appinventor.mit.edu). O ambiente App Inventor também fornece uma comunidade on-line com o objetivo de possibilitar o compartilhamento de aplicativos criados com o App Inventor, de forma que o trabalho de um usuário pode servir de inspiração para outros, e os usuários podem dar *feedback* uns aos outros, favorecendo a criatividade e ajudando os usuários a aprenderem e crescerem (Chan, 2019). Junto com os aplicativos compartilhados na galeria, são disponibilizados também os seguintes

metadados: o perfil do desenvolvedor do aplicativo (que contém apenas um nome e, opcionalmente, um link para mais informações), a data em que o projeto foi criado e alterado, uma descrição do aplicativo, um link opcional para um vídeo sobre o aplicativo, e créditos a autores de outros aplicativos nos quais o desenvolvedor se baseou.

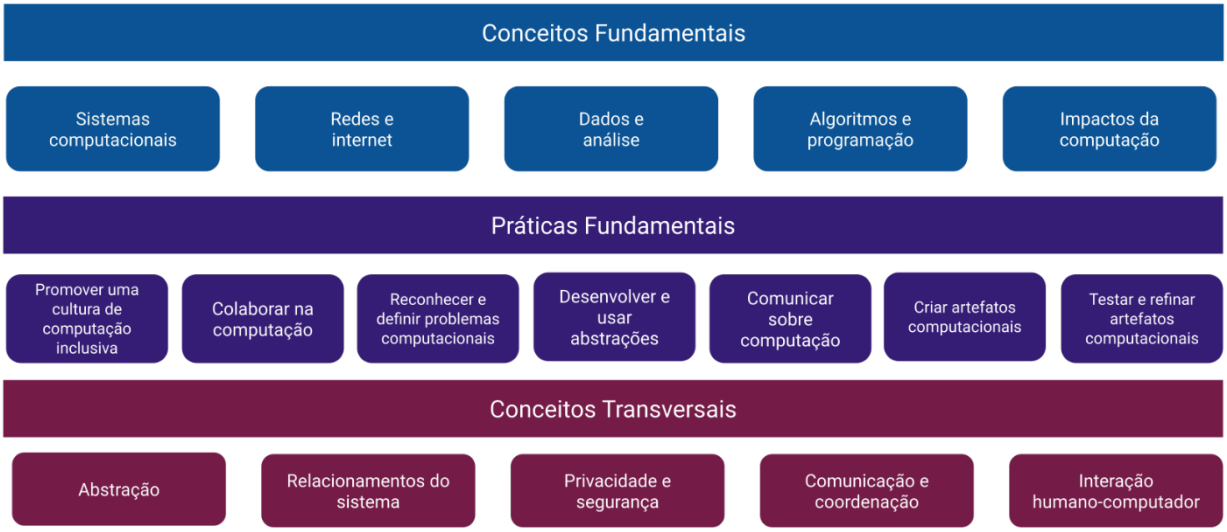


Figura 3: Conceitos e práticas de computação (CSTA, 2016).

O App Inventor foi criado principalmente para suportar o ensino de computação na educação básica e em cursos introdutórios de Ciência da Computação no ensino superior (Wolber et al., 2015). Mesmo focando principalmente no ensino de programação, já existem iniciativas para integrar o ensino de conceitos de design de interface como conceito transversal no ensino de computação com App Inventor (Ferreira, Gresse von Wangenheim, Missfeldt, Pinheiro, & Hauck, 2019) (Ferreira, Pinheiro, Missfeldt, & Gresse von Wangenheim, 2019).

O ensino de design é capaz de promover vários benefícios, desde a aprendizagem de métodos de pesquisa, habilidades de visualizar e de apresentar informações, até o incentivo à análise crítica, colaboração e formação de equipes. Soma-se a isso a vantagem de facilitar o aprendizado de habilidades cognitivas criativas e de habilidades manuais produtivas (AIGA, 2019). O ensino do design pode atuar, então, como incentivo aos alunos, pois os torna mais criativos e imaginativos, e também os ensina a aproveitar e a aplicar essa inventividade, além de colocá-la em prática (AIGA, 2019). Assim, a integração do ensino de design também se enquadra na *ação computacional*, uma visão do ensino de computação que enfatiza que jovens, enquanto aprendem computação, também precisam ter oportunidades de criar algo que tenha um impacto direto em sua vida e comunidade, visto que o ensino de computação na educação básica frequentemente é direcionado por uma ênfase em aprender “fundamentos” da programação (Tissenbaum, Sheldon & Abelson, 2019). Incorporar design de IU no ensino de computação também ajuda a ampliar a percepção de que a computação é mais do que apenas escrever código e, portanto, pode ajudar a aumentar a diversidade de estudantes que possam considerar uma carreira em TI (Ferreira, Gresse von Wangenheim, et al., 2019) (Agapie & Davidson, 2018).

3 Definição da Pesquisa e Coleta de Dados

O objetivo dessa pesquisa é analisar o estado da prática do design visual de aplicativos móveis criados com o App Inventor. Com esse objetivo, são investigadas as seguintes perguntas de análise:

P1. Quais componentes de IU são tipicamente usados em aplicativos criados com o App Inventor, e com que frequência?

P2. Em que medida o design visual de IU dos aplicativos criados com o App Inventor está em conformidade com guias de estilo?

P3. Qual é o grau de estética visual dos aplicativos criados com o App Inventor?

Para a análise dessas perguntas adotou-se uma metodologia multimétodo, realizando uma análise quantitativa de dados de aplicativos da galeria do App Inventor para responder às perguntas 1 e 2 e um *survey* para responder à pergunta 3.

Para obter os dados referentes às perguntas 1 e 2, com o apoio da equipe do MIT App Inventor, foram baixados todos os aplicativos que puderam ser acessados pela galeria do App Inventor em junho de 2018, totalizando 88.861 aplicativos, disponíveis sob a licença Creative Commons. Analisou-se o código de cada aplicativo utilizando o CodeMaster (Gresse von Wangenheim, Hauck, et al., 2018; Justen, 2019; Solecki, 2019), uma ferramenta que avalia de maneira automática o uso de conceitos de programação e o grau de conformidade com diretrizes de design de IU em aplicativos desenvolvidos com App Inventor. Durante a avaliação de cada aplicativo, a ferramenta registra dados sobre os elementos usados, os quais foram utilizados para determinar a frequência de uso dos componentes de design. O grau de conformidade com diretrizes de design é identificado com base em um conjunto de diretrizes básicas que foram extraídas do Material Design (https://material.io/design) e das diretrizes *Web Content Accessibility Guidelines* 2.0 (W3C, 2008) e adaptadas conforme as possibilidades de configuração dos componentes do App Inventor.

Para responder à pergunta 3, foi executado um estudo exploratório avaliando a estética de IUs. Seguindo Tractinsky (2013), consideramos a estética de IUs a beleza ou aparência agradável da interface de aplicativos móveis. Os avaliadores voluntários classificaram de forma subjetiva a estética visual de 110 IUs de aplicativos criados com App Inventor. Esses aplicativos foram selecionados da seguinte maneira: 100 aplicativos foram escolhidos aleatoriamente da galeria do App Inventor, e acrescentaram-se 10 IUs de aplicativos resultantes de unidades instrucionais aplicadas no contexto da Iniciativa Computação na Escola (http://www.computacaonaescola.ufsc.br/). A coleta foi realizada por meio de um questionário online incluindo uma explicação do estudo, pedido de consentimento do participante, questões demográficas e 110 perguntas ordenadas aleatoriamente sobre a estética visual das IUs. Cada uma dessas perguntas apresenta a imagem de uma IU de um aplicativo (239 x 425 *pixels* em *true color* de 24 bits). Como escala de resposta, foi adotada uma escala ordinal de 3 pontos com as alternativas “feia”, “mais ou menos” e “bonita”, adaptando a escala definida por Hassenzahl e Monk (2010) e também utilizada em outras pesquisas de estética visual (Hamborg, Hülsmann & Kaspar, 2014).

Os participantes responderam ao questionário online remotamente, no seu próprio computador. Embora não tenha sido imposto um limite de tempo, em geral, os participantes levaram cerca de 15 minutos respondendo.

O questionário foi divulgado amplamente em dezembro de 2018 entre acadêmicos e servidores da UFSC, e 108 respostas foram obtidas até fevereiro de 2019. Os questionários com respostas incompletas foram desconsiderados do estudo, pois as pontuações calculadas para as IUs (seção 4.3) seriam prejudicadas pelas respostas ausentes. Assim, manteve-se um total de 95 respostas completas.

4 Análise dos dados

Nas subseções seguintes são apresentados os resultados de cada uma das perguntas de análise.

4.1 Quais componentes de IUs são usados em aplicativos criados com o App Inventor, e com que frequência?

Analisando quantos aplicativos, dentre os 88.861 analisados, usam cada componente de IU (Figura 4), observa-se que os componentes mais comuns são *Botão* (usado por 85% dos aplicativos) e *Legenda* (63%), seguidos por *CaixaDeTexto* (31%) e *Imagem* (27%).

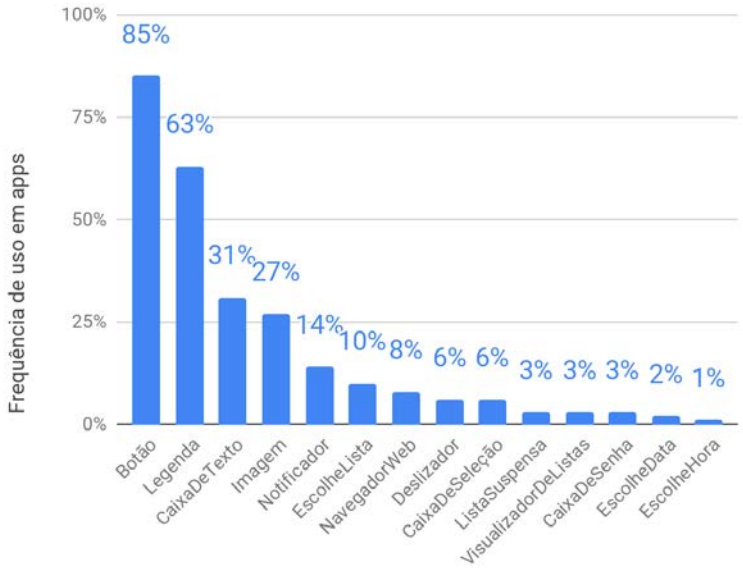


Figura 4: Frequência de aplicativos que contêm cada componente de IU (do total de 88.861 aplicativos).

Observando que imagens podem ser utilizadas também em outros componentes além do componente *Imagem*, por exemplo, para representar botões por meio de imagens, o total de aplicativos que usam imagens (em diversos componentes, não apenas no componente *Imagem*) aumenta para 54% (Tabela 2).

Tabela 2: Frequência de uso de componentes com imagem.

Componentes com imagem	Quantos aplicativos usam
Componente *Imagem*	23.597 (27%)
Botões com imagem	22.188 (25%)
Telas com imagem de fundo	20.510 (23%)
Total	47.671 (54%)

Alguns componentes são usados por uma parcela muito pequena dos aplicativos. Isso possivelmente indica que a função desses componentes é muito específica e que poucos aplicativos necessitam usá-los.

4.2 Em que medida o design visual de IU dos aplicativos criados com o App Inventor está em conformidade com guias de estilo?

Com base nas recomendações do Material Design e das *Web Content Accessibility Guidelines* 2.0 e levando em consideração as possibilidades de design visual fornecidas pelo App Inventor, sintetizou-se um conjunto básico de diretrizes, apresentado na Tabela 3. As diretrizes estão

divididas em categorias de acordo com as seções do Material Design. No restante desta seção, analisa-se a conformidade dos aplicativos por categoria.

Tabela 3: Diretrizes básicas de design visual de aplicativos criados com o App Inventor.

Diretriz	**Descrição**
Layout	
L1 Alinhamento	Devem-se usar organizadores em todas as telas
L2 *Layout* responsivo	O *layout* deve ser dimensionado de maneira responsiva
L3 Dimensões responsivas	Devem-se usar dimensões responsivas
Tipografia	
T1 Família da fonte	Deve-se usar uma família de fonte sem serifa
T2 Tamanho da fonte de botões	O texto de botões deve ter tamanho de fonte 14
T3 Tamanho da fonte de texto	Texto em geral deve ter um dos tamanhos de fonte recomendados
Escrita	
E1 Capitalização	Sentenças devem ter apenas a primeira letra maiúscula
E2 Texto personalizado	Componentes com texto devem usar um texto personalizado, não o padrão atribuído pelo App Inventor (p. ex., “Texto para Botão1”)
E3 Dica	Para toda caixa de texto, deve-se definir uma dica
E4 Uso dois pontos	Não se devem usar dois pontos ao final de legendas
Cores	
C1 Uso de cores	A quantidade de cores usadas deve ser limitada
C2 Contraste	Deve haver contraste suficiente entre texto e fundo
Imagens	
I1 Pixelização	Imagens não devem ficar pixelizadas
I2 Distorção	Imagens não devem ser distorcidas

4.2.1 Layout

Para o alinhamento de elementos no design de interface, o App Inventor fornece componentes chamados de organizadores, que servem para dispor outros componentes linearmente ou em grade de forma alinhada. Porém, analisando os 88.861 aplicativos, observa-se que menos da metade (46%) usa organizadores em todas as telas, enquanto outra parcela um pouco menor (39%) não possui nenhum organizador (Figura 5). Os demais aplicativos (15%) usam organizadores apenas em parte das telas. Isso indica que muitos aplicativos criados com o App Inventor não apresentam os elementos de interface de forma alinhada.

Figura 5: Quantidade de aplicativos por nível de conformidade com a diretriz de alinhamento.

O Material Design contém diretrizes sobre responsividade, que dizem respeito a como a IU se adapta a diferentes tamanhos de tela. O App Inventor disponibiliza uma configuração chamada “dimensionamento”, que pode ter o valor “fixo” ou “responsivo”. Observa-se que apenas 4,8% dos aplicativos têm o dimensionamento “responsivo”. Isso indica que a maioria dos aplicativos criados com o App Inventor não se adapta bem a diferentes tamanhos de tela de dispositivos.

Outra característica relevante para a responsividade é a forma como as dimensões são definidas, por exemplo, altura e largura de botões. Quando elas são especificadas como uma porcentagem da tela (e não como um valor exato), os componentes possivelmente se adaptam melhor a diversos tamanhos de tela. Apenas uma parcela muito pequena (1,8%) sempre define as dimensões em porcentagem.

4.2.2 Tipografia

Segundo os guias de estilo, recomenda-se o uso das famílias de fontes do sistema Android (Roboto ou Noto). Porém, como o App Inventor não fornece essas opções, a alternativa mais próxima é o uso de fontes sem serifa. Portanto, componentes com texto deveriam usar fonte “sem serifa” ou “padrão”, que geralmente corresponde a sem serifa. Analisando os aplicativos que têm esses componentes (ou seja, desconsiderando a parcela de aplicativos em que os componentes avaliados não estão presentes), observou-se que 88% sempre usam uma dessas opções (Figura 6 T1).

Para assegurar a legibilidade, o texto de botões deve ter tamanho de fonte 14, o que é atendido completamente pela maioria (68%) dos aplicativos que têm botões (Figura 6 T2). Quanto a outros componentes com texto, os guias de estilo especificam quais tamanhos devem ser usados conforme a semântica do texto (por exemplo, se é um título ou subtítulo). Devido à dificuldade de identificar a semântica do texto de um componente, analisou-se apenas se os tamanhos usados estão entre as alternativas de tamanho especificadas pelos guias de estilo. Mesmo com esse critério generalizado, mais de um terço (7% + 28%) dos aplicativos que têm esses componentes usam tamanhos de fonte inapropriados (Figura 6 T3).

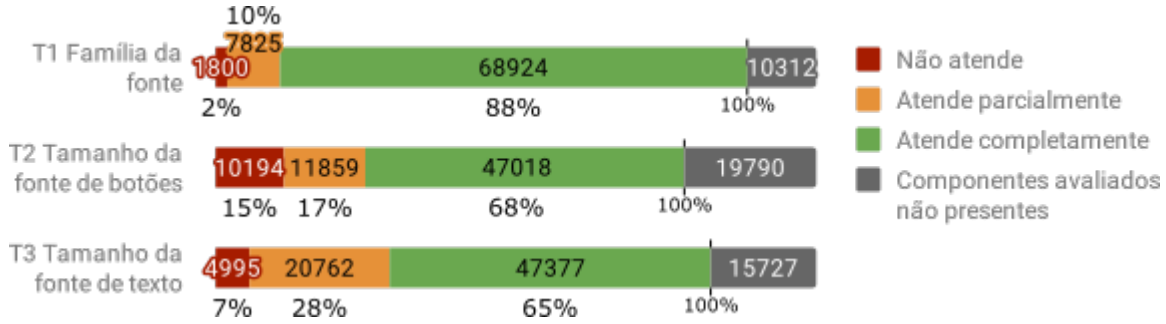


Figura 6: Quantidade de aplicativos por nível de conformidade com diretrizes de tipografia.

4.2.3 Escrita

Analisando o uso de letras maiúsculas em sentenças, observa-se que a maioria (62%) dos aplicativos que contêm sentenças tem apenas parte das sentenças com capitalização correta (Figura 7 E1). No entanto, ao analisar essa diretriz, não são consideradas siglas ou nomes próprios pela dificuldade de detectar essas situações. Portanto, é possível que o número real de aplicativos em total conformidade com essa diretriz seja maior.

Segundo os guias de estilo, todo texto deve ser claro e preciso. Portanto, componentes com texto não devem usar o texto genérico atribuído por padrão pelo App Inventor (p. ex., “Texto para Botão1”), mas um texto personalizado. Entre os aplicativos que têm componentes com texto atribuído por padrão, a maioria (86%) têm todos os textos personalizados (Figura 7 E2). Embora a alteração não garanta o uso de um texto apropriado, é uma evidência mínima da personalização do texto.

Toda caixa de texto pode possuir uma dica, que é um texto indicando como ela deve ser preenchida. Entre os aplicativos que têm caixas de texto, apenas 28% possuem dicas definidas em todas as caixas de texto (Figura 7 E3).

Para assegurar uma estética minimalista, os guias de estilo recomendam que não se usem dois pontos ao final de legendas. Entre os aplicativos que possuem legendas, 76% sempre evitam o uso de dois pontos (Figura 7 E4).

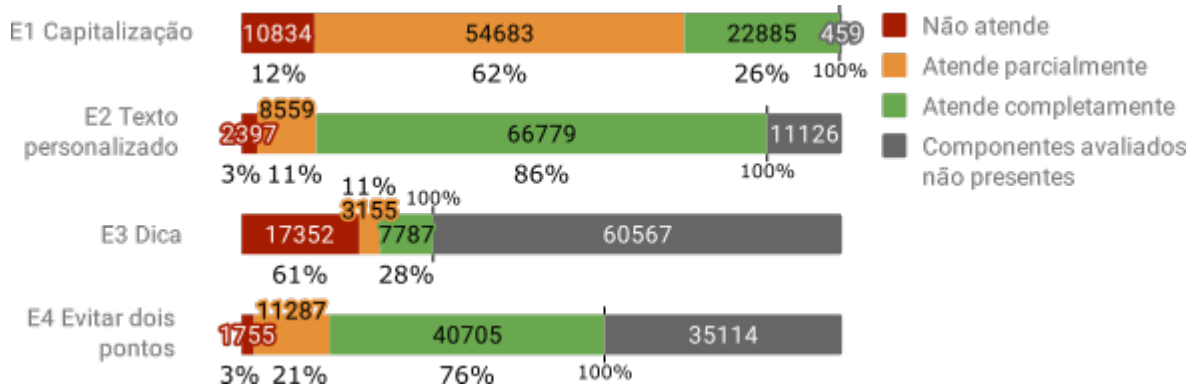


Figura 7: Quantidade de aplicativos por nível de conformidade com diretrizes de escrita.

4.2.4 Cores

Cores ajudam a expressar hierarquia, estabelecer a presença da marca, dar significado e indicar o estado de elementos. Especificamente para aplicativos Android, o Material Design recomenda o uso de uma paleta de cores composta basicamente por uma cor primária e uma cor secundária. Portanto, deve-se evitar em geral o uso de uma quantidade maior de cores. Observa-se que a maior parte dos aplicativos (80%) usa apenas duas ou menos cores (Figura 8). No entanto, grande parte usa as cores da paleta do App Inventor, que não estão entre as cores recomendadas pelo Material Design.

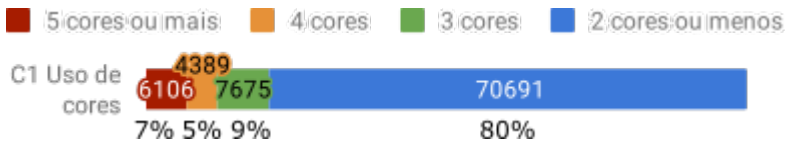


Figura 8: Quantidade de aplicativos por quantidade de cores usadas.

Seguindo os padrões de acessibilidade das WCAG 2.0 quanto ao contraste do texto, há dois níveis de conformidade: nível AA e nível AAA. Um aplicativo satisfaz o nível AA se todo texto possui contraste com a cor de fundo acima do mínimo recomendado pelas WCAG 2.0, e o nível AAA requer um contraste maior que o nível AA. Analisando os 78.623 aplicativos que contêm texto cuja cor pode ser configurada, observa-se que cerca de um terço (31%) possui texto com contraste insuficiente, ou seja, abaixo do mínimo requerido pelo nível AA (Figura 9).

Figura 9: Quantidade de aplicativos por nível de conformidade do contraste.

4.2.5 Imagens

Segundo o Material Design, imagens devem ter resolução suficiente para que não sejam exibidas de forma pixelizada, e definiu-se que isso ocorre quando a largura ou altura é ampliada em 50% ou mais. Além disso, imagens devem ser recortadas quando proporções diferentes da original são necessárias e, portanto, não devem ser distorcidas. Considera-se uma imagem distorcida quando a proporção original (largura/altura) não é mantida, com uma tolerância de 10% (largura/altura ≥ 1.1 ou altura/largura ≥ 1.1).

Dos 45.836 aplicativos com componentes avaliados quanto à pixelização, 45% (22% + 23%) contêm alguma imagem pixelizada, e dos 47.671 aplicativos avaliados quanto à distorção, 67%

(38% + 29%) contêm alguma imagem distorcida[1] (Figura 10). Portanto, a maioria dos aplicativos que usam imagens viola diretrizes de imagens.

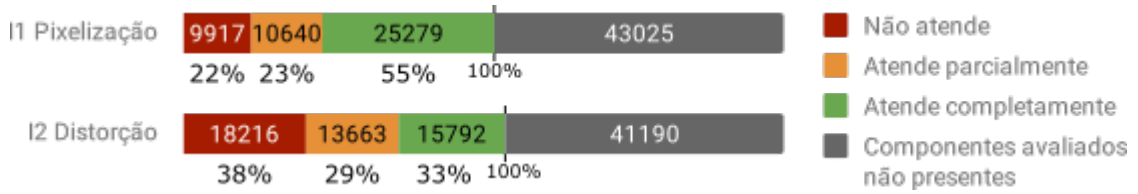


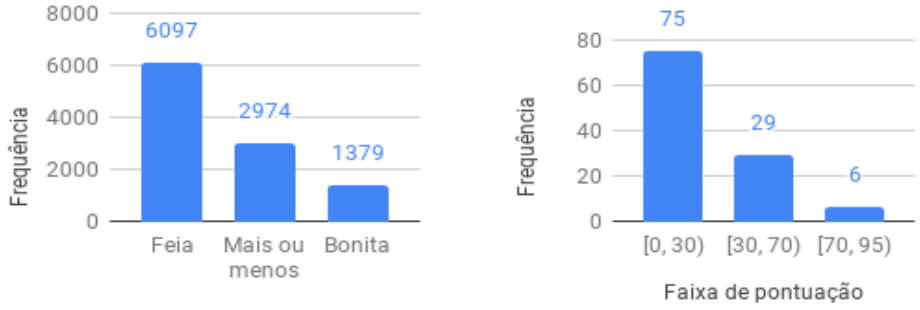
Figura 10: Quantidade de aplicativos por nível de conformidade com diretrizes de imagens.

Assim, as diretrizes com as quais há maior falta de conformidade são as de *layout* (L1, L2, L3), dica (E3), contraste (C2) e imagens (I1, I2). Quanto às diretrizes de tamanho da fonte (T3) e texto personalizado (E2), a falta de conformidade não é tão notável, mas possivelmente seria maior se as diretrizes pudessem ser analisadas de forma mais profunda, considerando a semântica do texto dos componentes.

4.3 Qual é o grau de estética visual dos aplicativos criados com o App Inventor?

Esta pergunta é analisada por meio de um *survey* com 95 participantes[2] avaliando 110 *screenshots* de IUs de aplicativos do App Inventor. Dois terços (65%) dos participantes são do sexo masculino, e a maioria (80%) é de usuários da plataforma Android.

Observa-se que, em geral, os participantes consideraram a maioria das IUs como “feia”, indicando um baixo grau de estética (Figura 11 a).



(a) (b)

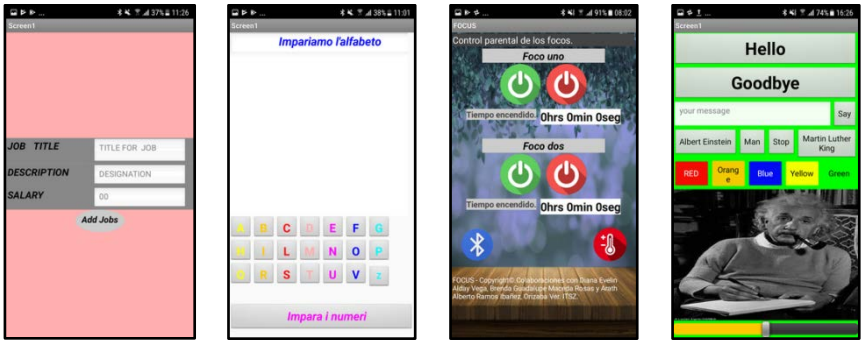
Figura 11: Resultados da avaliação estética: (a) frequência de avaliações por grau de estética e (b) número de IUs por faixa de pontuação.

Apesar da existência de um certo grau de concordância, observou-se que os participantes discordam sobre a estética de um número considerável de IUs de aplicativos Android (Gresse von Wangenheim, Porto, Hauck, & Borgatto, 2018). Portanto, calculou-se uma pontuação de grau de estética entre 0 e 95 para cada IU, associando uma pontuação parcial a cada avaliação (“feia” = 0 ponto; “mais ou menos” = 0,5 ponto; “bonita” = 1 ponto) e somando as pontuações parciais. As IUs foram então classificadas conforme a pontuação nos graus estéticos “feia” ([0, 30) pontos), “mais ou menos” ([30, 70) pontos) e “bonita” ([70, 95] pontos) (Figura 11 b). Para ilustrar os resultados dessa classificação, são apresentados alguns exemplos de IUs na Figura 12.

[1] Para componentes *Imagem* com imagem definida apenas em tempo de execução, pode-se avaliar a distorção (verificando a propriedade RedimensionarParaCaber), mas não a pixelização (que depende da imagem); por isso, o número de aplicativos avaliados difere

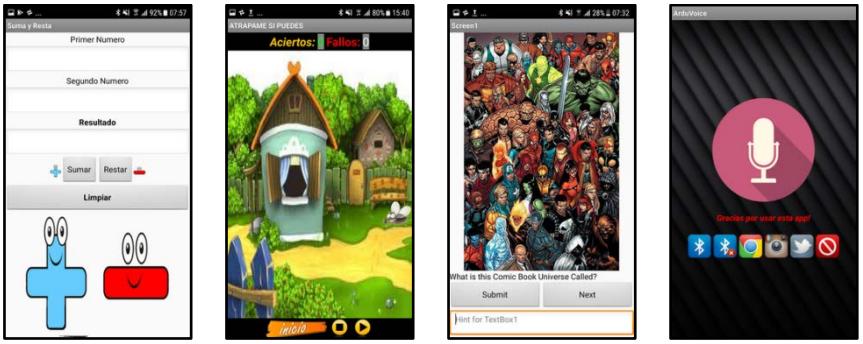
[2] Este estudo foi aprovado pelo Comitê de Ética da Universidade Federal de Santa Catarina (no. 2.903.849)

Exemplos de IUs avaliadas como feias - [0,30) pontos



(a) (b) (c) (d)

Exemplos de IUs avaliadas como mais ou menos - [30,70) pontos



(e) (f) (g) (h)

Exemplos de IUs avaliadas como bonitas - [70,95] pontos

(i) (j) (k) (l)

Figura 12: Exemplos de IUs por faixa de pontuação (grau de estética).

Comparando as IUs de diferentes graus estéticos, notam-se diferenças nos elementos visuais e na forma como esses elementos compõem a IU. As IUs feias possuem cores desarmônicas (*a* e *d*), tipografia e contrastes inadequados (*c*), iconografia mal elaborada (*c*) e *layout* desorganizado (*a* e *b*). As IUs avaliadas como mais ou menos também apresentam problemas, como imagens distorcidas (*e f*), mas de forma menos intensa. Por outro lado, as IUs bonitas apresentam problemas mais sutis, destacando-se um sistema de cores bem-definido, *layout* equilibrado e imagens adequadas, o que corresponde às diretrizes de design exploradas na seção 4.2.

5 Discussão

De forma geral, os resultados da análise indicam um baixo grau de conformidade dos aplicativos da galeria do App Inventor com guias de estilo em diversos aspectos. Isso se reflete na avaliação da estética visual, indicando que em geral as interfaces de aplicativos criados com o App Inventor são percebidas com baixo nível de estética. Entretanto, existem IUs com boa avaliação de estética, o que indica que é possível criar interfaces com um bom design visual com o App Inventor.

5.1 Implicações para o design de IUs com o App Inventor

Visto que a maioria dos aplicativos analisados apresenta um baixo nível de qualidade do design visual, identificou-se como o suporte fornecido pelo App Inventor pode ser melhorado para guiar e facilitar o desenvolvimento de IUs com um design visual melhor (Tabela 4).

Tabela 4: Sugestões de melhoria do suporte do App Inventor ao design visual.

Categoria	Sugestões de melhoria
Layout	Facilitar o agrupamento e alinhamento dos componentes da IU, o que atualmente pode necessitar de combinações complexas de organizadores.
	Incluir suporte a alinhamento usando grades.
	Possibilitar a criação de novas telas a partir de modelos predefinidos, com componentes já posicionados e alinhados (p. ex., modelo de tela de *login*).
	Suportar mais componentes padrões do Material Design.
Tipografia	Oferecer as fontes Roboto e Noto (fontes do sistema Android).
	Permitir que se associe uma semântica ao texto (título, subtítulo etc.) para orientar a escolha do tamanho da fonte.
Cores	Atualizar a paleta de cores com cores mais alinhadas ao Material Design.
	Orientar a escolha de cores harmônicas.
	Gerar uma paleta de cores personalizada com base na escolha de uma cor primária e uma secundária, incluindo variantes claras e escuras.
Imagens	Adicionar mecanismos para recortar as imagens quando necessário.
	Suportar o uso de imagens sem distorção, sempre mantendo a proporção original.
	Incorporar o acesso a catálogos de imagens e ícones do Material Design para aumentar a consistência da qualidade.

Adicionalmente, o próprio ambiente do App Inventor poderia indicar ao usuário possíveis violações das diretrizes enquanto o aplicativo é desenvolvido, auxiliando na prevenção dessas violações no design visual.

Considerando que o App Inventor é um software de código aberto, essas modificações podem ser feitas por qualquer interessado. Inclusive, motivado por essa pesquisa, já se iniciou a evolução de uma instância do App Inventor no contexto da Iniciativa Computação na Escola (http://www.computacaonaescola.ufsc.br/), aprimorando e complementando as seguintes funcionalidades para um suporte melhor ao design visual:

- Menu de cores. Foram substituídas as cores do menu do App Inventor pelas cores principais (valor de saturação 500) e três tonalidades de cada cor da paleta do Material Design (Google, 2019).
- Catálogo de ícones. Foi integrado no App Inventor um catálogo de ícones com base no Material Design (Google, 2019), permitindo a inclusão direta de ícones na interface do aplicativo.
- Definição da paleta de cores. Foi criado um suporte integrado no App Inventor para a definição de uma paleta de cores harmônica seguindo um processo sistemático:

- Definição de um painel semântico, permitindo o *upload* de uma imagem relacionada ao tema do aplicativo.
- Identificação automática da cor dominante nesse painel semântico.
- Definição da cor primária do aplicativo, identificando para isso a cor da paleta do Material Design mais próxima à cor dominante.
- Apresentação de opções de cores secundárias do aplicativo utilizando o círculo cromático, incluindo as opções de cor complementar, cor análoga ou uma variante da cor primária.
- Aplicação automática da paleta de cores aos elementos visuais do aplicativo que está sendo criado no App Inventor seguindo as regras de alocação das cores do Material Design e observando questões de contraste conforme as diretrizes do W3C (2019).

5.2 Implicações para o ensino de computação

De forma geral, o App Inventor tem se mostrado eficaz para o ensino de programação (Grover & Pea, 2013b). Adicionalmente, os resultados apresentados neste estudo revelam a existência de aplicativos com bom design visual, indicando que o App Inventor também tem potencial para ser usado no ensino de design de IU. Para isso, no entanto, seria importante facilitar e guiar o design visual mais alinhado às diretrizes de guias de estilo, de modo que os aprendizes pudessem focar em aspectos mais específicos do aplicativo em desenvolvimento. Se todos os elementos de design de IU recomendados estiverem prontamente disponíveis no App Inventor, o aprendiz será direcionado a utilizar princípios e conceitos de design visual de maneira apropriada, criando assim aplicativos com uma melhor estética visual.

Geralmente, as unidades instrucionais voltadas ao ensino de computação com App Inventor (Daniel, Gresse von Wangenheim, Medeiros, & Alves, 2017; Gomes & Melo, 2013; Leôncio, Sousa, Sousa, & Melo, 2017) não abordam o design visual de forma alinhada a princípios de diretrizes. No entanto, a falta de qualidade do design visual da maioria dos aplicativos analisados revela a necessidade de que esses princípios sejam abordados. Com isso, as competências necessárias para desenvolver um aplicativo móvel de sucesso seriam abrangidas de forma mais completa. Além disso, resultados de estudos pilotos integrando explicitamente conceitos de design visual no ensino de computação (Ferreira, Gresse von Wangenheim, et al., 2019) apontam que integrar esses conceitos no ensino de computação pode ter um impacto positivo na motivação, experiência e aprendizado dos alunos.

Nesse contexto, identifica-se também a necessidade de definir modelos de avaliação da aprendizagem de computação abrangendo princípios e conceitos de design visual. Assim, com base nas recomendações de guias de estilo, devem ser derivados sistematicamente instrumentos de avaliação baseada no desempenho para avaliar os aplicativos criados pelos alunos.

5.3 Ameaças à validade

Como em qualquer estudo empírico, existem fatores que ameaçam a validade deste estudo. Com relação à generalizabilidade, pode-se questionar se a amostra utilizada realmente reflete o estado da prática do design visual de aplicativos criados com o App Inventor. Os aplicativos foram obtidos da galeria do App Inventor, a plataforma oficial de distribuição, na qual todos os usuários podem compartilhar seus aplicativos. Além disso, utilizou-se uma amostra de tamanho significativo, incluindo 88.861 aplicativos para a análise automatizada e 110 IUs para o *survey*. Portanto, considera-se a amostra adequada para analisar o estado da prática.

As ameaças à validade de conclusão incluem a definição inadequada das diretrizes e erros na análise automatizada. Para minimizar essas ameaças, foi utilizado como base o principal guia de estilo para aplicativos Android, e a ferramenta CodeMaster foi ampla e sistematicamente testada em relação à corretude antes deste estudo (Gresse von Wangenheim, Hauck, et al., 2018).

Quanto à avaliação da estética das IUs, o tamanho da amostra de 95 participantes é considerado suficiente para obter resultados válidos. Foi utilizada uma medida de um único item usando uma escala ordinal, assumindo que o construto medido é unidimensional, visto que não existe consenso quanto à dimensionalidade ou uma escala para a avaliação da estética visual.

6 Conclusão

Este artigo apresenta o estado da arte de aplicativos criados com o App Inventor em relação ao design visual da IU. Em geral, os aplicativos possuem baixo nível de estética visual e frequentemente não estão em conformidade com diretrizes de guias de estilo. Porém, a existência de aplicativos com alto nível de estética evidencia que é possível criar interfaces com bom design visual no App Inventor.

Essa tendência possivelmente é reflexo de como conceitos de design de IU não fazem parte da formação dos usuários do App Inventor em geral. Evidencia-se, portanto, a necessidade de abordar o design de IU no ensino de computação envolvendo a criação de aplicativos móveis.

Com base nesses resultados, identifica-se também a necessidade de alterações no App Inventor para suportar de forma mais adequada o design visual alinhado a guias de estilo. Espera-se que o fornecimento de um suporte mais adequado auxilie no desenvolvimento de aplicativos com um melhor design visual.

Futuramente, pretende-se identificar quais diretrizes têm maior efeito sobre a qualidade do design visual, o que poderá direcionar o ensino de design visual para focar nos princípios e conceitos mais importantes para a qualidade do design visual da IU.

Referências

Agapie, E. & Davidson, A. (2018). Human-centered design charrettes for K-12 outreach. *Interactions, 25*(6), 74-77. [GS Search]

AIGA. (2019). The Professional Association for Design. Disponível em https://www.aiga.org/

Alves, N. d. C., Gresse von Wangenheim, C., Hauck, J. C. R., & Borgatto, A. F. (2019). *A Large-scale Analysis of App Inventor Projects*. Manuscrito submetido para publicação.

Chan, K. K. (2019). Building an Online Community of Creators Through MIT App Inventor. Dissertação de mestrado, Electrical Engineering & Computer Science, MIT, Cambridge, Estados Unidos. Disponível em https://appinventor.mit.edu/

CSTA. (2016). K–12 Computer Science Framework. Disponível em https://k12cs.org/

Daniel, G. T., Gresse von Wangenheim, C., Medeiros, G. & Alves, N. d. C. (2017). Ensinando a Computação por meio de Programação com App Inventor. *Anais do Computer on the Beach*. Florianópolis, Brasil. [GS Search]

Ferreira, M. N. F., Gresse von Wangenheim, C., Missfeldt Filho, R., Pinheiro, F. d. C. & Hauck, J. C. (2019). Learning user interface design and the development of mobile applications in middle school. *Interactions, 26*(4), 66-69. [GS Search]

Ferreira, M. N. F., Pinheiro, F. d. C., Missfeldt Filho, R., Gresse von Wangenheim, C. (2019). Ensinando Design de Interface de Usuário na Educação Básica: Um Mapeamento Sistemático do Estado da Arte e Prática. In *Anais do Congresso Brasileiro de Informática na Educação*. Brasília, Brasil.

Garrett, J. J. (2011). *Elements of user experience, the: user-centered design for the web and beyond*. Berkley: New Riders Press. [GS Search]

Gomes, T. C. & de Melo, J. C. (2013). App inventor for android: Uma nova possibilidade para o ensino de lógica de programação. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*. Campinas, Brasil. [GS Search]

Gresse von Wangenheim, C. G., Hauck, J. C., Demetrio, M. F., Pelle, R., Alves, N. d. C., Barbosa, H., & Azevedo, L. F. (2018). CodeMaster – Automatic Assessment and Grading of App Inventor and Snap! Programs. *Informatics in Education, 17*(1), 117-150. [GS Search]

Gresse von Wangenheim, C. G., Porto, J. V. A., Hauck, J. C., & Borgatto, A. F. (2018). Do we agree on user interface aesthetics of Android apps?. *arXiv:1812.09049 [cs.SE]*. [GS Search]

Grover, S. & Pea, R. (2013a). Computational thinking in K–12: A review of the state of the field. *Educational Researcher*, 42(1), 38-43. [GS Search]

Grover, S. & Pea, R. (2013b). Using a Discourse-Intensive Pedagogy and Android's App Inventor for Introducing Computational Concepts to Middle School Students. In *Proceedings of the 44th ACM Technical Symposium on Computer Science Education*. Minneapolis, Estados Unidos. [GS Search]

Hamborg, K. C., Hülsmann, J. & Kaspar, K. (2014). The interplay between usability and aesthetics: More evidence for the "what is usable is beautiful" notion. *Advances in Human-Computer Interaction, 2014*, 15. [GS Search]

Hassenzahl, M. & Monk, A. (2010). The inference of perceived usability from beauty. *Human–Computer Interaction, 25*(3), 235-260. [GS Search]

International Organization for Standardization. (2011). *ISO/IEC Standard 25010 - Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models*. Disponível em https://www.iso.org/standard/35733.html

Justen, K. A. (2019). *Desenvolvimento de um Analisador de Design de Interface no Contexto do Ensino de Computação com o App Inventor* (Trabalho de Conclusão de Curso). Disponível em https://repositorio.ufsc.br/handle/123456789/202490

Kumar, D. S., Purani, K. & Viswanathan, S. A. (2018). Influences of 'appscape' on mobile app adoption and m-loyalty. *Journal of Retailing and Consumer Services, 45*, 132-141. [GS Search]

Lee, D., Moon, J., Kim, Y. J. & Yi, M. Y. (2015). Antecedents and consequences of mobile phone usability: Linking simplicity and interactivity to satisfaction, trust, and brand loyalty. *Information & Management, 52*(3), 295-304. [GS Search]

Leôncio, N. N., Sousa, C. C., Sousa, R. P. & Melo, R. F. (2017). Programação em blocos com o MIT App Inventor: Um relato de experiência com alunos do ensino médio. In *Anais do Workshop de Informática na Escola*. Recife, Brasil. [GS Search]

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. [GS Search]

MIT. (2019). About Us. Disponível em http://appinventor.mit.edu/explore/about-us.html

Mustafaraj, E., Turbak, F. & Svanberg, M. (2017). Identifying original projects in App Inventor. In *Proceedings of the 30th International Flairs Conference.* Marco Island, Estados Unidos. [GS Search]

Okerlund, J. & Turbak, F. (2013). A preliminary analysis of app inventor blocks programs. In *Proceedings of the Conference on Visual Languages and Human Centric Computing*, San Jose, Estados Unidos. [GS Search]

Patton, E. W., Tissenbaum, M. & Harunani, F. (2019). MIT App Inventor: Objectives, Design, and Development. In: Kong SC., Abelson H. (eds) Computational Thinking Education. Springer, Singapore. [GS Search]

Rogers, Y., Sharp, H. & Preece, J. (2011). *Interaction design: beyond human-computer interaction* (3rd ed.). Chichester: Wiley. [GS Search]

Schlatter, T. & Levinson, D. (2013). *Visual usability: Principles and practices for designing digital applications.* San Francisco: Morgan Kaufmann [GS Search]

Solecki, I. S. (2019). *Uma abordagem para avaliação do design visual de aplicativos móveis criados com linguagens de programação baseadas em blocos* (Dissertação de mestrado). Manuscrito em preparação.

Tissenbaum, M., Sheldon, J. & Abelson, H. (2019). From computational thinking to computational action. *Communications of the ACM, 62*(3), 34-36. [GS Search]

Tractinsky, N., Katz, A. S. & Ikar, D. (2000). What is beautiful is usable. *Interacting with computers, 13*(2), 127-145. [GS Search]

Tractinsky, N. (2013). Visual Aesthetics. *The Encyclopedia of Human-Computer Interaction, 2nd Ed.* Disponível em: https://www.interaction-design.org/

Wolber, D., Abelson, H. & Friedman, M. (2015). Democratizing computing with app inventor. *GetMobile: Mobile Computing and Communications, 18*(4), 53-58. [GS Search]

World Wide Web Consortium. (2008). Web Content Accessibility Guidelines 2.0. Disponível em https://www.w3.org/TR/2008/REC-WCAG20-20081211/

World Wide Web Consortium. (2019). Mobile Accessibility at W3C. Disponível em https://www.w3.org/WAI/standards-guidelines/mobile/

Wilensky, U., Brady, C. E., & Horn, M. S. (2014). Fostering computational literacy in science classrooms. *Communications of the ACM, 57*(8), 24-28. [GS Search]

Xie, B., & Abelson, H. (2016). Skill progression in MIT app inventor. In Proceedings of the *IEEE Symposium on Visual Languages and Human-Centric Computing.* Cambridge, Inglaterra. [GS Search]

Xie, B., Shabir, I., & Abelson, H. (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*, Pittsburgh, Estados Unidos. [GS Search]

Zen, M., & Vanderdonckt, J. (2016). Assessing user interface aesthetics based on the inter-subjectivity of judgment. In *Proceedings of the 30th International BCS Human Computer Interaction Conference: Fusion!*, Bournemouth, Inglaterra. [GS Search]