

UNIVERSIDADE FEDERAL DE SANTA CATARINA  
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

Arthur Mesquita Pickcius

**DESENVOLVIMENTO DE UM *CHECKLIST* DE AVALIAÇÃO  
DE DESIGN VISUAL DE APLICATIVOS ANDROID  
EM CONFORMIDADE COM GUIAS DE ESTILO**

Florianópolis

2021

Arthur Mesquita Pickcius

**DESENVOLVIMENTO DE UM *CHECKLIST* DE AVALIAÇÃO  
DE DESIGN VISUAL DE APLICATIVOS ANDROID  
EM CONFORMIDADE COM GUIAS DE ESTILO**

Trabalho de Conclusão de Curso submetido  
ao Curso de Bacharelado em Ciências da  
Computação para a obtenção do Grau de  
Bacharel em Ciências da Computação.

Orientadora: Prof.<sup>a</sup> Dr.<sup>a</sup> rer. nat. Christiane  
Gresse von Wangenheim, PMP

Florianópolis

2021

Arthur Mesquita Pickcius

**DESENVOLVIMENTO DE UM *CHECKLIST* DE AVALIAÇÃO DE DESIGN VISUAL DE  
APLICATIVOS ANDROID EM CONFORMIDADE COM GUIAS DE ESTILO**

Trabalho de conclusão de curso submetido ao Departamento de Informática e Estatística da Universidade Federal de Santa Catarina para a obtenção do Grau de Bacharelado em Ciências da Computação.

Orientadora:

Prof.<sup>a</sup> Dr.<sup>a</sup> rer. nat. Christiane Gresse von Wangenheim, PMP

Banca Examinadora:

Prof. Dr. Adriano Ferreti Borgatto

Miriam Nathalie Fortuna Ferreira

Florianópolis

2021

## **Lista de Figuras**

Figura 1 - Tela inicial do aplicativo jetBlue

Figura 2 - Exemplo de uso de margens

Figura 3 - Exemplos de tipografia

Figura 4 - Interface de design do App Inventor 2

Figura 5 - Blocos de código do App Inventor 2

Figura 6 - Elementos de design visual do App Inventor 2

Figura 7 - Exemplos de aplicativos da iniciativa Computação na Escola

Figura 8 - Scree Plot referente ao CodeMaster: UI Design 2.0

## Lista de Tabelas

- Tabela 1 - Tipos de interação com dispositivos eletrônicos
- Tabela 2 - Descrição dos níveis de UX
- Tabela 3 - Meta-princípios da usabilidade visual
- Tabela 4 - Exemplos de combinações de cores em aplicativos Android
- Tabela 5 - Exemplos de uso inadequado de imagens
- Tabela 6 - Tamanhos de fontes
- Tabela 7 - Elementos do design visual em relação ao App Inventor
- Tabela 8 - Palavras-chave do mapeamento sistemático
- Tabela 9 - Números de resultados da busca
- Tabela 10 - *Checklists* encontrados
- Tabela 11 - Descrição dos dados a serem extraídos
- Tabela 12 - Guias base
- Tabela 13 - Elementos do design visual presentes nos *checklists*
- Tabela 14 - Uso de ferramentas nos *checklists*
- Tabela 15 - Métodos de avaliação
- Tabela 16 - Tamanhos de fonte (simplificado)
- Tabela 17 - *Checklist* Codemaster - UI Design 2.0
- Tabela 18 - Análise demográfica referente ao painel de especialistas
- Tabela 19 - Cohen's Kappa e nível de concordância
- Tabela 20 - Valores faltantes por item
- Tabela 21 - Correlação item-total e alfa de Cronbach com remoção de itens
- Tabela 22 - Resultados da análise da correlação policórica
- Tabela 23 - Carga fatorial em 6 fatores
- Tabela 24 - Carga fatorial em 1 fator

# Sumário

Abstract.....	8
Resumo.....	9
1. Introdução.....	10
1.1. Contextualização.....	10
1.2. Objetivos.....	13
1.3. Método de Pesquisa.....	14
1.4. Estrutura do trabalho.....	15
2. Fundamentação teórica.....	16
2.1. Usabilidade e design visual.....	16
2.1.1. Elementos do design visual.....	20
2.2. Guias de estilo e acessibilidade.....	24
2.3. Avaliação de usabilidade.....	27
2.4. Ensino de design visual na educação básica.....	29
2.5. App Inventor.....	29
3. Estado da arte.....	32
3.1. Definição do mapeamento.....	32
3.2. Execução da busca.....	34
3.3. Extração de informação.....	35
3.4. Discussão.....	36
4. Desenvolvimento do <i>checklist</i> CodeMaster UI Design 2.0.....	44
4.1. Definição conceitual.....	44
4.2. Checklist.....	46
5. Avaliação do checklist.....	50
5.1. Avaliação por painel de especialistas.....	50
5.1.1. Definição da avaliação por painel de especialistas.....	50

5.1.2. Execução da avaliação por painel de especialistas.....	51
5.1.3. Análise dos dados do painel de especialistas.....	52
5.2. Avaliação por estudo de caso.....	55
5.2.1. Definição da avaliação por estudo de caso.....	55
5.2.2. Execução da avaliação por estudo de caso.....	56
5.2.3. Análise dos dados do estudo de caso.....	57
5.3. Discussão.....	67
5.4. Ameaças à validade.....	68
6. Conclusão.....	69
Referências.....	70
Apêndice A – Tabelas da extração de dados.....	74
Apêndice B – Resumo dos elementos avaliados pelos <i>checklists</i> encontrados.....	80
Apêndice C – Artigo (Próxima página).....	82

## **Abstract**

Smartphones are becoming the most widespread personal consumer device. However, offering mobile access anywhere, anytime for anyone brings new challenges to usability. One way of evaluating the usability of interfaces is the heuristic evaluation. There are several sets of usability heuristics for this purpose. However, usability depends on the visual design among other characteristics. And there is a lack of tools and resources aimed specifically at evaluating visual design. The available Checklists for evaluating user interfaces usually focus on evaluating general utility, not providing features specifically aimed at visual elements. In order to fill this gap, this study aims to develop a visual design checklist for an evaluation of the user interface of Android applications. Thus, the objective of the present work is to systematically develop a checklist via literature survey mapping and unifying customized visual design guidelines to Android applications based on style guides. The results mainly include the development of a checklist and an assessment of its reliability and validity. The results include a satisfactory value of Cronbach's alpha and acceptable values of the factor loads for one factor. The checklist can be used in the development of Android applications by evaluating and indicating opportunities for improving visual design. As a consequence, it contributes to increasing the quality of applications and, in addition, allows users to achieve their interaction goals in an optimized way and / or even in teaching interface design.

**Keywords:** checklist, visual design, app development.

## Resumo

Os *smartphones* estão se tornando o dispositivo de consumo pessoal mais difundido. No entanto, oferecer acesso móvel em qualquer lugar, a qualquer momento para qualquer um traz novos desafios à usabilidade. Uma forma de avaliar a usabilidade de interfaces é a avaliação heurística. Existem vários conjuntos de heurísticas de usabilidade para esta finalidade. Porém, a usabilidade depende, entre outras características, do design visual. E faltam ferramentas e recursos voltados especificamente para suportar a avaliação do design visual. *Checklists* disponíveis para a avaliação de interfaces de usuário geralmente se concentram na avaliação de usabilidade geral, não fornecendo recursos especificamente voltados a elementos visuais. A fim de preencher essa lacuna, este estudo visa o desenvolvimento de um *checklist* de design visual para a avaliação da interface do usuário de aplicativos Android. Assim, o objetivo do presente trabalho é desenvolver sistematicamente um *checklist* por meio de um levantamento de literatura mapeando e unificando diretrizes de design visual customizadas a aplicativos Android com base em guias de estilo. Os resultados incluem principalmente o desenvolvimento de um *checklist* e a avaliação de sua confiabilidade e validade. Os resultados obtidos indicam que o modelo é válido e confiável. Os resultados incluem um valor satisfatório do alfa de Cronbach e valores aceitáveis das cargas fatoriais para um fator. O *checklist* pode ser utilizado no desenvolvimento de aplicativos Android avaliando e indicando oportunidades de melhoria do design visual. O que, por consequência, contribui para o aumento da qualidade de aplicativos e, além disso, permite que os usuários atinjam seus objetivos de interação de forma otimizada e/ou até no ensino de design de interface.

**Palavras-chave:** *checklist*, design visual, desenvolvimento de aplicativos.

## 1. Introdução

### 1.1. Contextualização

O uso dos smartphones para o consumo pessoal tem se popularizado e difundido cada vez mais. Atualmente existem mais de 154 milhões de smartphones no Brasil (CARRARA, 2020). Nos últimos anos, eles avançaram rapidamente de um dispositivo simples que apenas fazia chamadas telefônicas para sistemas interativos multimídia complexos. Tais mudanças nos dispositivos têm um impacto significativo na natureza da interação do usuário. Como os telefones celulares oferecem novas formas de interação (via gesto, sensores, câmera, voz, etc.), seus modelos de interação podem diferir significativamente dos tradicionais (WASSERMAN, 2010). Requisitos de tamanho e portabilidade de telefones celulares apresentam limitações, bem como as maneiras estranhas para a entrada de dados. Especialmente os telefones com *touchscreen* representam novos desafios por falta de *feedback* tátil, tamanho, etc. (BALAGTAS-FERNANDEZ; FORRAI; HUSSMANN, 2009). O design de interface é ainda mais complicado, pois os telefones móveis são usados por uma ampla gama de pessoas com objetivos diferentes a qualquer momento em qualquer local (HUANG; TIAN, 2018). Isso, por outro lado, torna a usabilidade um atributo de qualidade ainda mais importante nos aplicativos de software para *smartphones* (TREERATANAPON, 2012).

Assim, visando proporcionar aos usuários uma melhor interação com os aplicativos, a usabilidade deve ser uma prioridade no processo de desenvolvimento de qualquer interface, pois influencia o nível de aceitabilidade do usuário em relação ao sistema (KORTUM; SORBER, 2015). Usabilidade é a “medida em que um produto pode ser usado por usuários específicos para atingir objetivos específicos com eficácia, eficiência e satisfação em um contexto específico de uso” (“ISO 9241-210”, 2019). Em um contexto mais amplo, a experiência do usuário (UX) considera a satisfação antes, durante e após o uso (“ISO 9241-210”, 2019). Neste contexto, as abordagens de design centradas no ser humano para design e desenvolvimento de software visam tornar os sistemas interativos mais utilizáveis, abordando vários planos de uma maneira orientada ao processo (GARRETT, 2011). Assim, no contexto do desenvolvimento da interface do software, isso abrange vários níveis, que se referem à estratégia, escopo, estrutura, esqueleto e superfície, incluindo o

design de interação, design de interface e design visual (GARRETT, 2011).

A usabilidade depende, entre outras características, do design visual da interface de usuário (“ISO/IEC 25010”, 2011). 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 de usar (SHARP; ROGERS; PREECE, 2005). E é considerada um fator chave na usabilidade percebida e efetiva da interface do usuário, satisfação subjetiva do usuário, confiança, credibilidade e preferência (ZEN; VANDERDONCKT, 2016).

Diversos métodos tradicionais de avaliação de usabilidade não consideram problemas específicos de dispositivos móveis (JOYCE et al., 2016). Porém, atualmente, existem poucos estudos que avaliem a experiência de usuário de aplicativos mobile (HUANG; TIAN, 2018). Nesse contexto, destaca-se o guia de estilo Material Design (GOOGLE, 2014). O Material Design sintetiza princípios clássicos de design, podendo auxiliar no desenvolvimento de produtos de software com melhor usabilidade. Complementar, as *Web Content Accessibility Guidelines 2.0* (W3C, 2014) (W3C, 2013) apresentam diretrizes para que conteúdos da web sejam acessíveis para usuários portadores de deficiências, incluindo visão limitada, perda de audição, entre outros. Essas mesmas diretrizes se aplicam também a conteúdos apresentados em dispositivos móveis (W3C, 2015).

Guias de estilo auxiliam no desenvolvimento da interface fornecendo recomendações úteis para o design visual. No entanto, faltam ferramentas e recursos voltados especificamente para suportar a avaliação do design visual em alinhamento com esses guias. Por exemplo, as 10 heurísticas de Nielsen (1994a) não fornecem recomendações práticas sobre como atender aos princípios do design visual, mantendo o foco no nível de princípios. Por outro lado, *checklists* disponíveis para a avaliação de interfaces de usuário geralmente se concentram na avaliação de usabilidade geral (GRESSE VON WANGENHEIM et al., 2016), incluindo navegação, atributos não funcionais e até mesmo problemas de implementação. Porém, a maior parte destes *checklists* não fornece recursos especificamente voltados a elementos visuais. Vários desses *checklists* são voltados a domínios de aplicação específicos,

como aplicativos de saúde (STOYANOV et al., 2015).

O *checklist* desenvolvido no presente trabalho é voltado à avaliação de design visual de aplicativos Android e poderá ser utilizado tanto no desenvolvimento profissional de aplicativos quanto no ensino de computação. No âmbito do desenvolvimento profissional, o *checklist* pode ser utilizado por desenvolvedores e designers para avaliar, de forma rápida e com baixo custo, a conformidade do design visual dos seus protótipos de aplicativos com os principais guias de estilo e assim contribuir para a qualidade dos produtos sendo desenvolvidos.

No contexto educacional, o *checklist* pode ser utilizado por alunos e professores para avaliar os resultados de atividades ensinando a computação por meio do desenvolvimento de aplicativos com App Inventor. Como por exemplo o que vem sendo desenvolvido pela iniciativa Computação na Escola/INCoD/INE/UFSC (<https://computacaonaescola.paginas.ufsc.br/>). O *checklist* permite de forma objetiva avaliar a aprendizagem em relação a competências de design visual e o fornecimento de um *feedback* construtivo ao aluno.

Assim, a questão de pesquisa para o presente projeto é: se é possível criar um *checklist* para avaliar o design visual de aplicativos Android em conformidade com guias de estilo?

## 1.2. Objetivos

### **Objetivo geral**

O objetivo geral do presente projeto é definir um *checklist* para avaliar o design visual de aplicativos Android com base em guias de estilo.

### **Objetivos Específicos**

Para que o objetivo geral seja alcançado, faz-se necessário que os seguintes objetivos específicos sejam alcançados:

- O1.** Sintetizar a teoria da área de usabilidade e design visual em termos de princípios, elementos e guias de estilo.
- O2.** Levantar o estado da arte em relação à *checklists* voltados à avaliação do design visual.
- O3.** Mapear e unificar os *checklists* existentes e customizar o *checklist* com base nas diretrizes de guias de estilo com foco em aplicativos Android.
- O4.** Avaliar a confiabilidade e validade do *checklist* desenvolvido.

### **Delimitação e escopo do trabalho**

A proposta deste trabalho é a criação de um *checklist* voltado à aplicação no contexto de educação básica, visando contemplar os componentes principais do design visual, porém limitando-se a componentes com complexidade de implementação adequada à educação básica. Sendo assim, o *checklist* desenvolvido não é avaliado em relação a outros tipos de aplicação.

Além disso, este trabalho é desenvolvido considerando tablets e *smartphones*, portanto o termo mobile se refere apenas a estes dispositivos. O sistema operacional Android é o sistema alvo do desenvolvimento do *checklist*. Sendo assim, não são consideradas outras plataformas.

### 1.3. Método de Pesquisa

Do ponto de vista da natureza da pesquisa a ser empreendida, este trabalho se classifica como uma pesquisa aplicada (ou tecnológica), que tem por objetivo gerar produtos e/ou processos inéditos, com finalidades imediatas, com base em conhecimentos prévios. Quanto aos objetivos da pesquisa, este trabalho se caracteriza como uma pesquisa exploratória, pois visa proporcionar maior familiaridade com o problema investigado a fim de torná-lo explícito. A metodologia de pesquisa deste projeto é dividida nas seguintes etapas:

**Etapa 1.** Fundamentação teórica: É realizada uma análise de literatura na área de IHC (Interação Humano-Computador) em geral e enfatizando a área de usabilidade, design visual e de guias de estilo voltados a aplicativos Android.

**Atividade 1.1.** Analisar a área de Interação Humano-Computador, Engenharia de Usabilidade, Design Visual;

**Atividade 1.2.** Analisar guias de estilo voltados ao design visual de aplicativos móveis.

**Etapa 2.** Levantamento do estado da arte: Analisar o estado da arte em relação a *checklists* existentes voltados à avaliação do design visual de aplicativos Android. Na realização desta etapa foi aplicada a técnica de mapeamento sistemático de literatura (PETERSEN et al., 2008) para identificar e estudar guias de design de interface de ambientes de aplicativos Android atualmente sendo utilizados. No início do processo de mapeamento foi feita a definição do escopo da pesquisa, pergunta de pesquisa, palavras-chaves e critérios. Com o protocolo de busca definido, foi iniciada a execução da busca. Durante a execução da busca o conjunto de palavras-chaves a partir dos resumos e conclusões dos documentos são sempre reavaliados. Caso necessário, alterações no protocolo de busca são feitas e a execução é reiniciada. Em seguida foi realizada a extração das informações dos documentos, filtrando os documentos que respondem a pergunta de pesquisa definida no protocolo de busca. O mapeamento sistemático foi finalizado com a análise das informações dos documentos restantes da etapa anterior. Desta forma, a subdivisão desta atividade se resume em:

**Atividade 2.1.** Definir o protocolo da revisão sistemática;

**Atividade 2.2.** Executar a busca;

**Atividade 2.3.** Analisar e interpretar as informações extraídas;

**Atividade 2.4.** Documentar e discutir os resultados.

**Etapa 3.** Criação do *checklist*: Os *checklists* encontrados no mapeamento da literatura serão sistematicamente unificados integrando medidas derivadas a partir de guias de estilo.

**Atividade 3.1.** Mapear e unificar as heurísticas, diretrizes e *checklists* existentes;

**Atividade 3.2.** Integrar critérios de avaliação com base nos guias de estilo.

**Etapa 4.** Avaliação do *checklist*: O *checklist* desenvolvido é avaliado por um estudo de caso, por meio da coleta de aplicativos da Galeria App Inventor, os quais são analisados aplicando o *checklist*. A partir dos dados obtidos é realizada uma análise estatística sobre a confiabilidade do modelo por meio do coeficiente Alfa de Cronbach (CRONBACH, 1951) e a validade convergente por meio da matriz de correlação policórica (OLSSON, 1979). Além disso, a composição do modelo é analisada por meio de uma análise exploratória fatorial (GLORFELD, 1995).

**Atividade 4.1.** Definição da avaliação do *checklist*.

**Atividade 4.2.** Execução da avaliação.

**Atividade 4.3.** Análise dos dados obtidos.

#### 1.4. Estrutura do trabalho

No capítulo 2 é apresentada a fundamentação teórica para o desenvolvimento do *checklist*. No capítulo 3 é feita uma revisão sistemática da literatura sobre outros *checklists* de avaliação de design visual de aplicativos Android existentes. No capítulo 4 é feita uma proposta de um novo *checklist*, com base nos resultados obtidos na revisão sistemática. No capítulo 5 é apresentada a avaliação do *checklist* desenvolvido. O capítulo 6 apresenta uma discussão dos resultados obtidos no capítulo anterior e as ameaças à validade. No capítulo 7 são feitas considerações finais e conclusão.

## 2. Fundamentação teórica

Neste capítulo são apresentados os conceitos e definições do design visual, usabilidade e interação do humano com dispositivos eletrônicos, inserindo também avaliação e *checklists* no contexto de usabilidade de aplicativos Android.

### 2.1. Usabilidade e design visual

A interface de usuário, no contexto de dispositivos eletrônicos, é o meio pelo qual são realizadas comunicação e interação. Este meio faz uso de uma linguagem composta por sinais e símbolos para informar, direcionar e prover feedback (SCHLATTER; LEVINSON, 2013).

Comunicação envolve um emissor, uma mensagem, um sinal que transmite a mensagem e um receptor que interpreta essa mensagem. Ao avaliar a comunicação desta forma se torna evidente o fato de que designer e desenvolvedores não têm controle sobre toda a extensão do processo. Sendo assim, ao desenvolver uma interface selecionando elementos com base na expectativa dos usuários e combinando esses elementos com sinais visuais que estes usuários esperam e entendem, aumenta as chances de que a mensagem seja corretamente interpretada (SCHLATTER; LEVINSON, 2013).

A interação com dispositivos eletrônicos geralmente é classificada da forma exibida na Tabela 1.

TABELA 1: TIPOS DE INTERAÇÃO COM DISPOSITIVOS ELETRÔNICOS

Instrução	O usuário dá instruções ao produto para que este realize a operação desejada
Manipulação	Ocorre por meio de objetos reais ou virtuais (e.g. segurando ou movendo estes objetos)
Conversa	O usuário interage com o produto por meio da fala
Exploração	O usuário interage com um ambiente 3D por meio de realidade virtual ou realidade aumentada

Fonte: SHARP; ROGERS; PREECE, 2005.

Por ser voltado à avaliação do design visual de aplicativos, o presente trabalho se restringe às interações de instrução e de manipulação.

Em um contexto mais amplo, a experiência do usuário considera a satisfação

antes, durante e após o uso (“ISO 9241-210”, 2019). A experiência de usuário é composta por níveis e tenta assegurar que nenhum aspecto da interação do usuário ocorra sem o conhecimento e intenção do desenvolvedor, considerando cada ação do usuário e cada situação possível e sistematicamente feito um design de interface (GARRETT, 2011).

A usabilidade faz parte da experiência de usuário, sendo a “medida em que um produto pode ser usado por usuários específicos para atingir objetivos específicos com eficácia, eficiência e satisfação em um contexto específico de uso” (“ISO 9241-210”, 2019). Assim, a qualidade do design visual e da interação de usuário definem a usabilidade do produto.

A experiência de usuário pode ser dividida em níveis conforme apresentado na Tabela 2. Estes níveis foram definidos originalmente com foco em desenvolvimento *web* (GARRETT, 2011), porém esta divisão de níveis pode ser aplicada também ao desenvolvimento de aplicativos Android. Os níveis possuem diferentes graus de abstração e o desenvolvimento do produto ocorre partindo do nível mais abstrato para ao final atingir o nível mais concreto.

TABELA 2: DESCRIÇÃO DOS NÍVEIS DE UX

Estratégia	É definida em um alto nível de abstração, como o resultado da interação das expectativas do desenvolvedor e do cliente em relação ao produto
Escopo	É a tradução das expectativas de usuário e objetivos do produto em requisitos específicos para conteúdo e funcionalidades que o produto oferecerá aos usuários
Estrutura	Define os percursos da interação do usuário com o produto, no caso de aplicativos Android inclui as possíveis trajetórias de telas que o usuário faz ao utilizar o aplicativo
Esqueleto	É a disposição em que são colocados os botões, imagens, textos e demais elementos. Composto pelo design da interface, navegação e informação
Superfície	É o conteúdo sendo apresentado ao usuário, as imagens e o texto. Ou seja, o design visual do produto

Fonte: GARRETT, 2011.

O foco do presente trabalho está no nível de superfície, ou seja, no design visual do aplicativo. O objetivo deste nível é apresentar a disposição lógica que compõe o esqueleto do produto. É no design visual em que as funcionalidades e a

estética se misturam para desenvolver produtos que agradem ao usuário enquanto cumprem todos os objetivos dos outros níveis (GARRETT, 2011).

O design de interface significa definir as propriedades dos elementos do design visual, possibilitando a interação (GARRETT, 2011). Também pode ser visto como o desenvolvimento de produtos interativos para auxiliar a forma como as pessoas se comunicam e interagem em suas vidas cotidianas e profissionais, criando experiências de usuários que melhorem estes aspectos (SHARP; ROGERS; PREECE, 2005).

Quanto ao design visual, são considerados como base três meta-princípios (SCHLATTER; LEVINSON, 2013) apresentados na Tabela 3.

TABELA 3: META-PRINCÍPIOS DA USABILIDADE VISUAL

Consistência	O meta-princípio da consistência refere-se ao processo de guiar as expectativas do usuário em relação ao produto, por meio do uso de elementos com os quais ele está familiarizado.  Existem dois tipos de consistência: a consistência interna se refere ao próprio produto, e a consistência externa se dá em relação a outros produtos utilizados pelo mesmo público.
Hierarquia	A hierarquia trata sobre a percepção e interpretação da relação de importância entre os elementos. A percepção dessa hierarquia é influenciada pelo posicionamento, tamanho, cor, tipo e agrupamento entre os elementos.  No caso de aplicativos, essa hierarquia serve para facilitar o seu uso de forma que os usuários consigam atingir seus objetivos relacionados ao aplicativo intuitivamente.
Personalidade	Este meta-princípio se refere às impressões dos usuários em relação ao produto. Considerando as impressões que o usuário tem tanto de forma consciente, quanto inconsciente. Ao desenvolver um produto devem ser consideradas quais impressões querem ser passadas aos usuários e se o produto está atingindo esse objetivo.

Fonte: SCHLATTER; LEVINSON, 2013.

A partir dos meta-princípios se definem princípios de design, que auxiliam no desenvolvimento de experiências de usuário de qualidade.

“Outra maneira de conceitualizar a usabilidade se dá em termos de princípios de design. Trata-se de abstrações generalizáveis, destinadas a orientar os designers a pensar sobre aspectos diferentes de seus designs.” (SHARP; ROGERS; PREECE, 2005).

“Os princípios do design, quando usados na prática, normalmente são chamados de heurística. Esse termo enfatiza que algo deve ser feito

com esses princípios, quando aplicados a um dado problema.”  
(SHARP; ROGERS; PREECE, 2005).

Os princípios de design são diretrizes para o desenvolvimento de produtos com boa usabilidade. Também podem ser vistos como um conjunto de regras a serem seguidas (SHARP; ROGERS; PREECE, 2005).

**Visibilidade.** O princípio de visibilidade se refere à facilidade do usuário encontrar a função que procura no produto, estando relacionado à sua localização ou *layout*.

**Feedback.** É a resposta que o usuário recebe ao interagir com o produto, essa resposta está vinculada aos sentidos. No caso de aplicativos Android eles costumam envolver visão, audição e/ou tato (SHARP; ROGERS; PREECE, 2005).

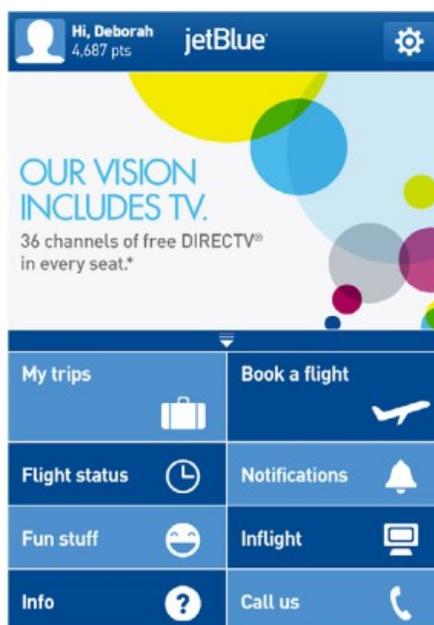
**Restrições.** São tudo aquilo que impede o usuário de realizar uma ação ao interagir com o produto. Elas podem ser separadas em restrições físicas, lógicas e culturais. Em aplicativos Android um exemplo de restrição física pode ser um botão que fica inativo enquanto o usuário não realiza uma outra ação. As restrições lógicas estão nos casos em que o usuário acaba não conseguindo utilizar uma função do produto porque o acesso a ela é feito de uma forma que não segue o senso comum. As restrições culturais são similares às lógicas, porém suas causas são convenções aprendidas, que podem variar de um local para outro (SHARP; ROGERS; PREECE, 2005).

**Mapeamento.** É a relação de causa e efeito entre as ações do usuário, ou alterações de estado do produto e a resposta que o produto fornece ao usuário (SHARP; ROGERS; PREECE, 2005).

**Consistência.** Complementando o meta-princípio apresentado anteriormente (SCHLATTER; LEVINSON, 2013), a consistência também pode ser vista como um dos princípios de design (SHARP; ROGERS; PREECE, 2005), com foco na consistência interna. É manter um padrão entre as interfaces, fazendo uso de elementos semelhantes para a realização de operações similares. Se um produto é projetado mantendo consistência é mais fácil para o usuário aprender como fazer uso dele. Se o usuário já utilizou uma função do produto em uma determinada interface, é criada uma expectativa de que seja simples encontrar a função que procura em outra interface, mesmo que não a tenha utilizado anteriormente. A Figura 1 ilustra um exemplo de consistência interna, em que a interface segue um

padrão de cores bem definido.

FIGURA 1: TELA INICIAL DO APLICATIVO JETBLUE



Fonte: SCHLATTER; LEVINSON, 2013.

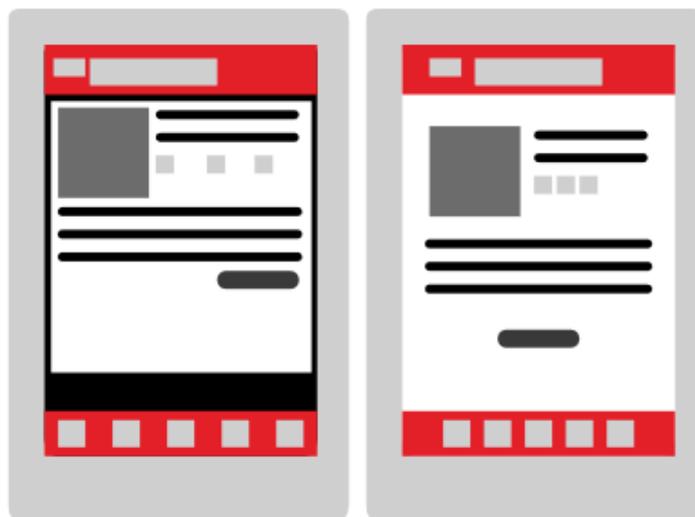
Para o conceito de *affordance*, neste trabalho é utilizada a definição feita por Norman (1999 apud PREECE; ROGERS; SHARP, 2005, p. 43). Ou seja, um atributo que se refere à facilidade de compreensão para o usuário de que ao realizar determinada ação ele obterá o resultado desejado.

### 2.1.1. Elementos do design visual

No contexto do desenvolvimento de aplicativos para Android, o design visual faz uso de elementos como *layout*, controles, cores, tipografia, imagens e escrita.

**Layout.** O *layout* se refere aos elementos que afetam a percepção e agrupamento daquilo que está sendo exibido ao usuário, como posicionamento, alinhamento e *grids* (SCHLATTER; LEVINSON, 2013), estando fortemente ligado ao princípio da visibilidade. A Figura 2 exemplifica como as alterações nas margens afetam o visual de um aplicativo.

FIGURA 2: EXEMPLO DE USO DE MARGENS



Fonte: SCHLATTER; LEVINSON, 2013.

**Controles.** Os controles são elementos de interface e métodos que o usuário utiliza para interagir com o aplicativo. Sua relação com o princípio da visibilidade busca facilitar ao usuário perceber as ações que podem ser operadas para realizar essa interação. Se um botão parece pressionável, ou um *slider* arrastável, etc. (SCHLATTER; LEVINSON, 2013).

**Cores.** São um fenômeno óptico, uma sensação causada pela luz que é percebida pelo olho humano, processada pelo cérebro. No contexto de aplicativos Android podem ser personalizadas a cor de fundo, cores dos elementos, cor do texto, etc. Existe uma lógica que rege a escolha de cores e uma estratégia em seu uso. Uma boa escolha de cores ajuda os usuários a associarem os conceitos apropriados ao contexto da aplicação, além de identificarem em qual parte do aplicativo se encontram e as interações que podem realizar (SCHLATTER; LEVINSON, 2013).

No design, teoria das cores é um termo ligado a um conjunto de regras que regem o uso das cores. Este conjunto de regras se baseia no círculo cromático de Isaac Newton, que originalmente separava as cores nas categorias:

- **Primárias:** Vermelho, azul e amarelo.
- **Secundárias:** (Criadas pela mistura das cores primárias) roxo, verde e laranja.
- **Intermediárias ou terciárias:** Criadas pela mistura de cores primárias com secundárias.

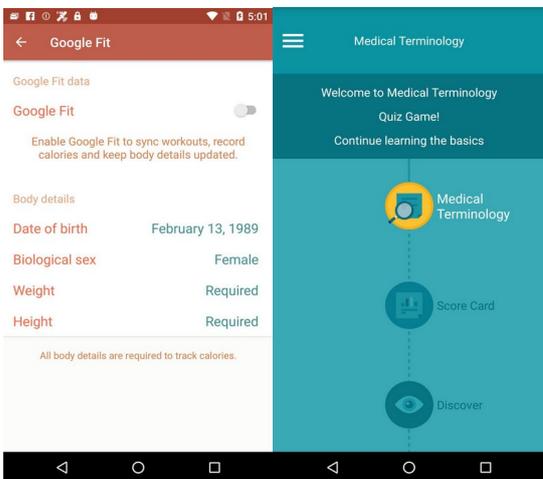
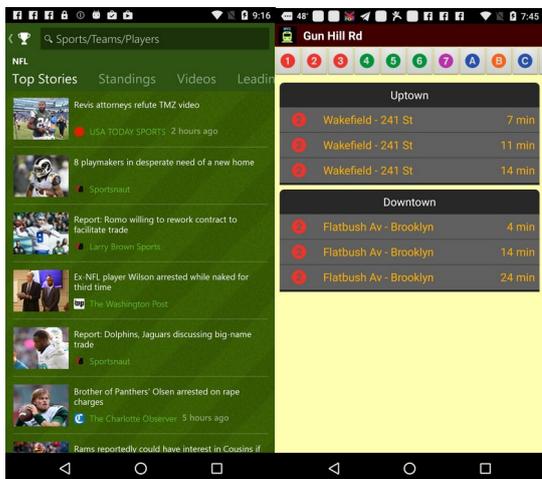
Existem esquemas de cores que servem para guiar a escolha de combinações harmoniosas de cores no design. Sendo alguns dos principais esquemas de cores:

- **Monocromático:** apenas uma cor, podendo variar a saturação.
- **Análogo:** cores próximas no círculo.
- **Complementar:** pares de cores opostas no círculo.
- **Complementar:** dividido: um par de cores equidistantes a uma outra cor.
- **Triádico:** três cores equidistantes no círculo.
- **Tetrádico:** dois pares de cores complementares.

A temperatura das cores também é um fator importante no design, afetando as respostas emocionais do usuário ao utilizar o aplicativo. Sendo importante que a escolha de cores seja feita considerando a finalidade do aplicativo e o seu público-alvo.

O contraste entre os elementos também influencia na experiência de usuário. Em um aplicativo ele pode ser utilizado para indicar quando um elemento está ativo, reduzindo o contraste quando não se pode interagir com aquele elemento. Porém, seu uso inadequado pode gerar dificuldade para o usuário visualizar determinados elementos como o texto. Isso fica evidente entre as combinações de cores inadequadas da Tabela 4.

TABELA 4: EXEMPLOS DE COMBINAÇÕES DE CORES EM APLICATIVOS ANDROID

Combinações de cores adequadas	Combinações de cores inadequadas
	

Fonte: Rico - base de dados do site *Interaction Mining*.

**Tipografia.** No desenvolvimento de aplicativos Android a tipografia envolve a

escolha de fonte, quantas fontes serão usadas e como é feita essa escolha. A escolha de fontes pode afetar a usabilidade de forma similar ao uso de cores.

Tipos de fonte com cantos pontiagudos são chamadas de *serif* e fontes sem essas pontas são *sans-serif*. Na Figura 3 são ilustrados dois tipos de fontes: (a) Helvetica Regular, que é uma fonte *sans-serif* e (b) Times New Roman, uma fonte *serif*. Outro tipo de fonte é o *monospace*, na qual cada caractere ocupa o mesmo espaço.

FIGURA 3: EXEMPLOS DE TIPOGRAFIA



Fonte: SCHLATTER; LEVINSON, 2013.

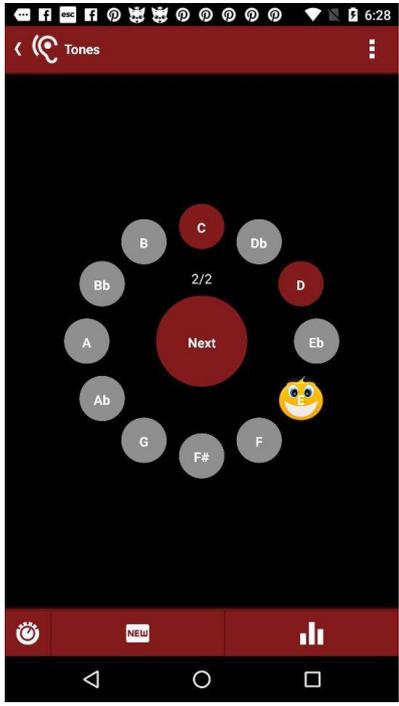
No sistema Android existe a unidade dp (*Density-independent Pixel*), que tenta manter um tamanho real independentemente da densidade de pixels da tela do dispositivo. A unidade padrão para tamanho de fonte é sp (*Scale-independent Pixel*), que é similar ao dp, porém possui um parâmetro nas configurações do sistema, sendo afetada por este.

**Imagens.** O uso de imagens também é uma ferramenta importante no design visual de aplicativos. Deve ser avaliado se será utilizada alguma imagem, que tipos de imagens, logos, fotos, ícones.

A consistência no uso de cores também se estende à paleta de cores das imagens. Sendo recomendável que esteja em consistência com a paleta de cores do aplicativo sempre que possível. Na Tabela 5 é apresentado um exemplo de uso indevido de ícone, em que o ícone está inconsistente com o tema do aplicativo e sua cor prejudica o contraste em relação ao texto.

O tamanho das imagens também é um fator que influencia na usabilidade. É difícil para o usuário visualizar interagir com elementos de tamanho inadequado. Imagens distorcidas pelo tamanho utilizado também prejudicam a experiência do usuário, como no exemplo da Tabela 5.

TABELA 5: EXEMPLOS DE USO INADEQUADO DE IMAGENS

Imagem distorcida pela escala	Ícones inconsistentes
	

Fonte: Rico – base de dados do site *Interaction Mining*.

**Escrita.** É um fator de grande influência na aceitação do público em relação ao aplicativo desenvolvido. Caso o aplicativo apresente erros gramaticais, ou termos inapropriados para o público-alvo, ele transmite a sensação de que não foi desenvolvido de forma profissional.

## 2.2. Guias de estilo e acessibilidade

Guias de estilo, também conhecidos como manuais de estilo, são conjuntos de regras sobre o design de um documento. Os guias podem ser especializados para determinados tipos de documentos ou finalidades. Este trabalho tem seu foco em guias que sejam relevantes para o desenvolvimento de aplicativos mobile, preferencialmente para o sistema Android.

## Material Design

“Material é um sistema adaptável de *guidelines*, componentes e ferramentas que fornecem suporte às melhores práticas no design de interface de usuário. Utilizando código aberto, Material agiliza a colaboração entre designers e desenvolvedores, e auxilia equipes a criarem belos produtos rapidamente.” (“Material Design”, 2014, tradução nossa).

É um sistema de diretrizes desenvolvido pelo Google desde 2014, com o objetivo de facilitar o design de interfaces. Vem sendo aplicado de forma consistente nos aplicativos Android do Google (GOOGLE, 2014).

O Material Design fornece ferramentas para auxílio no desenvolvimento de aplicativos. Em 2018 foram publicadas atualizações nas ferramentas, flexibilizando o seu uso ao introduzir o conceito de temas, ou seja, a propagação de elementos pelo aplicativo a ser desenvolvido (e.g. cores e tipografia) (LARDINOIS, 2018).

Um exemplo das *guidelines* do Material Design referente à tipografia está ilustrado na Tabela 6. Esta tabela corresponde a um *type scale*, que define padrões de fonte, tamanho, espaçamento e capitalização do texto.

TABELA 6: TAMANHOS DE FONTES

Scale Category	Typeface	Font	Size	Case	Letter spacing
H1	Roboto	Light	96	Sentence	-1.5
H2	Roboto	Light	60	Sentence	-0.5
H3	Roboto	Regular	48	Sentence	0
H4	Roboto	Regular	34	Sentence	0.25
H5	Roboto	Regular	24	Sentence	0
H6	Roboto	Medium	20	Sentence	0.15
Subtitle 1	Roboto	Regular	16	Sentence	0.15
Subtitle 2	Roboto	Medium	14	Sentence	0.1
Body 1	Roboto	Regular	16	Sentence	0.5
Body 2	Roboto	Regular	14	Sentence	0.25
BUTTON	Roboto	Medium	14	All caps	1.25
Caption	Roboto	Regular	12	Sentence	0.4
OVERLINE	Roboto	Regular	10	All caps	1.5

Fonte: “Material Design”, 2014

### W3C

Diretrizes de acessibilidade para conteúdo web (WCAG), que atualmente estão em sua versão 2.0. São desenvolvidas pelo *World Wide Web Consortium* (W3C), uma comunidade que desenvolve os padrões da internet.

O WCAG foi desenvolvido com objetivo de facilitar a acessibilidade do conteúdo web para a maior quantidade possível de pessoas com deficiência. Porém, normalmente ao seguir estas diretrizes o resultado obtido costuma ser uma melhor usabilidade também no caso geral (W3C, 2014).

A acessibilidade abrange uma vasta gama de deficiências, incluindo visual,

auditiva, física, de fala, intelectual, de linguagem, de aprendizagem e neurológica. Esse *checklist* abrange uma grande quantidade de casos, mas não todos os graus e tipos de deficiência (W3C, 2014).

O documento separa seu conteúdo em forma de camadas de princípios, diretrizes e critérios de sucesso para cada uma destas diretrizes. No documento também são fornecidos links para recursos e código fonte, exemplificando técnicas necessárias e sugeridas, bem como falhas comuns (W3C, 2014).

A W3C publicou notas sobre como interpretar a WCAG 2.0 fora do contexto web, e também no contexto de dispositivos mobile (W3C, 2013) (W3C, 2015). Estas notas servem como guias de caráter informativo, portanto não definem um conjunto de critérios, em relação à interpretação do conteúdo da WCAG.

Assume-se que a maioria dos critérios de sucesso da WCAG 2.0 pode ser aplicada para documentos e software fora do contexto web, sem nenhuma ou com mínimas alterações. Estas alterações em sua maioria são substituições de termos específicos do contexto web, que podem ser facilmente mapeados para conceitos de aplicativos mobile (e.g. conjunto de páginas web pode ser mapeado para um conjunto de aplicativos) (W3C, 2013).

### 2.3. Avaliação de usabilidade

Avaliação é um “processo sistemático de coleta de dados responsável por nos informar o modo como um determinado usuário ou grupo de usuários deve utilizar um produto para uma determinada tarefa em um certo tipo de ambiente” (SHARP; ROGERS; PREECE, 2005). Antes de avançar no desenvolvimento de um produto, é recomendado tentar prever se o público-alvo utilizará tal produto, e se o lançamento deste é viável (SHARP; ROGERS; PREECE, 2005).

Existem diversos tipos de avaliação, como: avaliações analíticas, sem presença de usuários; e avaliações empíricas, que é feita com participação dos usuários (NIELSEN, 1994a).

As avaliações empíricas são baseadas na vivência, experiência e no conhecimento do avaliador, possibilitando avaliar algo mais subjetivo. “Requerem

indiretamente um entendimento acerca das necessidades e da psicologia de cada um deles” (SHARP; ROGERS; PREECE, 2005). Enquanto as avaliações analíticas são baseadas em modelos. Neste tipo de avaliação são definidas condições mais concretas e a avaliação consiste em avaliar se o produto está cumprindo estas condições. São listados a seguir alguns dos principais métodos de inspeção (NIELSEN, 1994a) e (SAURO, 2019):

**Avaliação heurística.** Envolve múltiplos avaliadores examinando uma mesma interface de acordo com heurísticas, ou seja, um conjunto de regras ou princípios. Heurísticas são vistas como um conjunto de regras que servem para avaliadores como lembretes de áreas potencialmente problemáticas.

**Expert review.** Pode ser feita tanto por um avaliador quanto por um grupo. Este método realiza avaliações com diferentes níveis de rigor, normalmente com base em experiência pessoal e conhecimento sobre o produto (SAURO, 2019).

**Guideline review.** É feita por um avaliador guiado por um conjunto de diretrizes bem definidas, verificando a conformidade do produto com essas diretrizes. Em geral, as diretrizes são mais detalhadas do que heurísticas e podem ser vistas mais como *checklists* do que princípios (SAURO, 2019).

Este tipo de inspeção normalmente tem como entrada um *checklist* (NIELSEN, 1994a), que é desenvolvido a partir de uma análise de um ou mais *guidelines* relevantes para o projeto. Estes itens são normalmente derivados de heurísticas de usabilidade para operacionalizar a avaliação.

*Checklists* são listas de itens a serem conferidos se foram cumpridos ou não. Servindo para inspecionar algo de forma rápida. O critério de marcação dos itens é uma resposta do tipo “sim ou não” (p. ex. em uma lista de tarefas, a avaliação de cada item corresponde a responder à pergunta “esta tarefa foi concluída?”). Para facilitar a compreensão e/ou aplicação, os itens também podem ser separados por categoria de acordo com a necessidade.

Um *checklist* pode ser feito de maneira informal, elencando elementos importantes à medida que estes vão sendo lembrados. Este tipo de abordagem costuma ser utilizado quando o *checklist* é feito para ser utilizado de maneira pontual. Uma abordagem mais formal do desenvolvimento de um *checklist* seria realizá-lo de forma sistemática, fazendo uma melhor avaliação do contexto no qual

este será aplicado e refinando-o por meio de um processo de validação (JI et al., 2006).

No contexto de avaliação de usabilidade o *checklist* pode ser desenvolvido com base em guias de estilo. A divisão dos itens pode ser feita por princípio de design ou por elemento do design. No caso do presente trabalho os itens do *checklist* se referem à avaliação do design visual de aplicativos Android. Neste contexto, cada item do *checklist* a ser desenvolvido serve para auxiliar uma avaliação se o aplicativo está de acordo com o critério apresentado naquele item. Isto torna o processo de avaliação mais rápido e homogêneo.

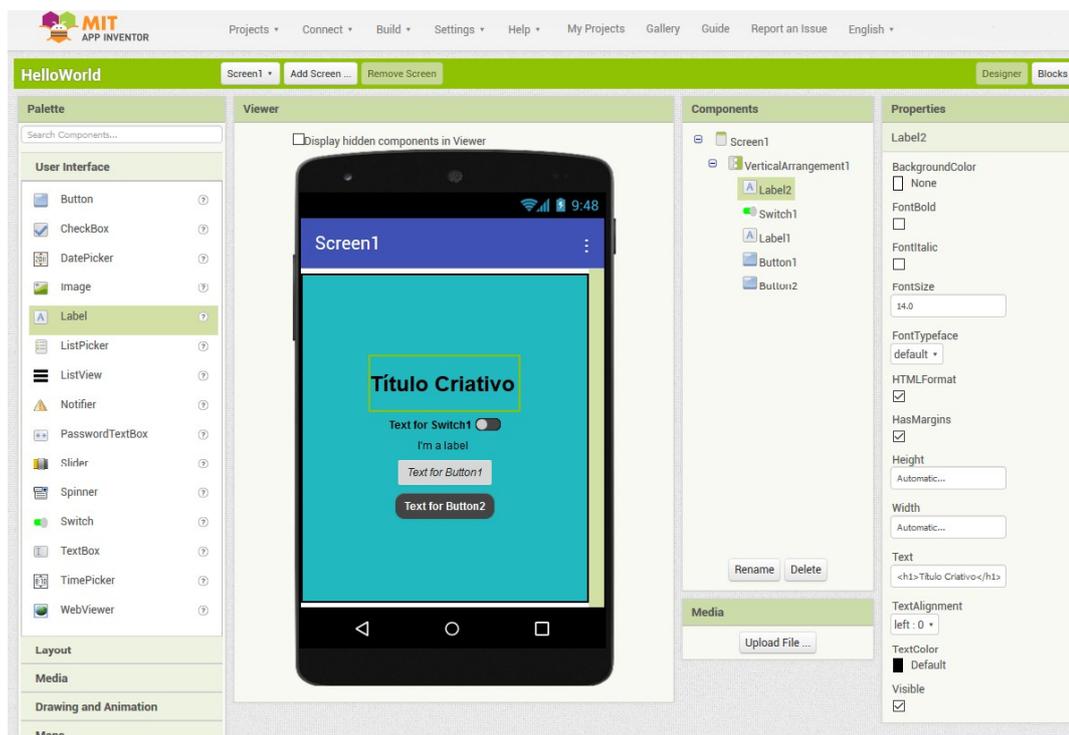
#### 2.4. Ensino de design visual na educação básica

O ensino de programação na educação básica cada vez mais vem sendo feito por meio do desenvolvimento de aplicativos para dispositivos móveis utilizando ambientes baseados em blocos, como o App Inventor. Porém, em geral, os aplicativos desenvolvidos pelos alunos apresentam usabilidade insatisfatória (FERREIRA et al., 2019b). Assim, o ensino de design visual no contexto do ensino de computação na educação básica pode ser utilizado como complemento ao ensino de programação, com o objetivo de estimular a capacidade criativa e divulgar esta área profissional para os jovens (FERREIRA et al., 2019b). Com o ensino de design visual em conjunto com a programação de aplicativos Android, os apps criados pelos alunos tendem a apresentar uma melhor usabilidade quando comparados com outros. O que pode despertar o interesse destes alunos tanto para as áreas de programação, quanto para o design visual.

#### 2.5. App Inventor

O App Inventor é uma ferramenta de desenvolvimento de aplicativos Android, teve sua primeira versão lançada em 2010 pelo Google e atualmente mantida pelo Instituto de Tecnologia de Massachusetts (MIT), lançando em 2013 o App Inventor 2 (Figura 4). Essa ferramenta foi desenvolvida com o objetivo de incentivar o desenvolvimento de aplicativos, permitindo que qualquer pessoa consiga desenvolvê-los, principalmente o público infantil (“App Inventor 2”, 2013).

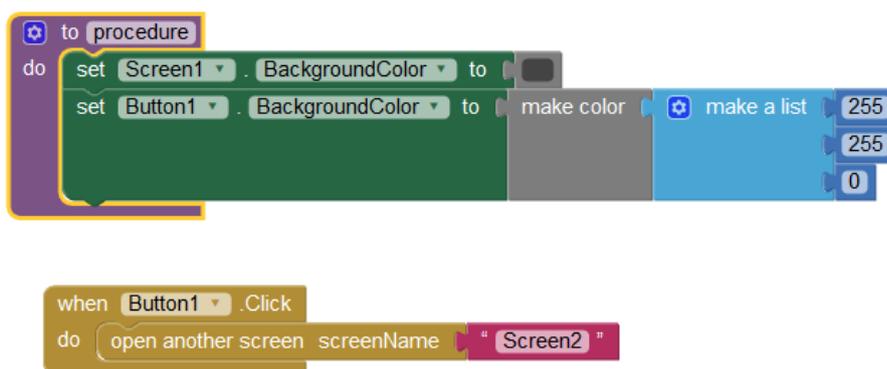
FIGURA 4: INTERFACE DE DESIGN DO APP INVENTOR 2



Fonte: Captura de tela do App Inventor, 2019.

O desenvolvimento de código no App Inventor é feito por meio do uso de blocos (Figura 5), que realizam de forma mais didática a mesma tarefa que um código textual.

FIGURA 5: BLOCOS DE CÓDIGO DO APP INVENTOR 2

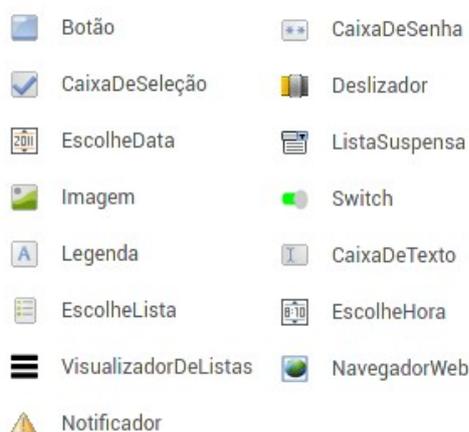


Fonte: Captura de tela do App Inventor, 2019.

O App Inventor permite tanto desenvolver a parte funcional de aplicativos usando programação baseada em blocos, quanto o design de interface destes aplicativos, fornecendo os elementos de UI apresentados na Figura 6. Para inserir um componente, tanto de UI, quanto de blocos de código, o desenvolvedor

simplesmente seleciona o componente em uma lista, arrastando-o para a área desejada.

FIGURA 6: ELEMENTOS DE DESIGN VISUAL DO APP INVENTOR



Fonte: App Inventor, 2019.

Em relação aos elementos de design visual, o App Inventor fornece opções de parametrização conforme apresentadas na Tabela 7.

TABELA 7: ELEMENTOS DO DESIGN VISUAL EM RELAÇÃO AO APP INVENTOR

<b>Layout</b>	Como elementos referentes ao <i>layout</i> existem no App Inventor opções de alinhamento como arranjos verticais, horizontais e em tabela. Além do posicionamento fixo, a documentação avisa que houve uma atualização permitindo posicionar e redimensionar os elementos em relação ao tamanho da tela.
<b>Controles</b>	Estão presentes no App Inventor os principais elementos de controle atualmente utilizados em aplicativos Android.
<b>Cores</b>	Existe uma paleta de cores padrão, mas existem formas de inserir cores utilizando RGB (e.g. Figura a) e RGBA. Permite alterar a cor do texto, do plano de fundo do aplicativo e de alguns elementos (e.g. botões e labels).
<b>Tipografia</b>	O App Inventor fornece pouca variedade de fontes, limitando-se apenas a uma fonte do tipo <i>sans serif</i> , uma <i>serif</i> e uma <i>monospace</i> . Também é possível configurar o tamanho do texto e utilizar os modificadores de negrito e itálico, mas somente englobando todo o texto do elemento.
<b>Imagens</b>	É possível inserir imagens e escolher parâmetros como altura e largura. Também é possível adicionar animações de deslizar.
<b>Escrita</b>	O App Inventor não possui ferramentas integradas para revisão de texto.

Fonte: App Inventor, 2019.

O App Inventor fornece uma interface para visualização e edição do design do aplicativo (Figura 6) e outra para a lógica do aplicativo. Este modelo de divisão entre as interfaces visa tornar o desenvolvimento dos aplicativos mais intuitivo.

### 3. Estado da arte

Para a avaliação do estado da arte no contexto do presente trabalho é realizado um mapeamento sistemático da literatura. O objetivo deste mapeamento é identificar e mapear as publicações relevantes ao escopo deste trabalho por meio de análise temática e classificação do material científico encontrado (PETERSEN et al., 2008).

#### 3.1. Definição do mapeamento

A questão central de pesquisa deste mapeamento é “**Quais *checklists* existem para avaliar o design visual de aplicativos Android desenvolvidos utilizando App Inventor e quais as suas características?**”

Como refinamento da questão de pesquisa, são definidas as seguintes perguntas de análise:

PA1. Quais *checklists* existem?

PA2. Quais as características do *checklist*?

PA3. Como o *checklist* foi desenvolvido?

PA4. Como foi realizada a avaliação do *checklist*?

**Critérios de inclusão.** Este mapeamento é realizado com foco em artigos publicados em periódicos ou anais de conferências. São considerados artigos científicos e artefatos que apresentem *checklist* ou rubrica para avaliação de design visual de aplicativos em dispositivos móveis.

- São incluídos apenas materiais em inglês e português;
- São incluídos apenas materiais acessíveis por meio do portal da CAPES;
- Por existirem poucas pesquisas especificamente voltadas ao App Inventor, a busca foi realizada sem este enfoque específico;
- Por haverem poucos resultados relevantes a pesquisa também foi estendida para *checklists* voltados para o sistema iOS.

**Critérios de exclusão.** Foram listadas as justificativas utilizadas para a remoção de itens que seriam relevantes segundo os critérios de inclusão, porém não foram incluídos nos resultados relevantes.

- O texto completo estar inacessível;
- Artigos duplicados;
- Artigos não relevantes (título, resumo ou palavras-chave não relacionadas aos objetivos deste mapeamento);
- Artigos não voltados para *tablets* ou *smartphones*;
- Não abordar de forma substancial o design visual.

**Crítérios de qualidade.** São considerados apenas materiais que apresentassem informações substanciais alinhadas ao contexto da questão central de pesquisa do presente trabalho.

**Fontes de dados e estratégias de pesquisa.** São considerados materiais disponíveis nas principais bibliotecas digitais do campo da computação (ACM Digital Library<sup>1</sup>, IEEE Xplore Digital Library<sup>2</sup>, ScienceDirect<sup>3</sup>, Scopus<sup>4</sup> e SpringerLink<sup>5</sup>). Com o propósito de ampliar a busca, o Google Scholar<sup>6</sup> é incluído como fonte de dados por indexar um grande conjunto de dados de diversas fontes de produção científica (HADDAWAY et al., 2015). São examinados materiais da literatura secundária (referências presentes nos resultados relevantes). Também foram realizadas buscas informais com o objetivo de obter informações sobre a situação dos *checklists* que estão sendo utilizados fora do ambiente acadêmico. Os resultados obtidos desta maneira estão evidentemente separados nas tabelas apresentadas no presente trabalho.

**Definição da *string* de busca.** A *string* de busca é composta a partir de conceitos relacionados à questão de pesquisa, incluindo sinônimos destes conceitos, como indicado na Tabela 8. Esta *string* foi refinada com base nos resultados obtidos em uma série de buscas de teste, com o objetivo de obter uma quantidade suficiente de resultados. Devido à baixa quantidade de resultados relevantes, a busca foi estendida para envolver o sistema iOS, por ser considerado relevante para o estudo.

1 [www.portal.acm.org](http://www.portal.acm.org)

2 [www.ieeexplore.ieee.org](http://www.ieeexplore.ieee.org)

3 [www.sciencedirect.com](http://www.sciencedirect.com)

4 [www.scopus.com](http://www.scopus.com)

5 [www.springerlink.com](http://www.springerlink.com)

6 [www.scholar.google.com](http://www.scholar.google.com)

TABELA 8: PALAVRAS-CHAVE DO MAPEAMENTO SISTEMÁTICO

Termos	Tradução e sinônimos utilizados (inglês)
<i>Checklist</i>	<i>Checklist, rubric</i>
Aplicativo	<i>Application, app, apps, applications</i>
Android	<i>Android, IOS</i>
<i>Design visual</i>	<i>Visual design, user interface, designing UI, UI design, aesthetics</i>

Fonte: Autoria própria

### String de busca genérica utilizada:

*(checklist OR rubric) AND (design OR designing) AND (“user interface” OR “designing UI” OR “UI design” OR “visual design” OR aesthetics) AND (android OR ios) AND (app OR apps OR application OR applications)*

### 3.2. Execução da busca

A busca foi realizada durante o mês de outubro de 2019, seguindo o protocolo de busca definido na seção 3.1. Mesmo após as buscas teste e a calibração da *string* de busca foram encontrados poucos resultados relevantes. A Tabela 9 mostra os números de resultados iniciais, analisados e potencialmente relevantes obtidos em cada uma das fontes de pesquisa.

TABELA 9: NÚMEROS DE RESULTADOS DA BUSCA

Fonte de dados	Resultado da pesquisa inicial	Resultados analisados	Resultados potencialmente relevantes	Resultados relevantes
ACM	212	100	5	3
Google scholar	~6.110	100	24	4
IEEE	1	1	1	1
ScienceDirect	11	11	1	1
SCOPUS	1	1	1	0
SpringerLink	58	58	3	1
<b>Total</b>	<b>6393</b>	<b>271</b>	<b>35</b>	<b>10</b>

Fonte: Autoria própria

Para a obtenção dos resultados potencialmente relevantes, foram avaliados os resultados iniciais, analisando o título, resumo e palavras-chave. Na etapa seguinte do mapeamento, os resultados potencialmente relevantes foram analisados

na íntegra, buscando por *checklists* que abordassem o design visual de aplicativos mobile. Como resultado foram encontrados 10 *checklists* propostos em artigos, e 4 originados de sites, todos listados na Tabela 10.

TABELA 10: CHECKLISTS ENCONTRADOS

Referência	Nome	Descrição
<b>Encontrados em trabalhos acadêmicos</b>		
Solecki et al. (2019) CnE (2019)	CodeMaster	Rubrica para avaliação de <i>design</i> de interface de aplicativos android desenvolvidos com App Inventor
Patil, Bhole e Shete (2016)	UI Automator Viewer	Ferramenta para avaliação de acessibilidade de aplicativos Android
Olsina, Santos e Lew (2014)	2Q2U 2.0	Definições de qualidade, qualidade no uso, usabilidade e experiência de usuário
Ballantyne et al. (2018) Ballantyne et al. (2019)	<i>A Comprehensive List of Accessibility Guidelines for Mobile Applications</i>	Um compilado de guidelines adaptado para <i>checklist</i> voltado à acessibilidade em aplicativos mobile
Xu et al. (2014)	<i>A Pilot Study of an Inspection Framework for Automated Usability Guideline Reviews of Mobile Health Applications</i>	Avaliação de um <i>framework</i> para <i>Guideline Reviews</i> de aplicativos mobile de saúde
Gómez, Caballero e Sevillano (2014)	<i>Heuristic Evaluation on Mobile Interfaces: A New Checklist</i>	<i>Checklist</i> abrangente para avaliação de interfaces de aplicativos mobile
Barros, Leitão e Ribeiro (2014)	<i>Design and evaluation of a mobile user interface for older adults</i>	Recomendações de usabilidade para o desenvolvimento de aplicativos voltadas para pessoas idosas
Xu (2013)	<i>Tablet application GUI usability checklist</i>	<i>Checklist</i> de avaliação de usabilidade para aplicativos em tablets
Wrede (2018)	<i>The Impact of Mobile Device Usability Guidelines on the Design of Help Documentation for Evernote's iOS Application</i>	<i>Checklist</i> para avaliar a usabilidade da documentação do Evernote
Riegler e Holzmann (2018)	<i>Measuring Visual User Interface Complexity of Mobile Applications With Metrics</i>	Ferramenta para avaliação automática da complexidade de interfaces visuais
<b>Encontrados em sites</b>		
Google (2013)	<i>Accessibility Testing Checklist</i>	(antigo) <i>Checklist</i> de teste de acessibilidade da Google
Babich (2017)	<i>The Guide to Mobile App Design: Best Practices for 2018 and Beyond</i>	Guia de usabilidade para aplicativos mobile
Shaporda ([2017])	<i>Mobile UX Checklist: 13 Things to Keep in Mind for UX Design</i>	Guia de usabilidade para aplicativos mobile
Roundhouse ([2017])	<i>18 Point Checklist For Your Mobile UX Design: Top 18 Keys to a Successful Mobile UX Design</i>	<i>Checklist</i> para auxiliar o desenvolvimento de aplicativos mobile

Fonte: Autoria própria

### 3.3. Extração de informação

Nesta seção são descritas as informações a serem extraídas dos resultados

recuperados na busca. A extração de dados busca evidenciar pontos relevantes para o presente trabalho nos resultados obtidos durante o mapeamento sistemático (seção 3.2). A Tabela 11 apresenta uma breve descrição dos itens a serem destacados em cada artefato obtido na busca. Os itens a serem destacados são referentes às perguntas de análise.

TABELA 11: DESCRIÇÃO DOS DADOS A SEREM EXTRAÍDOS

Pergunta de análise	Dados a extrair	Descrição
PA1. Quais <i>checklists</i> existem?	Nome	O nome do <i>checklist</i>
	Descrição	Breve descrição
	Referência	Referência bibliográfica
PA2. Quais as características do <i>checklist</i> ?	Guia base	O <i>checklist</i> foi feito com base em algum guia? Se sim, qual?
	Elementos avaliados	Cor, tipografia, ícones, imagens, layout, etc.
	Quantidade de itens	Quantidade de itens avaliados no <i>checklist</i>
	Possui ferramenta?	Se possui ferramenta online, e caso possua, a avaliação é automática ou manual?
PA3. Como o <i>checklist</i> foi desenvolvido?	Método de desenvolvimento do <i>checklist</i>	Se é com base em um estudo de caso, se foi sistemático, ou se não foi informado
PA4. Como foi realizada a avaliação do <i>checklist</i> ?	Tipo de estudo de avaliação	Indicação do tipo de estudo ( <i>research design</i> ) da avaliação do <i>checklist</i>
	Fatores avaliados	Indicação dos fatores que foram avaliados no estudo
	Método de coleta de dados	Indicação do(s) método(s) de coleta de dados adotado(s) na avaliação do <i>checklist</i>
	Tamanho de amostra	Quantidade de pontos de dados utilizadas na avaliação do <i>checklist</i>
	Método de análise de dados	Indicação do(s) método(s) de análise de dados utilizado(s) na avaliação do <i>checklist</i>
	Descobertas	Descrição dos principais resultados, pontos fortes e fracos identificados na avaliação do <i>checklist</i>

Fonte: Autoria própria

### 3.4. Discussão

Durante a busca formal, nas fontes de dados acadêmicas, foram encontrados poucos resultados relevantes. Por outro lado, a busca informal utilizou apenas os primeiros resultados encontrados, com o objetivo de obter uma breve noção do que seria encontrado não só como trabalho acadêmico mas também como produtos em *websites*. Nestas buscas foram encontrados mais alguns resultados, porém nenhum destes apontava método desenvolvimento ou estudo de avaliação do *checklist*. Esta seção foca apenas nos resultados obtidos por meio da busca formal descrita na

seção 3.1.

Os guias mais citados foram os da Google e da W3C, sendo que um dos guias da Google é antigo e está sendo contemplado no Material Design (2014). Como descrito na seção 2.2, o principal guia de acessibilidade da W3C para o contexto dos aplicativos é o WCAG 2.0, em conjunto com documentos que guiam sua aplicação para aplicativos *mobile*. O livro de Nielsen (1994b) também foi bastante citado entre os artigos avaliados, trazendo uma definição dos elementos do design visual. Entre os resultados encontrados em artigos, a maior parte dos estudos apresentam referências a livros ou outros artigos relacionados à interação humano computador e design de interfaces. A Tabela 12 lista os guias utilizados como base para o desenvolvimento dos *checklists* avaliados.

Para classificar os itens nos elementos do design visual é apresentada uma interpretação pelo autor de acordo com a definição presente na seção 2.1.1. Em relação aos elementos do design visual, conforme definido na seção 2.1.1, a divisão apresentada no *checklist* difere adicionando uma seção referente à escrita. Entre os *checklists* avaliados, mesmo que a definição de Nielsen (1994b) dos elementos do design visual seja citada, poucos abordam todos os elementos dessa definição. E o único a abordar a escrita foi o Codemaster (CNE, 2019).

Poucos *checklists* abordam diretamente o design visual. Normalmente eles avaliam usabilidade, como p. ex. Olsina, Santos e Lew (2014), focando em heurísticas como controle ao usuário. Alguns focam em acessibilidade, p. ex. Ballantyne et al. (2018), abordando conceitos como tamanho do texto e contraste.

TABELA 12: GUIAS BASE

Ref	Guia base
CodeMaster (2019)	-Material Design -WCAG 2.0
Patil, Bhole e Shete (2016)	- <i>Checklist</i> de acessibilidade da Google, 2013 -WCAG 2.0 -Prayag, S. -Panage
Olsina, Santos e Lew (2014)	-ISO 25010
Ballantyne, M. et al. (2018)	-WCAG 2.0
Xu et al. (2014)	-Health Information Management System Society (HIMSS) usability guidelines -Coarse Grained GUI Model (CGGM)
Gómez, Caballero e Sevillano (2014)	-Nielsen e Molich (1990) -Shneiderman (1987) -Pierotti (2005) -Constantine (2002) -Instone (1997) -Tognazzini (2003) -Hassan et al. (2004) -Olsina, Lafuente e Rossi (2001)
Barros, Leitão e Ribeiro (2014)	-Siek, Rogers e Connelly (1997) -"Navigation, orientation, and gestures for Windows Phone" Microsoft (2013) -"Button control design for Windows Phone" Microsoft (2013)
Xu (2013)	-"iOS Human interface guidelines" Apple (2012) -"User interface Guidelines" Google (2012) -"Designing UX for apps" Microsoft (2012) -"Windows Application UI Development" Microsoft (2012) -BlackBerry PlayBook tablets UI Guidelines (2012)
Wrede (2018)	-Budiu (2015) -Budiu (2016) -Gómez, Caballero e Sevillano (2014) -Meyer (2016) -Moumane, Idri e Abran (2016) -Nayebi (2012) -Nayebi (2013) -Nielsen (2011) -Rahmat, Zulzalil, Ghani e Kamaruddin (2015) -Schade (2015) -Thitichaimongkhon e Senivongs (2016)
Riegler e Holzmann (2018)	-Fu et al. (2007) -Zen and Vanderdonckt (2014) -Sears (2001) -Alemerien and Magel (2014) -Fu et al. (2007) -Comber and Maltby (1995) -Tullis (1981, 1983) -Bonsiepe (1968) -Ma et al. (2013)

Fonte: Autoria própria

A Tabela 13 lista os elementos do design visual que são avaliados em cada um dos *checklists* que fazem parte da extração de dados. Todos abordam o conceito de layout. Controles, cores e tipografia também são abordados pela maioria dos *checklists*. O uso de imagens e a escrita são os elementos menos avaliados. O Apêndice B apresenta um resumo do que é avaliado sobre cada elemento, separado por artigo.

TABELA 13: ELEMENTOS DO DESIGN VISUAL PRESENTES NOS *CHECKLISTS*

Referência	Elementos avaliados					
	Layout	Controles	Cores	Tipografia	Imagens	Escrita
CodeMaster (2019)	x	x	x	x	x	x
Patil, Bhole e Shete (2016)	x	x	x		x	
Olsina, Santos e Lew (2014)	x	x	x	x		
Ballantyne, M. et al. (2018)	x	x	x	x	x	x
Xu et al. (2014)	x	x	x	x		x
Gómez, Caballero e Sevillano (2014)	x	x	x	x	x	x
Barros, Leitão e Ribeiro (2014)	x	x			x	
Xu (2013)	x	x				
Wrede (2018)	x	x	x	x	x	
Riegler e Holzmann (2018)	x		x	x		
<b>TOTAL</b>	10	9	8	7	6	4

Fonte: Autoria própria

A quantidade de itens dos *checklists* avaliados varia de 9 (Riegler; Holzmann, 2018) a 313 (Wrede, 2018). A quantidade de itens pode ser influenciada por fatores como: objetivo do *checklist*, escopo, se é desenvolvido para ser aplicado por uma ferramenta, ou avaliador, entre outros. Os *checklists* que possuíam as maiores quantidades de itens foram criados para fins mais específicos. No caso do Wrede (2018), o *checklist* foi criado considerando exclusivamente a avaliação da documentação do Evernote.

Um ponto percebido no mapeamento foi a diferença entre a formulação dos itens dos *checklists* desenvolvidos para avaliação manual, e os que foram desenvolvidos para serem utilizados com ferramentas automáticas. Em avaliações

por ferramentas automáticas, podem ser avaliados itens mais técnicos (p. ex., conferir se o tamanho da fonte de um texto é 10 ou 12), que serão difíceis para um avaliador humano aferir. Por outro lado, enquanto outros itens podem ser difíceis de serem avaliados pela ferramenta, enquanto um avaliador humano pode facilmente realizar essa identificação (p. ex., uso de uma imagem já distorcida sem alterar a proporção ou estética).

Entre os *checklists* encontrados, existem tanto *checklists* voltados para avaliação automática, quanto para avaliação manual. A Tabela 14 detalha o uso de ferramentas na realização da avaliação de aplicativos nos trabalhos acadêmicos avaliados, e os que não fazem uso de ferramentas. A avaliação manual em alguns casos é realizada por especialistas, como no estudo realizado por Ballantyne et al. (2019). Em outros casos a avaliação é realizada por usuários, como em Barros, Leitão e Ribeiro (2014).

TABELA 14: USO DE FERRAMENTAS NOS *CHECKLISTS*

Referência	Uso de ferramentas
CodeMaster (2019)	Foi utilizada uma ferramenta desenvolvida para realizar a avaliação automática aplicando o <i>checklist</i> .
Patil, Bhole e Shete (2016)	O <i>checklist</i> foi desenvolvido para aprimorar a avaliação realizada por uma ferramenta já existente.
Olsina, Santos e Lew (2014)	Não informado.
Ballantyne, M. et al. (2018)	Quatro especialistas realizaram conjuntos diferentes de tarefas.
Xu et al. (2014)	Foi desenvolvida uma ferramenta para avaliação automática e os resultados obtidos pela ferramenta foram comparados com os resultados gerados por um avaliador especialista.
Gómez, Caballero e Sevillano (2014)	Não informado.
Barros, Leitão e Ribeiro (2014)	A avaliação foi realizada com base no feedback dos usuários.
Xu (2013)	A avaliação foi realizada por 3 especialistas.
Wrede (2018)	Não informado.
Riegler e Holzmann (2018)	Possui uma ferramenta que realiza a avaliação de parâmetros relacionados à complexidade da interface.

Fonte: Autoria própria

Os resultados desta busca apontam a falta de estudos de validação dos *checklists*, sendo normalmente realizados de forma ad-hoc, aplicando o *checklist* em poucos casos sem utilização de um método formal de avaliação. Em geral, os *checklists* encontrados em sites não apresentavam métodos de desenvolvimento, referências, e também não citavam estudos sobre sua validade, como pode ser visto na Tabela 15.

TABELA 15: MÉTODOS DE AVALIAÇÃO

Referência	Método de avaliação
CodeMaster (2019)	Estudo de caso
Patil, Bhole e Shete (2016)	Não informado
Olsina, Santos e Lew (2014)	Ad-hoc
Ballantyne, M. et al. (2018)	Não informado
Xu et al. (2014)	Estudo de caso
Gómez, Caballero e Sevillano (2014)	Ad-hoc
Barros, Leitão e Ribeiro (2014)	Não informado
Xu (2013)	Não informado
Wrede (2018)	Não informado
Riegler e Holzmann (2018)	Ad-hoc

Fonte: Autoria própria

De forma geral os *checklists* são similares na avaliação dos elementos do design visual, mas variam muito em termos de quantidade de itens e em relação ao suporte fornecido. Alguns são feitos para serem aplicados por ferramentas automatizadas, enquanto outros são voltados para avaliação manual.

Como resultado se observa que a maior parte dos *checklists* encontrados avaliam acessibilidade e usabilidade em geral, incluindo apenas alguns itens voltados ao design visual. Parte dos *checklists* encontrados não se baseia em guias de estilo, sendo que alguns são definidos com base na experiência dos autores, ou são compilados dos itens encontrados em outros *checklists*.

Alguns dos artigos encontrados citam o site Developer Android (2019), porém o site foi atualizado, e no momento, a página sobre usabilidade possui apenas links para páginas do Material Design (2019). A versão antiga do site fornecia um *checklist* de usabilidade. Neste *checklist*, considerando o contexto de design visual, foi localizado um item referente ao tamanho mínimo dos controles da interface de usuário.

O resultado encontrado que melhor atendeu aos requisitos do presente

trabalho foi o Codemaster (SOLECKI et al., 2019), por ser desenvolvido sistematicamente e ser formalmente avaliado por meio de critérios estatísticos. Este *checklist* separa os seus itens por elemento do design visual e avalia todos os elementos que estão sendo abordados no presente trabalho. Além dos elementos presentes existe nele o elemento da escrita, que também foi incluído neste trabalho. Porém o contexto educacional exige a avaliação de itens mais complexos que a avaliação automática ainda não é capaz de aferir.

Os *checklists* manuais conseguem abordar estes itens mais complexos (p. ex., legibilidade do texto em relação ao tipo de fonte escolhido), porém requerem um esforço considerável e conhecimento para serem aplicados. Um bom passo inicial para a melhoria das avaliações automáticas é a criação de um *checklist* para avaliação manual dos itens mais difíceis de serem automatizados. Assim os resultados do mapeamento indicam a necessidade de um *checklist* mais completo e sistematicamente devolvido, não se limitando a itens facilmente automatizáveis.

### **Ameaças à validade**

**Viés de publicação.** Uma ameaça comum para mapeamentos sistemáticos, está relacionada à seleção de publicações em que os resultados positivos têm uma chance maior de serem publicados do que os resultados negativos (KITCHENHAM, 2004). Para reduzir este risco, a busca é realizada com foco na existência dos *checklists* e não em suas avaliações. Este risco não deve ter um efeito substancial na pesquisa.

**Omissão de estudos relevantes.** Esta situação pode ocorrer caso a *string* de busca seja muito restritiva, ou caso seja abrangente demais, a ponto de trazer muitos resultados irrelevantes. Para minimizar este risco, foram realizadas buscas em várias fontes acadêmicas, materiais da literatura secundária foram incluídos e também foram realizadas buscas informais com o intuito de calibrar a *string* de busca. Além disso, na definição da *string* de busca, foram utilizados sinônimos para reduzir a chance de omissão de estudos relevantes.

**Seleção dos estudos relevantes.** Outra ameaça ao mapeamento sistemático é uma seleção de estudos incorreta. Para melhorar a precisão da seleção, foram documentados os critérios de inclusão e exclusão utilizados na identificação dos resultados relevantes, e a seleção foi revisada pela orientadora do trabalho.

**Imprecisão na extração dos dados.** É possível que haja uma imprecisão nos dados extraídos, pois quando estes não estão claramente descritos podem ocorrer erros de inferência. Para reduzir a chance de ocorrer este tipo de erro, tentou-se avaliar o material com base na terminologia comumente aceita.

#### 4. Desenvolvimento do *checklist* CodeMaster UI Design 2.0

O *checklist* desenvolvido tem por objetivo avaliar, em termos de qualidade, o design visual de aplicativos Android, com ênfase em aplicativos desenvolvidos na educação básica, utilizando o App Inventor 2 (MIT, 2019).

O ensino de programação na educação básica vem se mostrando uma necessidade cada vez mais importante. Como forma de complementar o ensino de programação, vem o ensino de usabilidade. Visando o apoio ao ensino de usabilidade na educação básica, é proposta a utilização do *checklist* como meio de auxílio ao desenvolvimento e avaliação de aplicativos Android.

##### 4.1. Definição conceitual

Na Educação Básica o ensino de design de UI foca no ensino de *design thinking*, design visual, usabilidade, design de UI, interação humano computador, design centrado no usuário e UX. Porém ainda há uma carência em relação a conceitos mais concretos, como cor, tipografia, etc. (FERREIRA et al., 2019a). Devido a este contexto específico e a características do App Inventor, para o *checklist* desenvolvido, são feitas algumas simplificações referentes às seguintes diretrizes de design visual:

- Uso de botões no formato oval: foi observado em trabalhos anteriores na iniciativa Computação na Escola que o uso do formato oval tem impacto negativo na estética.
- Uso de texto em itálico: o App Inventor atualmente não permite colocar parte do texto de um componente em itálico. Isso dificulta bastante o seu uso de forma pontual.
- Tamanhos de texto: como o checklist considera o contexto da educação básica, neste trabalho é utilizada uma versão simplificada dos tamanhos de fonte em relação ao recomendado pelo Material Design (2014)
- Uso de sinais de pontuação agregados: foi observado em trabalhos anteriores na iniciativa Computação na Escola que o uso de sinais de pontuação agregados tem impacto negativo na estética.

Como base para o *checklist* desenvolvido, foi utilizada a rubrica Codemaster (SOLECKI et al., 2020) (CNE, 2019). A partir deste foram adicionados itens

possivelmente relevantes, de acordo com os resultados da revisão da literatura e guias de estilo como do Material Design (2014) e do WCAG 2.0 (W3C, 2014).

O site do Material Design (2014) apresenta a Tabela 16, contendo um conjunto predefinido de categorias de tamanhos de fontes. O *checklist* desenvolvido, porém, se limita a avaliar o tamanho da fonte de forma simplificada, o uso de serifa e o uso de negrito e itálico.

Como o Material Design indica os tamanhos 14 e 16 para texto regular e texto de botão (Tabela 16), ambos foram definidos como 14sp, usando a fonte sans serif disponível no App Inventor. É dada preferência para o tamanho 14sp por ser o tamanho padrão no App Inventor. No *checklist* estão sendo considerados como títulos os textos de tamanho 20 ou superior que não estejam inseridos em botões. A Tabela 16 ilustra os tamanhos de texto que estão sendo considerados no *checklist*.

TABELA 16: TAMANHOS DE FONTE (SIMPLIFICADO)

	Tamanho	Capitalização do texto
<b>Título</b>	60	Sentença
<b>Título</b>	48	Sentença
<b>Título</b>	34	Sentença
<b>Título</b>	24	Sentença
<b>Título</b>	20	Sentença
Texto regular	14	Sentença
TEXTO DE BOTÃO	14	Letras maiúsculas

Fonte: autoria própria

#### 4.2. Checklist

O *checklist* é composto de um conjunto de critérios, perguntas com respostas associadas a níveis de *performance* de 0 a 2 pontos. Estes critérios estão divididos nas categorias: layout, controle, cores, tipografia, imagens e escrita. Desta forma, segue a divisão de categorias proposta na versão anterior do *checklist* (CNE, 2019). Cada critério está associado às referências que foram utilizadas para a composição daquele item.

Os itens obtidos na revisão da literatura foram analisados identificando os que estão incluídos no *checklist*, incluídos com modificações e os não incluídos. Para os itens que não foram incluídos são apontados os motivos principais por não terem sido incluídos. O motivo de não inclusão mais encontrado é o de subjetividade na avaliação do item. O ideal é que a avaliação dos itens apresente os mesmos resultados com avaliadores diferentes. Para isso os itens devem ser detalhados de forma concreta.

Outro motivo encontrado é não ser aplicável no contexto da educação básica, tanto por exigir detalhes que estão além da aprendizagem neste nível educacional, quanto pela dificuldade de implementação no App Inventor. Limitações do App Inventor também foram um fator de exclusão. Estas, em alguns casos, impediriam que algum tipo de erro fosse cometido, tornando desnecessária a sua avaliação. A Tabela 17 apresenta a versão 2.0 do *checklist* na íntegra.

Esta nova versão do *checklist* é então avaliada quanto à sua validade convergente e discriminante e influência de fatores subjacentes. E os resultados destas avaliações são apresentados na seção seguinte.

TABELA 17: CHECKLIST CODEMASTER - UI DESIGN 2.0

Id	Critério	-	0 pontos	1 ponto	2 pontos	Detalhamento da medição	Fonte (REF)
<b>Layout</b>							
L1	Qual o número mínimo e o número máximo de componentes visíveis em uma tela?	-	min < 2 OU max ≥ 20	min ≥ 2 E max entre 10 e 19	min ≥ 2 E max ≤ 9	Avaliação visual. Desconsiderar itens adicionados dinamicamente	L4. CnE(2019); Gómez, Caballero e Sevillano (2014); Xu et al. (2014); Riegler e Holzmann (2018)
L2	Todos os textos longos (3 linhas ou mais) estão com alinhamento à esquerda?	Não se aplica (não há textos longos)	Não	-	Sim	Avaliação visual. O App Inventor atualmente não possui o alinhamento justificado	
L3	O conteúdo de todos os elementos está dentro os limites do componente pai ou da largura da tela?	-	Não	-	Sim	Avaliação visual. Desconsiderar itens adicionados dinamicamente, caso exista scroll vertical	Patil, Bhole e Shete (2016); Ballantyne, M. et al. (2018); Xu et al. (2014); Wrede (2018)
L4	Algum espaço entre componentes visíveis ou bordas da tela ocupa mais do que 35% da tela?	-	Não	-	Sim	Avaliação visual ou conferir no código, caso necessário. Em largura ou altura	Xu et al. (2014); Riegler e Holzmann (2018); Gómez, Caballero e Sevillano (2014)
L5	Todo o texto possui margens laterais ( <i>padding</i> ) de no mínimo a largura de uma letra "n"?	Não se aplica (não há texto)	Não	-	Sim	Avaliação visual	Gómez, Caballero e Sevillano (2014)
L6	Todos os botões de voltar estão no canto superior esquerdo da tela?	Não se aplica (não há botões de voltar)	Não	-	Sim	Avaliação visual	Ballantyne, M. et al. (2018)
<b>Controles</b>							
C1	Todos os componentes alvos de toque têm largura e altura maior ou igual a 48x48 pixels dentro da área da tela?	-	Não	-	Sim	Avaliação visual ou conferir no código, caso necessário	L1. CnE(2019); Olsina, Santos e Lew (2014); Xu et al. (2014); Barros, Leitão e Ribeiro (2014); Riegler e Holzmann (2018); Xu (2013); Wrede (2018)
C2	Botões agrupados na mesma interface sempre têm o mesmo formato (retangular/arredondado)?	Não se aplica (não há botões agrupados)	Não	-	Sim	Ex: Todos os botões têm bordas arredondadas?	L2. CnE(2019); Wrede (2018)
C3	Botões agrupados na mesma interface têm sempre o mesmo tamanho independente de seu conteúdo?	Não se aplica (não há botões agrupados)	Não	-	Sim	Avaliação visual	L3. CnE(2019); Wrede (2018)
C4	Existem botões ou menus disponíveis na mesma posição em mais de uma tela?	Não se aplica (só há uma tela)	Não	-	Sim	Avaliação visual	Olsina, Santos e Lew (2014); Xu et al. (2014); Riegler e Holzmann (2018); Wrede (2018)
C5	Algum botão possui formato oval? (exceto ícone)	Não se aplica (não há botões)	Sim	-	Não	Avaliação visual	
<b>Cores</b>							
O1	Quantas cores são usadas no	-	4 ou mais, ou	3	1 ou 2	Avaliação automática utilizando o	C1. CnE(2019);

	aplicativo (além de preto, branco e cinza)?		nenhuma			Codemaster	Wrede (2018); Riegler e Holzmann (2018); Gómez, Caballero e Sevillano (2014)
O2	Qual o contraste entre texto e cor de fundo?	-	Ruim (Insuficiente)	Aceitável (Nível AA na WCAG)	Bom (Nível AAA na WCAG)	Avaliação automática utilizando o Codemaster	C2. CnE(2019); Patil, Bhole e Shete (2016); Olsina, Santos e Lew (2014); Ballantyne, M. et al. (2018); Xu et al. (2014); Riegler e Holzmann (2018); Wrede (2018)
O3	Usam-se apenas cores da paleta do Material Design?	-	Não	-	Sim	Avaliação automática utilizando o Codemaster	C3. CnE(2019)
O4	As tonalidades de cores usadas são harmônicas entre si (complementares, análogas ou triádicas)?	-	Não	-	Sim	Avaliação automática utilizando o Codemaster	C4. CnE(2019)
O5	Qual a maior quantidade de cores usadas no texto em uma tela?	Não se aplica (não há texto)	4 ou mais	3	Até 2	Avaliação visual	Olsina, Santos e Lew (2014)
O6	Qual a maior quantidade de níveis de intensidade de cada cor foi utilizada?	-	4 ou mais		Até 3	Avaliação visual. A intensidade da cor pode ser vista como a quantidade de cinza presente na cor. Se houver imagem com gradiente, considerar como 2 níveis	Gómez, Caballero e Sevillano (2014)
<b>Tipografia</b>							
T1	Todos os componentes usam família da fonte sem serifa?	Não se aplica (não há texto)	Não	-	Sim	Avaliação visual	T1. CnE(2019); Wrede (2018)
T2	Todos os títulos possuem tamanho 60, 48, 34, 24 ou 20pt?	Não se aplica (não há títulos)	Não	-	Sim	Conferir no App Inventor. Textos com tamanho 20 ou maior considerados títulos	Olsina, Santos e Lew (2014); Ballantyne, M. et al. (2018)
T3	Qual a maior quantidade de tamanhos de texto em uma tela?	Não se aplica (não há texto)	5 ou mais	-	De 1 a 4	Conferir no App Inventor	Olsina, Santos e Lew (2014); Xu et al. (2014)
T4	Todo o texto regular possui tamanho 14?	Não se aplica (não há texto regular)	Não	-	Sim	Conferir no App Inventor. Textos fora de botões, com tamanho menor que 20 são considerados texto regular	Olsina, Santos e Lew (2014); Ballantyne, M. et al. (2018); Wrede (2018)
T5	Todos os botões com texto têm tamanho da fonte igual a 14?	Não se aplica (não há botões com texto)	Não	-	Sim	Conferir no App Inventor	T2. CnE(2019); Olsina, Santos e Lew (2014); Wrede (2018)
T6	Existe algum elemento com texto todo em itálico?	Não se aplica (não há texto)	Sim	-	Não	Avaliação visual	T4. CnE(2019); Wrede (2018)
T7	Existe algum texto alinhado à direita?	Não se aplica (não há texto)	sim	-	Não	Avaliação visual	Gómez, Caballero e Sevillano (2014)
<b>Imagens</b>							
I1	Todos os ícones são do mesmo estilo (desconsiderando as cores)?	Não se aplica (não há ícones)	Não	-	Sim	Avaliação visual	I1. CnE(2019)
I2	Todos os ícones de elementos agrupados utilizam o mesmo esquema de cores?	Não se aplica (não há ícones)	Não	-	Sim	Avaliação visual	I1. CnE(2019)
I3	Existe alguma imagem ou ícone	Não se aplica (não	Sim	-	Não	Avaliação visual	I2. CnE(2019);

	<b>pixelizado?</b>	há imagens, nem ícones)					Gómez, Caballero e Sevillano (2014); Wrede (2018)
14	<b>Existe algum botão com imagem distorcida ou ícones distorcidos?</b>	Não se aplica (não há botões com imagen, nem ícones)	Sim	-	Não	Avaliação visual ou conferir no código, caso necessário	I3. CnE(2019)
15	<b>Existe algum componente do tipo imagem distorcido?</b>	Não se aplica (não há componentes do tipo imagem)	Sim	-	Não	Avaliação visual ou conferir no código, caso necessário, desconsiderando imagens de fundo e ícones	I4. CnE(2019)
16	<b>Existe alguma imagem de fundo de tela distorcida?</b>	Não se aplica (não há imagens de fundo)	Sim	-	Não	Avaliação visual ou conferir no código, caso necessário	I5. CnE(2019)
17	Existe alguma imagem exibida com tamanho maior que o tamanho da tela?	Não se aplica (não há imagens)	Sim	-	Não	Avaliação visual ou conferir no código, caso necessário	Patil, Bhole e Shete (2016); Wrede (2018)
18	Existe alguma imagem de texto?	Não se aplica (não há imagens)	Sim	-	Não	Avaliação visual	Ballantyne, M. et al. (2018)
19	Os ícones possuem contraste de no mínimo 3:1 com o fundo?	Não se aplica (não há ícones)	Não	-	Sim	Avaliação visual	Wrede (2018)
<b>Escrita</b>							
E1	<b>O texto de todos os botões está todo em maiúsculas?</b>	Não se aplica (não há botões com texto)	Não	-	Sim	Avaliação visual	E1. CnE(2019)
E2	<b>Todas as sentenças começam com letra maiúscula ou dígito?</b>	Não se aplica (não há texto)	Não	-	Sim	Avaliação visual	E2. CnE(2019)
E3	<b>Existem labels ou checkbox que terminam com pontuação (.,:;!?)?</b>	Não se aplica (não há labels, nem checkboxes com texto)	Pontuação diferente de interrogação	Apenas com interrogação	Não	Avaliação visual	E4. CnE(2019)
E4	<b>Existem sentenças que terminam com "." (ponto) (desconsiderando meio de parágrafo)?</b>	Não se aplica (não há texto)	Não	-	Sim	Avaliação visual	E5. CnE(2019)
E5	<b>O texto de botão mais longo tem quantos caracteres?</b>	Não se aplica (não há botões com texto)	15 ou mais	De 8 a 14	7 ou menos	Avaliação visual	E6. CnE(2019); Wrede (2018)
E6	O texto de item de menu mais longo tem quantos caracteres?	Não se aplica (não há itens de menu com texto)	20 ou mais	De 13 a 19	12 ou menos	Avaliação visual	Gómez, Caballero e Sevillano (2014); Wrede (2018)
E7	Existe algum caso de sinais de pontuação iguais agregados?	Não se aplica (não há texto)	Sim	-	Não	Avaliação visual	Wrede (2018)
E8	Existe algum texto todo em letras maiúsculas fora de botões (Exceto siglas)?	Não se aplica (não há texto fora de botões)	Sim	-	Não	Avaliação visual	Gómez, Caballero e Sevillano (2014); Wrede (2018)

Fonte: autoria própria

## 5. Avaliação do checklist

A rubrica proposta é avaliada de diversos pontos de vista. Nesta avaliação são analisadas as seguintes perguntas de análise:

PA1. O *checklist* CodeMaster: UI Design 2.0 está avaliando o design visual dos aplicativos em termos da *face validity*?

PA2. Qual o *interrater agreement* em relação a avaliação manual usando o *checklist* CodeMaster: UI Design 2.0?

PA3. O *checklist* CodeMaster: UI Design 2.0 apresenta consistência interna?

PA4. O *checklist* CodeMaster: UI Design 2.0 apresenta validade convergente e discriminante?

PA5. Qual a influência dos fatores subjacentes nas respostas aos critérios do *checklist* CodeMaster: UI Design 2.0?

As perguntas PA1 e PA2 são analisadas por meio de um painel de especialistas e as PA3, PA4 e PA5 são analisadas por estudo de caso. Nas seguintes seções são apresentadas as avaliações de acordo com as perguntas de pesquisa.

### 5.1. Avaliação por painel de especialistas

#### 5.1.1. Definição da avaliação por painel de especialistas

A *face validity* é utilizada para responder a PA1, sendo uma avaliação superficial que serve como indicativo para a questão “o *checklist* está avaliando o que deveria avaliar?”. As respostas positivas obtidas no painel de especialistas são indícios de que sim, o checklist está avaliando aquilo que se propõe a avaliar.

O *interrater agreement*, questionado na PA2, é a medida da concordância entre diferentes avaliadores. É avaliado utilizando o critério Kappa, sendo interpretado da seguinte forma: valores não positivos indicam que não há concordância, valores de 0,01 a 0,20 como baixa, 0,21 a 0,40 suficiente, 0,41 a 0,60 moderada, 0,61 a 0,8 substancial e 0,81 a 1,00 praticamente perfeita (MCHUGH,

2012).

### 5.1.2. Execução da avaliação por painel de especialistas

A avaliação por painel de especialistas foi realizada à distância, sendo divulgada no dia 30 de junho de 2020 e encerrada no dia 17 de julho de 2020. 28 avaliadores foram convidados a participar do painel de especialistas. Destes, 13 responderam o questionário representando uma taxa de resposta de 46%.

Tabela 18: ANÁLISE DEMOGRÁFICA REFERENTE AO PAINEL DE ESPECIALISTAS

Nível de conhecimento na área de UI design	Já realizou pesquisa com UI design	5
	Cursou disciplina relacionada	5
	Não possui experiência na área	2
	Finalizando mestrado em design	1
Nível de formação (concluído)	Ensino médio	4
	Graduação	2
	Mestrado	5
	Doutorado	2
	Especialização EAD e Novas Tecnologias	1
Já criou algum aplicativo utilizando App Inventor?	Sim	11
	Não	2

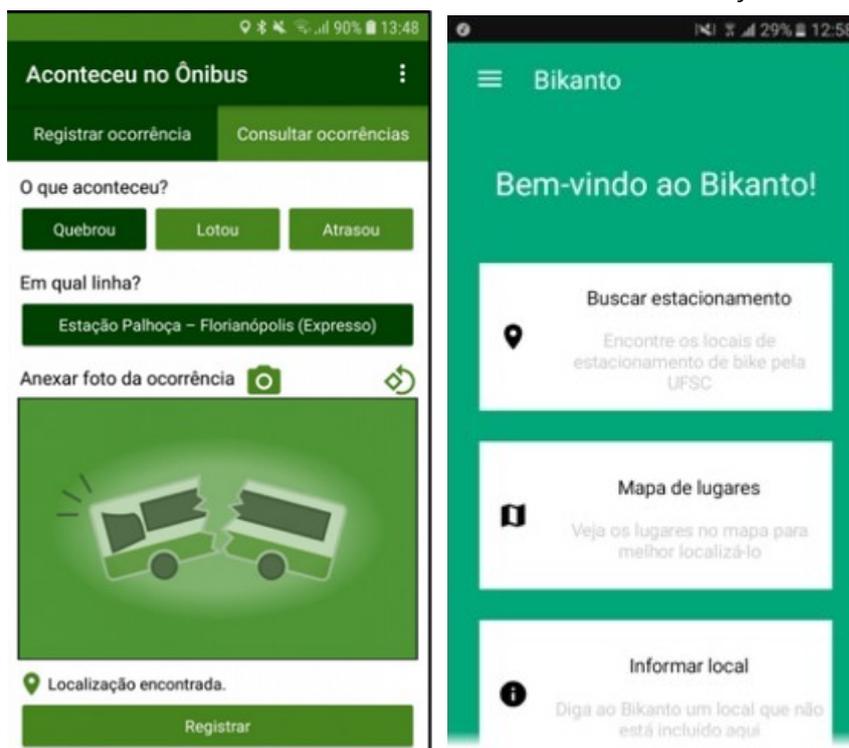
Fonte: autoria própria

O perfil demográfico dos participantes é resumido na Tabela 18. Entre os 13 participantes apenas dois destes não possuíam experiência na área e não haviam criado nenhum aplicativo utilizando o App Inventor.

Os avaliadores foram convidados a realizar a avaliação de um aplicativo utilizando o *checklist*. Após a realização desta avaliação é feita a avaliação do *checklist*, com questões sobre corretude, relevância, clareza e aplicabilidade na educação básica de cada um dos itens. Além disso, haviam questões avaliando a estrutura e abrangência do *checklist*.

Cada avaliador aferiu um aplicativo diferente, sendo que os aplicativos selecionados são aplicativos desenvolvidos na iniciativa Computação na Escola (Figura 7). A avaliação do *checklist* no painel de especialistas foi positiva, com algumas indicações de correções pontuais, principalmente quanto à clareza de alguns itens.

FIGURA 7: EXEMPLOS DE APLICATIVOS DA INICIATIVA COMPUTAÇÃO NA ESCOLA



Fonte: Computação na Escola

### 5.1.3. Análise dos dados do painel de especialistas

PA1. O *checklist* CodeMaster: UI Design 2.0 está avaliando o design visual dos aplicativos em termos da *face validity*?

As respostas obtidas por meio do painel de especialistas ajudaram a identificar itens do *checklist* que precisam de correções. O item “L2 – Todas as telas possuem componentes com alinhamento centralizado horizontalmente ou alinhado à esquerda?” foi alterado, pois a sua avaliação estava confusa e não representou melhoria para o design visual. Em substituição ao item removido, foi adicionado um item referente ao alinhamento de textos longos. Os seguintes itens foram modificados em relação à sua formulação para melhorar sua clareza:

- L1 – Qual o número mínimo e o número máximo de elementos em uma tela? (desconsiderando itens adicionados dinamicamente)
- L4 – Algum espaço entre elementos ou bordas da tela ocupa mais do que 35% da tela?
- L5 – Todo o texto possui margens (*padding*) de no mínimo 5px?
- O2 – Qual é o nível WCAG do aplicativo em relação ao contraste do texto?

- E3 – Existem *labels* que terminam com pontuação?

Em relação à *face validity*, as respostas obtidas apontam que o *checklist* está coerente em relação ao que se deseja avaliar. Sua estrutura e abrangência aparentam estar compatíveis com o esperado.

PA2. Qual o *interrater agreement* em relação a avaliação manual usando o *checklist* CodeMaster: UI Design 2.0?

O *interrater agreement* é avaliado utilizando os dados das avaliações realizadas no painel de especialistas em relação a avaliações realizadas pelo autor, avaliando os mesmos aplicativos que foram avaliados durante o painel de especialistas. A Tabela 19 apresenta o valor do Cohen's Kappa obtido para cada questão do *checklist*.

Como resultado, observa-se que 18 dos 41 itens apresentaram baixa concordância em relação ao *interrater agreement*. O baixo nível de concordância obtido nestes itens indica que diferentes avaliadores responderam aos itens de forma diferente para os mesmos aplicativos. Como isso é um resultado indesejado, foram realizadas alterações no *checklist* como melhorar as descrições destes itens e adicionar opções de “não se aplica” para alguns dos itens.

Houve uma falha no formulário e o item O1, na Tabela 19, referente à quantidade de cores, foi desconsiderado na avaliação do *interrater agreement* devido a essa falha. Observando esta falta de *interrater agreement*, sugere-se um estudo mais aprofundado, com participação de mais avaliadores; ou a automatização da avaliação, o que poderia eliminar essa questão.

Com base nesta avaliação inicial de *face validity* são adicionadas opções de “não se aplica”; também são realizadas alterações nos critérios dos itens L2, L5, O5, O6, T3, I9, E3, E5 e E6; e nas descrições dos itens E2, E7 e E8. Além disso, os itens O1, O2, O3 e O4, da categoria de cores, passaram a ser avaliados de forma automática utilizando a versão anterior do CodeMaster (CNE, 2019). Decidiu-se automatizar estes itens pois houve muita dificuldade em estabelecer critérios concretos e de fácil entendimento.

TABELA 19: COHEN'S KAPPA E NÍVEL DE CONCORDÂNCIA

	Item	Cohen's Kappa	Nível de concordância
L1	Número de elementos	0.23	Suficiente
L2	Alinhamento	<b>0.05</b>	<b>Baixa</b>
L3	Visibilidade	<b>0.00</b>	<b>Inexistente</b>
L4	Ocupação da tela	<b>-0.12</b>	<b>Inexistente</b>
L5	Margens do texto	<b>0.00</b>	<b>Inexistente</b>
L6	Posicionamento de botões	0.62	Substancial
C1	Tamanho mínimo	<b>0.20</b>	<b>Baixa</b>
C2	Formato dos controles	0.79	Substancial
C3	Padronização de tamanho	0.41	Moderada
C4	Consistência nos controles	0.31	Suficiente
C5	Botões com formato oval	1.00	Praticamente perfeita
O1	Quantidade de cores	-	<b>Não medido</b>
O2	Contraste do texto	<b>0.16</b>	<b>Baixa</b>
O3	Paleta de cores	<b>0.09</b>	<b>Baixa</b>
O4	Harmonia de cores	<b>-0.17</b>	<b>Inexistente</b>
O5	Quantidade de cores do texto	0.43	Moderada
O6	Níveis de intensidade de cores	1.00	Praticamente perfeita
T1	Família da fonte	1.00	Praticamente perfeita
T2	Padronização do tamanho dos títulos	<b>0.18</b>	<b>Baixa</b>
T3	Quantidade de tamanhos de texto	<b>0.00</b>	<b>Inexistente</b>
T4	Tamanho do texto regular	0.41	Moderada
T5	Tamanho de texto dos botões	0.67	Substancial
T6	Uso de itálico	<b>0.00</b>	<b>Inexistente</b>
T7	Alinhamento do texto	1.00	Praticamente perfeita
I1	Estilo dos ícones	0.53	Moderada
I2	Esquema de cores dos ícones	0.40	Suficiente
I3	Uso de imagens pixelizadas	<b>0.06</b>	<b>Baixa</b>
I4	Distorção de imagens (botões e ícones)	<b>0.16</b>	<b>Baixa</b>
I5	Distorção de imagens	0.33	Suficiente
I6	Distorção de imagens (plano de fundo)	<b>0.00</b>	<b>Inexistente</b>
I7	Ajuste de tamanho	0.21	Suficiente
I8	Uso de imagens de texto	0.40	Suficiente
I9	Contraste de ícones	0.56	Moderada
E1	Uso de caixa-alta em botões	0.83	Praticamente perfeita
E2	Capitalização do texto	<b>-0.12</b>	<b>Inexistente</b>
E3	Uso indevido de sinais de pontuação	0.26	Suficiente
E4	Pontuação	0.41	Moderada
E5	Quantidade de caracteres em botões	0.87	Praticamente perfeita
E6	Quantidade de caracteres em menus	<b>-0.14</b>	<b>Inexistente</b>
E7	Sinais de pontuação agregados	<b>0.11</b>	<b>Baixa</b>
E8	Uso de caixa-alta fora de botões	<b>0.06</b>	<b>Baixa</b>

Fonte: autoria própria

## 5.2. Avaliação por estudo de caso

Após a aplicação do *checklist* em uma quantidade suficiente de avaliações, é realizada uma análise estatística mais detalhada com base nos resultados destas avaliações.

### 5.2.1. Definição da avaliação por estudo de caso

Para buscar responder as perguntas de análise PA3, PA4 e PA5 é realizado um estudo de caso. Esse estudo é realizado aplicando o *checklist* na avaliação de um conjunto de 400 aplicativos. Nele são realizadas diferentes formas de análise de acordo com cada pergunta de análise.

PA3. O *checklist* CodeMaster: UI Design 2.0 apresenta consistência interna?

A confiabilidade do *checklist* é avaliada com a medição da consistência interna por meio do Alfa de Cronbach (CRONBACH, 1951). No caso deste *checklist*, todos os seus itens devem medir o mesmo fator: a conformidade do design de interface com as diretrizes de design visual. Portanto, espera-se que os itens avaliados pelo *checklist* apresentem um certo nível de consistência interna. Para isto, o valor do alfa de Cronbach entre 0,7 e 0,8 é considerado aceitável, de 0,8 a 0,9 é bom e superior a 0,90 é considerado excelente (DEVELLIS, 2003).

PA4. O *checklist* CodeMaster: UI Design 2.0 apresenta validade convergente e discriminante?

A validade convergente é avaliada por meio das intercorrelações entre os critérios, que é verificada pela análise da correlação policórica, que é a mais indicada para o caso de dados ordinais (OLSSON, 1979). E a validade discriminante é verificada pela correlação item-total. A correlação entre critérios é considerada satisfatória se seu coeficiente estiver acima de 0,29 (COHEN, 1998).

PA5. Qual a influência dos fatores subjacentes nas respostas aos critérios do *checklist* CodeMaster: UI Design 2.0?

Para a verificação da influência dos fatores subjacentes, inicialmente é realizado um teste do índice Kaiser-Meyer-Olkin (KMO). Neste teste, um valor inferior a 0,5 indica que não é possível realizar uma análise fatorial no conjunto de dados (BROWN, 2006). Verificada a possibilidade, é realizada uma análise fatorial

(GLORFELD, 1995). É utilizado o método de rotação Oblimin, no qual os fatores podem ser correlacionados (JACKSON, 2005).

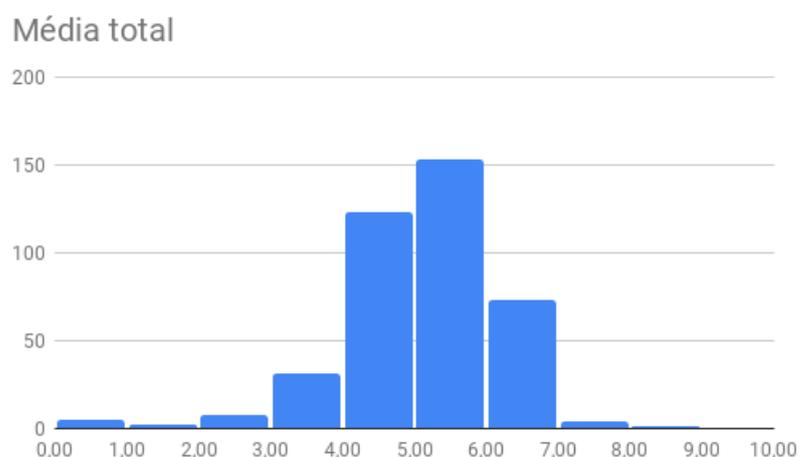
### 5.2.2. Execução da avaliação por estudo de caso

Para a realização do estudo de caso foram avaliados 400 aplicativos utilizando o *checklist*. Estas avaliações ocorreram entre junho de 2020 e fevereiro de 2021. Sendo que sua maioria foi realizada pelo autor, e as avaliações feitas durante o painel de especialistas também foram contabilizadas.

Quanto aos aplicativos avaliados, aqueles que foram utilizados durante o painel de especialista tiveram origem na iniciativa Computação na Escola (CNE, 2019), enquanto os demais fazem parte de um conjunto com 88 mil aplicativos da galeria do App Inventor. Foi utilizado um critério de haver no mínimo um botão e mais algum outro elemento para descartar aplicativos que fossem simples demais.

A Figura 8 apresenta um somatório da pontuação atribuída a cada item do *checklist* atribuindo notas no intervalo de 0 a 10. Observa-se que a grande maioria dos aplicativos obteve notas entre 4 e 7. Esta nota utiliza uma média aritmética simples, não definindo pesos para o quanto cada fator contribui para o design visual do aplicativo. Como foram evitados aplicativos muito triviais (com poucos componentes), não há tantos com nota abaixo de 3.

FIGURA 8: DISTRIBUIÇÃO DAS NOTAS ATRIBUÍDAS PELO *CHECKLIST*



Fonte: autoria própria

### 5.2.3. Análise dos dados do estudo de caso

Em diversos itens do *checklist*, é considerada a possibilidade daquele item não ser aplicável. Como p. ex. um aplicativo pode não possuir imagens, o que acaba impossibilitando a avaliação dos itens referentes a imagens. Isso é agravado por vários aplicativos serem jogos, ou aplicativos muito simples. A Tabela 20 apresenta a quantidade de valores faltantes, não aplicáveis, que foram atribuídos para cada item do *checklist* durante as avaliações.

As respostas marcadas como “não aplicável” foram consideradas como uma nova categoria de resposta para as análises realizadas no estudo de caso. E no cálculo das notas foi atribuído valor zero para estas respostas. Isso foi feito, em vez de remover aquele item do somatório, pois foi considerado que a ausência dos componentes também influencia no design visual.

TABELA 20: VALORES FALTANTES POR ITEM

	Item	Quantidade de valores faltantes
L1	Número de elementos	0
L2	Alinhamento	331
L3	Visibilidade	0
L4	Ocupação da tela	0
L5	Margens do texto	15
L6	Posicionamento de botões	327
C1	Tamanho mínimo	0
C2	Formato dos controles	130
C3	Padronização de tamanho	131
C4	Consistência nos controles	254
C5	Botões com formato oval	19
O1	Quantidade de cores	0
O2	Contraste do texto	0
O3	Paleta de cores	0
O4	Harmonia de cores	0
O5	Quantidade de cores do texto	16
O6	Níveis de intensidade de cores	0
T1	Família da fonte	23
T2	Padronização do tamanho dos títulos	229
T3	Quantidade de tamanhos de texto	16
T4	Tamanho do texto regular	117
T5	Tamanho de texto dos botões	55
T6	Uso de itálico	16
T7	Alinhamento do texto	16
I1	Estilo dos ícones	264
I2	Esquema de cores dos ícones	266
I3	Uso de imagens pixelizadas	110
I4	Distorção de imagens (botões e ícones)	227
I5	Distorção de imagens	234
I6	Distorção de imagens (plano de fundo)	268
I7	Ajuste de tamanho	115
I8	Uso de imagens de texto	112
I9	Contraste de ícones	262
E1	Uso de caixa-alta em botões	60
E2	Capitalização do texto	19
E3	Uso indevido de sinais de pontuação	49
E4	Pontuação	23
E5	Quantidade de caracteres em botões	58
E6	Quantidade de caracteres em menus	370
E7	Sinais de pontuação agregados	20
E8	Uso de caixa-alta fora de botões	78

Fonte: autoria própria

PA3. O *checklist* CodeMaster: UI Design 2.0 apresenta consistência interna?

O alfa de Cronbach considerando todos os itens do *checklist* CodeMaster: UI Design 2.0 é de **0,81**. Este valor, por estar entre 0,8 e 0,9, é considerado bom, indicando que o *checklist* apresenta consistência interna (DEVELLIS, 2003). A Tabela 21 aponta como ficaria o alfa de Cronbach caso ocorresse a remoção de cada um dos itens. Além disso, também é apresentada a correlação item-total para cada item.

O alfa de Cronbach obtido foi um pouco inferior ao da versão anterior do CodeMaster, que era de 0,84. Possivelmente essa redução no valor está relacionada aos valores faltantes e ao aumento da quantidade de itens do *checklist*. A Tabela 21 aponta que com a remoção dos itens L4 e O4, o valor do alfa pode aumentar. Estes itens já existiam na versão anterior do CodeMaster, porém o item L4 foi modificado, e o item O4 estava sendo avaliado automaticamente.

No caso do item L4, referente à ocupação da tela, decidiu-se manter o item no *checklist* por entender que o ponto avaliado por ele é bastante relevante para o design visual do aplicativo. O item O4, referente à harmonia das cores também apresentou correlação item-total negativa e aumento no alfa de Cronbach caso removido. Acredita-se que a categoria de cores é a mais afetada pela característica de muitos dos aplicativos da galeria do App Inventor serem apenas aplicativos de teste, criados sem uma preocupação com o design visual. Considerando isso, e o fato de o aumento no alfa ser muito pequeno, foi decidido manter o item.

TABELA 21: CORRELAÇÃO ITEM-TOTAL E ALFA DE CRONBACH COM REMOÇÃO DE ITENS

	Item	Correlação item-total	Alfa de Cronbach removendo o item
L1	Número de elementos	0,11	0,81
L2	Alinhamento	0,34	0,80
L3	Visibilidade	0,22	0,80
<b>L4</b>	<b>Ocupação da tela</b>	<b>-0,14</b>	<b>0,82</b>
L5	Margens do texto	0,38	0,80
L6	Posicionamento de botões	0,40	0,80
C1	Tamanho mínimo	0,01	0,81
C2	Formato dos controles	0,21	0,80
C3	Padronização de tamanho	0,17	0,81
C4	Consistência nos controles	0,42	0,80
C5	Botões com formato oval	0,11	0,81
O1	Quantidade de cores	0,23	0,80
O2	Contraste do texto	0,15	0,81
O3	Paleta de cores	0,00	0,81
<b>O4</b>	<b>Harmonia de cores</b>	<b>-0,04</b>	<b>0,81</b>
O5	Quantidade de cores do texto	0,50	0,80
O6	Níveis de intensidade de cores	0,17	0,80
T1	Família da fonte	0,30	0,80
T2	Padronização do tamanho dos títulos	0,40	0,80
T3	Quantidade de tamanhos de texto	0,42	0,80
T4	Tamanho do texto regular	0,41	0,80
T5	Tamanho de texto dos botões	0,47	0,79
T6	Uso de itálico	0,30	0,80
T7	Alinhamento do texto	0,29	0,80
I1	Estilo dos ícones	0,35	0,80
I2	Esquema de cores dos ícones	0,37	0,80
I3	Uso de imagens pixelizadas	0,28	0,80
I4	Distorção de imagens (botões e ícones)	0,24	0,80
I5	Distorção de imagens	0,31	0,80
I6	Distorção de imagens (plano de fundo)	0,28	0,80
I7	Ajuste de tamanho	0,34	0,80
I8	Uso de imagens de texto	0,55	0,79
I9	Contraste de ícones	0,32	0,80
E1	Uso de caixa-alta em botões	0,22	0,80
E2	Capitalização do texto	0,26	0,80
E3	Uso indevido de sinais de pontuação	0,32	0,80
E4	Pontuação	0,39	0,80
E5	Quantidade de caracteres em botões	0,40	0,80
E6	Quantidade de caracteres em menus	0,08	0,81
E7	Sinais de pontuação agregados	0,36	0,80
E8	Uso de caixa-alta fora de botões	0,40	0,80

Fonte: autoria própria.

PA4. O *checklist* CodeMaster: UI Design 2.0 apresenta validade convergente e discriminante?

Na Tabela 22 pode-se observar que há uma alta correlação interna entre a maioria dos itens nas categorias de tipografia, imagens e escrita. Enquanto nas categorias de layout, controles e cores são menos itens que apresentam maior correlação. É observada também uma certa correlação entre os itens das categorias de tipografia e escrita, o que sugere que essas categorias poderiam ser unificadas.

Existem diversos fatores que podem influenciar na correlação entre os itens, como p. ex. o próprio layout do App Inventor. Certas opções para customizar os componentes dos aplicativos por estarem juntas acabam sendo usadas em conjunto.

Alguns dos itens possuem características referentes a mais de uma categoria, como os itens L2 e L5, que estão na categoria de layout, mas por se referirem ao texto, acabam tendo uma forte correlação com tipografia e escrita.

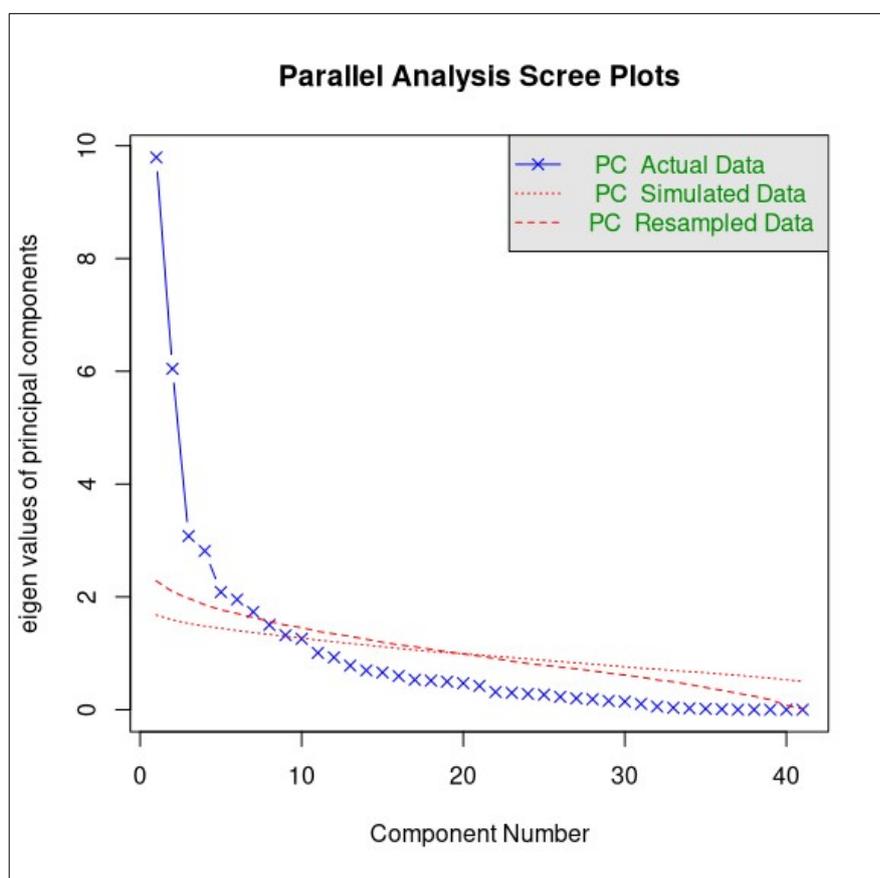
Observa-se que o item C1 apresenta correlação muito baixa com os demais itens da categoria de controles. Porém, por ele apresentar uma maior correlação com os itens da categoria de layout, foi identificado que este item poderia estar melhor alocado nesta categoria.



PA5. Qual a influência dos fatores subjacentes nas respostas aos critérios do *checklist* CodeMaster UI Design 2.0?

É realizado um teste sobre o conjunto de dados, utilizando o índice Kaiser-Meyer-Olkin (KMO). O valor obtido neste teste é de **0,78**. Por ser um valor superior a 0,5 isso indica que é apropriado realizar a análise fatorial.

FIGURA 9: SCREE PLOT REFERENTE AO CODEMASTER: UI DESIGN 2.0



Fonte: autoria própria

O *scree plot* apresentado na Figura 9 mostra 6 (seis) pontos acima da linha tracejada. Isso indica que podem haver 6 fatores subjacentes. Inicialmente foi feita a análise da carga fatorial utilizando 6 fatores. Como pode ser observado na Tabela 23, os valores em negrito são os valores que estão acima do ponto de corte de 0,3.

TABELA 23: CARGA FATORIAL COM 6 FATORES

	F1	F2	F3	F4	F5	F6
L1	0.03193	0.2148	-0.23075	-0.49633	0.50178	0.2436
L2	<b>-0.37119</b>	-0.3515	0.08726	-0.31435	0.23139	0.1976
L3	-0.04841	-0.5714	-0.23042	0.08158	-0.24691	0.2107
L4	0.08039	-0.1055	0.22703	-0.20648	0.24338	<b>-0.4358</b>
L5	-0.19844	-0.5131	0.43088	-0.00025	-0.03164	-0.2081
L6	<b>-0.44846</b>	-0.1471	-0.27345	0.26465	<b>-0.32724</b>	0.0446
C1	-0.02160	0.1553	<b>-0.31906</b>	0.18118	-0.07190	0.3443
C2	<b>-0.42661</b>	0.0409	0.18809	-0.14327	<b>-0.40146</b>	0.1960
C3	<b>-0.51288</b>	0.0662	0.27106	-0.30853	-0.16325	0.1585
C4	<b>-0.67920</b>	-0.0405	0.05898	-0.24267	0.06932	0.0229
C5	<b>-0.30474</b>	-0.1382	0.12324	-0.17435	<b>-0.48623</b>	0.2380
O1	-0.08515	-0.0924	<b>-0.67444</b>	0.07792	-0.17977	0.0995
O2	0.11839	0.2413	0.09203	-0.06369	0.59434	-0.1480
O3	-0.11324	<b>0.3189</b>	-0.13992	-0.17474	0.15052	<b>-0.5922</b>
O4	0.07234	0.0550	<b>-0.79740</b>	-0.06331	-0.05518	-0.0468
O5	0.17684	-0.9865	-0.16358	-0.00867	0.02464	-0.0105
O6	0.06307	-0.5560	-0.14610	-0.02274	<b>-0.34434</b>	0.1508
T1	0.10475	-0.8083	0.23104	0.19881	-0.01432	-0.0988
T2	-0.12941	-0.5871	0.30835	0.01411	0.28999	-0.1509
T3	0.05019	-0.8576	0.16739	0.24760	0.00392	-0.0563
T4	-0.16540	-0.6188	0.20218	-0.26431	0.09357	-0.0310
T5	-0.22777	-0.4801	0.07870	-0.37647	-0.00788	0.2446
T6	-0.22564	-0.7243	0.09110	-0.19263	0.06707	0.0751
T7	-0.00933	-0.8332	0.02734	-0.00580	-0.17686	0.1092
I1	0.13458	0.0568	-0.19523	-0.09204	<b>-0.63031</b>	0.1155
I2	-0.14457	-0.0173	0.01200	0.04182	<b>-0.72041</b>	0.0555
I3	0.10613	-0.0865	0.09310	<b>0.74704</b>	0.01838	-0.0305
I4	0.00843	0.1037	-0.24615	-0.00236	<b>-0.56682</b>	0.1203
I5	0.01705	0.0272	-0.09384	<b>0.62208</b>	0.01512	0.2590
I6	<b>-0.47003</b>	-0.0820	-0.21473	<b>0.31263</b>	<b>-0.30076</b>	0.1536
I7	-0.04641	0.0495	-0.14303	<b>0.63597</b>	-0.09622	0.1056
I8	0.07250	-0.1004	0.02049	<b>0.75571</b>	0.06869	-0.0627
I9	0.09995	0.1021	-0.19037	-0.06452	<b>-0.61684</b>	0.1227
E1	-0.20472	<b>-0.4864</b>	0.07704	-0.32686	-0.02595	0.2939
E2	-0.13124	<b>-0.8231</b>	-0.06600	0.05412	-0.06037	0.0451
E3	-0.24702	<b>-0.6398</b>	0.13528	-0.18841	0.15756	0.0187
E4	-0.11485	<b>-0.7850</b>	0.00893	0.07745	-0.01909	0.1050
E5	<b>-0.32196</b>	<b>-0.5044</b>	0.04659	-0.33864	-0.04394	0.1375
E6	-0.16093	<b>-0.5504</b>	-0.11227	-0.12944	-0.11079	0.1662
E7	-0.08119	<b>-0.7962</b>	0.06129	0.04781	-0.03090	0.0849
E8	<b>-0.44072</b>	-0.1157	0.30362	0.21293	-0.28896	-0.0568

Fonte: autoria própria

Na análise das cargas fatoriais (Tabela 24), para cada fator, todos os itens que apresentam o mesmo sinal e carga fatorial superior a 0,3 apresentam correlação com aquele fator. É observado que existem alguns casos em que a carga está acima do limiar, porém com o sinal oposto. Isso seria esperado em casos em que a resposta está em sentido contrário às demais (p. ex. uma resposta ser “discordo” enquanto as demais sendo “concordo” estão todas contribuindo para um mesmo construto).

Também é realizada a avaliação para 1 (um) fator como mostrado na Tabela 24. Essa redução para 1 (um) fator indica que todos os itens medem o mesmo construto subjacente. Este construto foi interpretado como sendo o design visual do aplicativo. Ao analisar a carga para 1 (um) fator, observa-se que a maioria (31 de 41) dos itens apresentam valor acima do limiar de 0,3.

No item O5 se observa um valor incomum, que está muito próximo de -1. Este item se refere à quantidade de cores utilizadas no texto. Nas avaliações praticamente não houve aplicativos que usassem mais do que a quantidade de cores definida no critério. Considerando isso, concluiu-se que este item não está avaliando corretamente o que se esperava. Uma alternativa seria uma redução na quantidade de cores considerada, juntamente a uma redução no escopo do item. Porém, devido a restrições do App Inventor, que permite apenas uma cor de texto por componente, isso também não avaliaria direito. É então feita a sugestão de remoção deste item.

TABELA 24: CARGA FATORIAL COM 1 FATOR

	<b>Critério</b>	<b>1 fator</b>
L1	Quantidade de componentes visíveis	0,00
<b>L2</b>	<b>Alinhamento</b>	<b>0,72</b>
<b>L3</b>	<b>Visibilidade</b>	<b>0,43</b>
L4	Ocupação da tela	-0,13
<b>L5</b>	<b>Margens do texto</b>	<b>0,67</b>
<b>L6</b>	<b>Posicionamento de botões</b>	<b>0,68</b>
C1	Tamanho mínimo	-0,09
C2	Formato dos controles	0,20
C3	Padronização de tamanho	0,17
<b>C4</b>	<b>Consistência nos controles</b>	<b>0,63</b>
<b>C5</b>	<b>Botões com formato oval</b>	<b>0,47</b>
O1	Quantidade de cores	0,14
O2	Contraste do texto	0,05
O3	Paleta de cores	-0,15
O4	Harmonia de cores	-0,18
<b>O5</b>	<b>Quantidade de cores do texto</b>	<b>0,73</b>
<b>O6</b>	<b>Níveis de intensidade de cores</b>	<b>0,49</b>
<b>T1</b>	<b>Família da fonte</b>	<b>0,69</b>
<b>T2</b>	<b>Padronização do tamanho dos títulos</b>	<b>0,66</b>
<b>T3</b>	<b>Quantidade de tamanhos de texto</b>	<b>0,91</b>
<b>T4</b>	<b>Tamanho do texto regular</b>	<b>0,65</b>
<b>T5</b>	<b>Tamanho de texto dos botões</b>	<b>0,62</b>
<b>T6</b>	<b>Uso de itálico</b>	<b>0,69</b>
<b>T7</b>	<b>Alinhamento do texto</b>	<b>0,87</b>
<b>I1</b>	<b>Estilo dos ícones</b>	<b>0,44</b>
<b>I2</b>	<b>Esquema de cores dos ícones</b>	<b>0,47</b>
<b>I3</b>	<b>Uso de imagens pixelizadas</b>	<b>0,39</b>
I4	Distorção de imagens (botões e ícones)	0,28
<b>I5</b>	<b>Distorção de imagens</b>	<b>0,44</b>
<b>I6</b>	<b>Distorção de imagens (plano de fundo)</b>	<b>0,35</b>
<b>I7</b>	<b>Ajuste de tamanho</b>	<b>0,41</b>
<b>I8</b>	<b>Uso de imagens de texto</b>	<b>0,56</b>
<b>I9</b>	<b>Contraste de ícones</b>	<b>0,41</b>
<b>E1</b>	<b>Uso de caixa-alta em botões</b>	<b>0,38</b>
<b>E2</b>	<b>Capitalização do texto</b>	<b>0,59</b>
<b>E3</b>	<b>Uso indevido de sinais de pontuação</b>	<b>0,62</b>
<b>E4</b>	<b>Pontuação</b>	<b>0,78</b>
<b>E5</b>	<b>Quantidade de caracteres em botões</b>	<b>0,50</b>
<b>E6</b>	<b>Quantidade de caracteres em menus</b>	<b>0,40</b>
<b>E7</b>	<b>Sinais de pontuação agregados</b>	<b>0,86</b>
<b>E8</b>	<b>Uso de caixa-alta fora de botões</b>	<b>0,69</b>

Fonte: autoria própria

Os itens que apresentaram cargas insuficientes apontados na Tabela 24 foram analisados com o objetivo de encontrar justificativas para este comportamento e apontar possíveis correções.

Em relação ao item L1, referente à quantidade de componentes visíveis na tela, foram encontrados durante as avaliações diversos aplicativos inacabados, com telas sem componentes ou com um único botão. Além disso, alguns dos aplicativos utilizavam a tela inicial como uma *splash screen*, com apenas uma imagem. Esses podem ter sido fatores importantes para explicar o resultado obtido. Porém, foi decidido manter este item por ter sido considerado um fator de grande influência no design visual.

Na categoria de controles, foi identificado que o item C1 apresenta carga negativa. Isso pode ser explicado pelo fato de que este item aparenta estar mais relacionado com a usabilidade do que com o design visual. Além disso, esse item é de difícil verificação manual quando a avaliação visual não é suficiente.

Os primeiros 4 (quatro) itens da categoria de cores apresentaram cargas baixas, em especial os itens O3 e O4. Esses itens foram avaliados de forma automática. Além disso, os itens O3 e O4 acabam sofrendo grande influência da paleta de cores fornecida no App Inventor. Isso pode ter sido causa de um viés no conjunto de dados, o que levou a este resultado. Assim, sugere-se a troca desta paleta na ferramenta.

Na categoria de imagens, o item I4 foi o único que apresentou carga insuficiente, porém bem próximo do limiar. Por haver dificuldade em separar o que é considerado como imagem e o que é considerado como ícone, este item poderia ser unificado com o item I5.

### 5.3. Discussão

Os resultados obtidos no painel de especialistas foram preliminares, tendo sido utilizados para identificar ajustes a serem feitos no *checklist*, antes do estudo de caso. Os resultados obtidos durante o estudo de caso indicam que o *checklist* CodeMaster: UI Design 2.0 pode ser considerado confiável e válido no contexto da avaliação de aplicativos Android produzidos com App Inventor.

Apesar de o *scree plot* (Figura 9) ter sugerido 6 (seis) fatores, o que corresponde com a quantidade de categorias do *checklist*, os itens apresentaram alto carregamento para apenas um fator. Os itens estão avaliando um único construto, que foi interpretado como sendo, a qualidade do design visual, ou design de interface.

Com base nos resultados obtidos durante o estudo de caso e na experiência proporcionada pelas avaliações realizadas são feitas as seguintes sugestões de alteração no *checklist*. Na categoria de imagens é proposta a unificação dos itens I4 e I5, pois ambos avaliam a distorção de imagens. O I4 avalia imagens em botões e ícones, enquanto o item I5 avalia as imagens que não se encaixarem nessa descrição e nem sejam plano de fundo. Como há um certo grau de dificuldade na diferenciação entre o que é avaliado em cada um, foi decidido unir os itens.

O item L2 se refere ao alinhamento de textos longos, sendo influenciado por uma das limitações do App Inventor. Ele não possui o alinhamento justificado, portanto, para seguir a recomendação do Material Design, o *checklist* considera como correto o uso de alinhamento à esquerda para textos longos. Este item se mostrou mais relacionado à categoria de tipografia do que à categoria de layout.

#### 5.4. Ameaças à validade

**Tamanho da amostra.** O tamanho da amostra pode influenciar no grau de confiança que se pode ter na generalização dos resultados. No painel de especialistas apenas 13 (treze) avaliadores participaram, o que gera uma baixa confiabilidade. Porém, esta foi uma etapa preliminar, servindo como orientação para ajustes no *checklist*. O estudo de caso utilizou 400 (quatrocentas) avaliações, o que pode ser considerado como um tamanho de amostra suficiente.

**Equilíbrio da amostra.** Os aplicativos produzidos na iniciativa Computação na Escola estão mais próximos do que se espera avaliar posteriormente com o *checklist* em relação ao design visual. Enquanto os aplicativos originados da galeria do App Inventor demonstram um menor cuidado com o design visual. Porém, como a quantidade de aplicativos desenvolvidos na iniciativa Computação na Escola era insuficiente para o estudo de caso, foram utilizados aplicativos da galeria do App Inventor para complementar o estudo.

**Viés de avaliador.** É desejável que quando um aplicativo for avaliado utilizando o *checklist* o resultado seja o mesmo, independente do avaliador. As avaliações utilizadas no estudo de caso foram realizadas por um único avaliador, evitando a influência do *interrater agreement*, porém pode causar viés por só haver a opinião de um avaliador.

**Valores faltantes.** Por ter sido incluída em diversos itens a opção do item “não se aplicar”, foi adicionada uma grande quantidade de valores faltantes nos dados avaliados no estudo de caso, que pode ter um impacto negativo na avaliação. Para mitigar este problema, esta opção é considerada no estudo de caso como uma nova categoria de resposta. Isto evita a perda de informação nas avaliações estatísticas.

**Qualidade dos aplicativos avaliados.** A maior parte dos aplicativos analisados no estudo de caso é da galeria do App Inventor. Muitos desses aplicativos foram criados seguindo tutoriais simples, o que leva a uma grande quantidade de aplicativos semelhantes entre si. Além disso, muitos dos aplicativos da galeria são bastante simples, não apresentando diversos dos itens que seriam avaliados pelo *checklist*, o que gerou os valores faltantes já citados. Esta diferença de qualidade provavelmente pelos aplicativos terem sido feitos por pessoas inexperientes em programação. Para tentar mitigar essa ameaça, foram utilizados os aplicativos desenvolvidos na iniciativa Computação na Escola (2019). Assume-se que estes aplicativos estariam mais próximos ao que se espera avaliar posteriormente com o *checklist*, em comparação com os aplicativos da galeria do App Inventor.

## 6. Conclusão

Como resultado do presente trabalho, foi realizada a atualização de um *checklist* para avaliar o design visual de aplicativos Android em conformidade com guias de estilo, respondendo à questão de pesquisa. O *checklist* atualizado foi o CodeMaster UI Design para a sua segunda versão.

Como parte deste trabalho, foi feita uma síntese da teoria da área de usabilidade e design visual, identificando guias de estilo a serem utilizadas (Objetivo 1). Foi realizado um levantamento do estado da arte em relação à *checklists* voltados à avaliação de design visual por meio de um mapeamento sistemático (Objetivo 2). Os *checklists* encontrados no mapeamento foram mapeados,

identificando itens relacionados ao design visual. Estes resultados foram unificados dando origem a um novo *checklist*, o qual é customizado com base nas diretrizes de guias de estilo (Objetivo 3). O *checklist* desenvolvido foi avaliado quanto à sua confiabilidade e validade por meio de um estudo de caso (Objetivo 4).

Como resultado deste trabalho, disponibiliza-se um *checklist* como ferramenta confiável e válido para avaliação de design visual de aplicativos em dispositivos Android em conformidade com guias de estilo, focada em aplicativos produzidos no App Inventor e voltada para uso na Educação Básica. O *checklist* disponibilizado pode ser utilizado para dar suporte ao ensino de design visual de aplicativos, contribuindo para o ensino de computação nas escolas.

Para trabalhos futuros sugere-se um estudo mais aprofundado sobre o *interrater agreement*, devido à pequena quantidade de avaliadores envolvidos. Também pode ser implementada a automatização da avaliação. Uma abordagem automatizada mitiga o problema relacionado ao *interrater agreement*, além do benefício de acelerar o processo de avaliação. Além disso, podem ser realizadas alterações no ambiente de programação App Inventor para induzir melhorias quanto ao design visual em suas opções padrão (p. ex. mudar a paleta de cores, ou facilitar o uso das opções de layout).

## Referências

**App Inventor 2.** Disponível em: <<https://appinventor.mit.edu/>>. Acesso em: 14 out. 2020.

BALAGTAS-FERNANDEZ, F.; FORRAI, J.; HUSSMANN, H. **Evaluation of User Interface Design and Input Methods for Applications on Mobile Touch Screen Devices.** In: GROSS, T. et al. (Eds.). Human-Computer Interaction – INTERACT 2009. Berlin, Heidelberg: Springer. v. 5726, p. 243–246, 2009.

BALLANTYNE, M.; JHA, A.; JACOBSEN, A.; HAWKER, J. S.; EL-GLALY, Y. N. **Study of Accessibility Guidelines of Mobile Applications.** Proceedings of the 17th International Conference on Mobile and Ubiquitous Multimedia. Anais... MUM 2018. New York, NY, USA: Association for Computing Machinery. 2018.

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

CARRARA, G. **Número de celulares conectados no Brasil cresce 24 milhões.**

Disponível em: <<https://noticias.r7.com/tecnologia-e-ciencia/numero-de-celulares-conectados-no-brasil-cresce-24-milhoes-06022020>>. Acesso em: 16 nov. 2020.

CNE. **CodeMaster V2.0**. Disponível em: <<http://apps.computacaonaescola.ufsc.br:8080/aluno.jsp>>. Acesso em: 13 out. 2020.

COHEN, Jacob. **Statistical Power Analysis for the Behavioral Sciences**. New York: Routledge Academic, 1998.

CRONBACH, L. J. **Coefficient alpha and the internal structure of tests**. *Psychometrika*, v. 16, n. 3, p. 297–334. 1951.

DEVELLIS, R. F. **Scale development: theory and applications**. Thousand Oaks: SAGE Publications, 2003.

FERREIRA, M. N. F.; PINHEIRO, F. C.; WANGENHEIM, C. G.; MISSFELDT FILHO, R. **Ensinando Design de Interface de Usuário na Educação Básica: Um Mapeamento Sistemático do Estado da Arte e Prática**. In: Anais do XXV Workshop de Informática na Escola. Brasília, DF, Brasil. 2019a.

FERREIRA, M. N. F.; WANGENHEIM, C. G.; MISSFELDT FILHO, R.; PINHEIRO, F. C.; HAUCK, J. C. R. **Learning user interface design and the development of mobile applications in middle school**. *Interactions*, v. 26, n. 4, p. 66–69. 2019b.

GARRETT, J. J. **The elements of user experience: user-centered design for the Web and beyond**. 2nd ed. Berkeley, CA: New Riders, 2011.

GLORFELD, L. W. **An Improvement on Horn's Parallel Analysis Methodology for Selecting the Correct Number of Factors to Retain**. *Educational and Psychological Measurement*, v. 55, n. 3, p. 377–393. 1995.

GOOGLE. **Material Design**. Disponível em: <<https://material.io/>>. Acesso em: 13 out. 2020.

GRESSE VON WANGENHEIM, C. et al. **A Usability Score for Mobile Phone Applications Based on Heuristics: International Journal of Mobile Human Computer Interaction**. v. 8, n. 1, p. 23–58. 2016.

HUANG, Z.; TIAN, Z. **Analysis and Design for Mobile Applications: A User Experience Approach**. In: MARCUS, A.; WANG, W. (Eds.). *Design, User Experience, and Usability: Theory and Practice*. Cham: Springer International Publishing. v. 10918p. 91–100, 2018.

**ISO 9241-210:2019**. Disponível em: <<https://www.iso.org/standard/77520.html>>. Acesso em: 14 out. 2020.

**ISO/IEC 25010:2011**. Disponível em: <<https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/03/57/35733.html>>. Acesso em: 14 out. 2020.

JACKSON, J. E. **Oblimin Rotation**. *Encyclopedia of Biostatistics*, John Wiley & Sons, Ltd 2005.

JI, Y. G.; PARK, J. H.; LEE, C.; YUN, M. H. **A Usability Checklist for the Usability Evaluation of Mobile Phone User Interface**. *International Journal of Human-Computer Interaction*, v. 20, n. 3, p. 207–231, 2006.

JOYCE, G.; LILLEY, M.; BARKER, T.; JEFFERIES, A. **Mobile Application Usability: Heuristic Evaluation and Evaluation of Heuristics**. In: AMABA, B. (Ed.). *Advances in Human Factors, Software, and Systems Engineering*. Cham: Springer International Publishing. v. 492p. 77–86, 2016.

KORTUM, P.; SORBER, M. Measuring the Usability of Mobile Applications for Phones and Tablets. *International Journal of Human-Computer Interaction*, v. 31, n. 8, p. 518–529, 2015.

LARDINOIS, F. **Google makes its Material Design system easier to customize****TechCrunch**. 2018. Disponível em: <<https://social.techcrunch.com/2018/05/08/google-makes-its-material-design-system-easier-to-customize/>>. Acesso em: 14 out. 2020.

MCHUGH, M. L. **Interrater reliability: the kappa statistic**. *Biochemia Medica*, v. 22, n. 3, p. 276–282, 2012.

NIELSEN, J. **Usability inspection methods**. Conference companion on Human factors in computing systems - CHI '94. Anais... In: CONFERENCE COMPANION. Boston, Massachusetts, United States: ACM Press, 1994a.

NIELSEN, J. **Usability engineering**. Boston, Massachusetts, United States: Academic Press, 1994b.

OLSINA, L.; SANTOS, L.; LEW, P. **Evaluating Mobileapp Usability: A Holistic Quality Approach**. In: CASTELEYN, S.; ROSSI, G.; WINCKLER, M. (Eds.). *Web Engineering*. Cham: Springer International Publishing. v. 8541p. 111–129, 2014.

OLSSON, U. **Maximum likelihood estimation of the polychoric correlation coefficient**. *Psychometrika*, v. 44, n. 4, 1979. p. 443–460, 1979.

PETERSEN, K. et al. **Systematic Mapping Studies in Software Engineering**. In: 12TH INTERNATIONAL CONFERENCE ON EVALUATION AND ASSESSMENT IN SOFTWARE ENGINEERING (EASE), 2008.

SAURO, J. **Understanding Expert Reviews and Inspection Methods****MeasuringU**. 2019. Disponível em: <<https://measuringu.com/inspection-methods/>>. Acesso em: 14 out. 2020.

SCHLATTER, T.; LEVINSON, D. A. **Visual usability: principles and practices for designing digital applications**. Amsterdam: Morgan Kaufmann, 2013.

SHARP, H.; ROGERS, Y.; PREECE, J. **Design de interação: além da interação humano-computador**. reimpressão ed. Porto Alegre, Brasil: Bookman, 2005.

SOLECKI, I.; PORTO, J. V. A.; ALVES, N. C.; WANGENHEIM, C. G.; HAUCK, J.; BORGATTO, A. F. **Automated Assessment of the Visual Design of Android Apps Developed with App Inventor**. *Proceedings of the 51st ACM Technical*

Symposium on Computer Science Education. Anais... SIGCSE '20. Portland, OR, USA: Association for Computing Machinery, 2020.

STOYANOV, S. R.; HIDES, L.; KAVANAGH, D. J.; ZELENKO, O.; TJONDRONEGORO, D.; MANI, M. **Mobile App Rating Scale: A New Tool for Assessing the Quality of Health Mobile Apps**. JMIR mHealth and uHealth, v. 3, n. 1, p. e27, 2015.

TREERATANAPON, T. **Design of the Usability Measurement Framework for Mobile Applications**. Proceedings of International Conference on Computer and Information Technology (ICCIT'2012). Anais... In: INTERNATIONAL CONFERENCE ON COMPUTER AND INFORMATION TECHNOLOGY (ICCIT'2012). Hammamet, Tunisia, 2012.

W3C. **Guidance on Applying WCAG 2.0 to Non-Web Information and Communications Technologies (WCAG2ICT)**. Disponível em: <<https://www.w3.org/TR/wcag2ict/>>. Acesso em: 14 out. 2020.

W3C. **Diretrizes de Acessibilidade para Conteúdo Web (WCAG) 2.0**. Disponível em: <<https://www.w3.org/Translations/WCAG20-pt-br/>>. Acesso em: 14 out. 2020.

W3C. **Mobile Accessibility: How WCAG 2.0 and Other W3C/WAI Guidelines Apply to Mobile**. Disponível em: <<https://www.w3.org/TR/mobile-accessibility-mapping/>>. Acesso em: 14 out. 2020.

WASSERMAN, A. I. **Software engineering issues for mobile application development**. Proceedings of the FSE/SDP workshop on Future of software engineering research – FoSER '10. Anais... In: THE FSE/SDP WORKSHOP. Santa Fe, New Mexico, USA: ACM Press, 2010.

ZEN, M.; VANDERDONCKT, J. **Assessing User Interface Aesthetics based on the Inter-subjectivity of Judgment**. In: Proc. of the 30TH International BCS Human Computer Interaction Conference, Bournemouth, GB, 2016.

## Apêndice A – Tabelas da extração de dados

Pergunta de análise	Dados a extrair	Descrição
PA1. Quais <i>checklists</i> existem?	Nome	CodeMaster
	Descrição	Rubrica para avaliação de <i>design</i> de interface de aplicativos Android desenvolvidos com App Inventor
	Referência	Solecki et al. (2019) CnE (2019)
PA2. Quais as características do <i>checklist</i> ?	Guia base	Material Design e WCAG 2.0
	Elementos avaliados	Layout, tipografia, controles, cores, imagens, escrita
	Quantidade de itens	28 no total, com 28 voltados ao design visual
	Possui ferramenta?	Possui ferramenta online para avaliação automática de código App Inventor (.aia) <a href="http://apps.computacaonaescola.ufsc.br:8080/aluno.jsp">http://apps.computacaonaescola.ufsc.br:8080/aluno.jsp</a>
PA3. Como o <i>checklist</i> foi desenvolvido?	Método de desenvolvimento do <i>checklist</i>	Análise do contexto; Definição dos objetivos de aprendizagem; definição dos critérios de avaliação; definição dos níveis de desempenho; avaliação da validade aparente
PA4. Como foi realizada a avaliação do <i>checklist</i> ?	Tipo de estudo de avaliação	Estudo de caso, seguindo a abordagem GQM
	Fatores avaliados	Consistência interna, validade convergente e validade discriminante
	Método de coleta de dados	Aplicativos disponíveis na galeria do App Inventor
	Tamanho de amostra	978
	Método de análise de dados	Alfa de Cronbach, intercorrelações entre os critérios e correlação critério-total
	Descobertas	Alfa de Cronbach = 0,71; Aumento no alfa de Cronbach ao excluir os itens E2, E6 e I4; Apenas o item C2 possui baixa correlação com os demais da mesma categoria; Baixa correlação critério-total nos itens E2, I1, I3 e I4

Pergunta de análise	Dados a extrair	Descrição
PA1. Quais <i>checklists</i> existem?	Nome	UI Automator Viewer
	Descrição	Ferramenta para avaliação de acessibilidade de aplicativos Android
	Referência	Patil, Bhole e Shete (2016)
PA2. Quais as características do <i>checklist</i> ?	Guia base	Checklist de acessibilidade da Google, 2013 WCAG 2.0 PRAYAG, S. PANAGE.
	Elementos avaliados	Cores, <i>layout</i> , acessibilidade
	Quantidade de itens	14 no total, com 5 itens voltados ao design visual
	Possui ferramenta?	Possui ferramenta de avaliação automática com conexão USB
PA3. Como o <i>checklist</i> foi desenvolvido?	Método de desenvolvimento do <i>checklist</i>	União dos itens do checklist de acessibilidade da Google e do WCAG 2.0
PA4. Como foi	Tipo de estudo de	Não informado

realizada a avaliação do <i>checklist</i> ?	avaliação	
	Fatores avaliados	Não informado
	Método de coleta de dados	Não informado
	Tamanho de amostra	Não informado
	Método de análise de dados	Não informado
	Descobertas	Não informado

Pergunta de análise	Dados a extrair	Descrição
PA1. Quais <i>checklists</i> existem?	Nome	2Q2U 2.0
	Descrição	Definições de qualidade, qualidade no uso, usabilidade e experiência de usuário
	Referência	Olsina, Santos e Lew (2014)
PA2. Quais as características do <i>checklist</i> ?	Guia base	ISO 25010
	Elementos avaliados	Usabilidade, aprendizagem, operabilidade, proteção contra erros de usuário, estética
	Quantidade de itens	23 no total, com 9 itens voltados ao design visual
	Possui ferramenta?	Não informado
PA3. Como o <i>checklist</i> foi desenvolvido?	Método de desenvolvimento do <i>checklist</i>	Análise do contexto; desenvolvimento do <i>checklist</i> ; aplicação e análise do resultado
PA4. Como foi realizada a avaliação do <i>checklist</i> ?	Tipo de estudo de avaliação	Ad-hoc
	Fatores avaliados	Não informado
	Método de coleta de dados	Não informado
	Tamanho de amostra	1
	Método de análise de dados	Não informado
	Descobertas	Não informado

Pergunta de análise	Dados a extrair	Descrição
PA1. Quais <i>checklists</i> existem?	Nome	<i>A Comprehensive List of Accessibility Guidelines for Mobile Applications</i>
	Descrição	Um compilado de guidelines adaptado para <i>checklist</i> voltado à acessibilidade em aplicativos mobile
	Referência	Ballantyne, M. et al. (2018) Ballantyne, M. et al. (2019)
PA2. Quais as características do <i>checklist</i> ?	Guia base	WCAG 2.0
	Elementos avaliados	Texto, áudio, vídeo, elementos da interface de usuário, controle de usuário, flexibilidade e eficiência, reconhecimento em vez de memorização, gestos, visibilidade do sistema, prevenção de erros, interações tangíveis
	Quantidade de	92 no total, com 30 itens voltados ao design visual

	itens	
	Possui ferramenta?	Não informado
PA3. Como o <i>checklist</i> foi desenvolvido?	Método de desenvolvimento do <i>checklist</i>	Criado visando a automação da avaliação, por meio da adaptação de itens do WCAG 2.0
PA4. Como foi realizada a avaliação do <i>checklist</i> ?	Tipo de estudo de avaliação	Não informado
	Fatores avaliados	Não informado
	Método de coleta de dados	Não informado
	Tamanho de amostra	Não informado
	Método de análise de dados	Não informado
	Descobertas	Não informado

Pergunta de análise	Dados a extrair	Descrição
PA1. Quais <i>checklists</i> existem?	Nome	<i>A Pilot Study of an Inspection Framework for Automated Usability Guideline Reviews of Mobile Health Applications</i>
	Descrição	Avaliação de um <i>framework para Guideline Reviews</i> de aplicativos mobile de saúde
	Referência	Xu et al. (2014)
PA2. Quais as características do <i>checklist</i> ?	Guia base	Health Information Management System Society (HIMSS) usability guidelines, Coarse Grained GUI Model (CGGM)
	Elementos avaliados	Layout, cores, tipografia, controles
	Quantidade de itens	18 no total, com 11 itens voltados ao design visual
	Possui ferramenta?	Informa que possui ferramenta, mas a ferramenta não foi encontrada
PA3. Como o <i>checklist</i> foi desenvolvido?	Método de desenvolvimento do <i>checklist</i>	Não informado
PA4. Como foi realizada a avaliação do <i>checklist</i> ?	Tipo de estudo de avaliação	Estudo de caso
	Fatores avaliados	Não informado
	Método de coleta de dados	Avaliação de aplicativos disponíveis na Google Play Store
	Tamanho de amostra	24 aplicativos
	Método de análise de dados	Comparação com <i>reviews</i> de usuários na Google Play Store
	Descobertas	Não informado

Pergunta de análise	Dados a extrair	Descrição
PA1. Quais <i>checklists</i> existem?	Nome	<i>Heuristic Evaluation on Mobile Interfaces: A New Checklist</i>
	Descrição	<i>Checklist</i> abrangente para avaliação de interfaces de aplicativos <i>mobile</i>
	Referência	Gómez, Caballero e Sevillano (2014)

PA2. Quais as características do <i>checklist</i> ?	Guia base	-Heuristic evaluation of user interfaces -Designing the user interface Strategies for Effective Human-Computer Interaction -Heuristic evaluation: a system checklist -Devlilish Details: Best Pratices in Web Design -Usability engineering for the web -First Principles of interaction design -Guía de Evaluación Heurística de sitios web; CITAR A REF (Hassan et al., 2004) -Diseño Web Centrado en el Usuario: Usabilidad y Arquitetctura de la Informaión -Specifying quality characteristics and attributes for websites
	Elementos avaliados	Visibilidade do status do sistema, correlação com o mundo real, controle de usuário, consistência, prevenção de erros, reconhecimento em vez de memorização, flexibilidade e eficiência, estética, tratamento de erros, documentação, habilidades, privacidade
	Quantidade de itens	230 no total, com 99 itens voltados ao design visual
	Possui ferramenta?	Não informado
PA3. Como o <i>checklist</i> foi desenvolvido?	Método de desenvolvimento do <i>checklist</i>	Definição do escopo; Compilação de heurísticas existentes; Gerar subitens mais específicos; Adicionar subitens específicos para mobile; Revisão textual; Validação
PA4. Como foi realizada a avaliação do <i>checklist</i> ?	Tipo de estudo de avaliação	Ad-hoc
	Fatores avaliados	Utilidade como ferramenta para designers e engenheiros de software sem conhecimentos prévios sobre usabilidade avaliarem interfaces de aplicativos
	Método de coleta de dados	Protótipos desenvolvidos para este fim
	Tamanho de amostra	2
	Método de análise de dados	Pesquisa de opinião
	Descobertas	Uma grande quantidade de itens ficou definida como "não aplicável", provavelmente pelas heurísticas serem genéricas e avaliarem muitos itens que não estarão presentes em todos os aplicativos

Pergunta de análise	Dados a extrair	Descrição
PA1. Quais <i>checklists</i> existem?	Nome	<i>Design and evaluation of a mobile user interface for older adults</i>
	Descrição	Recomendações de usabilidade para o desenvolvimento de aplicativos voltadas para pessoas idosas
	Referência	Barros, Leitão e Ribeiro (2014)
PA2. Quais as características do <i>checklist</i> ?	Guia base	-Siek, Rogers e Connelly (1997) -"Navigation, orientation, and gestures for Windows Phone" Microsoft (2013) -"Button control design for Windows Phone" Microsoft (2013)
	Elementos avaliados	Navegação, layout, input, conteúdo, ícones
	Quantidade de itens	9 no total, com 6 itens voltados ao design visual
	Possui ferramenta?	Não possui ferramenta
PA3. Como o <i>checklist</i> foi desenvolvido?	Método de desenvolvimento do <i>checklist</i>	As recomendações foram desenvolvidas com base em experimentos
PA4. Como foi realizada a avaliação do <i>checklist</i> ?	Tipo de estudo de avaliação	Não informado
	Fatores avaliados	Não informado
	Método de coleta	Não informado

	de dados	
	Tamanho de amostra	Não informado
	Método de análise de dados	Não informado
	Descobertas	Não informado

Pergunta de análise	Dados a extrair	Descrição
PA1. Quais <i>checklists</i> existem?	Nome	<i>Tablet application GUI usability checklist</i>
	Descrição	<i>Checklist</i> de avaliação de usabilidade para aplicativos em tablets
	Referência	Xu (2013)
PA2. Quais as características do <i>checklist</i> ?	Guia base	- "iOS Human interface guidelines" Apple (2012) - "User interface Guidelines" Google (2012) - "Designing UX for apps" Microsoft (2012) - "Windows Application UI Development" Microsoft (2012) - BlackBerry PlayBook tablets UI Guidelines (2012)
	Elementos avaliados	Navegação, contraste, conteúdo, controles, input
	Quantidade de itens	84 no total, com 44 itens voltados ao design visual
	Possui ferramenta?	Não possui ferramenta
PA3. Como o <i>checklist</i> foi desenvolvido?	Método de desenvolvimento do <i>checklist</i>	Pesquisa quantitativa de conteúdo para identificar fatores importantes; transformação qualitativa dos itens relevantes para a criação do <i>checklist</i>
PA4. Como foi realizada a avaliação do <i>checklist</i> ?	Tipo de estudo de avaliação	Ad-hoc
	Fatores avaliados	Não informado
	Método de coleta de dados	Foram escolhidos 3 aplicativos com base dos critérios: popularidade, fornecedor confiável e complexidade da interface de usuário
	Tamanho de amostra	3
	Método de análise de dados	Não informado
	Descobertas	Foram adicionados 6 itens ao <i>checklist</i>

Pergunta de análise	Dados a extrair	Descrição
PA1. Quais <i>checklists</i> existem?	Nome	<i>The Impact of Mobile Device Usability Guidelines on the Design of Help Documentation for Evernote's iOS Application</i>
	Descrição	<i>Checklist</i> para avaliar a usabilidade da documentação do Evernote
	Referência	Wrede (2018)
PA2. Quais as características do <i>checklist</i> ?	Guia base	- Budiu (2015) - Budiu (2016) - Gómez, Caballero e Sevillano (2014) - Meyer (2016) - Moumane, Idri e Abran (2016) - Nayebi (2012) - Nayebi (2013) - Nielsen (2011) - Rahmat, Zulzalil, Ghani e Kamaruddin (2015) - Schade (2015)

		-Thitichaimongkhol e Senivongs (2016)
	Elementos avaliados	Conteúdo, performance, design visual, feedback de sistema, público-alvo
	Quantidade de itens	313 no total, com 119 itens voltados ao design visual
	Possui ferramenta?	Não possui ferramenta
PA3. Como o <i>checklist</i> foi desenvolvido?	Método de desenvolvimento do <i>checklist</i>	Foi feita a união dos itens de diversos checklists, separando-os em categorias baseadas nos "elementos da experiência de usuário" (Garret). Foram removidos os itens duplicados e criadas subcategorias
PA4. Como foi realizada a avaliação do <i>checklist</i> ?	Tipo de estudo de avaliação	Não informado
	Fatores avaliados	Não informado
	Método de coleta de dados	Não informado
	Tamanho de amostra	Não informado
	Método de análise de dados	Não informado
	Descobertas	Não informado

Pergunta de análise	Dados a extrair	Descrição
PA1. Quais <i>checklists</i> existem?	Nome	<i>Measuring Visual User Interface Complexity of Mobile Applications With Metrics</i>
	Descrição	Ferramenta para avaliação automática da complexidade de interfaces visuais
	Referência	Riegler e Holzmann (2018)
PA2. Quais as características do <i>checklist</i> ?	Guia base	-Fu et al. (2007) -Zen and Vanderdonckt (2014) -Sears (2001) -Alemerien and Magel (2014) -Fu et al. (2007) -Comber and Maltby (1995) -Tullis (1981, 1983) -Bonsiepe (1968) -Ma et al. (2013)
	Elementos avaliados	Layout, cor, tipografia e consistência
	Quantidade de itens	8 no total, com 8 itens voltados ao design visual
	Possui ferramenta?	Possui ferramenta de avaliação da complexidade de interface visual
PA3. Como o <i>checklist</i> foi desenvolvido?	Método de desenvolvimento do <i>checklist</i>	Análise do contexto; Definição dos objetivos; definição dos critérios de avaliação; definição dos níveis de desempenho; avaliação da validade aparente
PA4. Como foi realizada a avaliação do <i>checklist</i> ?	Tipo de estudo de avaliação	Ad-hoc
	Fatores avaliados	O quanto cada métrica contribui para o resultado obtido
	Método de coleta de dados	Avaliação heurística de 3 aplicativos, com 12 avaliadores, e aplicação da ferramenta nestes aplicativos
	Tamanho de amostra	12 avaliadores, 3 aplicativos
	Método de análise de dados	Coefficiente de correlação de Spearman
	Descobertas	O sistema funciona e os itens avaliados são relevantes

## Apêndice B – Resumo dos elementos avaliados pelos *checklists* encontrados

Referências	Elementos do design visual	Resumo dos itens avaliados
CodeMaster (2019)	Layout	Dimensionamento, quantidade de elementos e uso de organizadores
	Controles	Uso de <i>tooltips</i> em caixas de texto, consistência no formato dos controles
	Cores	Avalia a conformidade com o WCAG e o Material Design, harmonia e quantidade de cores utilizadas
	Tipografia	Avalia tamanho da fonte, se a fonte é sem serifa, alinhamento do texto e uso de texto em itálico
	Imagens	Conformidade dos ícones com o Material Design, resolução e distorção das imagens
	Escrita	Uso de letras maiúsculas, quantidade de caracteres e pontuação
Patil, Bhole e Shete (2016)	Layout	Posicionamento dos elementos e divisão da página
	Controles	Minimização de entrada de texto
	Cores	Contraste
	Tipografia	-
	Imagens	Tamanho das imagens
Olsina, Santos e Lew (2014)	Layout	Posicionamento dos controles
	Controles	Consistência nos controles principais e contextuais
	Cores	Contraste, uniformidade e harmonia
	Tipografia	Tamanho da fonte
	Imagens	-
	Escrita	-
Ballantyne, M. et al. (2018)	Layout	Posicionamento dos elementos
	Controles	Clareza da funcionalidade dos controles, uso de links
	Cores	Contraste
	Tipografia	Tipo de fonte utilizado, alinhamento e espaçamento do texto
	Imagens	Uso de imagens contendo texto
	Escrita	Nível da linguagem utilizada, uso de termos técnicos, uso de abreviações e quantidade de caracteres
Xu et al. (2014)	Layout	Posicionamento e quantidade de elementos
	Controles	Imposição do uso de determinados controles
	Cores	Cor do texto e uso inapropriado da cor vermelha
	Tipografia	Consistência no estilo da fonte
	Imagens	-
	Escrita	Nível da linguagem e uso de termos técnicos
Gómez, Caballero e Sevillano (2014)	Layout	Correspondência entre a importância do conteúdo e a hierarquia dos elementos, margens, limite de tamanho da página
	Controles	Existência de feedback, desfazer ações, controles claramente sinalizados
	Cores	Relação com expectativas culturais, quantidade de cores
	Tipografia	Diferenciação entre <i>label</i> , entrada de texto e enfatizar texto

	Imagens	Relação com convenções culturais
	Escrita	Tipo de linguagem, uso de metáforas, consistência entre os títulos
Barros, Leitão e Ribeiro (2014)	Layout	Espaçamento e posicionamento dos controles
	Controles	Evitar determinados controles (panorama e <i>pivot</i> )
	Cores	-
	Tipografia	-
	Imagens	Uso de ícones em determinados controles
	Escrita	Tipo de linguagem
Xu (2013)	Layout	Posicionamento e visibilidade dos elementos
	Controles	Clareza na funcionalidade, consistência
	Cores	-
	Tipografia	-
	Imagens	Uso de ícones no lugar de texto
	Escrita	-
Wrede (2018)	Layout	Tamanho e posicionamento dos elementos
	Controles	Define alguns tipos de controles a serem preferencialmente utilizados, tamanho mínimo, sinalização clara
	Cores	Se refere à quantidade de cores a ser utilizada, contraste e uso das cores para enfatizar determinados elementos
	Tipografia	Enfatizar texto
	Imagens	Quantidade de ícones, nível de detalhe dos ícones, harmonia entre os ícones, contraste, preferência pelo uso de imagens
	Escrita	-
Riegler e Holzmann (2018)	Layout	Alinhamento, densidade, tamanho
	Controles	-
	Cores	Complexidade das cores (quantidade e combinação)
	Tipografia	Quantidade de tamanhos diferentes de texto e tamanho médio
	Imagens	-
	Escrita	-

**Apêndice C – Artigo (Próxima página)**

# Desenvolvimento de um *checklist* de avaliação de design visual de aplicativos Android em conformidade com guias de estilo

Arthur Mesquita Pickcius<sup>1</sup>, Christiane Gresse von Wangenheim<sup>1</sup>, Adriano Ferreti Borgatto<sup>1</sup>, Mirian Nathalie F. Ferreira<sup>1</sup>

<sup>1</sup>Departamento de Informática e Estatística  
Universidade Federal do Santa Catarina (UFSC) – Florianópolis, SC – Brasil

arthur.pickcius@grad.ufsc.br, c.wangenheim@ufsc.br,  
adriano.bordgatto@ufsc.br, nathalie@telemedicina.ufsc.br

**Abstract.** *In Basic Education, the teaching of programming has become increasingly relevant. As a complement to this, visual design provides an incentive for students to produce applications closer to those that are available on the market. The main objective of this project is to produce a tool to support the teaching of visual design. A great difficulty in the area of visual design is its evaluation, because visual design is an abstract concept, while we try to make measurements by establishing concrete criteria. In this work, an update is developed for the CodeMaster UI Design checklist, creating its version 2.0. And a case study is also carried out with 400 apps to assess the reliability and validity of the checklist.*

**Resumo.** Na Educação Básica o ensino de programação tem se tornado cada vez mais relevante. Como complemento a este, o design visual proporciona um incentivo aos estudantes para que produzam aplicativos mais próximos aos que estão disponíveis no mercado. O objetivo principal deste trabalho é produzir uma ferramenta de apoio ao ensino de design visual. Uma grande dificuldade na área do design visual é a sua avaliação, porque o design visual é um conceito abstrato, enquanto tentamos realizar medições estabelecendo critérios concretos. Neste trabalho é desenvolvida uma atualização para o *checklist* CodeMaster UI Design, criando sua versão 2.0. E também é realizado um estudo de caso com 400 aplicativos para avaliar a confiabilidade e validade do *checklist*.

## 1. Introdução

*Smartphones* são itens de consumo pessoal de grande relevância tecnológica. No Brasil, estima-se que o número total de celulares chegou a 154 milhões em 2020 [Carrara 2020]. Um dos desafios de design de interface para esse tipo de dispositivo é que os *smartphones* são utilizados por uma ampla gama de pessoas com objetivos diferentes [Huang e Tian 2018]. Por causa disso, o ensino de programação na Educação Básica tem se tornado cada vez mais relevante, e atrelado a ele, o design de interface. Além disso, na educação básica, o ensino de programação é realizado em grande parte com o uso de ferramentas de programação em blocos, em que as principais operações são todas listadas, podendo ser utilizadas clicando e arrastando.

Visando o ensino do design de interface, ou design visual, o presente trabalho tem por objetivo proporcionar uma ferramenta de apoio ao ensino de design visual. Esta

ferramenta é um *checklist* de avaliação de design visual. Este trabalho teve foco em aplicativos desenvolvidos no ambiente do App Inventor (2013) para o sistema Android. No caso do App Inventor, além de ser uma ferramenta de programação em blocos, a parte relacionada com o design visual também pode ser operada da mesma forma, arrastando os componentes listados até a área que representa a tela do *smartphone*.

Além do desenvolvimento do *checklist* é realizado um estudo de caso, avaliando sua confiabilidade e validade. Este estudo é feito por meio da avaliação semi-automatizada de 400 aplicativos.

## **2. Metodologia de pesquisa**

Como preparação para o desenvolvimento sistemático do *checklist* é realizada uma etapa de síntese da teoria da área de usabilidade e design visual em termos de princípios, elementos e guias de estilo. Após esta etapa é feito o levantamento do estado da arte em relação a *checklists* voltados à avaliação do design visual. Este levantamento é realizado por meio de um mapeamento sistemático utilizando como fontes de dados as principais bibliotecas digitais do campo da computação. É realizada então a etapa de criação do *checklist*, em que ocorre a unificação dos critérios e adequação aos guias de estilo. E, por fim, uma etapa de avaliação do *checklist*, por meio de um estudo de caso.

## **3. Desenvolvimento do *checklist***

### **3.1. Estado da arte**

O levantamento do estado da arte é realizado com o objetivo de identificar e mapear as publicações relevantes ao escopo do trabalho. Este mapeamento é produzido por meio de análise temática e classificação do material científico encontrado [Petersen et al. 2008].

A questão central de pesquisa é definida como “**Quais *checklists* existem para avaliar o design visual de aplicativos Android desenvolvidos utilizando App Inventor e quais as suas características?**”.

A Tabela 1 ilustra as quantidades de resultados encontrados em cada uma das etapas do mapeamento. Como resultado deste foram identificados 10 *checklists* que apresentaram características consideradas relevantes para o presente trabalho. No mapeamento foi observado que há uma falta de *checklists* específicos para avaliar o design visual. Existem vários que avaliam o aplicativo em geral, porém, ao refinar a busca se observa que vários avaliam a usabilidade, que é um conceito que engloba o design visual, mas poucos avaliam especificamente o design visual. Também foi observado pouco suporte ao contexto educacional. Além disso, poucos dos trabalhos apresentavam um estudo mais detalhado sobre a aplicação dos *checklists* propostos.

**Tabela 1. Quantidades de resultados da busca**

Fonte de dados	Resultado da pesquisa inicial	Resultados analisados	Resultados potencialmente relevantes	Resultados relevantes
ACM	212	100	5	3
Google scholar	~6.110	100	24	4
IEEE	1	1	1	1
ScienceDirect	11	11	1	1
SCOPUS	1	1	1	0
SpringerLink	58	58	3	1
<b>Total</b>	<b>6393</b>	<b>271</b>	<b>35</b>	<b>10</b>

### 3.2. Criação do *checklist*

Para a criação do *checklist*, são identificados em cada um dos *checklists* os itens referentes ao design visual. E então estes itens são avaliados quanto à sua inclusão ou não. Para cada um dos itens incluídos, são listadas as suas referências, identificando quais *checklists* endossam a existência daquele item. Quando um item não é incluído é então descrito o motivo da sua não-inclusão.

Durante a seleção dos itens, alguns são identificados como não estando de acordo com os guias selecionados. Além disso, após essa etapa os itens adicionados foram adaptados para seguirem as recomendações dos guias Material Design (2014) e WCAG 2.0 (2014). A Tabela 2 mostra os critérios dos itens do *checklist* desenvolvido e suas categorias de resposta.

**Tabela 2. Checklist Code Master UI Design 2.0**

Id	Critério	-	0 pontos	1 ponto	2 pontos
<b>Layout</b>					
L1	Qual o número mínimo e o número máximo de componentes visíveis em uma tela?	-	min < 2 OU max ≥ 20	min ≥ 2 E max entre 10 e 19	min ≥ 2 E max ≤ 9
L2	Todos os textos longos (3 linhas ou mais) estão com alinhamento à esquerda?	Não se aplica (não há textos longos)	Não	-	Sim
L3	O conteúdo de todos os elementos está dentro os limites do componente pai ou da largura da tela?	-	Não	-	Sim
L4	Alguns espaços entre componentes visíveis ou bordas da tela ocupa mais do que 35% da tela?	-	Não	-	Sim
L5	Todo o texto possui margens laterais ( <i>padding</i> ) de no mínimo a largura de uma letra "n"?	Não se aplica (não há texto)	Não	-	Sim
L6	Todos os botões de voltar estão no canto superior esquerdo da tela?	Não se aplica (não há botões de voltar)	Não	-	Sim

Controles					
C1	Todos os componentes alvos de toque têm largura e altura maior ou igual a 48x48 pixels dentro da área da tela?	-	Não	-	Sim
C2	Botões agrupados na mesma interface sempre têm o mesmo formato (retangular/arredondado)?	Não se aplica (não há botões agrupados)	Não	-	Sim
C3	Botões agrupados na mesma interface têm sempre o mesmo tamanho independente de seu conteúdo?	Não se aplica (não há botões agrupados)	Não	-	Sim
C4	Existem botões ou menus disponíveis na mesma posição em mais de uma tela?	Não se aplica (só há uma tela)	Não	-	Sim
C5	Algum botão possui formato oval? (exceto ícone)	Não se aplica (não há botões)	Sim	-	Não
Cores					
O1	Quantas cores são usadas no aplicativo (além de preto, branco e cinza)?	-	4 ou mais, ou nenhuma	3	1 ou 2
O2	Qual o contraste entre texto e cor de fundo?	-	Ruim (Insuficiente)	Aceitável (Nível AA na WCAG)	Bom (Nível AAA na WCAG)
O3	Usam-se apenas cores da paleta do Material Design?	-	Não	-	Sim
O4	As tonalidades de cores usadas são harmônicas entre si (complementares, análogas ou triádicas)?	-	Não	-	Sim
O5	Qual a maior quantidade de cores usadas no texto em uma tela?	Não se aplica (não há texto)	4 ou mais	3	Até 2
O6	Qual a maior quantidade de níveis de intensidade de cada cor foi utilizada?	-	4 ou mais	-	Até 3
Tipografia					
T1	Todos os componentes usam família da fonte sem serifa?	Não se aplica (não há texto)	Não	-	Sim
T2	Todos os títulos possuem tamanho 60, 48, 34, 24 ou 20pt?	Não se aplica (não há títulos)	Não	-	Sim
T3	Qual a maior quantidade de tamanhos de texto em uma tela?	Não se aplica (não há texto)	5 ou mais	-	De 1 a 4
T4	Todo o texto regular possui tamanho 14?	Não se aplica (não há texto regular)	Não	-	Sim
T5	Todos os botões com texto têm tamanho da fonte igual a 14?	Não se aplica (não há botões com texto)	Não	-	Sim
T6	Existe algum elemento com texto todo em itálico?	Não se aplica (não há texto)	Sim	-	Não
T7	Existe algum texto alinhado à direita?	Não se aplica (não há texto)	sim	-	Não
Imagens					
I1	Todos os ícones são do mesmo estilo (desconsiderando as cores)?	Não se aplica (não há ícones)	Não	-	Sim
I2	Todos os ícones de elementos agrupados utilizam o mesmo esquema de cores?	Não se aplica (não há ícones)	Não	-	Sim
I3	Existe alguma imagem ou ícone pixelizado?	Não se aplica (não há imagens, nem ícones)	Sim	-	Não
I4	Existe algum botão com imagem distorcida ou ícones distorcidos?	Não se aplica (não há botões com imagen, nem ícones)	Sim	-	Não
I5	Existe algum componente do tipo imagem distorcido?	Não se aplica (não há componentes do	Sim	-	Não

		tipo imagem)			
I6	Existe alguma imagem de fundo de tela distorcida?	Não se aplica (não há imagens de fundo)	Sim	-	Não
I7	Existe alguma imagem exibida com tamanho maior que o tamanho da tela?	Não se aplica (não há imagens)	Sim	-	Não
I8	Existe alguma imagem de texto?	Não se aplica (não há imagens)	Sim	-	Não
I9	Os ícones possuem contraste de no mínimo 3:1 com o fundo?	Não se aplica (não há ícones)	Não	-	Sim
<b>Escrita</b>					
E1	O texto de todos os botões está todo em maiúsculas?	Não se aplica (não há botões com texto)	Não	-	Sim
E2	Todas as sentenças começam com letra maiúscula ou dígito?	Não se aplica (não há texto)	Não	-	Sim
E3	Existem labels ou checkbox que terminam com pontuação (,;:;!)?	Não se aplica (não há labels, nem checkboxes com texto)	Pontuação diferente de interrogação	Apenas com interrogação	Não
E4	Existem sentenças que terminam com "." (ponto) (desconsiderando meio de parágrafo)?	Não se aplica (não há texto)	Não	-	Sim
E5	O texto de botão mais longo tem quantos caracteres?	Não se aplica (não há botões com texto)	15 ou mais	De 8 a 14	7 ou menos
E6	O texto de item de menu mais longo tem quantos caracteres?	Não se aplica (não há itens de menu com texto)	20 ou mais	De 13 a 19	12 ou menos
E7	Existe algum caso de sinais de pontuação iguais agregados?	Não se aplica (não há texto)	Sim	-	Não
E8	Existe algum texto todo em letras maiúsculas fora de botões (Exceto siglas)?	Não se aplica (não há texto fora de botões)	Sim	-	Não

#### 4. Avaliação do *checklist*

A avaliação do *checklist* é realizada por meio de um estudo de caso, com base nas seguintes questões de análise:

- O *checklist* apresenta consistência interna?
- O *checklist* apresenta validade convergente e discriminante?
- Qual a influência dos fatores subjacentes nas respostas aos critérios do *checklist*?

Para este estudo de caso foram realizadas avaliações semi-automatizadas em 400 aplicativos. A avaliação de 4 itens da categoria de cores foi feita de forma automatizada utilizando a implementação anterior do CodeMaster. Os critérios desses itens eram complexos e estavam gerando certa confusão quando avaliados por avaliadores diferentes. Os demais itens foram avaliados de forma manual, sendo que a maior parte dos aplicativos utilizados foi extraída da galeria do App Inventor. O restante destes aplicativos tiveram origem na iniciativa Computação na Escola.

Como métrica para a avaliação da confiabilidade do *checklist*, é utilizado o alfa de Cronbach. O valor obtido pelo conjunto dos itens foi de 0,81, que é considerado um

bom resultado segundo Cronbach. Também é realizada uma avaliação da correlação item-total e do impacto da remoção de itens no alfa de Cronbach (Tabela 3). Dois dos itens apresentam correlação item-total negativa e geram um aumento no alfa de Cronbach caso removidos. Mesmo assim, esses itens foram considerados importantes para a avaliação do design visual e foram mantidos.

Na análise da correlação (Tabela 4), o ideal é que os itens de uma categoria apresentem correlações altas entre si (validade convergente), enquanto itens de categorias diferentes devem apresentar baixas correlações (validade discriminante). As categorias de tipografia, imagens e escrita foram as que apresentaram as melhores intercorrelações. Para as demais categorias, que não apresentaram resultados tão bons, é esperado que com o avanço do ensino de design visual sejam obtidos resultados melhores. Porém, também são apontadas oportunidades de melhoria em alguns dos itens.

**Tabela 3. Correlação item-total e alfa de cronbach**

	Item	Correlação item-total	Alfa de Cronbach removendo o item
L1	Número de elementos	0,11	0,81
L2	Alinhamento	0,34	0,80
L3	Visibilidade	0,22	0,80
<b>L4</b>	<b>Ocupação da tela</b>	<b>-0,14</b>	<b>0,82</b>
L5	Margens do texto	0,38	0,80
L6	Posicionamento de botões	0,40	0,80
C1	Tamanho mínimo	0,01	0,81
C2	Formato dos controles	0,21	0,80
C3	Padronização de tamanho	0,17	0,81
C4	Consistência nos controles	0,42	0,80
C5	Botões com formato oval	0,11	0,81
O1	Quantidade de cores	0,23	0,80
O2	Contraste do texto	0,15	0,81
O3	Paleta de cores	0,00	0,81
<b>O4</b>	<b>Harmonia de cores</b>	<b>-0,04</b>	<b>0,81</b>
O5	Quantidade de cores do texto	0,50	0,80
O6	Níveis de intensidade de cores	0,17	0,80
T1	Família da fonte	0,30	0,80
T2	Padronização do tamanho dos títulos	0,40	0,80
T3	Quantidade de tamanhos de texto	0,42	0,80
T4	Tamanho do texto regular	0,41	0,80
T5	Tamanho de texto dos botões	0,47	0,79
T6	Uso de itálico	0,30	0,80
T7	Alinhamento do texto	0,29	0,80
I1	Estilo dos ícones	0,35	0,80
I2	Esquema de cores dos ícones	0,37	0,80
I3	Uso de imagens pixelizadas	0,28	0,80
I4	Distorção de imagens (botões e ícones)	0,24	0,80
I5	Distorção de imagens	0,31	0,80
I6	Distorção de imagens (plano de fundo)	0,28	0,80
I7	Ajuste de tamanho	0,34	0,80
I8	Uso de imagens de texto	0,55	0,79
I9	Contraste de ícones	0,32	0,80
E1	Uso de caixa-alta em botões	0,22	0,80
E2	Capitalização do texto	0,26	0,80
E3	Uso indevido de sinais de pontuação	0,32	0,80
E4	Pontuação	0,39	0,80
E5	Quantidade de caracteres em botões	0,40	0,80
E6	Quantidade de caracteres em menus	0,08	0,81
E7	Sinais de pontuação agregados	0,36	0,80
E8	Uso de caixa-alta fora de botões	0,40	0,80



possui o componente em questão, são adicionadas opções de “não se aplica”. Esta opção é considerada no estudo de caso como uma nova categoria de resposta, isto evita a perda de informação nas avaliações estatísticas.

## 6. Conclusão

Como resultado do presente trabalho, foi feita uma proposta de atualização do *checklist* CodeMaster UI Design, desenvolvendo uma versão 2.0. Sendo este um *checklist* voltado para o ensino de design visual atrelado ao ensino de programação, com o uso do App Inventor. O desenvolvimento deste checklist foi realizado considerando diretrizes e princípios do design visual, e teve como base trabalhos acadêmicos anteriores. Também, como resultado, existem os dados da avaliação do *checklist* que podem servir para pesquisas futuras.

## Referências

- App Inventor 2 (2020) Disponível em: <<https://appinventor.mit.edu/>>. Acesso em: 14 out. 2020.
- BALAGTAS-FERNANDEZ, F.; FORRAI, J.; HUSSMANN, H. (2009) Evaluation of User Interface Design and Input Methods for Applications on Mobile Touch Screen Devices. In: GROSS, T. et al. (Eds.). Human-Computer Interaction – INTERACT 2009. Berlin, Heidelberg: Springer. v. 5726, p. 243–246.
- BALLANTYNE, M.; JHA, A.; JACOBSEN, A.; HAWKER, J. S.; EL-GLALY, Y. N. (2018) Study of Accessibility Guidelines of Mobile Applications. Proceedings of the 17th International Conference on Mobile and Ubiquitous Multimedia. Anais... MUM 2018. New York, NY, USA: Association for Computing Machinery.
- BROWN, T. A. (2006) Confirmatory factor analysis for applied research. New York: The Guilford Press.
- CARRARA, G. (2020) Número de celulares conectados no Brasil cresce 24 milhões. Disponível em: <<https://noticias.r7.com/tecnologia-e-ciencia/numero-de-celulares-conectados-no-brasil-cresce-24-milhoes-06022020>>. Acesso em: 16 nov. 2020.
- CNE (2020) CodeMaster V2.0. Disponível em: <<http://apps.computacaonaescola.ufsc.br:8080/aluno.jsp>>. Acesso em: 13 out. 2020.
- COHEN, Jacob. (1998) Statistical Power Analysis for the Behavioral Sciences. New York: Routledge Academic.
- CRONBACH, L. J. (1951) Coefficient alpha and the internal structure of tests. Psychometrika, v. 16, n. 3, p. 297–334.
- DEVELLIS, R. F. (2003) Scale development: theory and applications. Thousand Oaks: SAGE Publications.

- FERREIRA, M. N. F.; PINHEIRO, F. C.; WANGENHEIM, C. G; MISSFELDT FILHO, R. (2019a) Ensinando Design de Interface de Usuário na Educação Básica: Um Mapeamento Sistemático do Estado da Arte e Prática. In: Anais do XXV Workshop de Informática na Escola. Brasília, DF, Brasil.
- FERREIRA, M. N. F.; WANGENHEIM, C. G; MISSFELDT FILHO, R.; PINHEIRO, F. C.; HAUCK, J. C. (2019b) R. Learning user interface design and the development of mobile applications in middle school. *Interactions*, v. 26, n. 4, p. 66–69.
- GARRETT, J. J. (2011) *The elements of user experience: user-centered design for the Web and beyond*. 2nd ed. Berkeley, CA: New Riders.
- GLORFELD, L. W. (1995) An Improvement on Horn's Parallel Analysis Methodology for Selecting the Correct Number of Factors to Retain. *Educational and Psychological Measurement*, v. 55, n. 3, p. 377–393.
- GOOGLE (2014) Material Design. Disponível em: <<https://material.io/>>. Acesso em: 13 out. 2020.
- GRESSE VON WANGENHEIM, C. et al. (2016) A Usability Score for Mobile Phone Applications Based on Heuristics: *International Journal of Mobile Human Computer Interaction*, v. 8, n. 1, p. 23–58.
- HUANG, Z.; TIAN, Z. (2018) Analysis and Design for Mobile Applications: A User Experience Approach. In: MARCUS, A.; WANG, W. (Eds.). . *Design, User Experience, and Usability: Theory and Practice*. Cham: Springer International Publishing. v. 10918p. 91–100.
- ISO 9241-210:2019 (2019) Disponível em: <<https://www.iso.org/standard/77520.html>>. Acesso em: 14 out. 2020.
- ISO/IEC 25010:2011. (2011) Disponível em: <<https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/03/57/35733.html>>. Acesso em: 14 out. 2020.
- JACKSON, J. E. (2005) Oblimin Rotation. *Encyclopedia of Biostatistics*, John Wiley & Sons, Ltd.
- JI, Y. G.; PARK, J. H.; LEE, C.; YUN, M. H. (2006) A Usability Checklist for the Usability Evaluation of Mobile Phone User Interface. *International Journal of Human-Computer Interaction*, v. 20, n. 3, p. 207–231.
- JOYCE, G.; LILLEY, M.; BARKER, T.; JEFFERIES, A. (2016) Mobile Application Usability: Heuristic Evaluation and Evaluation of Heuristics. In: AMABA, B. (Ed.). *Advances in Human Factors, Software, and Systems Engineering*. Cham: Springer International Publishing. v. 492p. 77–86.
- KORTUM, P.; SORBER, M. (2015) Measuring the Usability of Mobile Applications

- for Phones and Tablets. *International Journal of Human-Computer Interaction*, v. 31, n. 8, p. 518–529.
- LARDINOIS, F. (2018) Google makes its Material Design system easier to customize. *TechCrunch*. Disponível em: <<https://social.techcrunch.com/2018/05/08/google-makes-its-material-design-system-easier-to-customize/>>. Acesso em: 14 out. 2020.
- MCHUGH, M. L. (2012) Interrater reliability: the kappa statistic. *Biochemia Medica*, v. 22, n. 3, p. 276–282.
- NIELSEN, J. (1994a) Usability inspection methods. Conference companion on Human factors in computing systems - CHI '94. Anais... In: CONFERENCE COMPANION. Boston, Massachusetts, United States: ACM Press.
- NIELSEN, J. (1994b) Usability engineering. Boston, Massachusetts, United States: Academic Press.
- OLSINA, L.; SANTOS, L.; LEW, P. (2014) Evaluating Mobileapp Usability: A Holistic Quality Approach. In: CASTELEYN, S.; ROSSI, G.; WINCKLER, M. (Eds.). *Web Engineering*. Cham: Springer International Publishing. v. 8541p. 111–129.
- OLSSON, U. (1979) Maximum likelihood estimation of the polychoric correlation coefficient. *Psychometrika*, v. 44, n. 4, 1979. p. 443–460.
- PETERSEN, K. et al. (2008) Systematic Mapping Studies in Software Engineering. In: 12TH INTERNATIONAL CONFERENCE ON EVALUATION AND ASSESSMENT IN SOFTWARE ENGINEERING (EASE).
- SAURO, J. (2019) Understanding Expert Reviews and Inspection Methods. *MeasuringU*. Disponível em: <<https://measuringu.com/inspection-methods/>>. Acesso em: 14 out. 2020.
- SCHLATTER, T.; LEVINSON, D. A. (2013) Visual usability: principles and practices for designing digital applications. Amsterdam: Morgan Kaufmann.
- SHARP, H.; ROGERS, Y.; PREECE, J. (2005) Design de interação: além da interação humano-computador. reimpressão ed. Porto Alegre, Brasil: Bookman.
- SOLECKI, I.; PORTO, J. V. A.; ALVES, N. C.; WANGENHEIM, C. G.; HAUCK, J.; BORGATTO, A. F. (2020). Automated Assessment of the Visual Design of Android Apps Developed with App Inventor. Proceedings of the 51st ACM Technical Symposium on Computer Science Education. Anais... SIGCSE '20. Portland, OR, USA: Association for Computing Machinery.
- STOYANOV, S. R.; HIDES, L.; KAVANAGH, D. J.; ZELENKO, O.; TJONDRONEGORO, D.; MANI, M. Mobile App Rating Scale: A New Tool for

Assessing the Quality of Health Mobile Apps. JMIR mHealth and uHealth, v. 3, n. 1, p. e27, 11 mar. 2015.

TREERATANAPON, T. (2012) Design of the Usability Measurement Framework for Mobile Applications. Proceedings of International Conference on Computer and Information Technology (ICCIT'2012). Anais... In: INTERNATIONAL CONFERENCE ON COMPUTER AND INFORMATION TECHNOLOGY (ICCIT'2012). Hammamet, Tunisia.

W3C (2013) Guidance on Applying WCAG 2.0 to Non-Web Information and Communications Technologies (WCAG2ICT). Disponível em: <<https://www.w3.org/TR/wcag2ict/>>. Acesso em: 14 out. 2020.

W3C (2014) Diretrizes de Acessibilidade para Conteúdo Web (WCAG) 2.0. Disponível em: <<https://www.w3.org/Translations/WCAG20-pt-br/>>. Acesso em: 14 out. 2020.

W3C (2015) Mobile Accessibility: How WCAG 2.0 and Other W3C/WAI Guidelines Apply to Mobile. Disponível em: <<https://www.w3.org/TR/mobile-accessibility-mapping/>>. Acesso em: 14 out. 2020.

WASSERMAN, A. I. (2010) Software engineering issues for mobile application development. Proceedings of the FSE/SDP workshop on Future of software engineering research - FoSER '10. Anais... In: THE FSE/SDP WORKSHOP. Santa Fe, New Mexico, USA: ACM Press.

ZEN, M.; VANDERDONCKT, J. (2016) Assessing User Interface Aesthetics based on the Inter-subjectivity of Judgment. In: Proc. of the 30TH International BCS Human Computer Interaction Conference, Bournemouth, GB.