

Proposta de um *Checklist* de Avaliação de Usabilidade de Aplicativos Android no Contexto Educacional

João V. A. Porto¹, Heliziane Barbosa^{1,2}, Christiane Gresse von Wangenheim¹

¹Instituto Nacional para Convergência Digital (INCoD)/ Departamento de Informática e Estatística (INE)/ Universidade Federal de Santa Catarina (UFSC)

²Departamento de Expressão Gráfica/ Universidade Federal de Santa Catarina (UFSC)

{joao.porto, heliziane.barbosa}@grad.ufsc.com, c.wangenheim@ufsc.br

Abstract. *A captivating way of teaching computing can be by programming mobile applications with the block-based environment App Inventor. Yet, as the design of apps is crucial, this should also include the teaching of interface design concepts. Thus, in order to facilitate the assessment of such apps developed by students, we present a checklist that supports the evaluation of the conformity of the interface design with the style guide Material Design. The checklist has been systematically derived and evaluated with respect to face validity. It can be used as is by teachers to assess app designs and will also serve as basis for an automation of the assessment process.*

Resumo. *Uma forma cativante de ensinar computação pode ser pela programação de aplicativos móveis com o ambiente de programação baseado em blocos App Inventor. Entretanto, como o design dos aplicativos é crucial, isso também deve incluir o ensino de conceitos de design de interface. Assim, objetivando facilitar a avaliação de tais aplicativos desenvolvidos por alunos, apresentamos uma checklist que suporta a medição da conformidade do design de interface de aplicativos em relação ao guia de estilo Material Design. A checklist foi derivada e avaliada sistematicamente quanto a face validity. Pode ser usada por professores na avaliação do design de aplicativos e servirá como base para uma automatização do processo de avaliação.*

1. Introdução

Hoje, para ser um cidadão bem-educado num mundo permeado de Tecnologia da Informação, todos devem ter uma compreensão clara dos princípios e práticas da computação, o que vai além do simples uso da TI [CSTA 2017]. Este conhecimento é necessário para preparar os alunos para inserção no século XXI, independentemente da sua área final de estudo ou da carreira profissional escolhida [CSTA 2017]. A compreensão fundamental da computação permite aos alunos serem consumidores educados de tecnologia e criadores inovadores capazes de projetar sistemas de computação para melhorar a qualidade de vida de todas as pessoas.

Por isso, uma implementação rigorosa de ensino da computação nas escolas do Ensino Básico é essencial para criar essa ampla compreensão pública. Aprender computação envolve a aprendizagem do pensamento computacional, uma abordagem para a resolução de problemas, de forma que pode ser implementada com um

computador, usando um conjunto de conceitos, tais como abstração, recursão e iteração, processamento, análise de dados e criação de artefatos reais e virtuais [Wing 2006].

Uma estratégia para ensinar computação é por meio da programação de aplicativos para dispositivos móveis, uma vez que a população em geral já possui uma maior proximidade com os *smartphones* [O Estado de S. Paulo 2015]. Com esse objetivo foi desenvolvido o App Inventor [MIT 2017]. O App Inventor é um ambiente de programação baseado em blocos que permite a qualquer pessoa começar a programar e construir aplicativos para dispositivos Android. Em 2017 a comunidade do App Inventor já inclui mais de 6 milhões de usuários registrados em 195 países. Existem vários tutoriais de programação e/ou unidades instrucionais alinhadas as diretrizes do currículo do ensino de computação no Ensino Básico, como, por exemplo, a oficina do Jogo do Mosquito [Daniel et al. 2017], que lançam mão do App Inventor.

Porém, o desenvolvimento de aplicativos, envolve não somente competências de programação, mas também do design de interface, sendo considerado um conceito transversal no ensino de computação [CSTA 2017]. O design de interface visa maximizar a usabilidade e a experiência do usuário para tornar a interação do usuário efetiva, eficiente e satisfatória [Nielsen 1993]. Especialmente no contexto de aplicativos móveis, a usabilidade é um fator crítico para o sucesso desses [Yin et al. 2014] [Nascimento et al. 2016]. O design de interface com boa usabilidade pode ser orientado por guias de estilo, como o Material Design [Google 2017] voltado ao design de interface para aplicativos Android. O guia de estilo fornece recomendações para o design de diversos elementos de interface, como cores, fontes, layout e componentes. Seguir um guia de estilo previne problemas de usabilidade, pois o mesmo mantém a consistência entre designs para a mesma plataforma.

Assim, o conhecimento em relação as recomendações do Material Design no processo de desenvolvimento de aplicativos é essencial para assegurar a usabilidade. Dessa forma, é importante ensinar competências em relação ao design de interface em conformidade com Material Design. Para acompanhar esse processo de aprendizagem é essencial avaliar os trabalhos dos alunos e fornecer *feedback* formativo e somativo para guiar a sua aprendizagem [Hattie e Timperley 2007]. Assim, faz parte da avaliação dos projetos criados pelos alunos a avaliação em relação a usabilidade e a sua conformidade com os princípios do Material Design. Porém, essa avaliação do design de interface requer recursos substanciais, além de poder introduzir inconsistência ou viés [Zen et al. 2011]. Portanto, a formalização de critérios explícitos de avaliação e a sua automação por meio da análise de código é uma forma promissora de melhorar essa situação, auxiliando no ensino desses conceitos [Grigera et al. 2017] [Ivory e Hearst 2001].

Existem diversos conjuntos de heurísticas para avaliação de usabilidade, contudo, a maioria voltada para interface gráfica de computadores desktop, como, por exemplo, as dez heurísticas de Nielsen (1994). Há poucos estudos na área de heurísticas de usabilidade para aplicativos. Especificamente voltados ao design de interface de aplicativos Android desenvolvidos com App Inventor, em relação ao guia de estilo Material Design, não existem ainda *checklists* de avaliação. Sendo assim, o presente artigo apresenta uma proposta de *checklist* para a avaliação de design de interface de aplicativos Android criados no App Inventor no contexto do Ensino Básico visando a sua automatização no futuro.

2. Background: Design de interface de aplicativos

O design de interface de aplicativos no App Inventor é feito por meio do Designer que permite incluir diversos componentes visuais como botões, caixas de textos, imagens etc. (Figura 1). De acordo com o tipo da componente, permite, também, configurar propriedades tais como altura, largura, fontes, cor de fundo, entre outras.

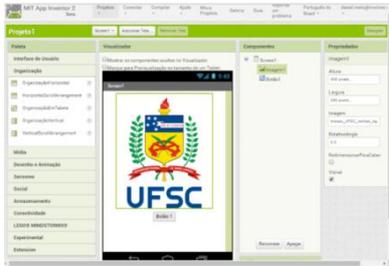


Figura 1. Área de trabalho do Designer do App Inventor

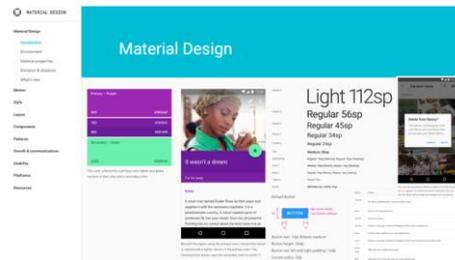


Figura 2. Exemplos de recomendações [Google 2017]

Dois aspectos fundamentais na área de desenvolvimento de aplicativos são a interação do usuário com o aplicativo e a qualidade dessa interação – a usabilidade. A usabilidade pode ser definida como “a medida pela qual um produto pode ser usado por usuários específicos para alcançar objetivos específicos com efetividade, eficiência e satisfação em um contexto de uso específico” [ISO 9241-11 2006]. No contexto de aplicativos móveis existem vários fatores que podem complicar a usabilidade. Por exemplo, a diferença de tamanho das telas e a natureza de interação do usuário, já que os dispositivos possibilitam o uso de aplicativos em qualquer lugar, em qualquer momento e por qualquer pessoa [Wasserman 2010] [Huang 2009].

Assim, para desenvolver aplicativos com um grau de usabilidade adequado é necessário seguir um processo sistemático de engenharia de usabilidade, tendo por base recomendações de design de interface. No contexto do design de interface de aplicativos Android, o Material Design visa criar uma linguagem visual que sintetiza princípios clássicos de design com inovação, tecnologia e ciência [Google 2017]. É, portanto, um guia de estilo que traz uma série de princípios com relação a elementos gráficos de aplicativos como, por exemplo, em relação a cores, iconografia, tipografia, proporção, tamanhos, espaçamento, animação, acessibilidade, componentes, entre outros. A Figura 2 apresenta alguns exemplos de recomendações.

3. Metodologia de pesquisa

A fim de desenvolver sistematicamente uma *checklist* de usabilidade, adotamos uma metodologia baseada em Rusu et al. (2011), como ilustrado na Figura 3.



Figura 3. Metodologia de pesquisa

A etapa exploratória consistiu em obter o estado da arte das *checklists* existentes para avaliação de usabilidade, conduzindo uma revisão sistemática da literatura. Durante a definição da revisão foram estabelecidos as perguntas de pesquisa e o

protocolo de revisão. A fase de execução foi realizada com base no protocolo de revisão, efetuando buscas nas fontes de dados definidas. Observada a ausência de trabalhos relevantes, foram executadas outras buscas, flexibilizando-se os critérios da busca sistemática. Os estudos encontrados foram analisados e os resultados sintetizados. Detalhes da revisão sistemática da literatura estão documentados em Porto et al. (2017). Como parte dessa etapa também foram levantados os elementos de design presentes no App Inventor, analisando-se as propriedades e limitações de tais elementos por meio do ambiente de programação. Por fim, considerando o contexto em que os aplicativos são desenvolvidos e o público-alvo, analisamos quais elementos de design são tipicamente usados em âmbito educacional.

A etapa descritiva e relacional consistiu em examinar os resultados captados anteriormente, observando-se os de maior relevância e estabelecendo relações entre os mesmos. Assim, os elementos de design do App Inventor foram relacionados com as diretrizes do Material Design, tendo-se como foco aqueles elementos utilizados com maior recorrência no contexto educacional. Como resultado, foi proposto um conjunto de critérios para avaliação do design de aplicativos com base no Material Design considerando as características do App Inventor.

Esta versão da *checklist* (v0.1) foi avaliada preliminarmente, verificando a validade de rosto e conteúdo. Seguindo Lawshe (1975), realizamos uma avaliação com um painel de especialistas de diferentes áreas de expertise. A partir do *feedback* dos especialistas, os critérios foram revisados definindo a versão 1.0 da *checklist*.

4. Estado da arte

Existem diversas propostas de heurísticas/*checklists* para avaliação da usabilidade de aplicativos para *smartphones* [Salazar et al. 2015]. Contudo, a maior parte delas consiste em adaptações de conjuntos de heurísticas tradicionais que têm por base a experiência de seus autores e/ou problemas de usabilidade observados com recorrência, de modo que não se fundamentam em nenhum guia de estilo. As poucas propostas que têm por base um guia de estilo não têm foco específico na usabilidade, além de serem projetadas de maneira a suportar um processo manual de avaliação e/ou de testes de usabilidade [Android 2017]. Assim, tais *checklists* não podem ser utilizadas para automatizar o teste de usabilidade dos aplicativos.

No âmbito educacional são encontradas principalmente rubricas voltadas a avaliações de projetos de aplicativos desenvolvidos por alunos, algumas específicas para aplicativos criados com App Inventor [Sherman et al. 2014] [Technovation 2014] e outras não restritas a esse [SLTP 2013]. Voltadas principalmente à avaliação de indicadores do pensamento computacional, essas rubricas geralmente não abordam critérios relacionados a usabilidade dos aplicativos ou os abordam de forma muito superficial, focando em avaliações manuais feitas pelos professores. Ademais, essas rubricas não se embasam em nenhum guia de estilo, de modo que possuem um caráter subjetivo, tendo em vista que cabe ao professor, de acordo com seu conhecimento e experiência, avaliar os aspectos de usabilidade.

Existem trabalhos na literatura que indicam a importância da automatização do processo de avaliação de usabilidade há algum tempo [Ivory e Hearst 2001]. Porém, observa-se que mesmo de forma mais ampla, propostas nesse sentido são escassas. A maior parte das propostas existentes apresenta ferramentas voltadas a avaliação da

usabilidade de *websites* e empregam diferentes conjuntos de diretrizes com ênfase nas *Web Content Accessibility Guidelines* [Web Accessibility Initiative 1999]. Não foi encontrada nenhuma ferramenta que avalia automaticamente a usabilidade de aplicativos desenvolvidos para *smartphones*.

Assim, atualmente não há somente ausência de *checklists* que suportam a medição da conformidade do design de interface de aplicativos criados com o App Inventor de acordo com as diretrizes do Material Design, mas observa-se a ausência de trabalhos que se fundamentam em quaisquer guias de estilo e tenham o objetivo de avaliar, de forma automatizada, a usabilidade de aplicativos para *smartphones*.

5. Análise de sistema e contexto

Com o objetivo de identificar os elementos de design de interface fornecidos pelo ambiente de programação App Inventor [MIT 2017], foi realizada uma análise do sistema. Foram incluídos neste levantamento todos os componentes visíveis cujas propriedades são editáveis. Esses elementos foram mapeados às seções relacionadas no guia de estilo Material Design [Google 2017] e são apresentados na Tabela 1.

Tabela 1. Relação dos elementos gráficos do App Inventor com itens do Material Design

Elemento do App Inventor (Designer)	Item Material Design
Estilo	
Diversos (botão, legenda, etc.)	Cor/Tipografia/Escrita
Imagem (ícone)	Ícones
Imagem	Imagens
Layout	
Organização horizontal	Princípios/Estrutura
Organização horizontal rolável	Princípios/Estrutura
Organização em tabela	Princípios/Estrutura
Organização vertical	Princípios/Estrutura
Organização vertical rolável	Princípios/Estrutura
Componentes	
Botão	Botões
Caixa de seleção	Controladores de seleção
Escolhe data	Pickers
Visualizador de listas	Listas
Notificador	Diálogos
Caixa de senha	Campos de texto
Deslizador	Deslizadores
Lista suspensa	Diálogos
Caixa de texto	Campos de texto
Escolhe hora	Pickers
Padrões	
Notificador	Notificações
Organização horizontal rolável	Técnicas de rolagem
Organização vertical rolável	Técnicas de rolagem

Com o objetivo de identificar quais desses elementos de design de interface do App Inventor são mais frequentemente usados no contexto educacional, ensinando a programação de aplicativos no Ensino Básico, foi realizado um estudo de caso. Foram aleatoriamente escolhidos 12 aplicativos criados por meninas com idades entre 14 e 18 anos no contexto da competição Technovation. Analisando esses aplicativos visualmente, foram registradas as quantidades de uso de cada um desses elementos de design de interface. A Figura 4 apresenta o resultado dessa análise, indicando a

frequência de uso e incluindo apenas componentes ou funcionalidades que são fornecidos pelo App Inventor e podem ser relacionados ao Material Design.

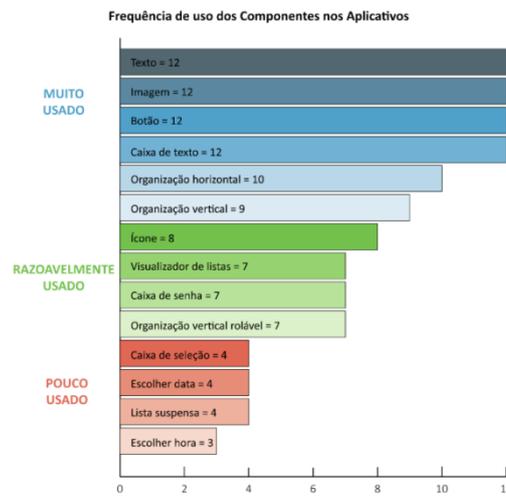


Figura 4. Frequência de uso dos componentes nos aplicativos

É possível observar que os componentes de texto, imagem, botão e caixa de texto são os mais utilizados, pois estão presentes em todos os aplicativos analisados. Já o elemento menos utilizado é a escolha de hora.

6. Definição dos critérios da *checklist*

Com o objetivo de suportar uma avaliação automatizada do design de aplicativos desenvolvidos com o App Inventor, foram selecionados os critérios de avaliação a partir do Material Design. Para a seleção dos critérios de avaliação foram levados em consideração somente critérios que:

- Avaliam a usabilidade;
- Avaliam elementos de design do App Inventor;
- Estão alinhados aos objetivos de aprendizagem referentes ao design de interface no Ensino Básico e que são frequentemente utilizados nesse contexto;
- Têm viabilidade para ser avaliados automaticamente.

A Tabela 2 apresenta todos os critérios selecionados, indicando o número e a categoria do elemento de design relacionados a cada um dos critérios.

Tabela 2. Critérios de avaliação do design de interface de aplicativos desenvolvidos com App Inventor em relação ao Material Design (v0.1)

No.	Categoria do elemento de design	Critério
1	Cores	Uso de somente duas cores diferentes (uma primária e outra secundária) em todo aplicativo, além de tons de preto, cinza e branco.
2		Contraste mínimo entre texto e fundo (cor ou imagem) para que seja legível.
3	Ícones	Ícones utilizados no aplicativo provêm do Material Design.
4	Imagens	Imagens precisam ter uma resolução/proporção mínima de qualidade, evitando pixelização ou distorções.
5	Tipografia	Mantém um padrão da tipografia utilizada em todo o aplicativo (por exemplo, uso da fonte Roboto ou estilo sans-serif em todos os componentes que possuem texto).
6		Mantém um padrão no tamanho da tipografia utilizada em todo o aplicativo.
7	Escrita	Usa um padrão de textos para botões e demais componentes análogos.
8		Títulos, rótulos e itens de menus devem usar sentenças de texto no estilo capitalizado (somente a primeira palavra da sentença capitalizada).

No.	Categoria do elemento de design	Critério
9		Evitar uso de ponto final em sentenças de texto solitárias nos seguintes componentes: rótulos, listas e corpo do texto de diálogos.
10		Pontos finais são usados em sentenças de texto com mais de uma linha ou sentenças seguidas por um link.
11		Evitar uso de dois pontos.
12	Tamanho do alvo de toque	Tamanho dos componentes nas dimensões horizontal e vertical é o mínimo para garantir um tamanho físico de cerca de 9mm independentemente do tamanho da tela.
13	Estrutura	A estrutura do aplicativo deve possuir uma barra permanente e um botão flutuante. Uma barra inferior pode ser inclusa para disponibilizar funcionalidades ou ações adicionais.
14		Usa sempre que possível os organizadores horizontais, verticais ou vertical rolável.
15	Barra do aplicativo	Formato da barra do aplicativo corresponde com o padrão.
16	IU responsivo	Usa especificação de tamanho de elementos em porcentagem (relativos) e não tamanhos fixos (absolutos), possibilitando design responsivo para tamanhos de telas diferentes.
17	Botão	Todos os botões devem conter texto em caixa alta.
18		Texto dos botões deve ter tamanho de 14pt.
19		Altura dos botões deve ser de no mínimo 48dp.
20		Botões usam cor secundária como cor de fundo (<i>raised buttons</i>) ou usam a mesma cor do fundo da área da tela (<i>flat button</i>)
21		Botões com ações assertivas/ confirmação são localizadas a direita e botões com ações negativa/conclusão a esquerda.
22	Listas	Elementos de listas são apresentados em uma ordem intrínseca (por exemplo, alfabética).
23	Menus	Nomes dos itens de menus começam com letra maiúscula.
24	Caixa de texto	Todas as caixas de texto devem possuir uma dica de preenchimento.
25	Deslizador	Todos os deslizadores devem utilizar as cores secundárias para realçá-los em meio aos demais componentes.
26	Confirmação	Confirmações seguem o padrão em termos de cor de fundo, fonte e botões.
27	Erros	Erros em relação a campos de textos são apresentados de forma <i>inline</i> .
28		Erros gerais do aplicativo são apresentados seguindo o padrão em termos de cor de fundo, fonte e botões.

7. Avaliação da *checklist*

7.1 Definição e execução da avaliação

Com o objetivo de verificar se a *checklist* pode auxiliar a avaliação da usabilidade/design de interface de aplicativos desenvolvidos com App Inventor no contexto educacional do Ensino Básico, foi realizada uma avaliação via painel de especialistas. Essa avaliação consistiu em analisar a *checklist* em termos de *face validity*, incluindo aspectos de completude, corretude, adequação para o contexto educacional, viabilidade de criação do elemento de design com App Inventor e viabilidade de automação da avaliação. A execução da avaliação da *checklist* foi realizada em outubro 2017. Participaram dessa avaliação 12 pesquisadores e profissionais com perfil apresentado na Figura 5, incluindo pesquisadores e profissionais com expertise nas áreas de design, desenvolvimento de software e ensino de computação no Ensino Básico.



Figura 5. Perfil dos participantes do painel de especialistas

7.2 Análise dos dados

O feedback dos avaliadores, em geral, confirmou a corretude da *checklist*. Somente o critério que definia a altura dos botões foi ajustado (alterando-se a unidade de medida de dp, do Material Design, para percentuais), levando-se em consideração questões de responsividade.

A *checklist* também foi considerada completa pela ampla maioria dos especialistas. Nesse sentido, uma das únicas sugestões foi a de incluir novos critérios para avaliar a conformidade do design do aplicativo com padrões de telas, detalhando-se o critério referente a estrutura das telas proposto na *checklist*.

Por outro lado, foi sugerida a exclusão de alguns critérios devido a diferentes fatores. Foi sugerida a exclusão dos critérios 15 e 27 por representarem conceitos de design avançados demais ao contexto do Ensino Básico. Foi sugerida, também, a exclusão dos critérios 13, 15, 26, 27 e 28, em decorrência da impossibilidade de criação dos elementos de design abordados nesses critérios com o App Inventor. Entretanto, a maioria das propostas de exclusão deveu-se a impossibilidade ou dificuldade de automatizar os critérios 7, 13, 15, 21, 22, 26, 27 e 28. Todos os critérios excluídos estão marcados em cinza na Tabela 2.

Assim, como resultado dessa avaliação, foi definida uma *checklist* com 19 critérios restantes, levando-se em consideração o feedback dos especialistas consultados.

7.3 Ameaças à Validade

Sendo somente uma avaliação inicial, diversos fatores podem ter influenciado ou ameaçando os resultados. Visando a redução de riscos em relação ao conteúdo, adotamos um processo sistemático, derivando sistematicamente as métricas do objetivo de avaliação, bem como o instrumento de coleta de dados. Apesar de um número razoável de participantes da avaliação, seria importante repetir a avaliação com um número maior, incluindo também especialistas de outras instituições. Ademais, essa avaliação inicial teve por foco somente a análise da *face validity*. Para obter resultados mais abrangentes em relação a outros tipos de validade, também faz-se necessária a realização de outros tipos de avaliação no futuro.

8. Conclusão

Para contribuir com ensino de computação no contexto do Ensino Básico, incluindo também o ensino de conceitos de design de interface, foi proposta uma *checklist* para avaliação do design de interface de aplicativos Android desenvolvidos com o App Inventor. A *checklist* guia uma avaliação desses designs de interface de forma objetiva, avaliando a sua conformidade em relação ao Material Design, um guia de estilo para aplicativos Android. Uma avaliação inicial forneceu os primeiros indícios de que a *checklist* é correta, completa e adequada nesse contexto. Assim, como resultado desta pesquisa, a *checklist* já pode ser utilizada por professores, de forma manual, para avaliar os projetos de aplicativos desenvolvidos por alunos. Além disso, no contexto da nossa pesquisa, direcionada ao ensino de computação, daremos continuidade automatizando o processo de avaliação com base na mesma.

Agradecimentos

Agradecemos todos os participantes da avaliação.

Este trabalho foi apoiado pelo CNPq, entidade do governo brasileiro focada no desenvolvimento científico e tecnológico.

Referências

- Android. Developers Guide (2017). Disponível em: <https://developer.android.com/develop/quality-guidelines/core-app-quality.html>
- Porto, J. V. A.; Gresse von Wangenheim, C.; Barbosa, H. (2017) Heurísticas de usabilidade de aplicativos Android desenvolvidos com App Inventor: Uma Revisão Sistemática da Literatura. Relatório Técnico INCoD/GQS.07.2017.P, Florianópolis/Brasil.
- Capelas, B (2017) Até o fim de 2017, Brasil terá um smartphone por habitante, diz FGV. O Estado de S. Paulo, 19 abr. 2017.
- CSTA. Standards - CSTA K-12 Computer Science Standards (2017) Disponível em: <https://sites.google.com/site/cstastandards/standards>. Acesso em: 04 out. 2017.
- Daniel, G. T.; Gresse Von Wangenheim, C.; Medeiros, G.; Alves, N. d. C. (2017) Ensinando a Computação por meio de Programação com App Inventor. Proc. of Computer on the Beach, Florianópolis/Brazil.
- Google. Introduction - Material Design. (2017) Disponível em: <https://material.io/guidelines/#introduction-principles>. Acesso em: 16 out. 2017.
- Grigera, J. et al. (2017) Automatic detection of usability smells in web applications. International Journal of Human-Computer Studies, 97, 129-148.
- Hattie, J., Timperley, H. (2007). The power of feedback. Review of educational research, 77(1).
- Huang, K. (2009) Challenges in Human-Computer Interaction Design for Mobile Devices. Proc. of the World Congress on Engineering and Computer Science, San Francisco, CA, USA.
- International Organisation for Standardisation, ISO 9241-110:2006, “Ergonomics of human-system interaction -- Part 110: Dialogue principles”.
- Ivory, M. Y., Hearst, M. A. (2001) The state of the art in automating usability evaluation of user interfaces. ACM Computing Surveys, 33(4), 470-516.
- Kitchenham, B., Charters, S. (2007) Guidelines for performing systematic literature reviews in software engineering Technical report, Keele University and University of Durham, UK.
- Lawshe, C.H. (1975). A quantitative approach to content validity. Personnel Psychology, 28, 563–575.
- Massachusetts Institute of Technology (2017) MIT App Inventor News & Events | Explore MIT App Inventor. Disponível em: <http://appinventor.mit.edu/>. Acesso em: 16 out. 2017.
- Nascimento, I. et al. (2016) An Empirical Study to Evaluate the Feasibility of a UX and Usability Inspection Technique for Mobile Applications. Proc. of 28th

- Int. Conf. on Software Engineering & Knowledge Engineering, California, USA.
- Nielsen, J. (1993) Usability Engineering. Morgan Kaufmann, 362 p.
- Qualidade do aplicativo principal | Android Developers (2017). Disponível em: <<https://developer.android.com/develop/quality-guidelines/core-app-quality.html>>. Acesso em: 04 set. 2017.
- Rubric: Nku Mobile App Design. Student Technology Leadership Program, Kentucky Department of Education. Jul. (2013). Disponível em: <<https://stlp.education.ky.gov/wp-content/uploads/2013/07/NKU-Mobile-App-Design-Challenge-rubric.pdf>>. Acesso em: 04 dez. 2017.
- Rusu, C. et al. (2011) A Methodology To Establish Usability Heuristics. Proc. of the 4th Int. Conf. on Advances in Computer-Human Interactions, Gosier, Guadeloupe, France.
- Salazar, L. H. A. et al. (2013) A Systematic Literature Review on Usability Heuristics for Mobile Phones. International Journal of Mobile Human Computer Interaction, 5(2), 50-61.
- Sherman, M. et al. (2014) App Inventor Project Rubric — Computational Thinking through Mobile Computing. Disponível em: <<https://nsfmoblect.files.wordpress.com/2014/09/mobile-ct-rubric-for-app-inventor-2014-09-01.pdf>>. Acesso em: 04 out. 2017.
- Technovation. Teacher and Mentor Lesson Guide (2017). Disponível em: <http://technovationchallenge.org/wp-content/uploads/2014/01/TeacherMentorLessonGuide_L1_6.pdf>. Acesso em: 04 out. 2017.
- Wasserman A. I. (2010) Software engineering issues for mobile application development. Proc. of Workshop on Future of Software Engineering Research, Santa Fe, New México, USA.
- Web Accessibility Initiative. W3C (2008). Web Content Accessibility Guidelines. Disponível em: <<https://www.w3.org/WAI/intro/wcag>>. Acesso em: 06 out. 2017.
- Wing, J. M. (2006). Computational thinking. Communications of the ACM, 49(3), 33-35.
- YIN, P. L. et al. (2014) Entrepreneurial Innovation: Killer Apps in the iPhone Ecosystem. American Economic Review, 104(5), 255-259.
- Zen, K., Iskandar, D.N.F.A., Linang, O. (2011). Using Latent Semantic Analysis for automated grading programming assignments. Proc. of the 2011 Int. Conference on Semantic Technology and Information Retrieval, Kuala Lumpur, Malaysia, 82 –88.