

# Ensinando Design de Interface de Usuário de Aplicativos Móveis no Ensino Fundamental

## *Teaching User Interface Design of Mobile Applications in Middle School*

M. Nathalie F. Ferreira  
Programa de Pós-Graduação em  
Design – Universidade Federal de  
Santa Catarina (UFSC)  
[nathalie.fortuna@posgrad.ufsc.br](mailto:nathalie.fortuna@posgrad.ufsc.br)

Fernando da Cruz Pinheiro  
Departamento de Informática e  
Estatística – Universidade Federal  
de Santa Catarina (UFSC)  
[fernando.pinheiro@posgrad.ufsc.br](mailto:fernando.pinheiro@posgrad.ufsc.br)

Christiane Gresse von  
Wangenheim  
Departamento de Informática e  
Estatística – Universidade Federal  
de Santa Catarina (UFSC)  
[c.wangenheim@ufsc.br](mailto:c.wangenheim@ufsc.br)

Raul Missfeldt Filho  
Departamento de Informática e  
Estatística – Universidade Federal  
de Santa Catarina (UFSC)  
[raul.missfeldt.filho@grad.ufsc.br](mailto:raul.missfeldt.filho@grad.ufsc.br)

Jean C. R. Hauck  
Departamento de Informática e  
Estatística – Universidade Federal  
de Santa Catarina (UFSC)  
[jean.hauck@ufsc.br](mailto:jean.hauck@ufsc.br)

### Resumo

A computação na Educação Básica geralmente é abordada pelo ensino de programação por meio de atividades práticas nas quais os alunos criam artefatos de software, como aplicativos móveis. Essa abordagem, no entanto, pode não abranger outras competências importantes, como o design da interface do usuário, que são essenciais para o desenvolvimento de software. Nesse contexto, este artigo apresenta o desenvolvimento, aplicação e avaliação de uma unidade instrucional que incorpora o ensino de competências de design de interfaces de usuário no ensino de computação. A unidade instrucional foi desenvolvida de forma sistemática seguindo um processo de design instrucional e aplicada e avaliada em uma escola pública brasileira. Os primeiros resultados indicam que a dinâmica pode ter um impacto positivo na motivação, na experiência do usuário e na aprendizagem assim como empoderar os alunos tornando-os criadores ativos de produtos de TI.

**Palavras-Chave:** Design de Interface de Usuário; Design Thinking; Ensino de Computação; Algoritmos e Programação, App Inventor, Ensino Fundamental

### Abstract

Teaching computing in K-12 is typically accomplished by teaching programming through practical activities in which students create software artifacts such as mobile applications. This approach, however, may not cover other important competencies such as user interface design, which are essential for software development. Within this context, this article presents the development, application and evaluation of an instructional unit that incorporates the teaching of UI design competencies into computing education. The instructional unit was developed in a systematic way following an instructional design process, and applied and evaluated in a Brazilian public middle school. First results indicate it can have a positive impact on motivation, user experience, and learning as well as empower students to become active creators of IT products.

**Keywords:** User Interface Design, Design Thinking, Computing Education; Algorithms and Programming; App Inventor; Middle school.

## 1 Introdução

Nos últimos anos, nota-se a crescente demanda do pensamento computacional como uma habilidade do século XXI (Grover & Pea, 2013; Wing, 2006). O pensamento computacional

Cite as: Ferreira, M. N. F., Pinheiro, F. da C., Gresse von Wangenheim, C., Missfeldt Filho, R., & Hauck, J. C. (2020). Teaching User Interface Design of Mobile Applications in Middle School (Ensinando Design de Interface de Usuário de Aplicativos Móveis no Ensino Fundamental). Brazilian Journal of Computers in Education (Revista Brasileira de Informática na Educação - RBIE), 28, 48-72. DOI: 10.5753/RBIE.2020.28.0.48

denota a ideia de desenvolver uma solução genérica para um problema decompondo-o, identificando variáveis e padrões relevantes, derivando um procedimento de solução algorítmica (Wing, 2006). Como tal, o pensamento computacional representa uma capacidade cognitiva para aplicar conceitos fundamentais e raciocínio que derivam da ciência da computação em geral e da programação/codificação de computadores em particular para diferentes outros domínios, incluindo atividades da vida real (Wang, 2015). Portanto, o ensino do pensamento computacional tem sido um foco dos esforços mundiais de educação de computação no Ensino Fundamental para preparar as crianças para demandas atuais e futuras de nossas sociedades do conhecimento (Grover & Pea, 2013; Kafai & Burke, 2013; Resnick et al., 2009). Muitas dessas iniciativas se concentram no ensino de programação, que não é apenas uma parte fundamental da computação, mas também uma ferramenta fundamental para apoiar as tarefas cognitivas envolvidas no pensamento computacional (Grover & Pea, 2013). A programação no Ensino Fundamental é tipicamente introduzida usando linguagens de programação visuais baseadas em blocos, como o App Inventor (<http://appinventor.mit.edu/explore>) para desenvolver aplicações móveis, concentrando no ensino de conceitos de programação, como eventos, condicionais, loops e assim por diante.

No entanto, essas iniciativas geralmente não cobrem outras competências importantes, como o design da interface do usuário (*user interface - UI*), que é essencial para o desenvolvimento de software e faz parte do corpo de conhecimento de computação (ACM, 2013). O design da UI visa maximizar a usabilidade e a experiência do usuário (Cooper, 2014). É um componente essencial da Interação Humano-Computador, uma disciplina focada no design, avaliação e implementação de sistemas de software interativo para uso humano e no estudo dos principais fenômenos que os cercam (Hewett et al., 1992).

As competências relacionadas ao design da UI são importantes não apenas para os profissionais de Tecnologia da Informação (TI), mas para qualquer pessoa, pois estão relacionadas às habilidades do século XXI (AIGA, 2013). Cabe ressaltar que estas habilidades estão em conformidade com as competências sugeridas pela Base Nacional Comum Curricular (BNCC) quando propõe o exercício da curiosidade intelectual e o uso de diferentes linguagens (visual e digital) (MEC, 2018). Ademais, diretrizes curriculares, como a *K-12 Computer Science Framework* (CSTA, 2017), indicam a importância do ensino do design da UI também na Educação Básica. Inclusive alguns países já começaram a incorporar o ensino de design para estudantes da Educação Básica, como China e Coréia do Sul (West-Knights, 2017; Ahn, 2012). A inserção de conhecimentos do design da UI na Educação Básica proporciona diversos benefícios, incluindo a aprendizagem de métodos de pesquisa, visualização e apresentação de informações, incentivo à análise crítica e colaboração e treinamento (AIGA, 2013). Aprender design pode tanto incentivar os alunos a serem imaginativos, quanto ensiná-los como aproveitar essa inventividade e colocá-la em prática (AIGA, 2013). Desta forma, a aprendizagem das competências de design fornece soluções e formas de engajamento com o mundo que permitem aos indivíduos atuarem como agentes de mudança e criação do século XXI (Kolodner, 2002; Christensen et al., 2016). Outro benefício da incorporação de conteúdos relacionados ao design da UI no ensino de computação é a ampliação da percepção de que a computação é mais do que apenas codificar e pode, assim, aumentar o interesse nesta área (Robinson, Pérez-Quinones & Scales, 2015) e ser uma motivação para uma carreira em TI (Yardi et al., 2008; Craig & Horton, 2009).

Embora existam vários esforços de ensino de design de uma forma criativa (projeto City<sup>1</sup> ou o laboratório K-12 da Stanford dschool<sup>2</sup>) no contexto do *maker movement*, basicamente não

---

<sup>1</sup> <http://www.cityxproject.com>

<sup>2</sup> <https://dschool.stanford.edu/programs/k12-lab-network>

existem iniciativas especificamente destinadas a ensinar conceitos básicos de design da UI (Ferreira et al., 2018). Assim, a fim de explorar a questão do ensino de design de UI na Educação Básica, desenvolvemos uma unidade instrucional “Faça seu próprio app”, com o objetivo de ensinar competências básicas de design visual de UI, incorporadas no ensino de computação, por meio do desenvolvimento de aplicações móveis. Aplicamos e avaliamos a qualidade dessa unidade instrucional por meio de um estudo de caso com alunos dos anos finais do Ensino Fundamental de uma escola pública.

## 2 Metodologia de pesquisa

O presente trabalho foi desenvolvido como pesquisa aplicada visando produzir um conhecimento que possa ser efetivamente aplicado no mundo real, para contribuir ao ensino de computação na Educação Básica. Para a realização da pesquisa foi utilizada uma abordagem multimétodo, conforme ilustrado na Figura 1, adotando diferentes métodos científicos adequados para cada etapa da pesquisa.

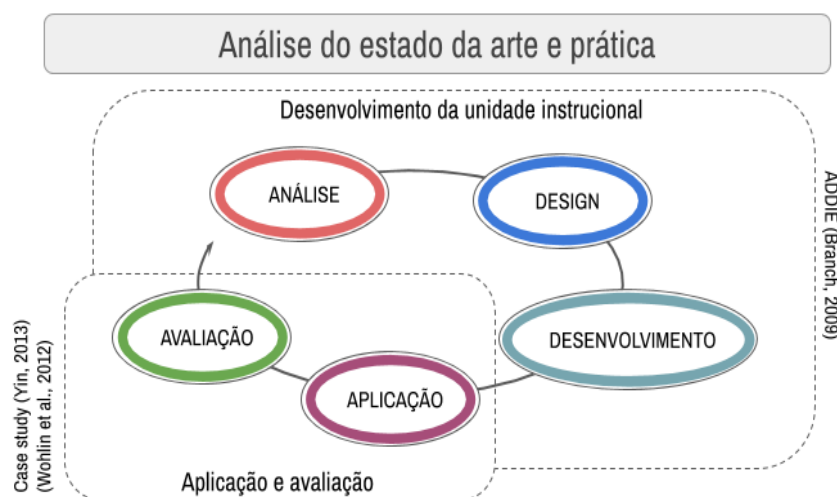


Figura 1: Metodologia de Pesquisa.  
Fonte: Elaborada pelos autores.

**Revisão da literatura.** Inicialmente, foi revisada a literatura existente sobre unidades instrucionais voltadas ao ensino de design visual de UI na Educação Básica (Ferreira et al., 2019), realizando um estudo de mapeamento sistemático seguindo Petersen et al. (2008). Foram analisados artigos publicados em Inglês e Português disponíveis na Web por meio das principais bibliotecas e bases de dados digitais (IEEE Xplore, Biblioteca Digital ACM, Wiley, Springer e Scopus), Google Scholar e plataformas de MOOCs, incluindo Udemy, Edx, Khanacademy e Coursera para minimizar os riscos de omissão. Os resultados dessa revisão são resumidos na Seção 3.

**Desenvolvimento da unidade instrucional.** Seguindo o modelo de design instrucional ADDIE (Branch, 2009), a unidade instrucional “Faça seu app” foi sistematicamente desenvolvida incorporando o ensino de competências de design visual de UI. Na primeira etapa foi analisado o contexto educacional e perfil dos alunos. A partir dessa análise e com base nas diretrizes de currículo foram definidos os objetivos de aprendizagem, estratégia instrucional e o material instrucional (utilizando o App Inventor como ambiente de programação).

**Aplicação e avaliação da unidade instrucional.** A unidade instrucional foi aplicada e avaliada por meio de um estudo de caso como parte de um projeto da Iniciativa Computação

na Escola em uma escola pública localizada em Florianópolis em 2018-2. O estudo de caso foi sistematicamente definido, planejado, executado e analisado seguindo o processo proposto por Wohlin et al. (2012) e Yin (2013). A avaliação foi definida adotando a abordagem *Goal/Question/Metric* (GQM) (Basili et al., 1994) uma abordagem de medição que sistematicamente decompôs um objetivo de medição em métricas via perguntas de análise. A definição da medição foi baseada no modelo dTECT (Gresse von Wangenheim et al., 2017) que visa a avaliação de unidades instrucionais para o ensino de computação na Educação Básica. O modelo dTECT avalia a percepção da qualidade pelos alunos em termos da qualidade da unidade instrucional, experiência de computação e percepção da aprendizagem. A avaliação também foi baseada no modelo MEEGA+KIDS (Gresse von Wangenheim et al., 2018), derivado do modelo MEEGA (Petri et al., 2017), que visa a avaliação da qualidade de jogos educacionais em termos de experiência do jogador e usabilidade. Esses modelos de avaliação foram escolhidos como base pois representam modelos amplamente adotados e/ou largamente avaliados em relação a sua confiabilidade e validade. Cabe destacar que os dados foram coletados por meio de vários instrumentos e analisados de forma descritiva.

### 3 Estado da arte e prática

Como resultado do estudo de mapeamento sistemático (Ferreira et al., 2019), identificamos 16 unidades instrucionais que abordam o design de interface de usuário incorporado ao ensino de computação na Educação Básica (Tabela 1).

Tabela 1: Unidades instrucionais abordando o ensino de design de UI.

Referência	Título
(Chen & Huang, 2017)	Design thinking in App inventor game design and development: A case study
(Sullivan, Reamon & Louie, 2003)	Girls embrace technology: A summer internship for high school girls
(Ke & Im, 2014)	A case study on collective cognition and operation in team-based computer game design by middle-school children
(Robinson & Pérez-Quñones, 2014)	Underrepresented middle school girls: on the path to computer science through paper prototyping
(Van Wart et al., 2014)	Apps for social justice: Motivating computer science learning with design and real-world problem solving
(Denner et al., 2005)	The girls creating games program: Strategies for engaging middle-school girls in information technology
(Edutopia, 2015)	Coding by design first approach
(Codelikeagirl, 2017)	Learning design by making games in scratch
(Tekkie Uni, 2018)	Build Your First App
(Createlab, 2017)	My createlab
(Code, 2018)	Curriculum code
(Robinson, Pérez-Quinones & Scales, 2015)	Understanding the attitudes of African American middle school girls toward computer science.
(Technovation, 2018)	Technovation Challenge
(Codehs, 2018)	Codehs
(Code.org/App lab, 2018)	Applab
(Get Started With Code 2, 2017)	Get started with code 2

Apesar da relevância do design visual de UI no desenvolvimento de software, evidencia-se a indigência do ensino de competências afins na Educação Básica. As poucas unidades instrucionais que existentes ensinam diversos aspectos relacionados ao design, incluindo *design thinking*, design de UI, design visual e UX. Integrado ao ensino de computação, tipicamente se adotam abordagens ágeis nesse nível educacional. No entanto, também

observamos que várias unidades instrucionais são limitadas à apresentação de princípios gerais de interface mas não aprofundam aos seus elementos. Isso indica a necessidade de projetar unidades instrucionais mais detalhadas e práticas que ajudem de maneira mais eficaz e de forma concreta os alunos no projeto das interfaces de usuário.

Outra deficiência está relacionada à carência de detalhes sobre a avaliação da aprendizagem dos alunos referente a essas competências. Diversos artigos relatam a avaliação com base nos artefatos criados (Chen & Huang, 2017; Technovation, 2018; Code.org, 2018), limitando muitas vezes a avaliação das competências de design visual de UI por meio de um único critério, quando se avalia competências de computação por meio de um conjunto de vários critérios de forma mais detalhada. Exames ou testes, incluindo perguntas sobre o design visual da UI, também foram relatados apenas em alguns casos (Codehs, 2018; Createlab, 2017).

Outro fator que pode impedir uma adoção mais difundida das unidades instrucionais relatadas é a falta de informações e/ou disponibilidade de acesso aos materiais instrucionais. A predominância de material apenas em Inglês pode também dificultar sua aplicação no Brasil.

Além disso, observamos que muitos artigos não apresentam informações essenciais sobre o(s) objetivo(s) de aprendizagem e/ou estratégia instrucional, nem a metodologia utilizada para o desenvolvimento das unidades instrucionais. Esse ponto fraco também pode ser observado em relação à avaliação da maioria das unidades instrucionais. Vários estudos não relatam avaliações e/ou apenas de forma *ad-hoc* com pouco rigor científico. Como resultado se evidencia a necessidade de mais pesquisas nessa área para guiar de forma sistemática e fundamentada o ensino de competências de design na Educação Básica.

## 4 Desenvolvimento da unidade instrucional "Faça seu próprio app"

### 4.1 Análise do contexto

Seguindo o processo de design instrucional, os alunos e o ambiente são caracterizados e os objetivos de aprendizagem são definidos.

**Alunos:** O público-alvo desta unidade de ensino são alunos de escolas públicas ou privadas dos anos finais do Ensino Fundamental, com idade entre 10 e 15 anos. Normalmente, a maioria dos alunos nessa idade já tem conhecimento e habilidades no uso de computadores e outros dispositivos que acessam a internet, redes sociais, jogos digitais, vídeos e música em casa. Os estudantes, geralmente, também sabem como usar dispositivos eletrônicos (telefones celulares, computadores, tablets) em casa e/ou por meio de aulas tradicionais de informática na escola. A maioria deles têm seu próprio telefone celular (predominantemente Android) e normalmente passa muito tempo *online*, especialmente em redes sociais e/ou jogos digitais. Em relação às habilidades relacionadas à computação, especificamente a programação, alguns alunos já têm uma compreensão de programas de computador, mas poucos sabem como criar um.

**Ambiente escolar:** O ensino da computação no Ensino Fundamental tem sido introduzido nas escolas como parte de aulas regulares de forma multidisciplinar ou como atividades extracurriculares (Bordini, et al., 2016). No entanto, como a computação não faz parte do currículo básico no Brasil, apesar da BNCC (MEC, 2018) sugerir competências afins, as unidades instrucionais que abordam esse conteúdo geralmente têm curta duração. As aulas tipicamente ocorrem em laboratórios de informática com computadores (ou *notebooks*) e acesso à internet. Esses laboratórios de informática normalmente são usados por professores de outras áreas para trabalhos digitais ensinando alfabetização de TI, como edição de textos, apresentações, etc. Às vezes, as escolas contam com um professor da sala informatizada apoiando o uso de computadores nas aulas, porém observa-se que a maioria das escolas não

possui professores especificamente formados na área de computação. As turmas têm em média de 25 a 40 alunos. Ainda, no contexto das escolas públicas, geralmente, há poucos recursos disponíveis para a aquisição de material instrucional.

**Ensino de computação:** Espera-se que o ensino de design visual de UI seja inserido em uma unidade instrucional com o objetivo de ensinar o pensamento computacional enfocando em conceitos básicos de algoritmos e programação por meio do desenvolvimento de aplicativos móveis com o App Inventor. A unidade deve ser baseada nas principais diretrizes de currículo voltadas ao ensino de computação e design na Educação Básica (ACM/IEEE, 2013; UXQB, 2018; CSTA, 2017; AIGA, 2008; SBC, 2018).

#### 4.2 Definição dos objetivos de aprendizagem

O objetivo da unidade instrucional “Faça seu próprio app” é ensinar o pensamento computacional com foco em algoritmos e programação, bem como competências de design visual de UI e conceitos de engenharia de software para estimular a aprendizagem de desenvolvimento sistemático de apps. Os objetivos de aprendizagem foram derivados das principais diretrizes curriculares para o contexto específico de aprendizagem, conforme apresentado na Tabela 2.

Tabela 2: Objetivos de aprendizagem.

ID	Objetivo de aprendizagem	Área de conhecimento	Fonte
OA1	Entender os algoritmos como um conjunto de instruções passo a passo para realizar tarefas.	Algoritmo e Programação	(CSTA, 2017: 1A-AP-08)
OA2	Criar sistemas de software interativos que incluem eventos, condicionais, variáveis, listas e sequências de caracteres usando uma linguagem de programação visual baseada em blocos.	Algoritmo e Programação	(CSTA, 2017: 2-AP-18)
OA3	Criar o design visual (cor, tipografia, imagem e composição) do sistema de software interativo.	Design visual, Algoritmos e Programação	(ACM/IEEE, 2013; CSTA, 2017: 2-IC-21; GARRETT, 2011)
OA4	Testar e refinar um sistema de software interativo com relação à funcionalidade e usabilidade.	Algoritmos, Programação e Engenharia de software	(CSTA, 2017: 2-AP-17, 1B-AP-15, 3A-AP-21)
OA5	Compartilhar o sistema de software interativo que foi desenvolvido	Comunicação	(CSTA, 2017: 1B-AP-12, 1B-AP-17, 2-AP-16; Lee et al., 2017)

Todos os objetivos de aprendizagem do curso foram cuidadosamente organizados buscando o aproveitamento dos benefícios diretos e indiretos da combinação de distintas áreas do conhecimento. No entanto, cabe destacar o foco do presente artigo principalmente no objetivo OA3 voltado a aprendizagem de conceitos de design visual.

#### 4.3 Definição da estratégia instrucional

Com base na análise de contexto, a unidade instrucional foi projetada como um curso de 12 encontros (cada uma de 3 h/a), conforme plano de ensino (Tabela 3). Considerando as necessidades típicas de ajustes na prática em termos de duração e nível de proficiência, definimos conteúdos principais e adicionais que podem ser aplicados em unidades mais longas e/ou com alunos mais avançados. A fim de proporcionar uma experiência da aprendizagem envolvente e motivadora, adotamos uma abordagem de ação computacional, um novo

enquadramento para a educação de computação socialmente relevante, que propõe que, os jovens ao aprenderem sobre computação, também tenham oportunidades de criar soluções de TI que tenham impacto direto em suas vidas e comunidades (Tissenbaum et al., 2019).

Tabela 3: Plano de ensino.

Aula (3 h/a)	Conteúdo	Objetivos de aprendizagem	Estratégia instrucional (conteúdo principal)	Estratégia instrucional (conteúdo adicional)
<i>Medição inicial do curso</i>				
<b>Introdução</b>				
1	Motivando a computação e o desenvolvimento de aplicativos	OA1	Aula expositiva, discussão	
	Conceitos básicos de computação: algoritmos e programação	OA1	Aula expositiva, aprendizagem baseada em jogos	Exercícios
2-3	Desenvolvimento de um primeiro app com <i>App Inventor</i> (app encontre-me)	OA2	Instrução interativa/prática, programação em pares	Evoluindo o app pré-definido
<b>Desenvolvimento de um novo app</b>				
4	Visão geral sobre o processo de desenvolvimento de aplicativos	OA1, OA2	Aula expositiva, brainstorming, revisão por pares, atividade prática	
5	Análise de contexto (usuários, tarefas, dispositivos e ambientes) e análise de requisitos incluindo as etapas de empatia e ideação do <i>design thinking</i>	OA1, OA2	Aula expositiva, atividade prática	Entrevistas de usuários
6	Design de baixa fidelidade com esboços (design de interface e mapa de navegação) abordando parte da prototipação do <i>Design Thinking</i>	OA2	Aula expositiva, atividade prática, revisão por pares	
7-8	Programação de <i>wireframes</i> com o App Inventor abordando parte da prototipação do <i>Design Thinking</i>	OA2, OA3	Atividade prática, programação em pares	
<i>Medição inicial de design visual</i>				
9	<i>Design</i> visual e programação de elementos de design visual abordando parte da prototipação do <i>Design Thinking</i>	OA3, OA4	Aula expositiva, exercícios, atividade prática, programação em pares	Revisão por pares
<i>Medição final de design visual</i>				
10	Teste de sistema e de usabilidade abordando o teste do <i>Design Thinking</i>	OA4	Aula expositiva, atividade prática	
11	Compartilhamento e apresentação do aplicativo (via .apk, <i>App Inventor Gallery</i> )	OA5	Aula expositiva, apresentação de pitch, atividade prática	
<b>Encerramento</b>				
12	Debriefing		Discussão	
<i>Medição final do curso</i>				

De forma geral, a unidade instrucional é dividida em 3 partes:

Parte 1. Introdução, motivando a computação e apresentando conceitos básicos de algoritmos e programação. Além disso, os alunos nessa parte também aprendem a programar um aplicativo pré-definido com o App Inventor, abordando conceitos básicos de programação.

Parte 2. Desenvolvimento de uma nova aplicação móvel. Durante essa parte, os alunos desenvolvem seu próprio aplicativo para resolver um problema identificado em sua comunidade. A fim de apoiar também a aprendizagem das habilidades do século XXI, como criatividade, resolução de problemas e pensamento crítico, adotamos a estratégia de construcionismo, na qual a aprendizagem do aluno se concentra na construção do conhecimento baseada na realização de uma ação concreta que resulta em um produto de software, que seja de interesse de quem o produz (Chella, 2016; Vasconcelos, et. al., 2005). O desenvolvimento da solução é guiado por um processo de software ágil que incorpora o *design thinking* que ajuda a planejar, administrar tarefas e criar inovações (Cooper 2014; Nielsen Norman Group, 2018; IDEO, 2015) (Figura 2).

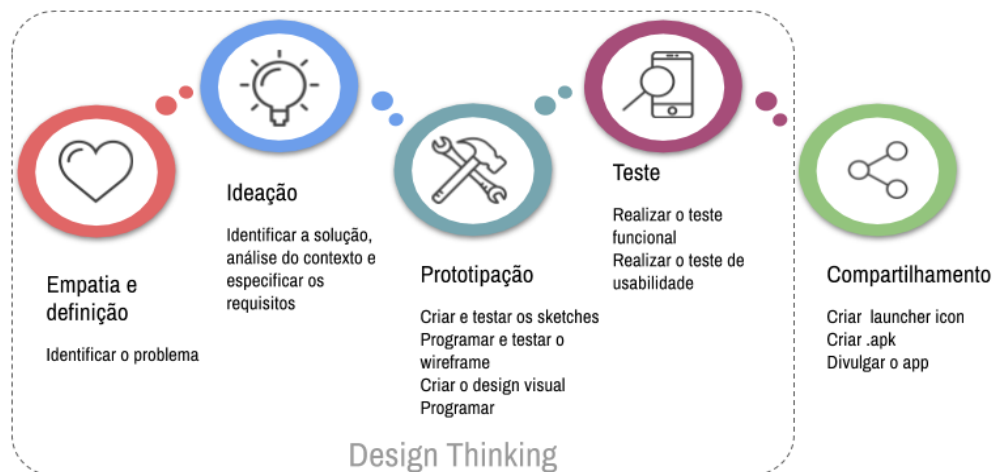


Figura 2: Processo de software.  
Fonte: Elaborada pelos autores.

Seguindo o processo de *design thinking* os alunos são orientados a identificar uma necessidade e empatizar com o público-alvo. A partir da necessidade identificada apresentam ideias de soluções e especificam os requisitos por meio de *user stories*. Durante a etapa da prototipação, os alunos começam desenhar *sketches* das telas do app e o mapa de navegação. A partir do design de baixa fidelidade, eles programam e testam o app no nível de *wireframe* no App Inventor. Em seguida os alunos precisam definir e programar o design visual do app. Completada a programação do app, eles realizam testes funcionais e de usabilidade para validar o atendimento dos requisitos definidos. No final desta parte os alunos aprendem também como compartilhar e divulgar o app por meio de apresentações a diversos públicos.




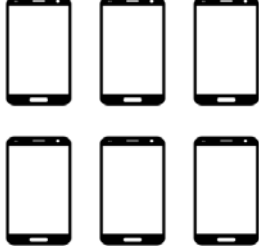
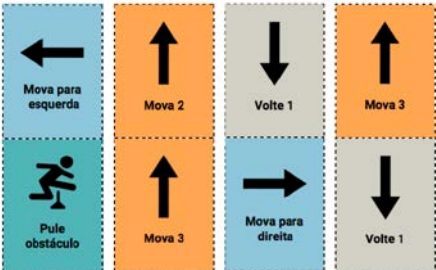





Parte 3. Encerramento. No final do curso é realizado o *debriefing* da unidade instrucional com os alunos.

Adotamos várias estratégias instrucionais predominantemente focadas na aprendizagem ativa com atividades práticas de programação para capacitar os alunos a praticar e explorar efetivamente conceitos de computação como parte do processo de aprendizagem (Wing, 2006; Grover & Pea, 2013; Lye & Koh, 2014). O ambiente de programação adotado foi o App Inventor que utiliza linguagem baseada em blocos (MIT; App Inventor, 2019).

De acordo com as estratégias instrucionais foram desenvolvidos vários materiais instrucionais em Português (Tabela 4).



Tabela 4: Exemplos de material instrucional.

<p>Slides</p>																										
<p>Modelos de artefatos a serem criados pelos alunos</p>		<p>Sketches</p> 																								
<p>Jogo educacional SplashCode (Gresse von Wangenheim et al., 2019).</p>																										
<p>Círculo cromático de papel para pintar</p>		 <table border="1" data-bbox="1155 1173 1347 1397"> <tr><td>R</td><td>Vermelho</td></tr> <tr><td>R-O</td><td>Vermelho e Laranja</td></tr> <tr><td>O</td><td>Laranja</td></tr> <tr><td>Y-O</td><td>Laranja e amarelo</td></tr> <tr><td>Y</td><td>Amarelo</td></tr> <tr><td>Y-G</td><td>Amarelo e verde</td></tr> <tr><td>G</td><td>Verde</td></tr> <tr><td>B-G</td><td>Azul e Verde</td></tr> <tr><td>B</td><td>Azul</td></tr> <tr><td>B-V</td><td>Azul e roxo</td></tr> <tr><td>V</td><td>Roxo</td></tr> <tr><td>R-V</td><td>Vermelho com roxo</td></tr> </table>	R	Vermelho	R-O	Vermelho e Laranja	O	Laranja	Y-O	Laranja e amarelo	Y	Amarelo	Y-G	Amarelo e verde	G	Verde	B-G	Azul e Verde	B	Azul	B-V	Azul e roxo	V	Roxo	R-V	Vermelho com roxo
R	Vermelho																									
R-O	Vermelho e Laranja																									
O	Laranja																									
Y-O	Laranja e amarelo																									
Y	Amarelo																									
Y-G	Amarelo e verde																									
G	Verde																									
B-G	Azul e Verde																									
B	Azul																									
B-V	Azul e roxo																									
V	Roxo																									
R-V	Vermelho com roxo																									
<p>Mini tutoriais de funcionalidades/ componentes específicos</p>																										

Para uma avaliação abrangente da aprendizagem dos alunos, adotamos múltiplos meios de acordo com os objetivos de aprendizagem:

**Avaliação de conceitos de design visual de UI baseada no desempenho e por meio de teste.** Para realizar a avaliação baseada no desempenho foi definida uma rubrica com critérios específicos de design visual (Tabela 5). Tais critérios foram definidos essencialmente com propósito de analisar tanto a aplicação dos elementos da UI de forma isolada (como cor, ícones e tipografia), quanto a composição da UI como um todo.

Tabela 5: Critérios de avaliação.

1	A paleta de cores está coerente com o tema escolhido	7	O tamanho das fontes auxiliam na hierarquia de informações
2	A organização das cores facilitam a interação do usuário	8	As imagens estão de acordo com o tema
3	As cores auxiliam na hierarquia de informações	9	As imagens estão nítidas (não estão pixeladas e distorcidas)
4	O contraste entre a cor do texto e do fundo da tela asseguram a legibilidade	10	Os ícones são fáceis de interpretar
5	Os textos apresentam tamanho de fonte agradáveis para leitura	11	As telas respeitam o mesmo padrão em relação aos elementos de cor, imagem e tipografia (tamanho, família, estilo, peso)
6	O alinhamento do texto contribui para leitura e harmonia do design visual	12	De modo geral o design visual está agradável e organizado

Como parte da rubrica foram definidos 5 níveis de desempenho para cada critério: totalmente atingido, atingido, parcialmente atingido, pouco atingido e não atingido (de 0 a 4 pontos).

Para complementar essa avaliação foi definido um teste, por meio de perguntas qualitativas (abertas) e de questões objetivas (fechadas), para aferir o vocabulário e o nível de conhecimentos dos alunos referente ao design de UI (Figura 3).

<p>1. A seguir apresenta-se a imagem da interface de um aplicativo que fornece informações sobre a qualidade da água das praias. Com base nessa interface responda:</p> <p>a. Descreva as características visuais dessa interface</p> <p>b. Você gostou dessa interface? Por quê?</p>	<p>4. Antes de programar o app, devemos fazer uma análise do usuário para:</p> <p>(a) identificar um problema na comunidade</p> <p>(b) entender as características de pessoas que vão utilizar o app</p> <p>(c) verificar se o app está correto</p> <p>(d) definir o design das telas</p>								
<p>2. A descrição “como Papai Noel eu preciso receber as cartas das crianças para que eu possa levar o presente de natal” é?</p> <p>(a) um programa de app</p> <p>(b) uma história de usuário</p> <p>(c) uma caracterização das pessoas que vão utilizar o meu app</p> <p>(d) o design visual do app</p>	<p>5. Descreva a história de alguma tarefa do seu app seguindo o modelo abaixo.</p> <table border="1" style="width: 100%;"> <tr> <td colspan="2"><b>História:</b> &lt;Título da história/nome da tarefa&gt;</td> </tr> <tr> <td><b>Como um(a)</b></td> <td>&lt;Papel/tipo de usuário&gt;</td> </tr> <tr> <td><b>Eu preciso</b></td> <td>&lt;tarefa que o usuário irá realizar&gt;</td> </tr> <tr> <td><b>Para que eu possa</b></td> <td>&lt;objetivo a ser atingido&gt;</td> </tr> </table>	<b>História:</b> <Título da história/nome da tarefa>		<b>Como um(a)</b>	<Papel/tipo de usuário>	<b>Eu preciso</b>	<tarefa que o usuário irá realizar>	<b>Para que eu possa</b>	<objetivo a ser atingido>
<b>História:</b> <Título da história/nome da tarefa>									
<b>Como um(a)</b>	<Papel/tipo de usuário>								
<b>Eu preciso</b>	<tarefa que o usuário irá realizar>								
<b>Para que eu possa</b>	<objetivo a ser atingido>								
<p>3. Quais elementos compõem o design visual de um app?</p> <p>(a) wifi, som, jogo</p> <p>(b) cores, tipografia, ícones e imagens</p> <p>(c) sensor de localização, mapa</p> <p>(d) som, áudio, vídeo</p>									

Figura 3: Teste.

As questões do teste foram simples, elaboradas levando em consideração o nível educacional dos estudantes e a intenção de proporcionar uma experiência de aprendizagem envolvente e lúdica.

**Avaliação de conceitos de algoritmos e programação baseada no desempenho.** A avaliação é complementada por meio da análise automatizada de conceitos de algoritmos e programação, baseada na rubrica definida por Alves (2019) que abarca conceitos básicos como operadores, variáveis, *strings*, etc. Essa avaliação é realizada pela ferramenta CodeMaster que apresenta os resultados a partir do código programado pelo aluno no App Inventor (Gresse von Wangenheim et. al, 2018; Alves, 2019).

## 5 Aplicação em escola pública

O curso “Faça seu próprio app” foi aplicado e avaliado no contexto do projeto Jovens Tutores com 10 alunos da Escola Básica Municipal Almirante Carvalhal durante o segundo semestre de 2018 (Figura 4). Participaram da aplicação 5 meninas e 5 meninos dos anos finais do Ensino

Fundamental (13 a 15 anos). As aulas foram ministradas por professores e alunos de graduação e pós-graduação do Departamento de Informática e Estatística – INE e Departamento de Expressão Gráfica – EGR da UFSC, suportado por profissionais de TI da empresa patrocinadora.

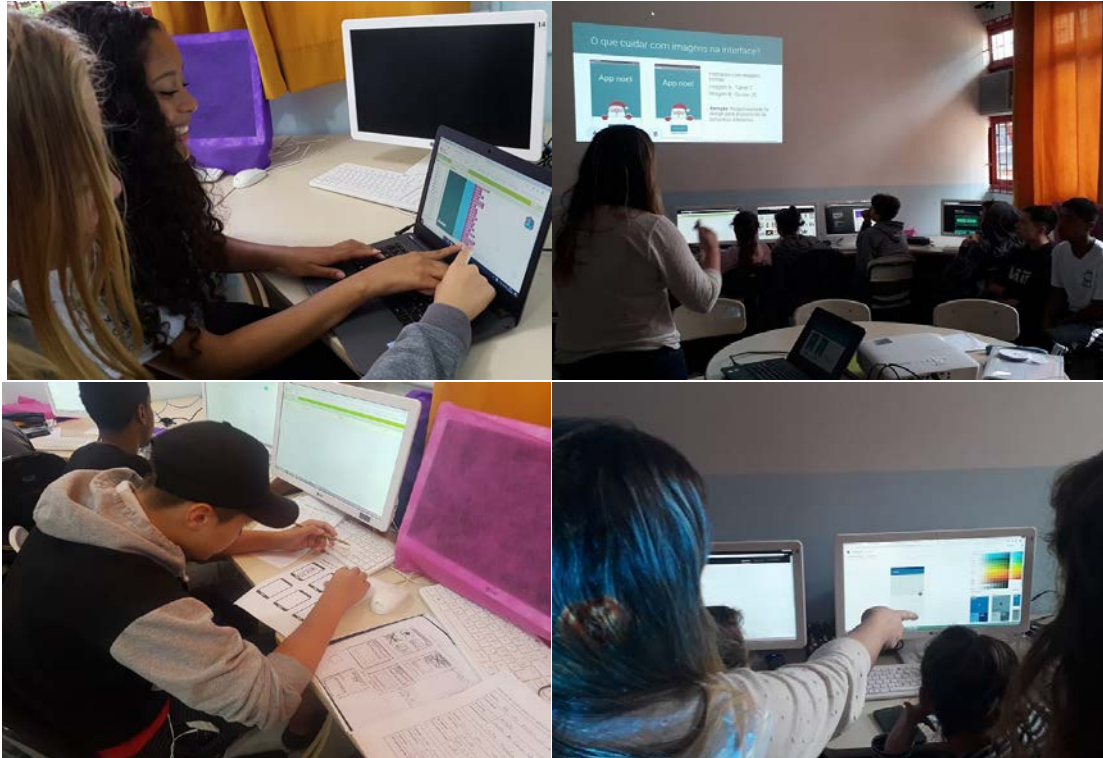


Figura 4: Alunos durante o curso.  
Fonte: Elaborada pelos autores.

Como uma parte introdutória, os estudantes aprenderam conceitos básicos de computação, começando com o conhecimento de algoritmos. Foi utilizado como material didático um conjunto de slides e o jogo SplashCode (Gresse von Wangenheim et al., 2019), praticado pelos alunos. Neste encontro os alunos também tiveram a oportunidade de conhecer o App Inventor e construir o app “Encontre-me” a partir de um tutorial e do suporte dos instrutores.

Na segunda parte do curso, adotamos uma abordagem de *design thinking* na qual os alunos debateram sobre problemas da comunidade e identificaram possíveis soluções de aplicativos. Eles analisaram o contexto em termos de usuários, tarefas, dispositivos móveis e o ambiente de uso. Os requisitos funcionais foram descritos em forma de histórias de usuários e os requisitos de usabilidade foram especificados com relação à eficácia, eficiência e satisfação para cada uma das tarefas identificadas usando modelos predefinidos. Os alunos desenharam esboços e um mapa de navegação das interfaces da tela. Então, eles começaram a programar e testar um protótipo de *wireframe* com o App Inventor. Além disso, os alunos definiram e programaram os elementos visuais das interfaces, incluindo paleta de cores, ícones, logotipo, tipografia e imagens (Figura 5).

### Identificação do problema

Qual é o problema identificado?	Falta de organização em escola!
Quem é afetado pelo problema e como é afetado?	Alunos de 6ºs. Não sabem as frequências da escola
Um aplicativo pode resolver este problema?	Sim

### User story

História: *Quero saber o dia da semana da aula*

Como um(a) *Novo aluno*

eu preciso *Saber todos alunos*

para que eu possa *Fazer minha matrícula*

### Identificação da solução

Nome do seu app: *ALMIRANTE CARVALHAL - ESCOLA*

Quem vai fazer o que com o app? *Sei todos os tipos de escolas.*

Por que seu app é impressionante? *Por que os professores os alunos.*

### Persona

### Requisitos de usabilidade

Tarefa: *Saber os professores de cada matéria*

- Todos os usuários conseguem completar a tarefa.
- Todos os usuários conseguem completar a tarefa em no máximo 5 segundos/minutos
- Todos os usuários respondem a pergunta "essa tarefa foi difícil, difícil, nem fácil-nem difícil, fácil, muito fácil" no mínimo com Facil (muito difícil, difícil, nem fácil-nem difícil, fácil...).

### Sketches

### Resultados teste de interface

O que funcionou...	Judo
Dúvidas...	Penultima
O que pode ser melhorado...	Desing <small>do conteúdo do lista</small>
Idéias...	Penultima

### Resultados teste de usabilidade

Tarefa: *Ver a lista de alunos da sua turma*

- Todos os usuários conseguem completar a tarefa.
- Todos os usuários conseguem completar a tarefa em no máximo 15 seg. minutos/segundos.
- Todos os usuários acham que foi Muito fácil realizar essa tarefa.

### Wireframe

### Resultados teste funcional

Teste cada funcionalidade do seu app para cada ação e entrada.

Funcionalidade:	Ação	Entrada	Resultado esperado	Resultado retornado
Quero ver os 6ºs. e Professores etc	Selecionar a turma e escolher a lista de alunos	Volto embora	Aparecer a lista de alunos	OK
	Selecionar a turma e escolher a lista de horários		Aparecer a lista de horários	OK
	Selecionar a turma e escolher a lista de professores		Aparecer a lista de professores	OK
	Clicar em voltar		Voltar uma tela	OK

### Design final

Figura 5: Exemplos de artefatos criados pelos alunos.  
Fonte: Elaborada pelos autores.



Em seguida, os alunos realizaram um teste simples de sistema e de usabilidade para validar o alcance dos requisitos. Eles também aprenderam como gerar uma versão .apk do aplicativo para publicar no Google Play e como divulgar os seus apps por meio de cartazes e página web. Os resultados também foram apresentados a públicos da escola e de uma empresa local de TI (Figura 6).



Figura 6: Apps desenvolvidos pelos alunos.  
Fonte: Elaborada pelos autores.

O tema para o desenvolvimento dos aplicativos foi livre, cada dupla focou em uma ideia diferente, variaram desde necessidades de sociais (procurar empregos e telefones de emergência) até finalidades informacionais (dicas de plantio, praias e informações da escola).

## 6 Avaliação da unidade instrucional

### 6.1 Definição da avaliação

O objetivo da avaliação é analisar se a unidade instrucional “Faça o seu próprio app” permite atingir os objetivos de aprendizagem e se, como uma estratégia instrucional, proporciona uma experiência de aprendizagem motivadora e envolvente. Para atingir esse objetivo, avaliamos a unidade instrucional por meio de um estudo de caso, que permite a pesquisa em profundidade de um indivíduo, grupo ou evento (Wohlin et al., 2012; Yin, 2017).

O objetivo da avaliação é decomposto em três fatores de qualidade e suas dimensões (Figura 7), a saber: **experiência de aprendizagem** - definida como um fator de qualidade que abrange um envolvimento profundo do aluno na atividade de aprendizagem, incluindo seus sentimentos, prazeres e interações (Savi et al., 2011; Keller, 1987); **aprendizagem** - medida por meio da aprendizagem baseada no desempenho e testes e da aprendizagem percebida pelos alunos (Savi et al., 2011; Sindre & Moody, 2003); **Usabilidade** - grau em que a unidade instrucional pode ser usada pelos alunos para atingir seus objetivos especificados com eficácia e eficiência em termos da aprendizagem (Savi et al., 2011; ISO/IEC, 2019).

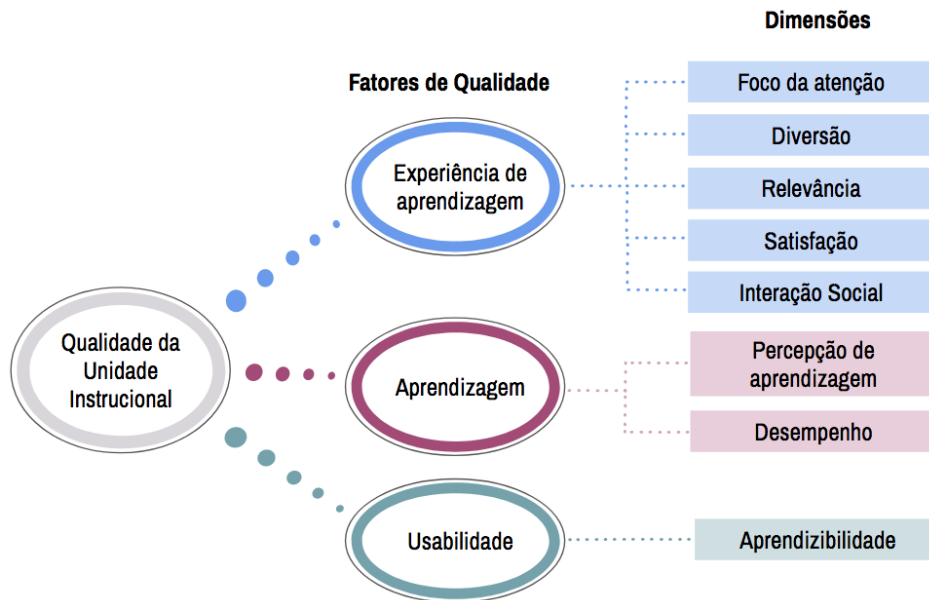


Figura 7: Decomposição dos fatores e dimensões de qualidade.  
 Fonte: Elaborada pelos autores.

A coleta de dados é operacionalizada por meio de avaliação baseada no desempenho, teste e questionários respondidos pelos alunos a fim de coletar informações sobre a unidade instrucional.

## 6.2 Análise dos dados

### 6.2.1 Avaliação da experiência da aprendizagem

Em geral a experiência de aprendizagem foi positiva um vez que a maioria dos alunos achou divertido aprender os conteúdos abordados (Figura 8). Dentre os conteúdos questionados o “fazer o design visual” se destacou positivamente pois foi indicado pelo maior número de alunos como muito divertido. Por outro lado, “programar e testar” foi o único tópico que foi considerado muito chato por um aluno.

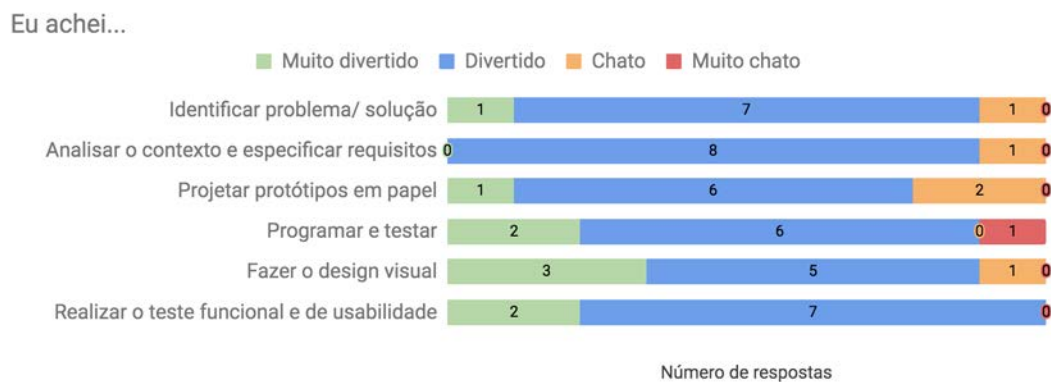


Figura 8: Distribuição da frequência de respostas relacionadas a diversão.  
 Fonte: Elaborada pelos autores.

Ainda, a grande maioria dos alunos considera os conteúdos ensinados fáceis de aprender (Figura 9). Poucos alunos reconheceram uma dificuldade em identificar um problema/solução (3), programar e testar (4) e fazer o design visual (2).

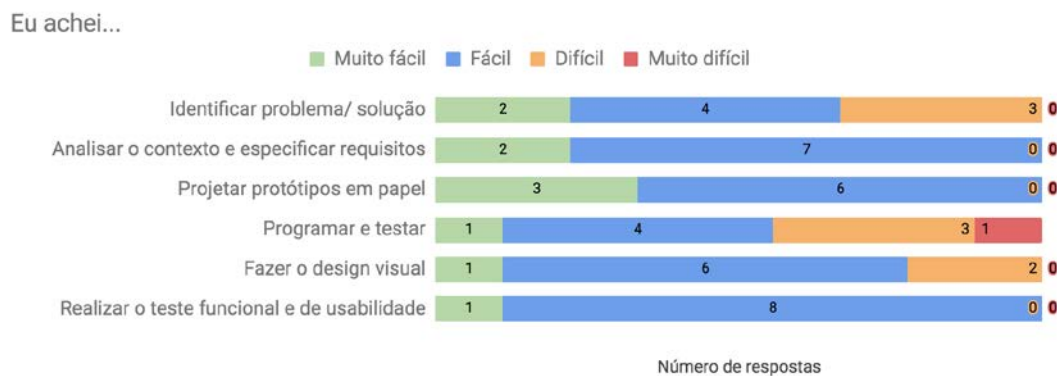


Figura 9: Distribuição da frequência de respostas relacionadas a sua facilidade de aprendizagem.

Fonte: Elaborada pelos autores.

Apenas um aluno achou que as aulas passaram devagar, os demais afirmaram que não sentiram o tempo passar durante as aulas. Esse resultado pode ser um indicativo que os alunos ficaram concentrados no conteúdo e não notaram a passagem do tempo. Vale destacar que todos os alunos demonstraram satisfação no processo de desenvolvimento dos apps e interesse em compartilhar os aplicativos criados.

Quase todos os alunos (8) avaliaram os encontros como excelentes ou bons e somente um qualificou o encontro do design visual como regular. O aluno que avaliou negativamente este conteúdo não possuía interesse nessa área.

Foi questionado também o reconhecimento da importância dos conteúdos ensinados (Figura 10). A partir dos dados foi possível perceber que todos os alunos reconhecem a relevância da programação, testes e do design visual para o processo de desenvolvimento de apps. Eles também compreenderam a importância do processo de desenvolvimento e da prototipação. Apenas a análise do contexto e especificação dos requisitos foi considerado não tão importante por quatro alunos.

Eu considero esse assunto importante...

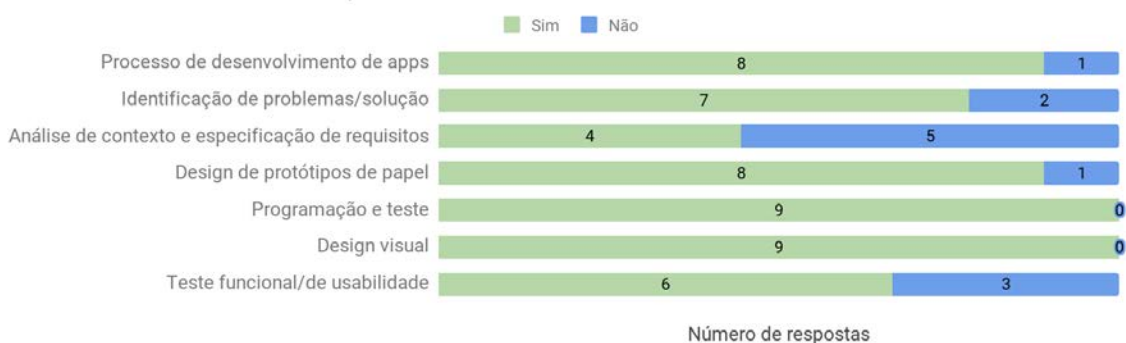


Figura 10: Distribuição da frequência de respostas relacionadas a importância de assuntos.

Fonte: Elaborada pelos autores.

Ao final da unidade instrucional os alunos demonstram um interesse parcial em aprender mais sobre os conteúdos ensinados (Figura 11). Sob esse viés, foi possível perceber que os tópicos que mais se destacaram foram: o processo de desenvolvimento de apps e o design visual. Esse resultado indica que o design de interface é um tema que pode contribuir para despertar o interesse no ensino de computação. Por outro lado, os outros conteúdos indicam uma divisão entre o interesse nas respostas.

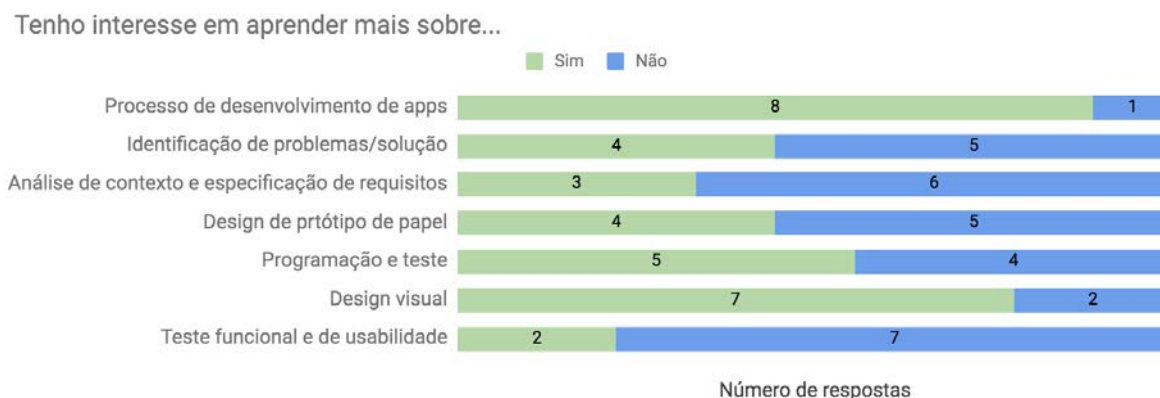


Figura 11: Distribuição da frequência de respostas relacionadas ao interesse de aprender.

Fonte: Elaborada pelos autores.

### 6.2.2 Avaliação da aprendizagem

Como resultado da unidade instrucional os alunos em pares criaram apps funcionais com App Inventor (Figura 6). Considerando a ênfase do presente trabalho no design de UI, enfocamos nessa questão na avaliação da aprendizagem relacionada ao objetivo de aprendizagem OA3 por meio de uma avaliação baseado no desempenho dos alunos. Esta avaliação foi realizada utilizando uma rubrica que analisou a interface dos aplicativos desenvolvidos pelos alunos em relação a critérios relacionados ao design visual. Os resultados dessa avaliação são apresentados na Tabela 6.

Tabela 6: Avaliação baseado no desempenho do design visual dos aplicativos criados.

Crítérios	Achei o seu Emprego	Healthy Plants	App Phone	Floripa Praias	Info Carvalhal
A escolha de cores (paleta de cores) está coerente com o tema escolhido	4	4	3	3	4
A organização das cores facilitam a interação do usuário	3	4	3	4	4
A organização das cores facilitam a interação do usuário	4	4	3	3	4
O contraste entre cor e texto e fundo da tela facilita a leitura das informações (asseguram a legibilidade)	3	4	3	3	4
Os textos apresentam tamanho das letras (fonte) agradáveis para leitura	4	4	4	4	2
O alinhamento do texto contribui para leitura e harmonia do design visual	2	3	3	4	4
O tamanho das letras (fontes) auxiliam no destaque de informações (hierarquia)	4	3	4	4	3
As imagens estão de acordo com o tema	4	4	4	4	4
As imagens estão nítidas (não estão pixeladas e/ou distorcidas)	4	4	4	4	4
Os ícones são fáceis de interpretar	4	2	4	4	2
As telas respeitam o mesmo padrão em relação aos elementos de cor, imagem e tipografia (tamanho, família, estilo, peso)	2	4	1	2	3
De modo geral o design visual está agradável e organizado	3	4	3	4	4
<b>Pontuação total (de máx. de 48 pontos)</b>	<b>41</b>	<b>44</b>	<b>39</b>	<b>43</b>	<b>42</b>
<b>Média</b>	<b>8,54</b>	<b>9,17</b>	<b>8,13</b>	<b>8,96</b>	<b>8,75</b>

A partir da análise das interfaces de usuário aferimos que, de forma geral, os cinco aplicativos desenvolvidos pelos alunos apresentam cores coerentes com seus temas e contraste



(entre a cor do texto e fundo da tela) que contribuem para a interação do usuário. Somente dois aplicativos, Floripa Praias e App Phone, não mantiveram a uniformidade de cores entre as telas. Quanto às tipografias utilizadas, todos os aplicativos possuem fontes sem serifa e a maioria das telas apresenta tamanho que assegura a legibilidade e hierarquia das informações. Somente um aplicativo, o InfoCarvalho, em uma única tela, utilizou fontes pequenas que poderiam prejudicar a leitura. Acrescenta-se que as imagens (fotografias e/ou ícones) utilizadas em todos os apps são nítidas e coerentes com os temas dos escolhidos. Considerando as limitações da ferramenta App Inventor a respeito do tamanho de botões, *grid* e tipografias, pode-se observar que a composição de todos os aplicativos estava coerente. Dessa forma podemos concluir que os alunos aprenderam conceitos básicos de design visual e inclusive conseguiram transferir esse conhecimento aos seus próprios aplicativos.

Destaca-se também o excelente resultado em relação a estética visual dos apps criados pelos alunos. Com base nos resultados de um *survey* com 95 participantes<sup>3</sup> avaliando os screenshots de UIs de aplicativos criados pelos alunos e de outros 105 apps aleatoriamente escolhidos da Galeria do App Inventor (Solecki et al., 2019), classificando as telas como (“feia” = 0 ponto; “mais ou menos” = 0,5 ponto; “bonita” = 1 ponto). Observa-se que a pontuação de grau de estética somando as respostas dos apps criados pelos alunos está muito acima da média (Tabela 7).

Tabela 7: Resultado da avaliação estéticas.

Aplicativo	Pontuação
Achei o seu emprego	56,25
HealthyPlants	84,5
AppPhone	59,25
FloripaPraias	75,5
InfoCarvalho	60
<b>Média dos apps da Galeria do App Inventor</b>	<b>26,5</b>

Em geral, os apps desenvolvidos pelos alunos se destacam por um grau de estética maior, principalmente em relação a um sistema de cores bem-definido, imagens adequadas e composição equilibrada.

Completando a avaliação da aprendizagem de conceitos de design de UI por meio de um teste ao final da unidade instrucional foi também confirmado um grau satisfatório de conhecimento (Figura 12).

#### Questões do teste



Figura 12: Questões do teste.  
Fonte: Elaborada pelos autores.

<sup>3</sup> Este estudo foi aprovado pelo Comitê de Ética da Universidade Federal de Santa Catarina (no. 2.903.849).

As questões sobre elementos do design visual e teste funcional foram corretamente respondidas pelos estudantes. No entanto, o índice de acerto foi menor nas questões sobre a interpretação de uma história de usuário e análise de usuário, o que nos permite inferir que estes conteúdos não foram muito bem compreendidos pelos estudantes ou houve má interpretação da pergunta. Por fim, a última questão, que solicitava o preenchimento de uma história de usuário em relação aos apps criados, sete alunos preencheram corretamente o que indica que estes alunos aprenderam a aplicar os conceitos relacionados a análise de contexto do app.

Quanto aos dados dos questionários, diante das respostas dos alunos percebe-se que eles reconhecem que aprenderam o conteúdo da unidade instrucional (Figura 13). Baseado nos resultados de uma autoavaliação os conceitos em que mais os alunos percebem ter alto nível de conhecimento no final da unidade instrucional foram: design do protótipo em papel, programar e testar, design visual, teste funcional e usabilidade. Por outro lado, programação e análise do contexto e especificação dos requisitos são os conteúdos em que alunos mais se sentem iniciantes. Houve apenas uma resposta indicando nenhum conhecimento aprendido em relação ao design visual. Acredita-se que as respostas entre “Programação” e “Programar e testar” possuem diferença em razão do tema programação é algo muito mais amplo e complexo, já programar e testar foi um dos temas das aulas aplicadas em que os alunos tiveram bons resultados.

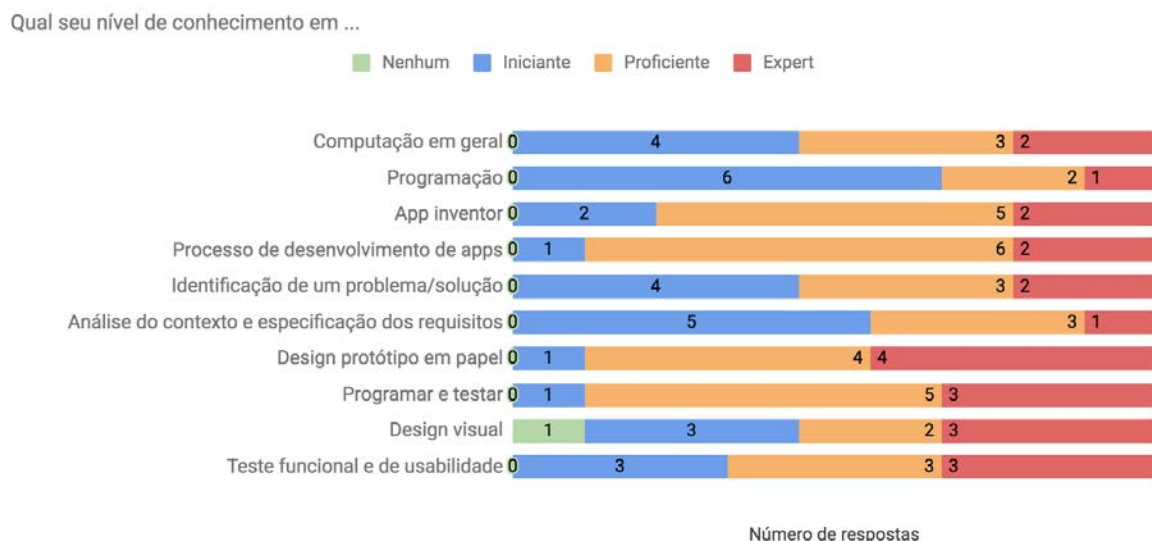


Figura 13: Distribuição da frequência de respostas relacionadas a percepção da aprendizagem.

Fonte: Elaborada pelos autores.

Os alunos também se consideram capazes de explicar a maioria dos conceitos para outros, indicando inclusive níveis de aprendizagem mais altos (Figura 14). Especialmente em “Fazer protótipos em papel” e “Programar e testar”, que obtiveram apenas uma respostas negativa, indicando serem os mais compreendidos. Já na opção “Realizar análise do contexto e especificar requisitos” a turma ficou dividida e cinco alunos responderam não ter segurança em explicar para outra pessoa.

Eu consigo explicar a um(a) amigo(a)...

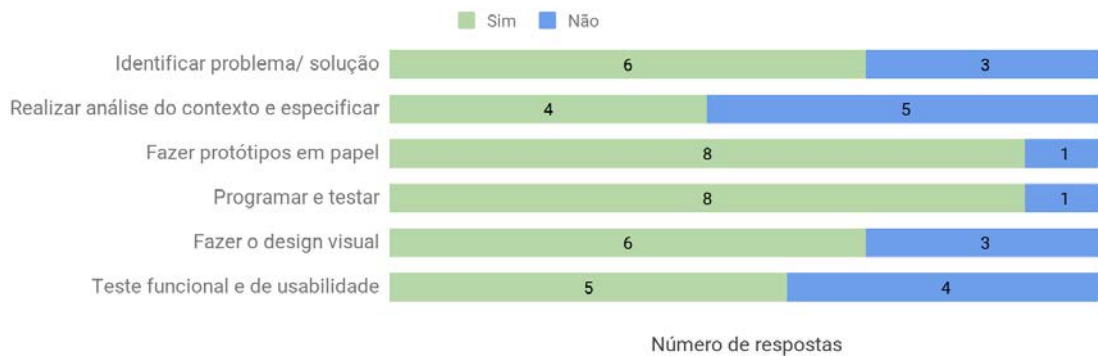


Figura 14: Distribuição da frequência de respostas relacionadas a aprendizagem.  
 Fonte: Elaborada pelos autores.

### 6.2.3 Avaliação da usabilidade da unidade instrucional

Os alunos avaliam a qualidade das aulas e materiais utilizados de maneira muito positiva (Figura 15). A maioria acha excelente ou bom o material instrucional e também a forma como as aulas foram realizadas. Todos consideraram a unidade instrucional excelente.

O que você achou?



Figura 15: Distribuição da frequência de respostas relacionadas às aulas e o curso em geral.  
 Fonte: Elaborada pelos autores.

### 6.2.4 Comentários gerais

Os comentários gerais dos alunos (Tabela 8) indicaram uma grande apreciação em todas as etapas envolvidas na criação de um app.

Tabela 8: Comentários qualitativos (números indicando a frequência de citações).

O que <b>mais</b> gostei no curso de fazer um app foi?	Design Visual, paleta de cores e etc. Programar Aprender a fazer um app Design (2) Tudo (3) Design visual
O que <b>menos</b> gostei no curso de fazer um app foi?	Escolher as plantas e pesquisar descrição delas Nada (5) Protótipo de papel Fazer questionário Pesquisar plantas, e descrições

A parte do design do app destacou-se por ser mencionado por vários alunos como o que mais gostaram no curso e também programar e criar um app. Já o que os alunos menos gostaram foi buscar informações para apresentar no app, construir o protótipo em papel (e responder os questionários de avaliação). Vários alunos indicaram que não houve nada de que não gostassem no curso.

## 7 Discussão

Os resultados indicam uma avaliação muito positiva da unidade instrucional. Durante o curso, os alunos participaram ativamente e puderam seguir as instruções. Os alunos expressaram seu envolvimento e satisfação, especialmente durante as atividades práticas, desenvolvendo seu próprio aplicativo. A instrução passo-a-passo com atividades práticas imediatas ajudou a manter os alunos imersos no processo de aprendizagem. Também observamos que, especialmente, as atividades de design de UI têm sido uma maneira mais concreta de ensinar um processo sistemático para desenvolver aplicativos, em vez de se concentrar em atividades mais abstratas de engenharia de software, como a definição de casos de uso.

A utilização do ambiente de programação baseado em blocos App Inventor também mostrou facilitar a programação dos aplicativos e permitir a criação de aplicativos Android. No entanto, embora o App Inventor suporte a implementação do design da UI, observamos várias deficiências, incluindo falta de suporte em conformidade com os guias de estilo do Android, como o *Material Design* (2019) da Google, por exemplo, em relação às cores, fontes e formatos de botões assim como a falta de acesso direto a ícones ou imagens. Além disso, como o App Inventor se concentra no suporte para programação e teste, não fornece suporte adicional para outras etapas do processo (como a definição de *user stories* ou a seleção de uma paleta de cores). Observando essas deficiências, estamos trabalhando atualmente no desenvolvimento do App Inventor para fornecer um suporte mais abrangente ao processo de desenvolvimento completo, incluindo o design de UI.

Os resultados criados pelos alunos foram muito além da nossa expectativa, especialmente ao comparar a qualidade do design da interface com outros aplicativos criados com o App Inventor. Os alunos puderam transferir conhecimentos sobre os meta-princípios de consistência, hierarquia e personalidade dos elementos de interface, proporcionando a associação entre harmonia e utilidade.

Os alunos gostaram muito do curso, expressando isso em seus comentários. Os alunos também acharam o curso fácil de seguir e gostaram da estratégia e material instrucional. Os alunos gostaram de trabalhar em pares e se divertiram enquanto aprendiam. No final do curso, eles se sentiram capazes de criar programas de computador por conta própria e demonstraram interesse em aprender mais sobre programação e design de UI. Observamos também, durante a apresentação dos aplicativos, um efeito positivo na autoestima dos alunos, demonstrando-se mais confiantes de que são capazes de criar aplicativos atraentes e utilizáveis. No final do curso, os alunos estavam interessados em continuar em casa e/ou por meio de outros cursos.

A adoção de um processo baseado no *design thinking* também serviu como incentivo para que os alunos se tornassem mais criativos e imaginativos, criando aplicativos funcionais e também ensinando-os a usar essa inventividade e colocá-la em prática. Desta forma, a aprendizagem do design fornece soluções e formas de se envolver com o mundo, permitindo que os indivíduos atuem como agentes de mudança e criação no século XXI. A incorporação de conteúdo relacionado ao design de UI no ensino de computação também ajuda a ampliar a percepção de que a computação é mais do que apenas codificar e, portanto, pode ajudar a atrair pessoas que normalmente não consideram uma carreira em TI.

**Ameaças a validade.** O projeto do estudo de caso aplicado nesta pesquisa pode causar vários problemas em relação à validade dos resultados. Um dos problemas é a falta de um *benchmark* de conhecimento dos alunos antes do curso para fazer uma comparação e analisar as diferenças. Isso é ainda mais exacerbado pelo fato de que não há grupo controle para comparar os efeitos identificados. No entanto, devido às características deste projeto específico, só foi possível realizar um estudo de caso em vez de um experimento. Além disso, o fato de a unidade

instrucional ter sido aplicada apenas com um pequeno conjunto de participantes e dentro da mesma escola reduz a possibilidade de generalização dos resultados. Entretanto, considerando a natureza exploratória de nossa pesquisa neste momento, consideramos aceitável o rigor científico de um estudo de caso cuidadosamente definido. Outra possível ameaça é que aspectos como diversão e satisfação são difíceis de medir e capturados por medidas subjetivas. Para contornar essa ameaça à validade, os itens dos questionários foram sistematicamente derivados com base no instrumento de medição padronizado MEEGA+ que foi avaliado em termos de validade e confiabilidade em larga escala (Petri et al., 2017).

## 8 Conclusão

Este artigo apresenta o desenvolvimento, aplicação e avaliação de uma unidade instrucional voltada à integração do ensino de competências de design de UI no ensino de computação nos anos finais no Ensino Fundamental. Adotando uma estratégia de aprendizagem construtivista, seguindo o *framework* de ação computacional, os alunos são desafiados a desenvolver seu próprio aplicativo móvel para resolver um problema em sua comunidade. Incorporando o ensino de conceitos de design de UI no desenvolvimento de software, seguindo um processo ágil baseado em *design thinking*, os alunos aprendem passo a passo como criar um aplicativo móvel com o App Inventor. Resultados de um estudo de caso exploratório em uma escola pública de Florianópolis, fornecem uma primeira indicação de que a unidade instrucional pode contribuir para a aprendizagem de computação e design de UI. Os alunos avaliaram a experiência de aprendizagem e usabilidade de uma forma muito positiva, mostrando que a unidade instrucional pode ser divertida, eficiente e eficaz. Com base nesse sucesso, estamos revisando o material instrucional e evoluindo o App Inventor para facilitar a aprendizagem do design de UI. Também estamos começando a desenvolver cursos de desenvolvimento profissional para professores em serviço, a fim de promover uma aplicação mais ampla.

## Agradecimentos

Queremos agradecer aos alunos por fornecer feedback, bem como aos mentores voluntários pela ajuda.

Este trabalho foi apoiado pelo CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico - [www.cnpq.br](http://www.cnpq.br)), entidade do governo brasileiro voltada ao desenvolvimento científico e tecnológico e a pela empresa Involves Soluções Tecnológicas S/A, patrocinadora do projeto Jovens Tutores.

## Referências

- ACM/IEEE. (2013) Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science. Nova York, NY, EUA, p. 518. Retrieved from [https://www.acm.org/binaries/content/assets/education/cs2013\\_web\\_final.pdf](https://www.acm.org/binaries/content/assets/education/cs2013_web_final.pdf)
- Aiga. (2013). The Professional Association for Design. Retrieved from <https://www.aiga.org>.
- Ahn, S. (2014). A Study on Information Science Curriculum of Productivity Tools to Increase Ability for Problem Solving in Elementary and Middle School. *Journal of The Korean Association of Information Education*, 18(2), 235-242. doi: <https://doi.org/10.14352/jkaie.2014.18.2.235> [GS Search]

- Alves, N. d. C. (2019). CodeMaster: Um Modelo de Avaliação do Pensamento Computacional na Educação Básica através da Análise de Código de Linguagem de Programação Visual. *Dissertação (Programa de Pós-Graduação em Ciência da Computação (PPGCC))* – Universidade Federal de Santa Catarina. [[GS Search](#)]
- Basili, V. R., Caldiera, G. & H. Dieter Rombac (1994). Goal, Question Metric Paradigm. In *J. J. Marciniak, Encyclopedia of Software Engineering*, Wiley-Interscience, New York, NY, USA. [[GS Search](#)]
- Bordini, A. et al. (2016). Computação na educação básica no brasil: o estado da arte. *Revista de Informática Teórica e Aplicada*, 23 (2), 210-238. doi: <https://doi.org/10.22456/2175-2745.64431> [[GS Search](#)]
- Branch, R. M. (2009). *Instructional Design: The ADDIE Approach*. New York: Springer. [[GS Search](#)]
- Chen, P. & Huang, R. (2017). Design thinking in App inventor game design and development: A case study. In: *Proc. of 17th Int. Conf. on Advanced Learning Technologies*. Timisoara, Romania, 139-141. doi: <https://doi.org/10.1109/ICALT.2017.161> [[GS Search](#)]
- Chella, M. T. (2016). Ori: Plataforma para Robótica Educacional de Baixo Custo. In: *VI Mostra Nacional de Robótica*, Sergipe, 778-781. [[GS Search](#)]
- Christensen, K. S., Hjorth, M., Iversen, O. S. & Blikstein, P. (2016). Towards a formal assessment of design literacy: Analyzing K-12 students' stance towards inquiry. *Design Studies*, 46, 125-151. doi: <https://doi.org/10.1016/j.destud.2016.05.002> [[GS Search](#)]
- Code like a Girl. (2017). Retrieved from <https://code.likeagirl.io/learning-design-by-making-games-in-scratch-6b90cd9e8a83>
- Code. (2018). Code.org. Retrieved from <https://curriculum.code.org/csd-1718/unit4/>
- Code.org/App Lab. (2018). Retrieved from <https://code.org/educate/applab>
- Codehs. (2018). Retrieved from <https://codehs.com/info/curriculum>
- Cooper, A. et al. (2014). *About face: the essentials of interaction design*. Indianapolis: John Wiley e Sons. [[GS Search](#)]
- Craig, M. & Horton, D. (2009, March). Gr8 designs for Gr8 girls: a middle-school program and its evaluation. *ACM SIGCSE Bulletin*, 41(1), 221-225). doi: <https://doi.org/10.1145/1539024.1508949> [[GS Search](#)]
- Createlab. (2017). Retrieved from <http://www.mycreatelab.com/ais/>.
- CSTA. (2017). K-12 Computer Science Framework. Retrieved from <http://k12cs.org/wp-content/uploads/2016/09/K%E2%80%9312-Computer-Science-Framework.pdf>.
- Denner, J., Werner, L., Bean, S. & Campe, S. (2005). The girls creating games program: Strategies for engaging middle-school girls in information technology. *Frontiers: A Journal of Women Studies*, 26(1), 90-98. [[GS Search](#)]
- Edutopia. (2015). Edutopia. Retrieved from <https://www.edutopia.org/blog/coding-by-design-first-approach-douglas-kiang>.
- Ferreira, M. N.F., Missfeldt Filho, R., Pinheiro, F. d. C., Gresse von Wangenheim, C. & Hauck, J. C. R. (2018). Ensinando Design de Interface de Usuário na Educação Básica: Um Mapeamento Sistemático do Estado da Arte e Prática. In: *Anais do Workshop de Informática na Escola no Congresso Brasileiro de Informática da Educação*, Brasília/DF, 2019. doi: <http://dx.doi.org/10.5753/cbie.wie.2019.511> [[GS Search](#)]
- Garrett, J. J. (2011). *The Elements of User Experience: User-Centered Design for the Web and Beyond*, (2nd ed.). Berkeley: New Riders. [[GS Search](#)]



- Get started with code 2. (2017). Retrieved from <https://itunes.apple.com/us/book/get-started-with-code-2/id1226776857?mt=11>.
- Gresse von Wangenheim, C., Petri, G.; Zibetti, A. W., Borgatto, A. F., Hauck, J. C. R., Pacheco, F. S. & Missfeldt Filho, R. (2017). dTECT: A Model for the Evaluation of Instructional Units for Teaching Computing in Middle School. *Informatics in Education*, 16(2), 301-318. [GS Search]
- Gresse von Wangenheim, C., Petri, G. & Borgatto, A. F. (2018). MEEGA+KIDS: A Model for the Evaluation of Educational Games for Computing Education in Secondary School. Technical Report INCoD/GQS.06.2018.E, INCoD/INE/UFSC, Florianopolis/Brazil. [GS Search]
- Gresse von Wangenheim, C. Hauck, J. C. R., Demetrio, M. F., Pelle, R. Alves, N. d. C.; Barbosa, H. & Azevedo, L. F. (2018). CodeMaster – Automatic Assessment and Grading of App Inventor and Snap! Programs. *Informatics in Education*, 17(1), 117-150. [GS Search]
- Gresse von Wangenheim, C., Araújo e Silva de Medeiros, G. A, Missfeldt Filho, R., Petri, G. , Da Cruz Pinheiro, F., Ferreira, M. N. F. & Hauck, J. C. R. (2019). SplashCode– A Board Game for Learning an Understanding of Algorithms in Middle School. *Informatics in Education*, 18(2), 259-280. doi: [10.31235/osf.io/2qbnp](https://doi.org/10.31235/osf.io/2qbnp) [GS Search]
- Grover, S. & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher*, 42(1), 38-43. doi: <https://doi.org/10.3102/0013189X12463051> [GS Search]
- Hewett, T. et al. (1992). ACM SIGCHI Curricula for Human-Computer Interaction. ACM. [GS Search]
- Ideo, (2015). *The field guide to human-centered design*. IDEO.org. Retrieved from <https://www.designkit.org/resources/1>
- Iniciativa Computação na Escola. (2019). Retrieved from <http://www.computacaonaescola.ufsc.br>.
- ISO 9241-220. (2019). Ergonomics of human system interaction-part 220: Human-centred design for interactive systems. International Standardization Organization (ISO). Suíça. [GS Search]
- Kafai, Y. B. & Burke, Q. (2013). Computer programming goes back to school. *Phi Delta Kappan*, 95(1), 61-65. doi: <https://doi.org/10.1177/003172171309500111> [GS Search]
- Ke, F. & Im, T. (2014). A case study on collective cognition and operation in team-based computer game design by middle-school children. *International Journal of Technology and Design Education*, 24(2), 187-201. [GS Search]
- Keller, J. (1987). Development and Use of the ARCS Model of motivational Design. *Journal of Instructional Development*, 10(3), 2-10. [GS Search]
- Kolodner, J. L. (2002). Facilitating the learning of design practices: Lessons learned from an inquiry into science education. *Journal of Industrial Teacher Education*, 39(3), 9-40. [GS Search]
- Lee, S., Hooshyar, D., Ji, H., Nam, K. & Lim, H. (2017). Mining biometric data to predict programmer expertise and task difficulty. *Cluster Computing*, 22, 1–11. [GS Search]
- Lye, S. Y., Koh & J. H. L. (2014) Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, 41(C), 51-61. [GS Search]
- Material Design (2019). Retrieved from <https://material.io/design/>.
- MEC. (2018) Base Nacional Comum Curricular. Retrieved from [http://basenacionalcomum.mec.gov.br/images/BNCC\\_EI\\_EF\\_110518\\_-versaofinal\\_site.pdf](http://basenacionalcomum.mec.gov.br/images/BNCC_EI_EF_110518_-versaofinal_site.pdf).
- MIT; App Inventor. (2019). App Inventor. Retrieved from <http://appinventor.mit.edu/explore/about-us.html>.
- Nielsen Norman Group (2018). Nielsen Norman Group. Retrieved from <https://www.nngroup.com/>.

- Petersen, K. et al. (2008). Systematic mapping studies in Software engineering. In: *Proc. of the 12th Int. Conference on Evaluation and Assessment in Software Engineering*, Swindon, Inglaterra, 68-77. [GS Search]
- Petri, G., Gresse von Wangenheim, C. & Borgatto A.F. (2017). Evolução de um Modelo de Avaliação de Jogos para o Ensino de Computação. In: *Anais do 25º Workshop sobre Educação em Computação (CSBC/WEI)*, São Paulo, Brasil. [GS Search]
- Petri, G., Gresse von Wangenheim, C. & Borgatto A.F. (2018). MEEGA+, Systematic Model to Evaluate Educational Games. In: Lee N. (Eds.) *Encyclopedia of Computer Graphics and Games*, Springer. [GS Search]
- Resnick, M. et al. (2009). Scratch: Programming for all. *Communications of the Acm*, 52(11), 60-67. [GS Search]
- Robinson, A. & Pérez-Quñones, M. A. (2014). Underrepresented middle school girls: on the path to computer science through paper prototyping. In: *Proc. of the 45th ACM Technical Symposium on Computer Science Education*, Atlanta, GA, EUA, 97-102. doi: <https://doi.org/10.1145/2538862.2538951> [GS Search]
- Robinson, A., Pérez-Quñones, M. A. & Scales, G. (2015). Understanding the attitudes of African American middle school girls toward computer science. In: *Proc. of Research in Equity and Sustained Participation in Engineering, Computing, and Technology*, Charlotte, NC, EUA. doi: [10.1109/RESPECT.2015.7296507](https://doi.org/10.1109/RESPECT.2015.7296507) [GS Search]
- Savi, R., Gresse von Wangenheim, C. & Borgatto, A. F. (2011). A Model for the Evaluation of Educational Games for Teaching Software Engineering. In: *Proc. of the Brazilian Symposium on Software Engineering*, São Paulo, Brasil. doi: [10.1109/SBES.2011.27](https://doi.org/10.1109/SBES.2011.27) [GS Search]
- SBC. (2018). Diretrizes para ensino de Computação na Educação Básica. Sociedade Brasileira de Computação. Retrieved from <http://www.sbc.org.br/educacao/diretrizes-para-ensino-de-computacao-na-educacao-basica>.
- Sindre, G. & Moody, D. (2003). Evaluating the Effectiveness of Learning Interventions: an Information Systems Case Study. In: *Proc. of the 11th European Conference on Information Systems*, Paper 80. Naples, Italy. [GS Search]
- Solecki, I. S., Justen, K. A., Porto, J. V. A., Gresse von Wangenheim, C., Hauck, J. C. R. & Borgatto, A. F. (2019). Estado da Arte do Design Visual de Aplicativos Móveis desenvolvidos com App Inventor. Submetido para publicação.
- Sullivan, J. F., Reamon, D. & Louie, B. (2003). Girls embrace technology: A summer internship for high school girls. In: *Proc. of Frontiers in Education*, Westminster, CO, EUA. [GS Search]
- Technovation. (2018). Technovation Challenge. Retrieved from <http://www.technovationbrasil.org/curriculo>.
- Tekkie Uni (2018). Build Your First App. Retrieved from <https://tekkieuni.com/courses/build-your-first-app/>.
- Tissenbaum, M.; Sheldon, J. & Abelson, H. (2019). From Computational Thinking to Computational Action. *ACM Interactions*, 62(3), 34-36. [GS Search]
- UXQB (2018). CPUX-F Curriculum and Glossary. Retrieved from [https://uxqb.org/wp-content/uploads/documents/CPUX-F\\_EN\\_Curriculum-and-Glossary.pdf](https://uxqb.org/wp-content/uploads/documents/CPUX-F_EN_Curriculum-and-Glossary.pdf).
- Van Wart, S. J., Vakil, S. & Parikh, T. S. (2014). Apps for social justice: Motivating computer science learning with design and real-world problem solving. In: *Proc. of the Conference on Innovation e Technology in Computer Science Education*, Uppsala, Sweden, 123-128. [GS Search]
- Vasconcelos, F. H. L. et. al. (2005). Inclusão Digital e Social: Um Exemplo da Formação Profissionalizante para Jovens com o uso de Tecnologias Computacionais. In: *Anais do Workshop de*



*Informática na Escola*, São Leopoldo, Brasil. doi: <http://dx.doi.org/10.5753/cbie.wie.2005.%25p> [GS Search]

Wang, P. S. (2015). *From Computing to Computational Thinking* (1st ed.), New York: Chapman and Hall/CRC. [GS Search]

West-Knights, I. (2017). Why are schools in China looking west for lessons in creativity? *Financial Times*. [GS Search]

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35. doi: <https://dl.acm.org/doi/10.1145/1118178.1118215> [GS Search]

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B. & Wesslén, A. (2012). *Experimentation in Software Engineering*. New York: Springer-Verlag Berlin Heidelberg. [GS Search]

Yardi, S., Krolikowski, P., Marshall, T. & Bruckman, A. (2008). An IHC Approach to Computing in the Real World. *Journal on Educational Resources in Computing*, 8(3). [GS Search]

Yin, R. K. (2017). *Case study research: design and methods* (6th ed.). Thousand Oaks: Sage Publications, Inc. [GS Search]