

DESENVOLVIMENTO DE UMA UNIDADE INSTRUCIONAL PARA ENSINAR O DESENVOLVIMENTO DE APPS NO ENSINO FUNDAMENTAL COM O APP INVENTOR

Raul M. Filho¹, Christiane G. von Wangenheim¹, Jean R. Hauck¹, Giselle A. S. Medeiros¹

¹Departamento de Informática e Estatística
Universidade Federal de Santa Catarina - Florianópolis, SC - Brazil

raul_mf96@hotmail.com, c.wangenheim@ufsc.br, jean.hauck@ufsc.br,
gisellearaujo.ufsc@gmail.com

Abstract. *Computing has become increasingly influential nowadays, helping and improving people's lives. For this reason, computing should be popularized so that students can learn key computer ideas and how to code so that they can solve problems in the easiest and most efficient possible way in their interest areas. While there are currently several initiatives to teach computing in elementary school, there are basically no instructional units covering other important concepts such as interface design. In this context, is systematically developed an instructional unit to teach programming and apps interface design as an important competency in the area of Computing following ACM/CSTA K-12 Computer Science Framework and SBC curricula in Elementary Schools. This instructional unit covers the teaching of app programming by following a software process, design thinking and interface design through the development of apps for Android phones, using the App Inventor visual programming environment. The results confirm the unit's success, achieving the learning objectives, facilitating learning and arousing interest in the students.*

Resumo. *A computação vem se tornando cada vez mais influente nos dias atuais, auxiliando e melhorando a vida das pessoas. Por esse motivo, a partir do Ensino Fundamental, a computação deve ser popularizada para que estudantes possam aprender ideias-chave de computação e a programação resolvendo problemas da maneira mais fácil e eficiente possível em suas áreas de aplicação. Mesmo existindo atualmente várias iniciativas para ensinar computação no Ensino Fundamental, basicamente não existem unidades instrucionais abrangendo também outros conceitos importantes, como o design de interface. Neste contexto, é desenvolvida uma unidade instrucional de maneira sistemática para ensinar a programação e design de interface de apps como competência importante na área de Computação alinhado aos currículos de referência do ACM/CSTA K-12 Computer Science Framework e da SBC no Ensino Fundamental. Esta unidade instrucional contempla o ensino de programação de apps seguindo um processo de software, design thinking e design de interface por meio do desenvolvimento de apps para celulares Android, utilizando o ambiente de programação visual App Inventor. Os resultados confirmam o sucesso da unidade, atingindo os objetivos de aprendizagem, facilitando a aprendizagem e despertando o interesse nos alunos.*

1. Introdução

A influência da computação é sentida por todos individualmente, em sociedade e também de forma global [CSTA 2016]. A área de computação vem possibilitando grandes inovações em todos os campos de estudo e também na vida diária das pessoas. Diversos tipos de informações podem ser transformados e processados pelos computadores para criar apps, jogos, carros autônomos, robôs inteligentes e muito mais [CSTA 2016]. Portanto, é muito importante que as pessoas já aprendam desde pequenas os conceitos da computação.

Com o aprendizado da Computação, o estudante desenvolve seu raciocínio lógico, pensamento algorítmico, design e uma estruturada forma de resolução de problemas, conceitos e habilidades que podem ser usados muito além das salas de aula [CSTA 2011]. Essas competências proporcionam ao aluno a capacidade de implementar, testar e implantar uma solução, lidando com as restrições do mundo real, e essas habilidades são praticáveis em muitos contextos [CSTA 2011]. Esse cenário enfatiza a necessidade de oferecer uma educação que envolva diferentes conhecimentos de computação, de forma que computadores devem ser vistos muito além de ferramentas, mas, sim, um meio acessível de expressar ideias e expor a criatividade [Cabraia e Scaico 2013].

O ensino de conceitos básicos de computação nas escolas é fundamental para desenvolver o raciocínio computacional e lógico das crianças, pelo seu caráter transversal às demais ciências [Nunes 2011]. É necessário o desenvolvimento de habilidades computacionais na educação básica, pois se caracteriza como algo importante intelectualmente. Pode promover múltiplos caminhos profissionais futuros, desenvolver a capacidade de resolução de problemas, apoiar e relacionar-se com outras ciências e motivar os estudantes [CSTA 2011]. O raciocínio lógico deveria ser ensinado desde cedo, pois aumenta a capacidade de dedução e conclusão de problemas [Sica 2011]. Porém, o ensino dessas competências é reservado muitas vezes somente para o ensino superior que optam pelos cursos relacionados, mas as diversas outras áreas do saber ficam deficientes neste quesito [Sica 2011]. Atualmente o que é ensinado em escolas é o que se chama de “computer literacy” [CSTA 2016], que se trata do ensino do uso de aplicativos tais como editores de texto, imagens, apresentações, etc. O objetivo é que seja ensinado nas escolas a proficiência digital, “IT fluency” [CSTA 2011], que se refere a possibilitar os alunos a terem a capacidade de aprender e aplicar novas tecnologias de forma produtiva.

Já existem diversas iniciativas mundialmente focando no ensino de computação explorando diversas maneiras de ensino, como unidades escolares, *code clubs*, *summer camps*, etc. [MIT(c) 2019], focando principalmente no ensino de computação via programação de jogos, robôs ou *apps*. Tipicamente, neste estágio escolar, a forma como é ensinada a programação é via linguagens de programação baseadas em blocos, como *Scratch* [Scratch 2019] por exemplo.

O *App Inventor* [MIT(a) 2019] é um ambiente de programação visual baseado em blocos e permite qualquer pessoa, até mesmo uma criança, começar a programar e construir aplicativos completos para dispositivos *Android*. Criando *apps* com o *App Inventor*, estudantes aprendem a pensar criativamente, a trabalhar de forma colaborativa e a pensar de forma sistemática na solução de problemas.

Já existem unidades instrucionais voltadas ao ensino de programação de *apps* com *App Inventor*. A maioria envolve unidades instrucionais ensinando a programação de um determinado tipo de *app* por meio de vídeos explicativos [MIT(b) 2019] e tutoriais *online*, como o “Magic 8-ball” que faz previsões do futuro [MIT(d) 2019] ou oficinas [Daniel 2016]. A grande parte destes são em Inglês, com poucas exceções, como por exemplo o jogo do

mosquito [Daniel 2016]. Por outro lado, surgiram também iniciativas como a *Technovation*, que estimula meninas a aprenderem a programar por meio de um desafio, que dispõe um currículo e material didático guiando o desenvolvimento de um app incluindo conceitos de negócio e de design [Technovation 2019]. Contudo, grande parte das iniciativas ensina somente a programação dos *apps*, muitas vezes sem abordar outras áreas importantes da computação como engenharia de *software*, *design thinking*, e design de interfaces. O *design thinking* fornece um processo dinâmico, participativo, e criativo para empatizar, definir, idealizar, prototipar e testar um sistema de software [Razzouk e Shute 2012] focando na resolução de problemas que valoriza e explora várias perspectivas de um problema [Chen e Huang 2017].

Desta forma, este projeto desenvolve uma unidade instrucional voltada ao ensino de computação integrando conceitos de engenharia de *software*, *design thinking* e design de interfaces para alunos do Ensino Fundamental no Brasil. A unidade é alinhada ao *framework* de currículo da CSTA [CSTA 2016] e do guia de currículo da SBC [SBC 2018]. A unidade ensina a computação por meio de programação de *apps* com *App Inventor* no contexto de uma aprendizagem baseada em problemas, abrangendo desde a parte do levantamento do problema, definição de solução, design de interface até o desenvolvimento de um aplicativo funcional. Espera-se que esta unidade instrucional criada possa ser aplicada amplamente em escolas do Brasil no ensino fundamental, popularizando não só competências de programação, mas também os conceitos de design e engenharia de *software* como elementos do aprendizado de computação.

2. Design Instrucional

O objetivo geral desta experiência relatada é o ensino de conceitos de computação por meio da programação de apps para dispositivos móveis Android utilizando a ferramenta App Inventor no Ensino Fundamental de escolas públicas brasileiras. Dessa maneira, o contexto de aplicação é formado pelo público alvo de alunos do Ensino Fundamental de escolas brasileiras com idade entre dez e quinze anos. Atualmente, os alunos nesta idade tipicamente possuem capacidade de utilizar e controlar dispositivos eletrônicos, sobretudo *smartphones* [D'Angelo 2017]. O ambiente de aplicação são escolas públicas municipais de Florianópolis que possuem uma sala informatizada com computadores e acesso à Internet. Estas sala possuem um professor responsável que normalmente possuem formação em alguma licenciatura com alguma especialização em tecnologias na educação [PMF 2019], mas não costumam ter conhecimentos na área de computação.

Os objetivos de aprendizagem envolvem principalmente os conceitos de programação e o pensamento computacional, mas também os de engenharia de *software* e design de interfaces, conteúdos focados nesta experiência. Adicionalmente, também são incluídos objetivos de aprendizagem relacionados à sustentabilidade, pelo caráter interdisciplinar da unidade instrucional com a disciplina de Ciências.

São criadas duas versões da unidade instrucional proposta, possibilitando a aplicação de forma interdisciplinar em outras disciplinas da educação básica e/ou no contra turno ou como atividade extracurricular. A versão mais curta é projetada para a disciplina de Ciências com o tema desenvolvimento e consumo sustentável, o qual compreende ensino de conceitos de computação ao mesmo tempo que os assuntos de sustentabilidade são envolvidos. A unidade em sua versão longa é projetada para ser aplicada durante um tempo mais prolongado, com diversas aulas, cada uma abordando conceitos específicos do processo de desenvolvimento de apps. As versões da UI são projetadas com o objetivo de serem ministradas de forma presencial, sendo coordenadas por um professor da educação básica com

auxílio ou não de um mentor capacitado nos assuntos envolvidos na unidade. Há também a possibilidade da unidade ser aplicada à distância.

O conteúdo da unidade instrucional inclui o ensino competências de Engenharia de *Software*, de programação, de design thinking e design de interface, por meio do desenvolvimento de aplicativos com a ferramenta App Inventor. As versões da unidade são projetadas para a aplicação em uma duração de 30 e 10 horas/aula nas versões longa e curta, respectivamente. O material instrucional envolve slides, um jogo, tarefas de casa, workbooks para a documentação dos artefatos criados, vídeos, etc. (Figura 1).



Figura 1. Materiais instrucionais

3. Metodologia

O objetivo deste trabalho é o desenvolvimento, aplicação e avaliação de uma unidade instrucional com o intuito de ensinar conceitos de computação no Ensino Fundamental. Dessa forma, é realizado um estudo sobre os resultados da unidade por meio de um caso de estudo exploratório para compreender os fatos observados e coletados durante as aplicações, possibilitando a orientação para trabalhos futuros.

O estudo de caso é realizado conforme os procedimentos propostos por Wohlin et al. (2012) e Yin (2013).

Definição do estudo: a definição do estudo é realizada com base na proposição da pergunta de pesquisa, hipóteses e design de pesquisa. Da pergunta de pesquisa são, então, derivadas quatro perguntas de análise as quais são sistematicamente decompostas em medidas para a coleta de dados conforme a abordagem GQM [Basili et al. 1994]. A coleta de dados é realizada por meio sobretudo de questionários.

Execução do estudo: como abordagem de design instrucional é utilizado o modelo ADDIE [Branch 2009], o qual possibilita o desenvolvimento sistemático da unidade. Aqui são realizadas duas etapas, a primeira se caracteriza pelo design da unidade, levantando as características dos aprendizagem e do ambiente. Com isso, são definidos e/ou produzidos os objetivos de aprendizagem, planos de ensino, materiais instrucionais, métodos e estratégias instrucionais a serem utilizados. Na segunda etapa, a unidade instrucional é aplicada em suas duas versões, coletando os dados sobre sua realização com as formas definidas para tal.

Análise e interpretação do estudo: os dados coletados das duas aplicações da unidade instrucional são, então, analisados e interpretados com base em métodos quantitativos e qualitativos, gerando os resultados e conclusões finais.

4. Resultados

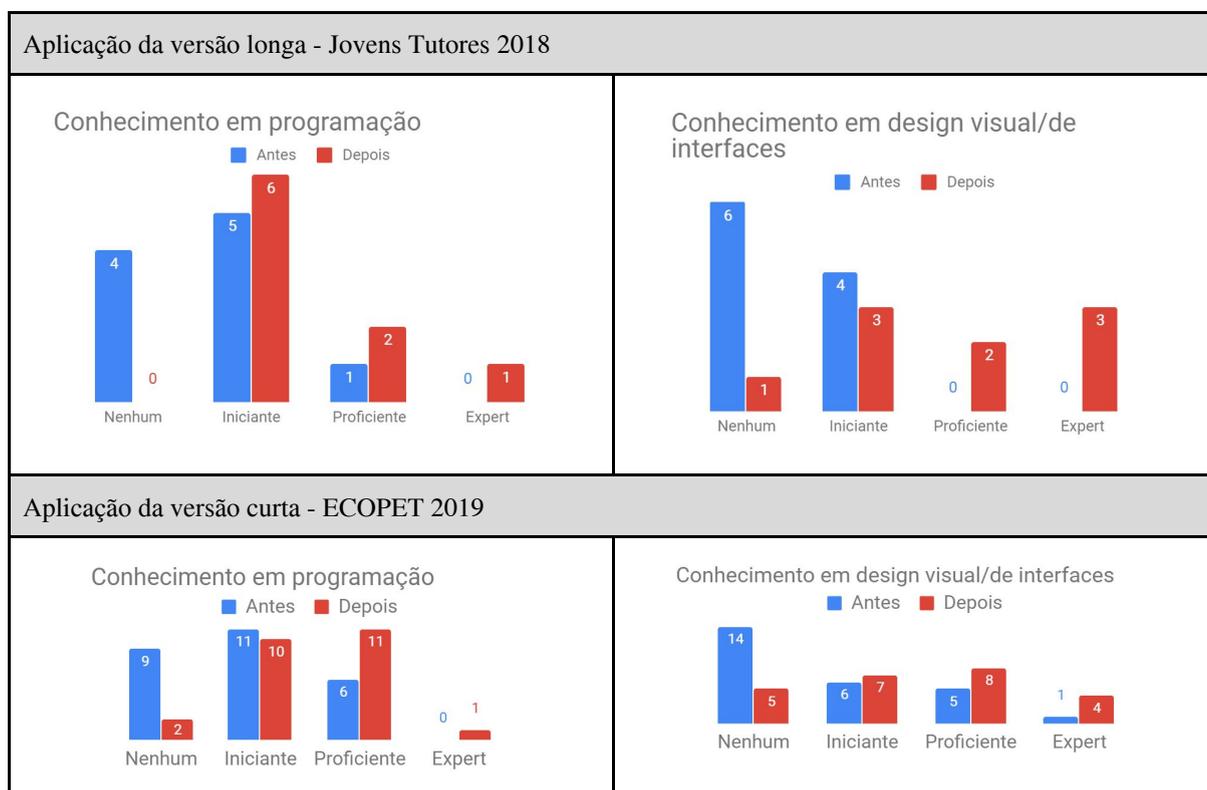
Foram realizadas duas aplicações das versões da unidade instrucional nos anos de 2018 e 2019 na escola Almirante Carvalhal na cidade de Florianópolis. A quantidade de respostas e os dados demográficos podem ser observados na Tabela 1.

Tabela 1. Informações das duas aplicações realizadas

Aplicação da versão longa - Jovens Tutores 2018	
Quantidade de respostas	9
Quantidade de meninas	5
Quantidade de meninos	5
Média de idade	14 anos
Aplicação da versão curta - ECOPET 2019	
Quantidade de respostas	24
Quantidade de meninas	12
Quantidade de meninos	14
Média de idade	10 anos

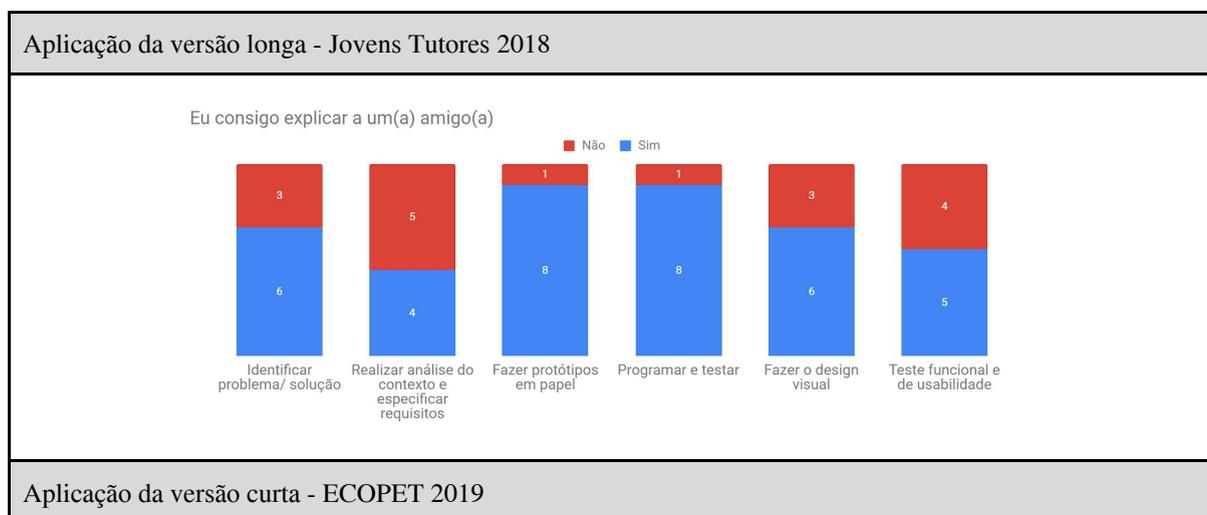
Os resultados da análise dos dados coletados durante as duas aplicações indicam uma avaliação positiva da unidade instrucional (Tabela 2). Em ambas, os alunos seguiram todo o processo de desenvolvimento proposto e obtiveram um acréscimo em seu aprendizado e um ótimo resultado nos apps desenvolvidos. Foi notável que a maioria dos alunos estava motivada e interessada em aprender mais sobre os conceitos de computação, o que acarretou em uma grande experiência para eles. A percepção do aumento do aprendizado foi algo notório pela grande maioria dos alunos. Os resultados indicam, para as duas aplicações, que antes e depois dos cursos há uma significativa diferença na aprendizagem percebida pelos alunos. Ter havido alunos que disseram ter nível “expert” em certos conteúdos, faz crer que os resultados foram ainda melhor do que o esperado e que estes alunos têm confiança de que o que aprenderam pode ser fielmente reproduzido por eles.

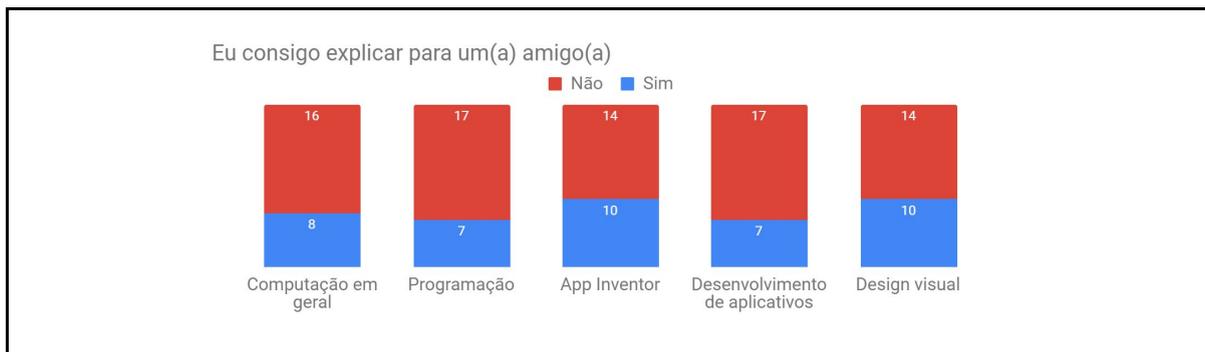
Tabela 2. Autoavaliação sobre o conhecimento nos conceitos de computação



Enquanto na primeira aplicação os alunos, em geral, se sentem capacitados a explicar a maioria dos conteúdos, na segunda, os participantes não demonstram habilidade e segurança para ensinar os conteúdos ensinados (Tabela 3). Uma das respostas para esta diferença é que os alunos na primeira aplicação tiveram mais tempo para aprender os conteúdos e também um contato mais direto com os instrutores, por serem em menor quantidade. Também, na segunda aplicação os alunos eram em média 4 anos mais novos e, tipicamente, não é esperado que alunos nessa idade tenham a habilidade de transferir conhecimentos, ainda mais algo tão novo e complexo para eles.

Tabela 3. Capacidade percebida pelos alunos em ensinar os conceitos vistos nas aulas





O desempenho dos alunos na primeira aplicação foi algo que chamou muito a atenção, superando expectativas pela qualidade dos *apps* criados (Tabela 4). Este resultado demonstra que a utilização de um processo de desenvolvimento de *apps* bem definido, juntamente aos conceitos de *design thinking*, permite construir aplicativos úteis e com uma excelente interface com o usuário. Pela avaliação dos artefatos desenvolvidos com base nos *workbooks*, é possível perceber que os alunos conseguiram compreender e produzir as etapas necessárias no processo de desenvolvimento de uma maneira adequada, comprovando a qualidade do resultado final. Em relação aos conceitos de engenharia de *software* como histórias de usuário e testes funcionais os alunos demonstram um significativo grau de conhecimento, demonstrando que foram conteúdos bem aceitos por eles. Já os conteúdos envolvendo casos de uso e funcionamento das tarefas dos *apps*, os alunos tiveram grande dificuldade para aplicar e mesmo com certa ajuda os resultados não foram satisfatórios. Na segunda aplicação, o desempenho dos alunos em relação aos *apps* desenvolvidos não foi tão satisfatória quanto na primeira (Tabela 4). Os alunos demonstraram mais dificuldade em aplicar os conteúdos estudados em sala e o resultado foi percebido nos *apps* construídos. Um dos fatores que influenciou este resultado negativo é a quantidade de alunos em relação ao tamanho da sala. Pelo fato da sala estar mais cheia, para prender a atenção dos alunos é muito mais difícil e distrações afetam o aprendizado dos conteúdos ensinados. Outro fato que fez a diferença é a idade dos alunos, diversos fatores podem ser afetados por causa disso. O interesse dos alunos se torna mais limitado, a capacidade de prestar atenção é afetada, a experiência em resolver problemas e a preferência pela diversão em vez de um bom resultado. Como o tempo geral do curso também foi reduzido em relação à primeira aplicação, as aulas tiveram que ser comprimidas ou não ensinadas, justificando os resultados inferiores.

Tabela 4. Apps desenvolvidos nas duas aplicações

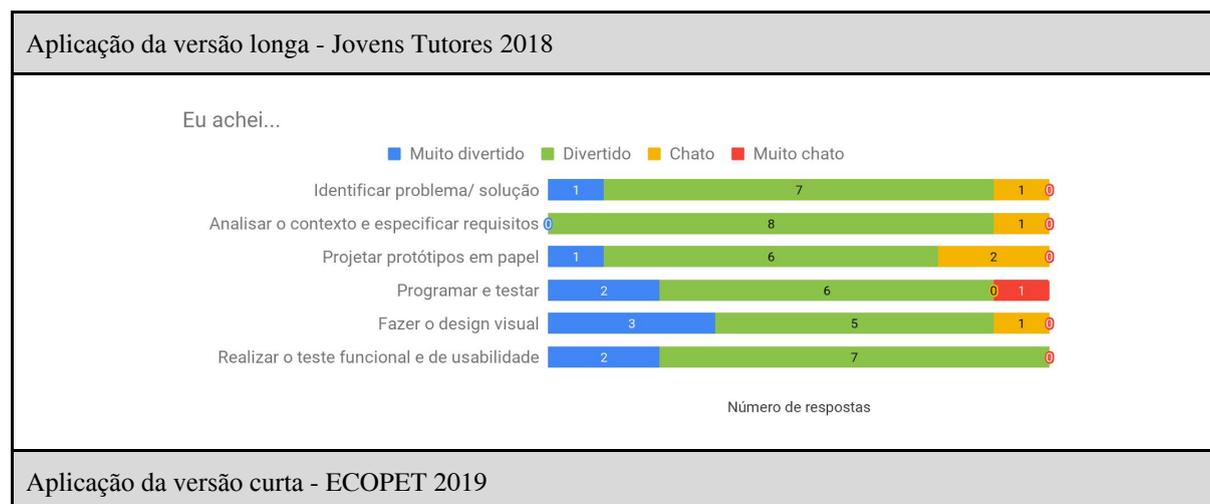


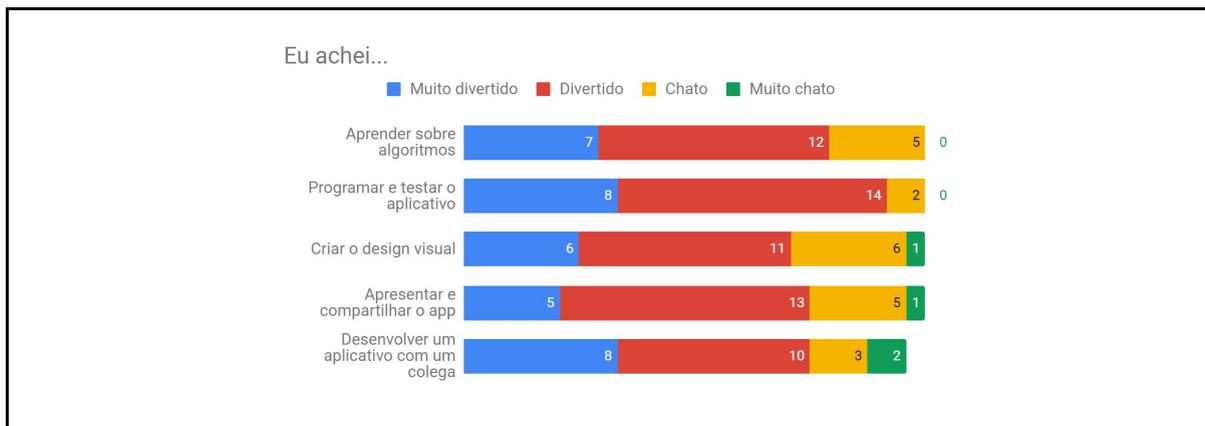
Aplicação da versão curta - ECOJET 2019



A participação nas aplicações da unidade instrucional se mostrou algo muito prazeroso para os alunos, que acreditam ter sido uma experiência divertida (Tabela 5). Este fato demonstra que as aulas foram aplicadas de uma forma agradável e o nível de ensino dos conteúdos foi adequado a maioria dos alunos. A percepção do tempo durante as aulas foi outro parâmetro que indica a boa experiência do curso. Boa parte dos estudantes aponta que não sentiu a passagem do tempo durante o ensino dos conteúdos em sala, o que demonstra que eles voltaram seu foco na instrução e estavam interessados no aprendizado dos conceitos apresentados. Outro ponto que chama atenção é a apreciação pelo desenvolvimento dos apps com um colega, apresentada pela grande fração dos alunos. O desenvolvimento em equipe estimula a discussão entre os alunos, sempre buscando a melhor maneira de resolver os problemas ou na hora de decidir decisões de projeto. Além disso, a construção do app fica sujeita a análise por duas pessoas, o que leva a um resultado com mais qualidade e menos defeitos.

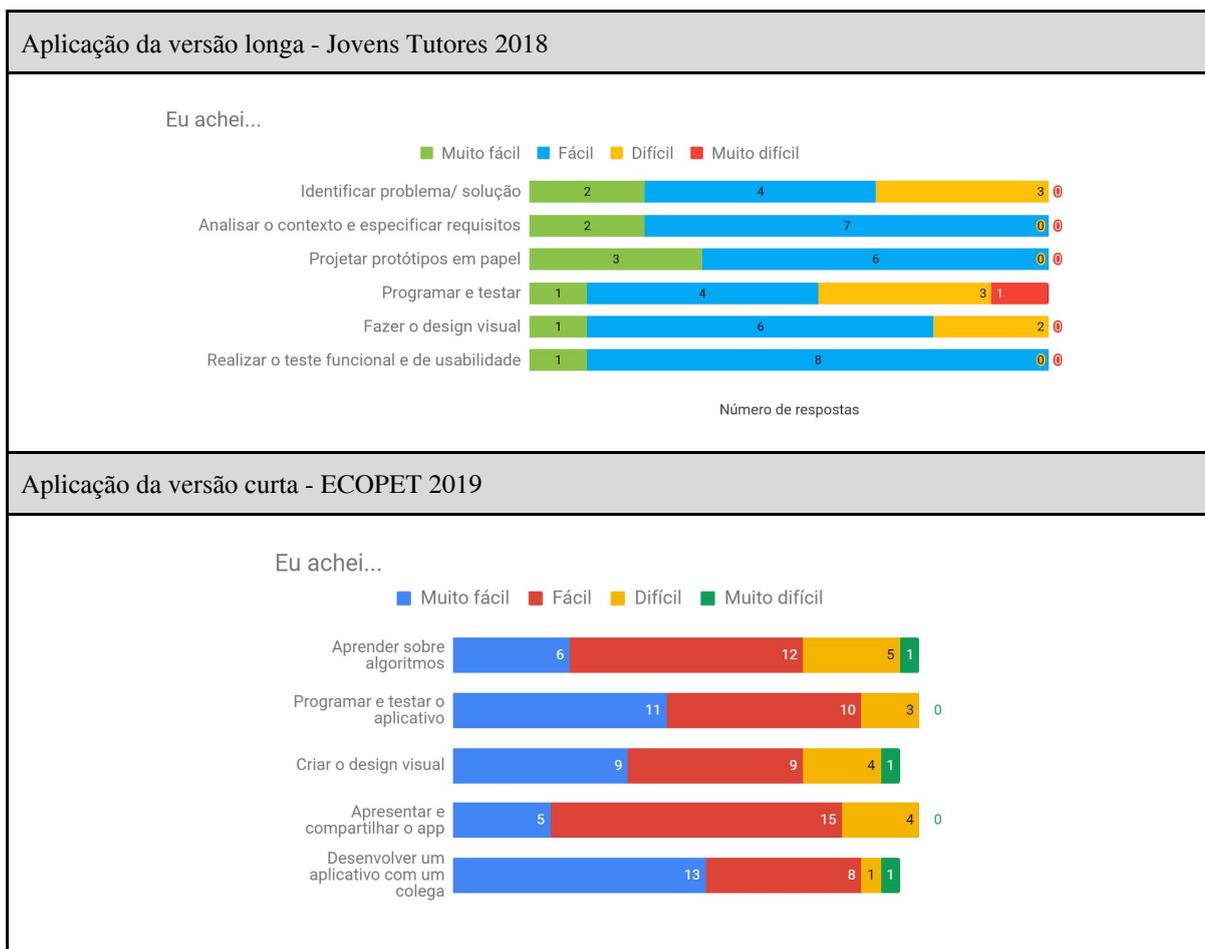
Tabela 5. Experiência de aprendizado nas duas aplicações





A utilização da ferramenta *App Inventor* e a forma como as aulas foram planejadas auxiliou para que os alunos avaliassem o aprendizado dos conteúdos como fáceis (Tabela 6). Por mais que a maioria dos conceitos ensinados tenham um grau de complexidade elevado, os participantes das aplicações sentiram um nível adequado de aprendizagem, caracterizando com um certo grau de facilidade.

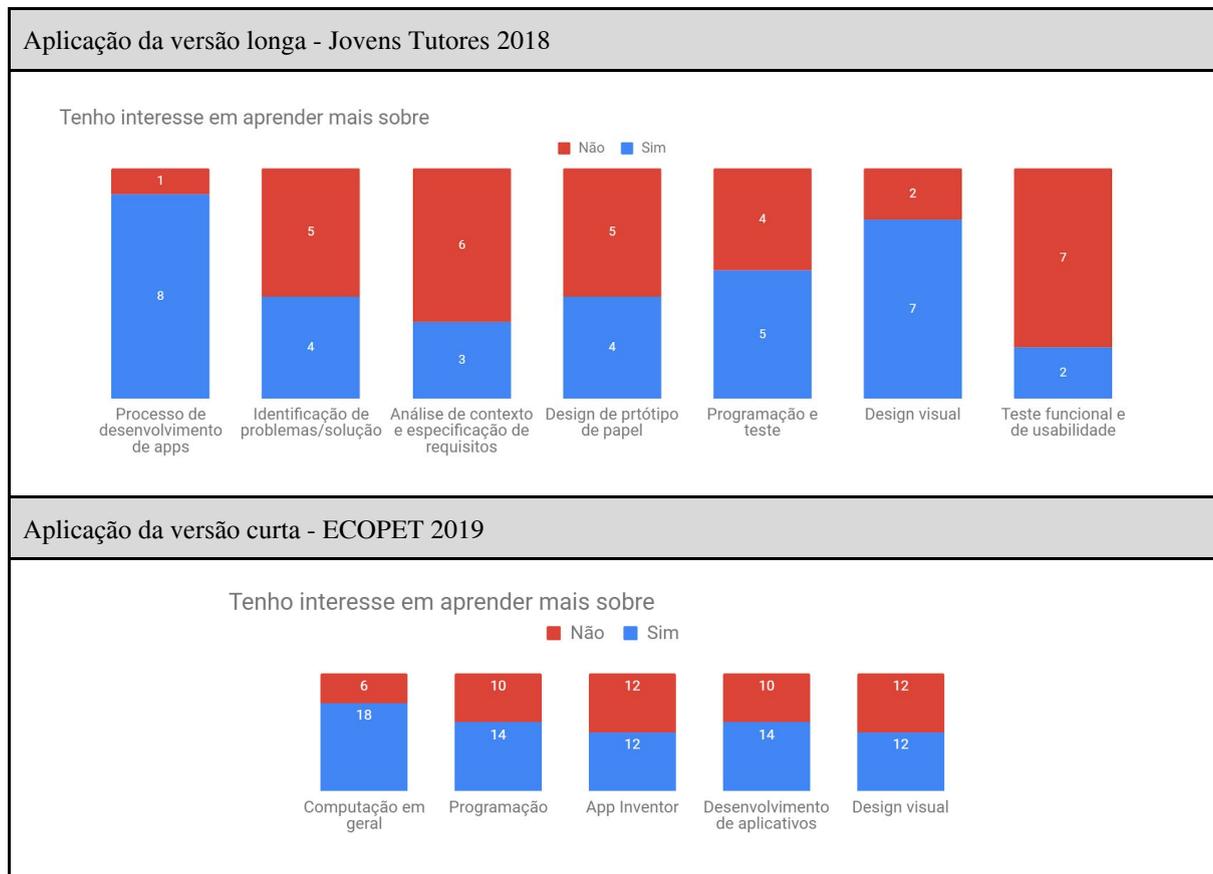
Tabela 6. Facilidade de aprendizado nas duas aplicações



A adoção de um processo de desenvolvimento de *apps* com *design thinking* envolvendo conceitos design visual serviu como incentivo para que os alunos se

interessassem mais pelos conteúdos, tornando-os mais criativos e participativos (Tabela 7). Também permitiu a construção de diversos *apps* úteis que resolvem problemas da sociedade, permitindo que os alunos se envolvam na mudança do mundo em que vivem de maneira prática. A percepção da importância dos conteúdos ensinados também comprova que os alunos compreendem que no futuro essa aprendizagem se tornará necessária a diferença no mundo em que vivem.

Tabela 7. Interesse em aprender mais sobre os conteúdos nas duas aplicações



5. Conclusão

A finalidade deste trabalho foi o desenvolvimento de uma unidade instrucional para ensinar o desenvolvimento de aplicativos para alunos do ensino fundamental de escolas brasileiras com a ferramenta App Inventor, envolvendo conceitos de engenharia de software e design de interfaces. A unidade foi aplicada em dois momentos diferentes, em uma versão longa/completa e outra mais curta.

A versão longa envolveu 10 alunos do ensino fundamental, com idade entre 13 e 15 anos, na produção de um aplicativo funcional. A avaliação desta aplicação permitiu notar que as expectativas foram cumpridas e os resultados foram muito além do esperado. A maioria dos conteúdos ensinados foram bem aceitos pelos alunos e as competências e o interesse dos alunos tiveram um acréscimo notável. Na aplicação mais curta, foram envolvidos 28 alunos do ensino fundamental, com idade entre 9 e 11 anos, no aprimoramento de um aplicativo já desenvolvido focando mais na construção do design de interfaces de maneira interdisciplinar com a disciplina de Ciências com o tema de sustentabilidade. A avaliação desta aplicação

mostrou que o tempo mais curto, a idade e a quantidade de alunos tiveram forte influência para que os resultados não fossem tão bons quanto na primeira aplicação. Entretanto, em geral, os alunos se mostraram interessados em aprender mais sobre os conteúdos e acreditam que foi uma experiência divertida ter participado do curso.

Como resultado deste trabalho, é disponibilizada uma unidade instrucional para ensinar o processo de desenvolvimento de apps por meio do desenvolvimento de aplicativos abrangendo conceitos de engenharia de software e design de interfaces. Indica-se que esta unidade seja aplicada para alunos a partir do ensino fundamental, com idade acima de 14 anos de idade com ou sem experiência prévia em computação/programação. A ideia geral desta unidade é capacitar os alunos à produção de *apps* utilizando um processo de desenvolvimento adequado, desenvolvendo habilidades de programação e design. Sua aplicação pode ser inserida tanto de forma interdisciplinar quanto extracurricular e, até mesmo, a distância.

Referências

- Basili, V. R. et al. (1994) “The Goal Question Metric Approach”, <http://www.cs.umd.edu/~mvz/handouts/gqm.pdf>, jul. 2019.
- Branch, R. M. (2009) *Instructional Design: The ADDIE Approach*. New York, New York, USA, Springer Science & Business Media.
- Cambráia, A. C. e Scaico, P. D. (2017) “Os desafios da Educação em Computação no Brasil: um relato de experiências com Projetos PIBID no Sul e Nordeste do país”. *Revista Espaço Acadêmico*, n. 194.
- CSTA. (2016) “K-12 Computer Science Framework”, <http://k12cs.org/wp-content/uploads/2016/09/K%E2%80%9312-Computer-Science-Framework.pdf>, jul. 2019.
- CSTA. (2011) “K-12 Computer Science Standards” <https://www.csteachers.org/page/standards>, jul. 2019.
- D'Angelo, P. (2017) “Pesquisa sobre smartphones: a relação dos pais e das crianças com smartphones no Brasil”, <https://blog.opinionbox.com/pesquisa-smartphones-criancas/>, jul. 2019.
- Daniel, G. T. (2016) *Design de unidade instrucional de desenvolvimento de aplicativos para o ensino fundamental*, Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Departamento de Informática e Estatística, Universidade Federal de Santa Catarina, Florianópolis.
- MIT (a). (2019) “About us”, <http://appinventor.mit.edu/explore/about-us.html>, jul. 2019.
- MIT (b). (2019) “Beginner Video Tutorials”, <http://appinventor.mit.edu/explore/ai2/beginner-videos.html>, jul. 2019.
- MIT (c). (2019) “News & Events”, <http://appinventor.mit.edu/explore/news-events.html>, jul. 2019.
- MIT (d). (2019) “Magic 8-Ball Tutorial”, <http://appinventor.mit.edu/explore/teach/magic-8-ball.html>, jul. 2019.

- Nunes, D. J. (2012) *Ciência da Computação na Educação Básica*. Anais do XX Workshop sobre Educação em Computação (WEI), Curitiba, Paraná.
- PMF. (2019) “Processo Seletivo De Substitutos - Edital nº 004/2018”, http://substituto2019.fepese.org.br/?go=download&arquivo=2018_PMF_Substitutos_Ed_04.pdf, jul. 2019.
- SBC. (2017) “Referenciais de Formação em Computação: Educação Básica”, <http://www.sbc.org.br/files/ComputacaoEducacaoBasica-versaofinal-julho2017.pdf>, jul. 2019.
- Scratch. (2019) “Acerca do Scratch”, <https://scratch.mit.edu/about>, jul. 2019.
- Sica, C. (2011) “Ciência da Computação no Ensino Básico e Médio”, <http://www.odiario.com/blogs/carlossica/2011/10/07/ciencia-da-computacao-no-ensino-medio/>, jul. 2019.
- Technovation. (2019) “Currículo”, <http://www.technovationbrasil.org/curriculo>, jul. 2019.
- Wohlin, C. et al. (2012) *Experimentation in Software Engineering*. Berlin: Springer-Verlag Berlin Heidelberg, p. 235.
- Yin, R. K. (2009) *Case Study Research: Design and Methods*. SAGE, 4 ed., p. 219.