

INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO DO PROBLEMA

A computação está cada vez mais presente no nosso cotidiano por meio dos diversos dispositivos digitais que estão se tornando indispensáveis para as nossas tarefas. A maioria das profissões, em maior ou menor grau, também utilizam soluções por meio dos computadores para cumprir suas funções. Assim, é necessário que as pessoas, independentemente da sua área de conhecimento, tenham uma compreensão do pensamento computacional (CSTA, 2011). Esta compreensão possibilita resolver problemas de forma computacional, aumentando a produtividade, criatividade e inventividade do profissional, tornando competitivo na sua área (BLIKSTEIN, 2008).

A Sociedade Brasileira de Computação (SBC, 2017) entende que a Computação é uma ciência que deve ser ensinada desde cedo para que no futuro tenhamos recursos humanos qualificados para enfrentar os desafios que advirão. Este conhecimento é também fundamental para que os alunos possam ser educados tecnologicamente e possibilitar criar novas tecnologias visando ajudar a vida das pessoas (WANGENHEIM, 2014). Além disso, estimula o interesse das pessoas nesta área, sendo que no Brasil há uma necessidade de formar profissionais para o setor de *software*, pois existe um déficit na área (CARDOSO e DE DAVID, 2017). Porém, alguns empecilhos dificultam o ensino da computação no Ensino Básico, como a falta de materiais de estudos e recursos humanos qualificados (INEP, 2010 a 2014). Ademais, este ensino não faz parte da cultura da educação básica no Brasil, se restringindo apenas aos cursos técnicos e superiores.

Existem diversos currículos com o objetivo de disseminar o conhecimento da computação no Ensino Básico, dentre eles está o modelo de currículo referencial CSTA/ACM K-12 (CSTA, 2016). O currículo define diretrizes que ensina as principais competências da computação, dentre elas as estão o pensamento computacional (resolver problemas em uma forma que pode ser implementada no computador) e a programação (competência de criar programas utilizando linguagem de programação), conhecimentos chaves para resolver problemas por meio dos computadores (CSTA, 2011).

Visando a inclusão do ensino da computação na Educação Básica, existem diversas iniciativas que desenvolveram ferramentas para auxiliar a programação, por exemplo, *Scratch* (SCRATCH; MIT, 2013) e *App Inventor* (*App Inventor*, MIT, 2013). Ainda, existem diversas unidades instrucionais, por exemplo por meio de programação de jogos ou animações com *Scratch* (OLIVEIRA, 2014), exercícios de programação com blocos (WILSON, 2015) ou robótica (BENITTI, 2009). Uma alternativa é ensinar o desenvolvimento de *apps* em celulares utilizando *App Inventor* (MIT, 2014). Neste contexto já existem alguns tutoriais (MIT, 2017) e também unidades instrucionais que ensinam fazer um *app*. Estas unidades instrucionais tipicamente enfocam no ensino da programação, não abordando o ensino de conceitos de Engenharia de *Software* (ES) e/ou Engenharia de Usabilidade (EU).

A ES é definida como uma sistemática de abordagens quantificáveis para o desenvolvimento, operação, e manutenção de *software* (IEEE CS, 2014). Assim, para ensinar computação de forma mais completa é importante também ensinar competências de ES, como: atividades de análise de requisitos e testes de *software* (CSTA, 2016). Ainda, é necessário o ensino das competências relacionada a EU, como por exemplo, a análise de contexto, prototipação de telas e a realização de testes de usabilidade (CSTA, 2016). A integração destes conceitos no ensino de computação é essencial para assegurar o desenvolvimento de aplicativos com confiabilidade e usabilidade, fatores determinantes de sucesso dos *apps* (PREECE et al., 2005). Apesar dessas duas áreas de conhecimento divergirem no foco durante desenvolvimento do *software*, percebe-se que é possível desenvolver sistemas considerando práticas de ambas as áreas (DA SILVA, 2004).

Atualmente, o ensino de computação no nível de Ensino Básico foca muito na programação. Além disso, o ensino de conceitos de ES e/ou EU se restringe ao ensino superior. Assim, faltam unidades instrucionais que sistematicamente integram estes conceitos de forma apropriado no Ensino Básico. Neste contexto, este trabalho pretende desenvolver um modelo para ensinar o desenvolvimento de aplicativos integrando práticas da ES e EU no contexto do Ensino Fundamental 2. Estas práticas estão alinhadas com o modelo de currículo de referencia CSTA/ACM K-12 (CSTA, 2016) e com base no grupo de processos das normas ABNT NBR ISO/IEC 12207:2009 e ABNT NBR ISO/IEC 9241-11. Este modelo é feito por meio de uma unidade instrucional (UI), no qual irá conter materiais didáticos, como por exemplo,

slides, roteiros, folhas de tarefas, avaliações. Segundo Gonçalves (2017, p. 28), entende-se por UI como um conjunto de aula para atingir objetivos de desempenho para um público-alvo. Esta aula é composto por materiais, atividade e exercícios para alunos e instrutores visando a aprendizagem de um assunto levando em consideração o seu contexto de aprendizagem (apud BRANCH, 2009; HILL, ROWAN, & BALL, 2005; ROOIJ, 2011). Por fim, o trabalho aplica e avalia a UI desenvolvida em escolas avaliando-as em relação à aprendizagem dos alunos, bem como sua efetividade. Nesta avaliação é utilizado o modelo dTECT (Wangeminet, 2017), que visa avaliar sistematicamente a qualidade das UIs focado o ensino da computação básica com base na percepção dos alunos.

1.2 PERGUNTA DE PESQUISA

A pergunta de pesquisa definida para o presente trabalho é: como é possível ensinar ES e EU por meio do desenvolvimento de aplicativos móveis focado no Ensino Fundamental 2 com qualidade em relação aos aspectos de conteúdo, materiais, estratégias instrucionais, e experiência do usuário.

1.3 OBJETIVOS

O objetivo geral e os objetivos específicos são descritos nesta seção.

Objetivo geral

Este trabalho tem como objetivo o desenvolvimento sistemático de um modelo de ensino de ES e EU para o desenvolvimento de aplicativos móveis no Ensino Fundamental 2 com o ambiente de programação *App Inventor*. O desenvolvimento deste modelo engloba a definição de um processo de desenvolvimento de *apps* integrando sistematicamente práticas de ES e EU, que será ensinado como parte da unidade instrucional.

Objetivos específicos

Os objetivos específicos são:

OE1. Análise da fundamentação teórica sintetizando os conceitos básicos em relação ao ensino de computação no ensino fundamental 2, ao ambiente de programação *App Inventor*, e à ES e EU.

OE2. Levantamento do estado da arte e prática por meio da revisão sistemática de literatura para entender como atualmente os conceitos de ES e EU são ensinados no nível do ensino fundamental.

OE3. Evolução de uma unidade instrucional para o ensino de desenvolvimento de aplicativos integrando o ensino de ES e EU.

OE3.1. Definição de um processo de ES e EU voltado ao desenvolvimento de aplicativos no contexto do Ensino Fundamental 2.

OE3.2. Desenvolvimento de material didático, como por exemplo, *slides*, roteiros, folhas de tarefas, avaliações.

OE3.3. Adaptação/evolução do *App Inventor* para apoiar o ensino do processo de desenvolvimento de *software*.

OE4. Aplicação e avaliação da unidade instrucional desenvolvida em escolas avaliando a unidade em relação à aprendizagem dos alunos por meio de uma série de estudos de caso, bem como sua efetividade, conforme o modelo de avaliação dTECT (WANGENHEIM et al., 2017).

DELIMITAÇÕES

Este trabalho tem como principal foco o ensino de práticas de EU e ES na Educação Básica, pois, analisando o estado da arte observa-se que as UIs criadas para o ensino da computação tem tipicamente enfoque para os cursos técnicos e superiores, ou então foca apenas em ensinar programação. As práticas de ES estão alinhado a norma ABNT NBR ISO/IEC 12207:2009, já a de EU com a norma ABNT NBR ISO/IEC 9241-11. Assim, não serão incluídos na UI práticas/métodos de ES e EU abordado por outros autores ou normas relacionado a estas áreas de conhecimento.

O escopo da UI visa o ensino de desenvolvimento de *apps* para os alunos do Ensino Fundamental 2, isto é, apenas para o período do 6º ao 9º ano. Este período está relacionado ao modelo de matriz curricular CSTA-K12, mais especificamente ao seu nível 3. A UI limita-se ao desenvolvimento para as plataformas *mobiles* que

utilizam *Android* como sistema operacional. Uma vez que o celular está se tornando comum entre os adolescentes (faixa etária de 10 a 15 anos), visto que 84% dos adolescentes usam o dispositivo (TELECO, 2016).

Por fim, este trabalho aborda ensino da computação utilizando exclusivamente a ferramenta de desenvolvimento *App Inventor*. Esta ferramenta foi escolhida por ser indicada para as pessoas que nunca programaram e desejam desenvolver seus primeiros aplicativos (WOLBER, 2012). Portanto, o trabalho não abrange a utilização de outras ferramentas, como por exemplo, *Strach*, *Hour of code*.

1.4 ADERÊNCIA À CIÊNCIA DA COMPUTAÇÃO

O tema deste trabalho está aderido conforme os objetivos do Programa de Pós-graduação em Ciência da Computação da Universidade Federal de Santa Catarina (PPGCC/UFSC), mais especificamente nos tópicos de Processo de Desenvolvimento de *Software* e Qualidade de *Software*, conforme a definição da área de Engenharia de *Software* da SBC (SBC, 2005) e alinhado com as normas ABNT NBR ISO/IEC 12207:2009 e ABNT NBR ISO/IEC 9241-11.

O regimento interno nº 01/PPGCC/2013, publicado em 01/10/2013, em seu artigo 1º, define os objetivos do programa como “desenvolvimento de novos conhecimentos em Ciência da Computação”. Dentre os conhecimentos que integram a Ciência da computação está a disciplina de ES. Esta aborda diversos tópicos, dentre elas a Qualidade de *Software* que tem como principal conteúdo a usabilidade, que contribuem para a qualidade (IEEE CS, 2014). Para o uso sistemático da usabilidade é necessário a Engenharia de Usabilidade, como então sub área de ES.

Neste sentido, este trabalho integra à Ciência da Computação ao desenvolver UIs contendo conhecimentos de ES e EU. Esses conhecimentos são relevantes, pois a UI foi desenvolvida para atender os objetivos educacionais definidos pelo currículo de referência CSTA K-12 voltado para o ensino da Ciência da Computação no Ensino Básico. Outros conhecimentos relevantes para Ciência da Computação também foram produzidos, tais como, o ensino da programação, levantamento atualizado do estado da arte e da prática sobre o ensino da computação no ensino básico por meio de uma Revisão Sistemática da Literatura (RSL) e um *Survey*.

Analisando com mais detalhe aos objetivos da linha de pesquisa em Engenharia de *Software* do PPGCC/UFSC, observa-se a aderência da presente tese a estes objetivos, quais são:

“Engenharia de *Software*: tem como objetivo formar indivíduos capazes de conduzir o processo de desenvolvimento de *software* e de investigar novas metodologias, técnicas e ferramentas para a concepção de sistemas.” (PPGCC/UFSC, 2015).

Posto isto, esta dissertação aborda primordialmente o ensino ES e EU no desenvolvimento de *apps*. Considerado que a ES é parte fundamental do processo de desenvolvimento de *software* de qualidade, estando presente entre os conhecimentos definidos pelo *Software Engineering Body of Knowledge* (SWEBOK) (IEEE CS, 2014), entende-se que há correlação do tema da tese a este objetivo da linha de pesquisa.

1.5 METODOLOGIA DE PESQUISA

1.5.1. Contexto de pesquisa e classificação

O contexto de pesquisa deste trabalho se baseia no método científico em camadas (*research-process onion*) (SAUDERS, LEWIS, & THORNHILL, 2009), conforme ilustra a Figura 1.

O contexto da pesquisa é definido nas seguintes camadas (descritas da mais interna para mais externa): horizonte de tempo, método, estratégia, abordagem e filosofia. Neste sentido, segue o contexto desta pesquisa:

- Cross-sectional: é feita análise dos indivíduos, no caso os alunos, durante o período de aplicação da UI;
- Multi-método: pois ao longo desta pesquisa são aplicados diversos métodos, tais como, Revisão Sistemática da Literatura (RSL) (KITCHENHAM, 2004), ADDIE (BRANCH, 2009), normas ABNT NBR ISO/IEC 12207:2009 e ABNT NBR ISO/IEC 9241-11, relativo a ES e EU, respectivamente.

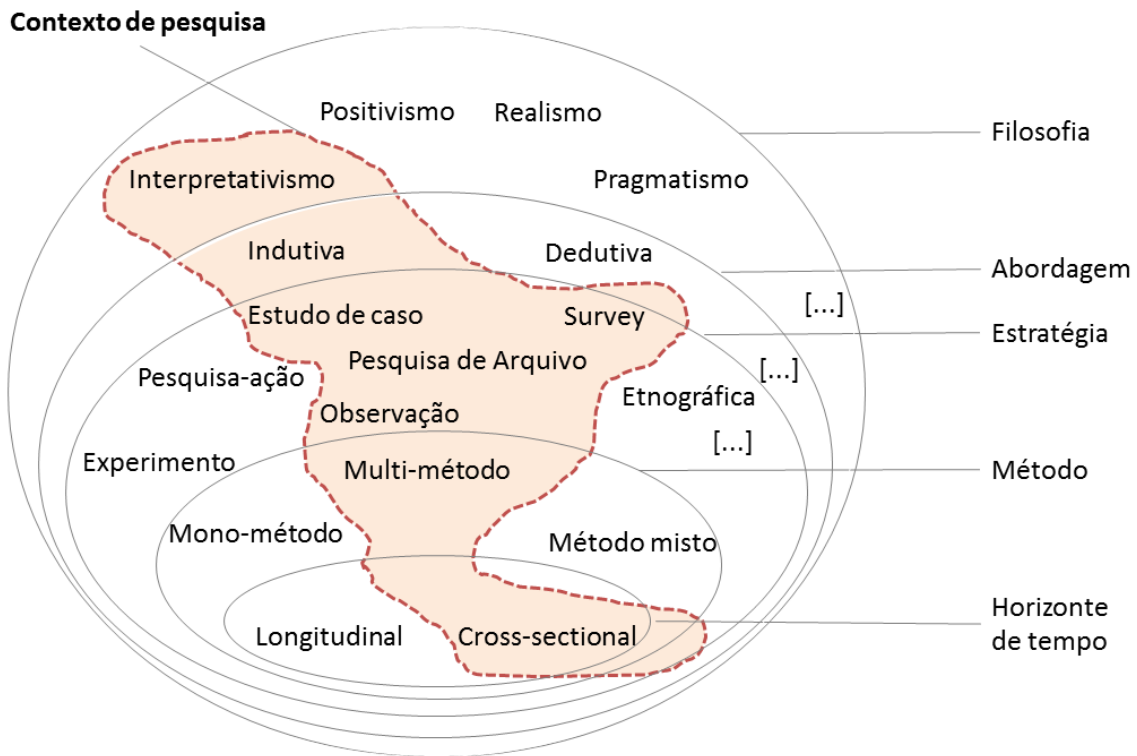


Figura 1. Contexto da pesquisa (SAUDERS, LEWIS, & THORNHILL, 2009)

- Estratégias: Para realizar esta pesquisa, são utilizadas diversas estratégias, tais como, estudos de caso (YIN, 2014), *survey* (KASUNIC, 39 2005), observação, e pesquisa de arquivo;
- Indutiva: pois são analisados estudos de caso particulares de aplicação da UI para inferir as conclusões gerais e para se basear na solução do problema.
- Interpretativista: é interpretativista porque para atingir o objetivo desta pesquisa é preciso interpretar as informações coletadas durante a aplicação da UI.

A Tabela 1 apresenta a classificação da pesquisa realizada neste trabalho.

Critério	Classificação	Justificativa
Natureza	Aplicada	A UI desenvolvida neste trabalho poderá ser aplicada por tutores que ensinam computação no Ensino Básico.
Objetivos	Exploratória	Para atingir o objetivo deste trabalho é preciso analisar o estado da arte (OE2 – RSL); levantamento do estado da prática nas escolas (OE2 – <i>survey</i>).
Abordagem	Qualitativa e Quantitativa	Pertinente ao OE2 (RSL/survey) e OE4 (estudo de caso) que incluem análise de dados coletados de forma qualitativa e quantitativa (YIN, 2014).
Procedimentos	Bibliográfica, Documental, DI, Estudo de caso	É utilizado diversos métodos para atingir o objetivo deste trabalho, como: Bibliográfica (OE1 – Fundamentação teórica), Documental (OE2 – RSL), normas (OE3.1), estudo de caso (OE4), etc.

Tabela 1. Classificação da pesquisa

1.5.2. Etapas da pesquisa

Este trabalho separa em 4 etapas de pesquisa para o desenvolvimento, aplicação e avaliação da UI, conforme ilustra a Figura 2.

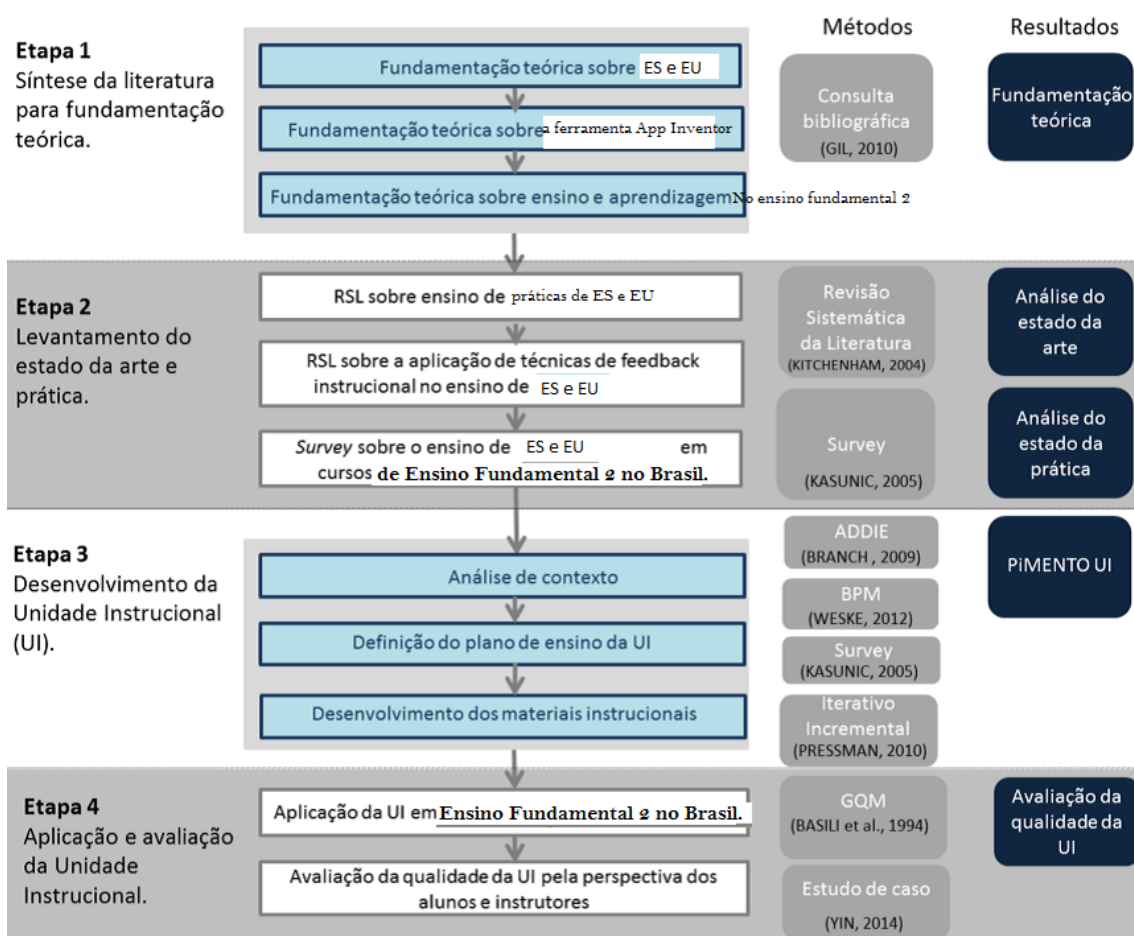


Figura 2. Etapas da pesquisa

O detalhamento do trabalho realizado em cada etapa é detalhado a seguir:

Etapas 1 - Síntese da literatura para fundamentação teórica

O objetivo desta etapa é fazer um estudo das literaturas que abordam atividades das disciplinas de ES e EU. Também é feita uma análise teórica sobre computação móvel e de programação de aplicativo por meio da ferramenta *App Inventor*. Após, é feito um estudo teórico sobre o Ensino Básico e do currículo de referência CSTA K-12, que serão integrados para introduzir a computação no Ensino Fundamental 2.

Etapas 2 - Levantamento do estado da arte e prática

Nesta etapa, será feito o levantamento do estado da arte para entender como o ensino da computação vem sendo realizado e avaliado nas escolas. Esta etapa é dividida em duas partes: 1) a Revisão Sistemática da Literatura (RSL) identifica e analisa as principais unidades instrucionais/estratégias de ensino da computação na Educação Básica. Este trabalho segue o processo de RSL de Kitchenham (KITCHENHAM, 2004), no qual é feito o planejamento da RSL, realizando o levantamento das palavras-chaves do estudo, definição das perguntas de pesquisa e fonte de estudo. Após é realizada a execução de busca dos trabalhos, extração e análise das informações dos assuntos em questão, e 2) O *Survey*, no qual é feito o levantamento de como o ensino da computação está sendo realizado no Ensino Básico. Seguindo o processo de Kasunic (KASUNIC, 2005), o *survey* também tem a etapa de planejamento, no qual é definido o objetivo de pesquisa, o público-alvo, planejamento da amostra, definição das questões de análise métricas, projeto do questionário, e teste piloto do questionário. Por fim, é feita a distribuição do questionário e a coleta de dados, no qual vai servir de base para interpretação do resultado.

Etapas 3 - Desenvolvimento da UI

Para desenvolvimento da UI foi utilizado o método de *design* instrucional abordado por ADDIE (BRANCH, 2009). Nesta abordagem, são definidas fases que por sua vez contêm as suas atividades pertinentes. Segue as fases abaixo:

1. Análise: é identificado e caracterizado o público-alvo após coletar as informações de um *survey* sobre o ensino da computação no Ensino Básico.

2. Projeto: Esta fase realiza o projeto da UI, no qual é explicitado as técnicas de ensino utilizadas e definir e sequenciar o conteúdo da UI. O conteúdo deve estar conforme o que o currículo CSTA K-12 necessita em relação ao ensino de ES e EU para o ensino da computação no Ensino Básico. Nesta fase também serão definido a estratégias de *feedback* instrucionais, e como ocorrerá a avaliação da UI. Por fim, é feito um levantamento dos recursos necessário para o processo de aprendizagem.
3. Desenvolvimento: nesta fase são criados os materiais instrucionais, que inclui a elaboração de um processo de desenvolvimento de um *app* integrando práticas de ES e EU. Essas práticas serão baseadas nos levantamentos e classificação das técnicas/métodos de ES e EU com base no grupo de processos das normas ABNT NBR ISO/IEC 12207:2009 e ABNT NBR ISO/IEC 9241-11, respectivamente. Por fim, é feito a implementação da adaptação/evolução da ferramenta *App Inventor* para apoiar o ensino do processo de desenvolvimento de *software*.

Etapa 4 - Aplicação e avaliação da UI

Nesta etapa a UI desenvolvida é colocada em prática e avaliada em uma escola que oferece o Ensino Fundamental 2, que servirá como estudo de caso. Primeiramente é definido e planejado a sua aplicação e avaliação. Feito isso, é realizado a aplicação conforme definido no plano de aula, mas antes é preciso verificar se o ambiente físico e computacional está adequado para o ensino. Depois, ocorre a coletado dos dados para que a aplicação do ensino possa ser analisado. Por fim, a qualidade da UI é avaliado por meio de instrumentos de medições do modelo de avaliação dTECT (WANGENHEIM et al., 2017).

1.6 CONTRIBUIÇÕES

A realização desta dissertação produz contribuições nos âmbitos científico, tecnológico e social.

1.6.1 Contribuição no âmbito científico

Este trabalho tem como principal contribuição científica a elaboração de um modelo de ensino de Engenharia de *Software* e Engenharia de Usabilidade para o

Ensino Fundamental 2. Este modelo possibilitará os alunos a aplicar os conceitos dessas disciplinas na construção de *apps* confiáveis e fáceis de usar.

Dentre outras contribuições, está o levantamento do estado da arte atual em relação aos trabalhos que já apresentaram propostas de desenvolver uma UI para o ensino da computação no ensino básico. Analisando estas unidades instrucionais, observa-se que todos focam em ensinar os alunos a programar, não inserindo nos seus objetivos ensinar a EU e ES. Diante disso, este trabalho define de um processo de desenvolvimento de *apps* integrando práticas de ES e EU customizado ao contexto do Ensino Fundamental 2, incluindo materiais instrucionais para introduzir práticas de ES e EU no ensino de computação no nível de Ensino Fundamental 2 alinhando ao currículo CSTA (2016).

Por fim, com o intuito de avaliar a UI em relação à aprendizagem dos alunos, bem como sua efetividade, este trabalho contribui cientificamente aplicando e avaliando a unidade instrucional desenvolvida em escolas por meio de estudo de caso seguindo o modelo de avaliação dTECT (WANGENHEIM et al., 2017).

1.6.2 Contribuição no âmbito tecnológico

Em relação ao âmbito tecnológico a importância deste trabalho é o aprimoramento da ferramenta *App Inventor* para suportar adequadamente o processo definido e o modelo de ensino. O aprimoramento desta ferramenta permitirá introduzir a EU por meio da evolução de seus componentes visuais deixando amigáveis e com maior variedade.

1.6.3 Contribuição no âmbito social

No âmbito social as escolas de ensino básico poderão utilizar a UI desenvolvida, incluindo todo material didático e a versão aprimorada do *App Inventor*. Dessa forma, os tutores poderão ensinar, além da programação, a ES e EU, tornando o ensino da computação mais completa.

O projeto ainda visa a popularização da computação contribuindo de forma prática à sua aplicação e para o crescimento dessa área de conhecimento. Além disso, contribui com a formação da população em geral, como também estimula o interesse para a atuação nesta área.

Referências

- ABNT NBR ISO/IEC 12207:2009 - **Engenharia de sistemas e software - Processos de ciclo de vida de software**. Disponível em: <http://www.abntcatalogo.com.br/norma.aspx?ID=38643>. Acesso em: Maio 2017.
- ABNT NBR ISO/IEC 9241-11. **Requisitos ergonômicos para trabalho de escritório com computadores**: Parte 11 — Orientação sobre usabilidade. ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. Rio de Janeiro: sn, p. 21, 2002.
- BENITTI, F. B. V. et al. **Experimentação com Robótica Educativa no Ensino Médio**: ambiente, atividades e resultados. In: Anais do Workshop de Informática na Escola. p. 1811-1820, 2009.
- BLIKSTEIN, P. **O Pensamento Computacional e a Reinvenção do Computador na Educação**, 2008. Disponível em: <<http://bit.ly/1IXIbNn>>. Acesso em: Junho de 2015.
- BRANCH, R. (2009). ***Instructional design: The ADDIE approach*** (2nd ed.). USA: Springer.
- CARDOSO, E.; DE DAVID, T. **A falta de profissionais de tecnologia de informação no mercado de trabalho**. Uma Nova Pedagogia para a Sociedade Futura, p. 697-700, 2017.
- CHAO, G. Human-computer interaction: **The usability test methods and design principles in the human-computer interface design**. In: Computer Science and Information Technology. Beijing: China. 2009. p. 283 – 285
- CSTA, 2016. **CSTA K–12 Computer Science Standards**. The CSTA Standards Task Force - Revised, ACM, New York/USA.
- DA SILVA, André Constantino et al. **Aplicabilidade de Padrões de Engenharia de Software e de IHC no Desenvolvimento de Sistemas Interativos**. In: IV Congresso Brasileiro de Computação-CBComp. 2004. p. 118-123.
- GONÇALVES, Rafael Queiroz. **Ensino de Gerenciamento de projetos de software mediado por ferramentas**. 2017. p. 28. Tese (Doutorado em Ciência da Computação) - Universidade Federal de Santa Catarina, Florianópolis, 2017.

HILL, H., ROWAN, D., & BALL, D. (2005). ***Effects of teachers' mathematical knowledge for teaching on student achievement***. American Educational Research Journal, 42(2), pp. 371-406.

IEEE CS, 2014. **SWEBOK - guide to the Software Engineering body of knowledge** (3th ed.). USA: IEEE

KASUNIC, M. (2005). ***Designing an effective survey***. SEI. Hanscom AFB/ISA: Carnegie Mellon.

KITCHENHAM, B. (2004). ***Procedures for performing systematic reviews***. UK: Keele University.

LIANG L.; DENG X.; WANG Y. **Usability Evaluation Driven by Cooperative Software Description Framework**. In: Computational Sciences and Optimization. 2009. p. 364 – 366.

MIT, *APP INVENTOR*. Disponível em: <<http://appinventor.mit.edu/explore/about-us.html>>. Acesso em: 20 de abril de 2017.

MIT. **Tutorials for App Inventor**. Disponível em: <<http://appinventor.mit.edu/explore/ai2/tutorials.html>>. Acesso em: Maio/2017

PREECE, J. et al. ***Design de interação: além da interação homem-computador***. Porto Alegre: Bookman, 2005.

ROOIJ, S. (2 de 2011). ***Instructional design and project management: complementary or divergent?*** Educational Technology Research and Development, 59(1), pp. 139-158.

SBC, 2017. **Plano de Gestão para a SBC Biênio Agosto 2015 – Julho 2017**. Disponível em: <<http://www.sbc.org.br/documentos-da-sbc/send/135-eleicoes/999-plano-de-gestao-para-a-sbc-bienio-agosto-2015-julho-2017>> Acesso em: Maio/2017.

SBC, 2005. **Currículo de Referência da SBC para Cursos de Graduação em Bacharelado em Ciência da Computação e Engenharia de Computação**. Sociedade Brasileira de Computação.

TELECO, 2016. **Perfil dos Usuários de Celular**. Disponível em: http://www.teleco.com.br/ncel_usu.asp. Acesso em Junho de 2017

WILSON, C. **Hour of code---a record year for computer science**. ACM Inroads, v. 6, n. 1, p. 22-22, 2015.

WANGENHEIM, C. G. V. et al. **dTECT: Um Modelo para a Avaliação de Unidades Instrucionais para o Ensino de Computação na Educação Básica**. INCoD/GQS.02.2017.P (May/2017).

WOLBER, D. **AppInventor.org**. 2012. Disponível em: <<http://www.appinventor.org/course-in-a-box>>. Acesso em 25 jun. 2015.

YIN, R. (2014). **Case study research: design and methods** (5th ed.). SAGE Publications.