PLANO DE TRABALHO PARA O MESTRADO

DESENVOLVIMENTO DE UM MODELO INSTRUCIONAL PARA O ENSINO DE ENGENHARIA DE *SOFTWARE* E USABILIDADE

Universidade Federal de Santa Catarina – UFSC

Programa de Pós-Graduação em Ciência da Computação – PPGCC

Professora Orientadora: Christiane A. Gresse von Wangenheim

1. Introdução

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. Dessa forma, é essencial que os profissionais do século XXI, independentemente da sua área de conhecimento, tenham uma compreensão dos princípios da Computação. O pensamento computacional, por exemplo, auxilia a análise e desenvolvimento de soluções para os problemas que podem ser resolvidos computacionalmente (CSTA, 2016). Além disso, no Brasil, há uma necessidade de formar profissionais para o setor de *software*, pois existe um déficit na área (CARDOSO et al, 2017).

Com esta visão existe a tendencia de começar ensinar computação já no ensino Basico. Mais uma frase e ref sobre isto? Para este fim existem diversos unidade instrucional, por exemplo por meio de programação de jogos ou animações com Scratch REF , exercícios de programação com blockly (REF:Hour of code) ou robótica REF. . uma alternativa é ensinar o desenvolvimento de apps em celulares utilizando App Inventor (MIT, ano?). Neste contexto já existem alguns tutoriais (MIT REF) 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).

Porem para ensinar computação de forma mais completa (algo deste tipo?) é importante também ensinar competências de ES, como: atividades de análise de requisitos e testes de *software* (CSTA,2016), como tambem competências

relacionada a EU, como por exemplo, a análise de contexto, prototipação de telas e a realização de testes de usabilidade (CSTA, 2016) (PREECE et al., 2005). A integração destes conceitos no ensino de computação mesmo no nível do Ensino básico é essencial para assegurar o desenvolvimento de aplicativos com confiabilidade e usabilidade, fatores determinantes de sucesso dos apps.

Porém atualmente o ensino de computação no nível de Ensino Basico foca muito na parte da programação. O ensino de conceitos de ES e/ou EU se restringe ao ensino superior. Desta forma faltam? Ainda unidades instrucionais que sistematicamente integram estes conceitos de forma apropriado no ensino básico.

1. Objetivos

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.

O objetivo desse trabalho está inserido na linha de pesquisa de Engenharia de *Software* do PPGCC dentro dos tópicos de Processo de Desenvolvimento de *Software* conforme a definição da área de Engenharia de *Software* da SBC (SBC, 2005) e alinhado a norma ABNT NBR ISO/IEC 12207:2009.

Objetivos específicos:

O1. Analise 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.

O2. 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.

O3. Evoluçao de uma unidade instrucional para o ensino de desenvolvimento de aplicativos integrando o ensino de ES e EU.

O3.1. Definiçao de um processo de ES e EU voltado ao desenvolvimento de aplicativos no contexto do ensino fundamental 2.

O3.2. Desenvolvimento de material didático, como por exemplo, slides, roteiros, folhas de tarefas, avaliações.

O3.3. Adaptação/evolução do App Inventor para apoiar o ensino do processo de desenvolvimento de *software*.

04. Aplicaçao e avaliacao da unidade instrucional desenvolvida em escolas avaliando a unidade em relação à aprendizagem dos alunos, bem como sua efetividade, conforme o modelo de avaliação dETECT (WANGENHEIM et al., 2017).

3. Contribuições científicas potenciais.

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. Além disso, prevê-se as seguintes contribuições científicas:

- Levantamento do estado de arte e prática de forma sistemática fornecendo uma visão geral sobre esta questão de pesquisa;

- Definição de um processo de desenvolvimento de apps integrando práticas de ES e EU customizado ao contexto do ensino fundamental;

- Desenvolvimento de uma unidade instrucional para introduzir práticas de ES e EU no ensino de computação no nível de ensino fundamental 2.

- Dados e resultados de avaliação sistemática do ensino de ES e EU por meio de estudo de caso;

Uma contribuição tecnologia do presente trabalho será oAprimoramento do ambiente de programação App Inventor para suportar adequadamente o processo definido e o modelo de ensino.

Como impacto social o presente projeto visa a popularização da computação contribuindo de forma prática à sua aplicação contribuindo para o crescimento dessa área de conhecimento. Contribuindo a formação da população em geral, como também estimular o interesse paa a atuação nesta área xxxx

**Referências (Rever acho que cortei umas fora)**

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.

BOURQUE, P.; FAIRLEY, R. E. **Guide to the *Software* Engineering Body of Knowledge**. Versão 3.0. IEEE Computer Society, 2014. Disponível em: <www.swebok.org>. Acesso em: out. 2016.

CARDOSO, E. et al. **A falta de profissionais de tecnologia de informação no mercado de trabalho**. Uma Nova Pedagogia para a Sociedade Futura, p. 697-700, 2017.

CSTA (2016). **CSTA K–12 Computer Science Standards**. The CSTA Standards Task Force - Revised 2016, ACM, New York/USA (2016).

DANIEL, G. et al. **Ensinando a Computação por meio de Programação com App Inventor**. Anais do Computer on the Beach, p. 357-365, 2017.

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

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

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

SOMMARIVA, W. **Usabilitygame**: **game simulator to support the teaching of usability**. 2012. 210 f. Dissertação (Mestrado em Computação Aplicada) - Universidade do Vale do Itajaí, São José, 2012.

SOUZA, L. S.; SPINOLA, M. de M. **Requisitos de usabilidade em projetos de interface centrado no usuário de *software* de dispositivos móveis**. Associação Brasileira de Engenharia de Produção, 2006.

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