

03

Analisando Processos de Desenvolvimento de Software Educativo a partir de uma Revisão Sistemática de Literatura

Maelson Marques de Lima

Alex Ribeiro de Lima

Alanna C. Coêlho Monteiro

Edson Holanda Cavalcante Júnior

Luciana de Queiroz Leal Gomes

Resumo

Este artigo apresenta uma revisão de literatura, realizada com base no guia de Kitchenham e Charters, seguindo-o como roteiro para condução de uma revisão sistemática da literatura (RSL). Esta pesquisa foi realizada a fim de obter informações sobre processos de desenvolvimento de software educativo, e ainda, levantar as técnicas, métodos e os processos utilizados para desenvolver software educativo. Os resultados obtidos apresentam tendências no desenvolvimento de softwares educativos, e servem como ponto de partida para o início de estudos na área e da evolução dos próprios processos, podendo ser utilizado como um referencial de estudos primários para futuras pesquisas.

Palavras-Chave: Processo de desenvolvimento de software educativo, software educacional, técnicas de desenvolvimento de software educacional.

Abstract

This paper presents a literature review, performed based on Kitchenham and Charters' guide for conducting a systematic literature review (SLR). We conducted this research in order to obtain information on educational software development processes. Also raise the techniques, methods and processes used to develop educational software. The result show trends in the development of educational software, and we can apply it as a starting point for the beginning of studies in this subject. The result of this research also can be applied as a reference of primary studies for future research.

Keywords: Educational software development process, Educational software, educational software development techniques.

1. Introdução

O processo de globalização vem tornando a tecnologia cada vez mais presente na vida das pessoas. O baixo custo e os diferentes tipos de dispositivos encontrados no mercado faz com que as pessoas de diferentes classes sociais sejam incluídas no mundo digital, antes restrito a pessoas com condições financeiras mais favoráveis.

Partindo desse ponto de vista e indo em direção à educação, é natural a introdução da tecnologia na vida acadêmica dos alunos, tanto na sala de aula como fora dela. Os professores, com o passar do tempo, também vão se tornando mais favoráveis ao seu uso. Como deve ser o uso da tecnologia em sala de aula? Como a tecnologia pode melhorar o ensino? Esses são alguns dos fatores mais estudados e mais pesquisados pelos cursos de licenciatura, e por grandes pensadores do mundo acadêmico, como José Manuel Moran (MORAN, 2001).

Com isso vem crescendo a necessidade de *softwares* educativos no processo de ensino/aprendizagem. Mas um grande desafio para o desenvolvimento desses sistemas é que os engenheiros de *softwares* devem se preocupar com o lado pedagógico e com a maneira como funcionamos processos cognitivos como elementos para a educação. É necessária uma constante interação entre pedagogos, alunos, psicólogos e desenvolvedores, na qual os engenheiros de *software* devem estar preparados e dispor de ferramentas adequadas para atender essa nova demanda.

O objetivo deste artigo é apresentar os resultados obtidos a partir de uma Revisão Sistemática de Literatura (RSL). Com a RSL foram levantados na literatura científica de conferências e revistas, quais são os métodos e as técnicas utilizadas em processos de desenvolvimento de *software* educativo, além dos próprios processos utilizados para este fim, no intuito de compreender melhor sua concepção e contribuir para sua melhoria.

2. Visão geral dos processos de desenvolvimento de *software*

Com o fácil acesso à tecnologia e a diversidade de dispositivos tecnológicos que se tem nos dias atuais, além do crescente número de usuários, os problemas em relação ao desenvolvimento de *softwares* para suprir essas novas ferramentas também começam a aparecer, especialmente quando se tratam de tecnologias educacionais.

O desenvolvimento de *softwares* educacionais não é uma tarefa trivial, são diversos os fatores que devem ser levados em consideração ao produzi-los, já que esses estão diretamente ligados ao desenvolvimento cognitivo e ao processamento de informações, necessários para o processo de ensino/aprendizagem de seus usuários. O que implica na necessidade inerente de uma equipe de desenvolvimento formada por profissionais que atuam em diferentes áreas e de técnicas sofisticadas de engenharia de *software*, condizentes com a taxonomia do *software* educacional que se quer desenvolver.

A pobre documentação que faz referências aos métodos, técnicas e ferramentas utilizadas para o desenvolvimento de *softwares* educacionais é um dos principais fatores que influenciam para a fraca padronização de modelos de processos destes.

Dois abordagens de desenvolvimento de *software* já consagradas são energicamente citadas, os Modelos Prescritivos de Processo e os Modelos Ágeis de Processo, cujas boas práticas são sugeridas pelo Manifesto para o Desenvolvimento Ágil de Software (PRESSMAN, 2006).

2.1 Processos prescritivos

Para solucionar os inúmeros problemas na construção de *software* de alta qualidade no prazo e dentro do orçamento estimado é que foram propostos os modelos prescritivos de processos, que podem ser definidos como processos que prescrevem um conjunto de elementos do processo (atividades arcabouço, ações de Engenharia de *Software*, tarefas, produtos de trabalho) e um fluxo de trabalho (PRESSMAN, 2006).

Existem diversos modelos prescritivos de processo (Cascata, Espiral, Evolucionário, Incremental, dentre outros), cada um sugerindo um fluxo de processo diferente, mas que pode e deve ser adaptado à equipe de trabalho, à situação problema e ao próprio projeto. Em comum, todos os modelos têm o mesmo conjunto de atividades genéricas de arcabouço: concepção, planejamento, modelagem, construção e implantação.

2.2 Processos ágeis

Para tentar melhorar a forma de se desenvolver *software*, aproveitando boas práticas da Engenharia de *Software*, e buscando por novas formas de desenvolvimento que se adaptassem melhor às mudanças, foi elaborado o manifesto para

Desenvolvimento Ágil de *Software*.

A engenharia de *software* ágil sugere mudanças revolucionárias nos processos de desenvolvimento de *software*, com uma filosofia estruturada em suas diretrizes que priorizam indivíduos e interações em vez de processos e ferramentas, *software* funcionando em vez de documentação abrangente, colaboração junto ao cliente em vez de negociação de contratos, respostas a modificações em vez de seguir um plano (PRESSMAN, 2006).

Dessa maneira, os processos ágeis buscam diminuir a distância entre cliente e desenvolvedor, de modo que os clientes passam a ser considerados parte da equipe de desenvolvimento e começam a dar suas contribuições desde o início do processo. A comunicação deve ser enfatizada de modo que seja ativa e contínua entre todos os membros que compõem a pequena equipe de desenvolvimento. As entregas de funcionalidades do *software* são mais rápidas e essas funcionalidades podem ser utilizadas em ambiente de produção.

A equipe de desenvolvimento precisa estar motivada e bem entrosada para que consiga sincronizar bem as tarefas a serem desenvolvidas, entregar os incrementos de *software* operacional dentro do prazo e custo especificados. Além disso, a equipe precisa ser capaz de suportar qualquer mudança que venha ocorrer seja no *software* ou na própria equipe.

3. O *software* educativo

Software educativo é definido como sendo um sistema computacional e interativo, intencionalmente concebido para facilitar a aprendizagem de conceitos específicos (GOMES; PADOVANI, 2005). Ainda pode-se dizer que um *software* tem caráter educativo quando proporciona algum tipo de aprendizado a quem o utiliza.

Um *software* voltado para a educação deve criar um ambiente propício à aprendizagem. Dessa maneira, o maior desafio no processo de desenvolvimento de um *software* educativo é transcrever as atividades educacionais de modo que a equipe de desenvolvimento possa ser guiada a ponto de ter suas metas cumpridas, e, os educadores, suas necessidades e ideias atendidas.

Outro problema quando se fala em desenvolvimento de *softwares* educativos é que tanto o educador e equipe pedagógica quanto a equipe de desenvolvimento devem levar em conta os diversos processos pelos quais o aluno passa durante a formação do conhecimento.

A equipe de desenvolvimento de um *software* educativo deve dispor de métodos e técnicas para planejar um sistema que instigue a curiosidade e faça a interação entre o conteúdo que está sendo ministrado e a formação do saber. Assim, a qualidade do produto a ser desenvolvido depende diretamente das pessoas envolvidas, da forma como o planejamento será realizado, da finalidade e dos procedimentos que serão utilizados.

4. Metodologia aplicada

A revisão sistemática da literatura (RSL) descrita neste documento tem como objetivo identificar, na literatura científica de conferências e revistas, quais são as técnicas, métodos e ferramentas utilizadas em processos de desenvolvimento de *software* educativo. O objetivo é identificar e considerar a aplicação das abordagens utilizadas. Este trabalho segue como base as diretrizes específicas para RSL (KITCHENHAM; CHARTERS, 2007). O protocolo de pesquisa para a RSL descrita neste documento é resumido a seguir.

O primeiro passo é formular uma pergunta de pesquisa primária como parte do protocolo de pesquisa. O objetivo desta revisão é encontrar evidências na literatura para responder à seguinte pergunta:

Questão de Pesquisa: *Quais métodos são usados para executar um processo de desenvolvimento de software educativo?*

Da questão de pesquisa, derivamos as palavras chave **Method** = *tool, technique*; e **Educational software process development** = *educative software process development*. Finalmente, os sinônimos para as palavras-chave e termos booleanos (AND, OR) foram utilizados para formar o seguinte termo de busca: (*Method OR Process OR tool OR technique*) AND (“*educational software process development*” OR “*educative software process development*”).

As bases escolhidas para realizar buscas automáticas foram: ACM Digital Library¹ e IEEE Xplore². Buscas não automatizadas foram realizadas nas bases disponibilizadas pela Sociedade Brasileira de Computação (SBC): Revista Brasileira de Informática

¹dl.acm.org

²http://ieeexplore.ieee.org

³http://www.br-e.org/pub/index.php/rbie/issue/archive

⁴http://www.br-e.org/pub/index.php/sbie/issue/archive

na Educação (RBIE)³ e Simpósio Brasileiro de Informática na Educação (SBIE)⁴.

Os critérios utilizados para a inclusão e exclusão de estudos primários foram: não responder à questão de pesquisa, não estar escrito em Inglês ou Português, não estar disponível integralmente, estudos duplicados, e estudos não relevantes (excluídos pelo título, resumo, palavras-chave não relacionados aos objetivos desta RSL).

Os critérios utilizados para a exclusão de estudos primários foram: não responder à questão de pesquisa, não estar escrito em Inglês ou Português, não estar disponível integralmente, estudos duplicados, e estudos não relevantes (excluídos pelo título, resumo, palavras-chave não relacionados aos objetivos desta RSL).

O processo de seleção dos estudos primários teve como primeiro passo a leitura de títulos, resumos e palavras-chave, excluindo estudos não relevantes. A segunda parte da seleção foi realizada com base na leitura dos resumos, introduções e conclusões dos estudos selecionados no primeiro passo. Finalmente, uma lista com os estudos relevantes para a revisão foi criada.

5. Evolução dos processos de desenvolvimento de *software* educativo (1984 - 2012)

Como resultado de busca foram identificados 724 estudos relevantes. Destes, foram excluídos alguns com base nos títulos, resumos e palavras-chave, resultando 47 estudos. Com a leitura dos artigos, a partir dos 47 artigos filtrados, foram obtidos 29 estudos primários relacionados a processos de desenvolvimento de *software* educativo. Doze desses estudos são produto de pesquisas realizadas no Brasil, retirados dos anais do SBIE e da RBIE. Os resultados apresentam um número considerável de estudos realizados pelos Estados Unidos. O uso de bases nacionais (SBIE e RBIE) representa boa parte dos estudos encontrados no Brasil (11 dos 12).

A Tabela 1 apresenta a relação de estudos primários levantados e o que foi obtido em cada estudo.

Tabela 1 Relação dos estudos primários selecionados

ID	Referência	Fonte	País	O que foi obtido
ID01	(WONG, 1993)	IEEE	China	Uso de prototipagem para produção de <i>software</i> educativo.
ID02	(LUDI, 1996)	IEEE	EUA	Estudo conduzido com crianças para identificar aspectos relacionados ao desenvolvimento de <i>software</i> educativo.
ID03	(BEYNON, 1997)	IEEE	Reino Unido	Apresenta a Modelagem Empírica como abordagem para desenvolver <i>software</i> educativo.
ID04	(CAVALCANTE; DOMINGUES, 1998)	RBIE	Brasil	Metodologia tradicional com processos prescritivos de desenvolvimento de <i>software</i> educativo.
ID05	(SQUIRES, 1999)	IEEE	Reino Unido	Design volátil (muda com o contexto de uso) de interface de <i>software</i> educativo construtivista.
ID06	(BANDOHO <i>et. al.</i> , 2000)	IEEE	Japão	Técnicas de design de <i>software</i> educativo que permitem a sincronização e participação do aluno e do professor.
ID07	(BATANOV; DIMMITT; CHOOKITTIKUL, 2000)	IEEE	Tailândia	Utilização do modelo Q&A juntamente com um <i>framework</i> específico para o desenvolvimento de <i>software</i> educativo.
ID09	(LAGE; ZUBENKO; CATALDI, 2001)	IEEE	Argentina	Design, desenvolvimento, avaliação e suporte de softwares educativos com a participação de professores no processo.
ID10	(PIZZOL; ZANATTA, 2001)	SBIE	Brasil	Design de Interface com uso de <i>storyboards</i> na construção de jogos RPG educativos.
ID11	(POLLARD; DUKE, 2002)	IEEE	Austrália	Design e modelagem dos processos para facilitar o desenvolvimento de <i>software</i> de matemática.
ID12	(BATTAIOLA <i>et. al.</i> , 2002)	SBIE	Brasil	Design de interfaces para jogos educativos.

ID13	(GOMES; WANDERLEY, 2003)	SBIE	Brasil	Processo integrado e sistemático de elicitação de requisitos de <i>software</i> educativo.
ID14	(MARCZAK <i>et. al.</i> , 2003)	SBIE	Brasil	Modelagem de ambientes educacionais com agentes inteligentes.
ID15	(GIRAFFA; MARCZAK; PRIKLADNICKI, 2005)	SBIE	Brasil	Processo de desenvolvimento de <i>software</i> educativo PDS-E.
ID16	(IVAN; JOSUÉ, 2007)	IEEE	México	Descreve uma prática efetiva de desenvolvimento de <i>softwares</i> educativos voltados para instrumentação eletrônica virtual.
ID17	(RANKIN <i>et. al.</i> , 2008)	ACM	EUA	<i>Design</i> do jogo centrado no usuário.
ID18	(KAM <i>et. al.</i> , 2008)	ACM	EUA/ Índia	Práticas ágeis de <i>design</i> e prototipagem.
ID19	(PERES, MEIRA, 2008)	SBIE	Brasil	Desenvolvimento de interfaces para <i>softwares</i> educativos utilizando a noção de gênero discursivo. Apresenta práticas ágeis.
ID20	(BRAGA; QUEIROZ; GOMES, 2008)	SBIE	Brasil	<i>Design</i> de interface.
ID21	(BARBA <i>et. al.</i> , 2009)	ACM	EUA	Desenvolvimento de jogos educativos com realidade aumentada utilizando técnicas ágeis e evolucionárias de prototipagem e <i>design</i> de jogos.
ID22	(DOUCET; SRINIVASAN, 2010)	ACM	EUA	<i>Design</i> de jogos educativos que simulam ambientes reais, com base em conceitos educacionais de metáforas.
ID23	(MEERBAUM-SALANT; HAZZAN, 2010)	ACM	Israel	Processo <i>Agile Constructionist Mentoring Methodology</i> (ACMM), usado como guia de desenvolvimento de projetos de <i>softwares</i> educativos.
ID24	(COSTA, LOUREIRO; REIS, 2010)	IEEE	Portugal	<i>Design</i> de interface com base no usuário. Usa métodos ágeis com equipes multidisciplinares.

ID25	(MARÇAL <i>et. al.</i> , 2010)	SBIE	Brasil	Levantamento e análise de requisitos para o desenvolvimento de aplicativos educacionais em dispositivos móveis.
ID26	(OLIVEIRA <i>et. al.</i> , 2010)	SBIE	Brasil	Apresenta a técnica de levantamento de requisitos árvore de características.
ID27	(FIORAVANTI; NAKAGAWA; BARBOSA, 2010)	SBIE	Brasil	Apresenta uma arquitetura de referência para ambientes educacionais.
ID28	(ZHANG; LIU, 2012)	IEEE	China	Processo de desenvolvimento de <i>software</i> que combina prototipação rápida e <i>middleware</i> para aumentar a produtividade e o tempo de lançamento do produto no mercado.

As pesquisas nos engenhos de busca foram realizadas em 2011 e retomadas em 2012. Os estudos encontrados datam de 1993 a 2012, para que fosse obtida a evolução da área dentro dos tópicos pesquisados, como pode ser visto na Figura 1. O maior número de estudos foi publicado em 2010 (6 estudos), sendo seguido do ano de 2008 (4 estudos) o que pode representar um aumento de interesse no estudo de métodos e processos para desenvolvimento de *software* educativo.

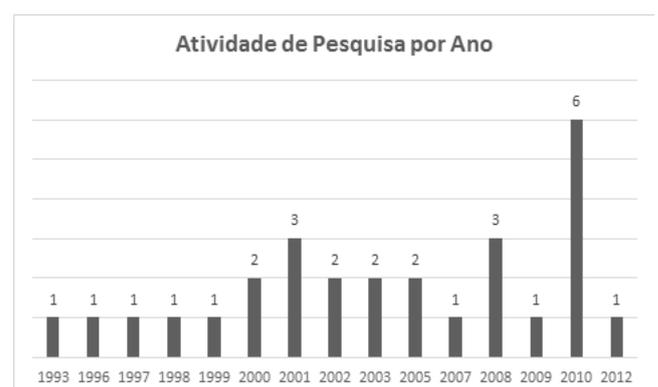


Figura 1 - Frequência de Pesquisa nos Tópicos Processo, Ferramenta, Técnica e Método Relacionada a Software Educativo Realizada por Ano (de 1993 a 2012)

A Figura 2 apresenta a quantidade de estudos por tópicos pesquisados. Esses tópicos foram derivados do termo de busca (Processo, Método, Ferramenta e Técnica). A maioria dos estudos obtidos apresenta processos de desenvolvimento de *software* que são

uma adaptação de processos já existentes. Dentre as técnicas obtidas, a maioria está relacionada ao desenvolvimento de interfaces e ao levantamento de requisitos.

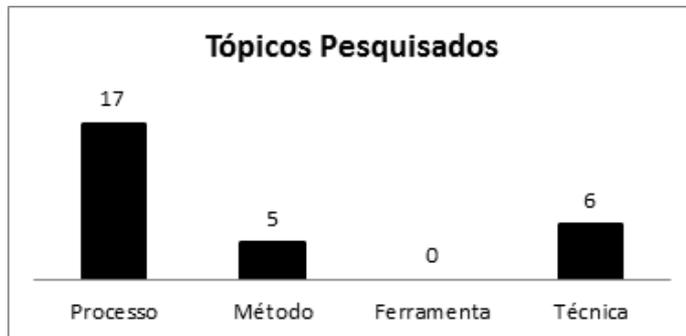


Figura 2 - Quantidade de Estudos por Tópico Pesquisado

De acordo com Pressman (2006), métodos, do ponto de vista da Engenharia de *Software*, são abordagens estruturadas para o desenvolvimento de *software*, que incluem descrições de modelo do sistema, notações, regras, recomendações e diretrizes de processo. Os métodos podem ser:

- a) **Descrições de modelos:** descrições de modelos gráficos que podem ser produzidos;
- b) **Regras:** restrições aplicadas aos modelos de sistema;
- c) **Recomendações:** conselhos para a boa prática de projeto;
- d) **Diretrizes de processo:** quais atividades devem ser seguidas.

Considerando isto, foram encontrados cinco métodos na pesquisa realizada, de acordo com a Tabela 2.

Tabela 2 Métodos Utilizados no Desenvolvimento de Software Educativo

Referência	Método	País
(FIORAVANTI; NAKAGAWA; BARBOSA, 2010)	EDUCAR, descreve o modelo de uma arquitetura de referência para ambientes educacionais (representação genérica e abstrata que ajuda a definir os elementos do sistema)	Brasil
(DOUCET; SRINIVASAN, 2010)	<i>Design</i> de jogos educativos, que simulam ambientes reais, com base em conceitos educacionais de metáforas.	EUA
(BRAGA; QUEIROZ; GOMES, 2008)	Projeto de <i>software</i> educacional seguindo como recomendação um modelo teórico que foi identificado a partir da teoria dos campos conceituais.	Brasil

(RANKIN <i>et. al.</i> , 2008)	<i>Design</i> centrado no usuário, na perspectiva da construção de jogos sérios.	EUA
(KAM <i>et. al.</i> , 2008)	Propõem o ciclo receptivo – prática – ativação, que pode ser usado como modelo conceitual para o projeto de jogos educacionais.	EUA/ Índia

Além disso, foram obtidas seis técnicas relacionadas ao desenvolvimento de *softwares* educativos. A relação entre as técnicas encontradas, seu uso e os países que as publicaram encontram-se na Tabela 3.

Tabela 3 Técnicas Utilizadas no Desenvolvimento de Software Educativo

Referência	Técnica	Uso	País
(BEYNON, 1997)	Modelagem empírica	Todo o processo	Reino Unido
(BANDOH <i>et. al.</i> , 2000)	Sincronização e participação do aluno e do professor	Projeto (<i>design</i>) e <i>design</i> de interface	Japão
(PIZZOL; ZANATTA, 2001)	RPG (<i>Braistorm</i> e <i>Storyboard</i>)	Levantamento de requisitos e interface	Brasil
(PERES; MEIRA, 2008)	Gênero discursivo	Desenvolvimento de interface	Brasil
(COSTA; LOUREIRO; REIS, 2010)	<i>Design</i> de interface com base no usuário	Desenvolvimento de interface	Portugal
(OLIVEIRA <i>et. al.</i> , 2010)	Árvore de características	Levantamento de requisitos	Brasil

Como pode ser visto pelos resultados, métodos ágeis e técnicas de construção de jogos vêm sendo utilizados para o desenvolvimento de *software* educativo. Isso pode ser explicado por algumas características de ambos serem também de *softwares* educativos (interdisciplinaridade, fator humano ressaltado, colaboração do cliente, preocupações relacionadas com interface e motivação para o uso, entre outras). Também, novas técnicas foram elaboradas, a exemplo da árvore de características (OLIVEIRA *et. al.*, 2010) e um guia fundamentado em desenvolvimento ágil para construção de *softwares* educativos (MEERBAUMSALANT; HAZZAN, 2010).

7. Tendências no desenvolvimento de software educativo

O interesse pelo desenvolvimento de *software* educacional vem aumentando com o passar dos

anos. Isto ocorre principalmente devido às inovações tecnológicas e com o amadurecimento da ideia de que computadores podem influenciar a educação de maneira positiva, quando inseridos de forma concisa com os métodos de ensino. Com a revisão realizada nesta pesquisa, notou-se o interesse significativo pelas metodologias ágeis nos processos de desenvolvimento de *softwares* educacionais, principalmente pelo fato desta considerar mutações ou evoluções constantes, o que permite a fácil adaptação aos ambientes em uso, ou seja, os próprios métodos se permitem alterar (COSTA; LOUREIRO; REIS, 2010).

As metodologias ágeis vêm ganhado credibilidade com o passar dos anos por se mostrarem seguras em suas diretrizes, pelo amadurecimento e evolução de suas técnicas e, é claro, por seus métodos conseguirem gerar bons resultados.

Nesta revisão foi observado que, dentre as fases do processo de desenvolvimento, as que mais se destacaram nos estudos encontrados foram: **levantamento de requisitos** (GOMES; WANDERLEY, 2003), (MARÇAL *et. al.*, 2010), (OLIVEIRA *et. al.*, 2010); **projeto de software** (BANDOH, 2000), (RANKIN *et. al.*, 2008), (KAM *et. al.*, 2008), (FIORAVANTI; NAKAGAWA; BARBOSA, 2010) e **design de interfaces** (SQUIRES, 1999), (LAGE; ZUBENKO; CATALDI, 2001), (IZZOL; ZANATTA, 2001), (BATTAIOLA *et. al.*, 2002), (BRAGA; QUEIROZ; GOMES, 2008), (PERES; MEIRA, 2008), (BARBA *et. al.*, 2009), (DOUCET; SRINIVASAN, 2010), (COSTA; LOUREIRO; REIS, 2010). Isto acontece pelo fato destas serem imprescindíveis para a satisfação do cliente, além de, é claro, possibilitar o devido entendimento da ideia central do *software* por parte dos desenvolvedores. Técnicas de **testes**, **prototipagem**, **modelagem**, **avaliação** e **suporte** (Kamet. *al.*, 2008), (BARBA *et. al.*, 2009), (MEERBAUM-SALANT; HAZZAN, 2010), (MARCZAK *et. al.*, 2003), (BATANOV; DOMMITT; CHOOKITTIKUL, 2000), (LAGE; ZUBENKO; CATALDI, 2001), (FIORAVANTI; NAKAGAWA; BARBOSA, 2010) também foram citadas na literatura analisada, porém com menos ênfase às características dos processos utilizados para sua concepção.

No Brasil existe o interesse por ambientes virtuais de aprendizagem, principalmente devido ao aumento significativo de projetos fomentando o ensino à distância (EAD). Isto acontece porque o governo e empresas privadas vêm buscando meios de atingir

um público alvo que necessita de maneiras especiais de lidar com o processo de ensino/aprendizagem. Logo, surgiram algumas técnicas e modelos que buscam padronizar o processo de desenvolvimento dessa área específica. É o caso de Oliveira *et. al.* (2010) e Fioravanti, Nakagawa e Barbosa (2010), que sugerem, respectivamente, boas práticas ou técnicas para levantamento de requisitos e arquitetura de referência para o desenvolvimento de ambientes educacionais.

A crescente busca por jogos educativos, que facilitam a aprendizagem do aluno, é tendência em todos os países que buscam agregar a tecnologia à educação, e isto se reflete no processo de desenvolvimento do *software* educativo (PIZZOL; ZANATTA, 2001), (BATTAIOLA *et. al.*, 2002), (BARBA *et. al.*, 2009), (DOUCET; SRINIVASAN, 2010). Segundo Braga (2000), o estudante apresenta certa distância da leitura ou do estudo, preferindo atividades esportivas diversas ou outros tipos de jogos, a passar algumas horas por dia estudando ou pesquisando um assunto. Isto é um ponto a ser considerado no desenvolvimento de *software* educativo: fornecer um *software* educacional que prenda a atenção do estudante. Este é um objetivo a ser perseguido e alcançado no desenvolvimento de *software* educacional, o de trazer um ambiente prazeroso ao estudante para que os *softwares* que mais se pareçam com jogos atraiam um público cada vez maior.

8. Conclusões e trabalhos futuros

Esta pesquisa procurou construir a linha do tempo do desenvolvimento de *software* educativo, na perspectiva dos processos de desenvolvimento de *software*. Considerando estes, foram identificados também técnicas e ferramentas que auxiliaram na busca dessas evidências na literatura.

Analisando os resultados obtidos, foi possível identificar o uso de processos e técnicas já conhecidos no desenvolvimento de *softwares* educativos. Outros também foram criados para auxiliar no desenvolvimento de *software* educativo, e são apresentados como resultados desta pesquisa.

Foi possível observar por meio desta pesquisa que técnicas aplicadas no desenvolvimento de jogos são bem aceitas e utilizadas na produção de jogos educativos. Foi realizado um mapeamento e caracterização da evolução dos processos de *software* educativo. Essa evolução apresenta trabalhos que

consideram desde apenas a importância singular do uso de um processo de desenvolvimento na atividade de criação de um *software* educativo até processos que se preocupam com a qualidade do *software* produzido e a melhoria do próprio processo de desenvolvimento. Apesar de não ter sido realizada uma validação do *string* de busca e uma preocupação maior com ameaças à validade deste trabalho, os autores consideram como maior contribuição deste a própria revisão da literatura, que apresenta a evolução da maneira de pensar e executar o desenvolvimento de *software* educativo. Espera-se que os resultados obtidos sirvam de ponto de partida para várias outras pesquisas, por meio dos estudos primários e resultados reportados. Como trabalhos futuros, sugere-se uma complementação desta revisão, com *strings* de busca mais elaborados e com uma análise mais aprofundada dos estudos primários encontrados, em direção a um processo de *software* que melhor suporte o desenvolvimento de *softwares* educativos.

Referências

- BANDOH, H.*et al.* “Development of educational software for whiteboard environment in a classroom” In International Workshop on Advanced Learning Technologies, 2000.
- BARBA, E.*et al.* “Lessons from a Class on Handheld Augmented Reality Game Design” In Proceedings of the 4th International Conference on Foundations of Digital Games, 2009.
- BATANOV, D. N.; DIMMITT, N. J.; CHOOKITTIKUL, W. “Q&A Teaching/Learning Model as a New Basis For Developing Educational Software” In 30th Annual Frontiers in Education Conference, 2000.
- BATTAIOLA, A. L.*et al.* “Desenvolvimento de um Software Educacional com base em Conceitos de Jogos de Computador”. In: Anais do XIII Simpósio Brasileiro de Informática na Educação, 2002.
- BEYNON, W. M. Empirical Modelling for Educational Technology, In: International Conference on Cognitive Technology, 1997.
- BRAGA, J. M. Aventurando pelos caminhos da leitura e escrita de jogadores de Role Playing Game (RPG). Disponível em: <www.anped.org.br/1604t.htm>. Acesso em: 15 out. 2000.
- BRAGA, M. M.; QUEIROZ, A. E. M.; GOMES, A. S. “Design de Software Educacional Baseado na Teoria dos Campos Conceituais”. In: Anais XIX do Simpósio Brasileiro de Informática na Educação, 2008.
- Cavalcante, R. J.; DOMINGUES, T. S. “Metodologia De Desenvolvimento De Software Educativo: Aplicação Em Educação Ambiental e Microinformática”. In: IV Congresso RBIE, 1998.
- COSTA, A. P.; LOUREIRO, M. J.; e REIS, L. P. “Hybrid User Centered Development Methodology”. In: 5th Iberian Conference on Information Systems and Technologies, 2010.
- Cota, M. P., DamiAn, A. R., Dacosta, J. G., e Boo, S.C. (1999) “Maja: A Way To Design Multimedia/Hipermedia Software To Teach Engineering” In Frontiers in Education Conference.
- DOUCET, L.; SRINIVASAN, V. “Designing Entertaining Educational Games Using Procedural Rhetoric: A Case Study”. In: Proceedings of the 5th ACM SIGGRAPH Symposium on Video Games, 2010.
- FIORAVANTI, M. L.; NAKAGAWA, E. Y.; BARBOSA, E. F. “EDUCAR: Uma Arquitetura de Referência para Ambientes Educacionais”. In: Anais do XXI Simpósio Brasileiro de Informática na Educação, 2010.
- GIRAFFA, L.; MARCZAK, S.; PRIKLADNICKI, R. “PDS-E: Em direção a um processo para desenvolvimento de Software Educacional”. In: Anais do XVI Simpósio Brasileiro de Informática na Educação, 2005.
- GOMES, A. S.; PADOVANI, S. Usabilidade no ciclo de desenvolvimento de software educativo. In: Anais do XVI Simpósio Brasileiro de Informática na Educação, 2005.
- GOMES, A. S.; WANDERLEY, E. G. “Elicitando requisitos em projetos de Software Educativo”. In:

- Anais XIV do Simpósio Brasileiro de Informática na Educação, 2003.
- IVAN, G.; e JOSUÉ, G. “Implementing Virtual Practices using an Alternative Methodology to Develop Educational Software”. *In: Second International Conference on Systems and Networks Communications, 2007.*
- KITCHENHAM, B.; CHARTERS, S. Guidelines for performing Systematic Literature Reviews in Software Engineering. Keele University and Durham University Joint Report, 2007.
- KLIMICK, C.; ANDRADE, F. RPG & Educação, São Paulo. Disponível em: <<http://www.akrito.com.br/educ/>>. Acesso em: 22 out. 2000.
- LAGE, F. G.; ZUBENKO, Y.; CATALDI, Z. “An Extended Methodology For Educational Software Design: Some Critical Points”. *In: 31st Annual Frontiers in Education Conference, 2001.*
- LUDI, S. Children and Educational Software: Software Design Under the Microscope. *In: IEEE International Conference on Multi Media Engineering Education, 1996.*
- MARÇAL *et al.* “Da Elicitação de Requisitos ao Desenvolvimento de Aplicações de Mobile Learning em Matemática”. *In: Anais do XXI Simpósio Brasileiro de Informática na Educação, 2010.*
- MARCZAK, S.*et al.* “Modelando um ambiente de aprendizagem na Web: a importância da formalização do processo de desenvolvimento”. *In: Anais do XIII Simpósio Brasileiro de Informática na Educação, 2003.*
- KAM, M.*et al.* “Designing E-Learning Games for Rural Children in India: A Format for Balancing Learning with Fun”. *In: Proceedings of the 7th ACM conference on Designing interactive systems, 2008.*
- MORAN, J. M. Saberes e Linguagens de educação e comunicação, editora da UFPel, Pelotas, 2001. p. 19-44.
- MEERBAUM-SALANT, O.; HAZZAN, O. “An Agile Constructionist Mentoring Methodology for Software Projects in the High School”. *In: ACM Transactions on Computing Education, 2010.*
- OLIVEIRA, C. C. *et al.* “Árvore de Características de Software Educativo: Uma Proposta para Elicitação de Requisitos pelo Usuário”. *In: Anais do XXI Simpósio Brasileiro de Informática na Educação, 2010.*
- PERES, F.; MEIRA, L. “Dialogismo: a ideia de gênero discursivo aplicada ao desenvolvimento de software educativo”. *In: Anais do XIX Simpósio Brasileiro de Informática na Educação, 2008.*
- PIZZOL, C. D.; ZANATTA, A. L. “O RPG como técnica na construção de software educacional: A Revolução Farroupilha”. *In: Anais do XII Simpósio Brasileiro de Informática na Educação, 2001.*
- POLLARD, J.; DUKE, R. “A Software Design Process to Facilitate the Teaching of Mathematics”. *In: International Conference on Computers in Education, 2002.*
- PRESSMAN, R. S. **Engenharia de Software**. 6ª edição, McGraw-Hill, 2006. 752 p.
- RANKIN, Y. A.*et al.* “User Centered Game Design: Evaluating Massive Multiplayer Online Role Playing Games for Second Language Acquisition”. *In Proceedings of the 2008 ACM SIGGRAPH Symposium on Video games, 2008.*
- RICARDO, C. “Educational Software Features: What Do Educators Want?”. *In: ProceedingsEdCompCon '84, 1984.*
- SQUIRES, D. “Educational Software and Learning: Subversive Use and Volatile Design”. *In: Proceedings of the 32nd Annual Hawaii International Conference on Systems Sciences, 1999.*
- WONG, S. “A Quick Prototyping Approach to Educational Software Development - The Hypermedia Experience”. *In: IEEE Region 10 Conference on Computer, Communication, Control and Power Engineering, 1993.*
- ZHANG, H.; LIU, H. Educational software process

improvement model and strategy. International Conference on Computer Science and Information Processing, 2012.