

CONCEPÇÃO E EVOLUÇÃO DA ENGENHARIA DIDÁTICO- INFORMÁTICA:

Uma proposta de metodologia para a produção de software educativo

Conception and evolution of Didactical-Computational Engineering:

A proposal for a methodology for the production of educational software

Ricardo Tiburcio

Doutor em Educação Matemática e Tecnológica
Prefeitura da Cidade do Recife – Pernambuco
rico.tiburcio@gmail.com
<https://orcid.org/0000-0001-8601-3993>

Franck Bellemain

Doutor em Didactique des Mathématiques
Universidade Federal de Pernambuco – UFPE
franck.bellemain@ufpe.br
<https://orcid.org/0000-0001-5358-2057>

Amanda Rodrigues

Mestre em Educação Matemática e Tecnológica
Universidade Federal de Pernambuco – UFPE
amandarodrigues.eg@hotmail.com
<https://orcid.org/0000-0002-9129-992X>

Resumo

Apresentamos neste artigo os resultados de uma pesquisa que teve como objetivo desenvolver uma metodologia para a produção de software educativo. Essa metodologia fundamenta-se nas contribuições do campo de estudos e investigações da Educação Matemática articulada aos indicativos e orientações da Engenharia de Software. Trazemos neste texto a problematização do desenvolvimento de tecnologias digitais para o ensino e a aprendizagem da Matemática e uma possível solução para questões oriundas desta problemática: a Engenharia Didático-Informática. Explicitamos aqui a origem da Engenharia Didático-Informática e como ela foi estruturada sob as óticas de alguns pesquisadores que se empenharam em compreender como o desenvolvimento de software educativo pode ser guiado, considerando contribuições sobre o ensino e a aprendizagem aliados aos avanços das tecnologias computacionais.

Palavras-Chave: Engenharia Didático-Informática. Desenvolvimento de software educativo.

Abstract

In this article we present the results of a research that aimed to develop a methodology for the production of educational software. This methodology is based on contributions from the field of

studies and investigations of Mathematical Education linked to the indications and orientations of Software Engineering. We bring in this text the problematization of the development of digital technologies for the teaching and learning of Mathematics and a possible solution to issues arising from this problem: Didactical-Computational Engineering. We explain here the origin of Didactical-Computational Engineering and how it was structured under the optics of some researchers who endeavored to understand how the development of educational software can be guided considering contributions on teaching and learning combined with advances in computational technologies.

Keywords: Didactical-Computational Engineering. Development of educational software.

Introdução

A utilização de recursos tecnológicos digitais para auxiliar o ensino e a aprendizagem de conhecimentos é uma realidade global: tutoriais, jogos, simuladores, linguagem de programação, robótica, entre outros recursos, são utilizados com frequência em ambientes de aprendizagem (virtuais ou presenciais). Entretanto, a qualidade de muitos deles é questionada devido à forma com que são produzidos (BENITTI; SCHLINDWEIN; SEARA, 2005; SANTOS, 2009), além disso, em algumas investigações (TIBURCIO, 2014, 2016, 2020; TIBURCIO, BELLEMAIN, 2016, 2018; BELLEMAIN; SILVA; TIBURCIO, 2017), foram verificadas dificuldades na produção de softwares educativos. O presente artigo traz à tona a problemática do desenvolvimento desses recursos: existe a necessidade de uma engenharia de software que dê suporte para a construção dos programas.

Durante a revisão de literatura, observou-se a necessidade de articulação entre as diversas áreas de conhecimentos envolvidas na concepção de software educativo – Cognição, Didática, Design, Epistemologia etc. - bem como a conexão entre as contribuições teóricas sobre o ensino e a aprendizagem e os referenciais da Engenharia de Software.

De acordo com Bourque e Fairley (2014), um processo de engenharia de software está preocupado com as atividades de trabalho realizadas pelos engenheiros para conceber, desenvolver, manter e operar um software, tais como levantamento de requisitos, design, construção, testes, gerenciamento de configurações etc. Um processo pode ser utilizado para a criação de tecnologias educativas, contudo, carece de especificações que a Educação exige. Nesse sentido, define-se a Engenharia de Software Educativo como o conjunto de técnicas e organização metódica com a finalidade de idealizar e desenvolver interfaces e artefatos, visando contribuir para a melhoria das relações de ensino e aprendizagem de diversas áreas do conhecimento (TIBURCIO, 2016).

No contexto educativo, consideramos que a contribuição da Engenharia Didática (ARTIGUE; 1996, 2002, 2009), para a criação e análise de sequências de ensino, configura-se como importante referencial metodológico, auxiliando professores e pesquisadores na sistematização da produção de situações de ensino para a investigação, assim como para a aprendizagem, em diversos níveis educacionais. Como metodologia de pesquisa, a Engenharia Didática tem como princípio a observação e análise de sequências de ensino, bem como a idealização e criação prévias dessas sequências para validar hipóteses, na confrontação entre análise a priori e a posteriori. Contudo, os aspectos de criação de recursos digitais, tais como softwares, programas, jogos, entre outros, perpassam conhecimentos sobre ensinar e aprender, fazendo-se necessários conhecimentos mais específicos de tecnologias digitais.

Isto posto, nesse contexto, ao se verificar a necessidade de relacionar teorias (e construtos teóricos) com os referenciais da Engenharia de Software, para desenvolver produtos tecnológicos que contemplem as minúcias das relações educativas, em pesquisa de Mestrado, Tiburcio (2016) apresentou uma proposta de metodologia de desenvolvimento de software educativo que articula a Engenharia de Software (BOURQUE; FAIRLEY, 2014; SOMMERVILLE, 2011) e a Engenharia Didática (ARTIGUE, 1996, 2002, 2009; ALMOULOU; COUTINHO, 2008; ALMOULOU; SILVA, 2012; PERRIN-GLORIAN, 2009): a Engenharia Didático-Informática – EDI (TIBURCIO, 2016; 2020; TIBURCIO; BELLEMAIN, 2018; SILVA, 2016).

A EDI emergiu num cenário de tentativas de modelização de processos de desenvolvimento de softwares educativos que considerem aspectos teóricos e práticos na produção de recursos para o ensino e a aprendizagem. O termo “Engenharia Didático-Informática” constitui-se da percepção em utilizar os procedimentos metodológicos e as reflexões teóricas das duas engenharias mencionadas. Utilizamos tal expressão para designar esse referencial, pois ele fundamenta uma engenharia de software com os contributos teóricos e metodológicos da Engenharia Didática.

A necessidade de articulação entre as engenharias fora verificada ao serem observadas algumas limitações das mesmas para o desenvolvimento de softwares educativos: as limitações da Engenharia Didática – ED são verificadas no momento em que tal metodologia não contempla, em suas contribuições, a totalidade das necessidades para o desenvolvimento de softwares educativos. A Engenharia de Software – ES, por sua vez, não contempla especificidades que os softwares educativos necessitam, quando se observa que os modelos

padronizados de desenvolvimento foram criados para produtos comerciais, bancários, domésticos etc.

Com isso, o presente texto tem por objetivo apresentar a construção da Engenharia Didático-Informática: uma proposta de engenharia de softwares educativos fundamentada em contribuições teóricas oriundas da Educação Matemática em conexão com a Engenharia de Softwares, que tem por finalidade o desenvolvimento de recursos tecnológicos respaldados em contribuições teóricas e potencialidades tecnológicas.

Indícios de como se produzir software educativo

Estudos indicam que muitos recursos tecnológicos digitais são desenvolvidos centrados nas possibilidades oriundas da tecnologia ou apenas nas teorias sobre a aprendizagem dos conhecimentos. Mesmo quando há a tentativa de articulação entre teorias educativas e possibilidades tecnológicas, ainda assim a engenharia é frágil (TIBURCIO, BELLEMAIN, RAMOS, 2015; TIBURCIO, BELLEMAIN, 2018).

Segundo Tiburcio (2016), de forma geral, os softwares educativos são desenvolvidos de três formas: 1. utilizando-se de metodologias padronizadas provenientes da Engenharia de Software (não educativos): método em cascata; Desenvolvimento Iterativo e Incremental, método de Prototipagem; método em Espiral, Metodologias Ágeis, entre outros. Nessa vertente, os interessados no desenvolvimento (equipes, professores, pesquisadores, entre outros) utilizam as metodologias citadas, sem realizar alterações, para o desenvolvimento dos recursos; 2. realizando uma adaptação de metodologias padronizadas. Nesse caso, as equipes justificam que as metodologias padronizadas auxiliam no desenvolvimento de softwares educativos, porém, reconhecem que são necessárias adaptações visto que existem diferenças entre produtos tecnológicos para fins educativos e produtos para outros fins. E, por último, 3. elaborando uma integração de áreas. Nessa vertente de desenvolvimento, as equipes observam as contribuições teóricas, as tendências de ensino e aprendizagem, o que se discute na academia, orientações de documentos oficiais e tentam conectar com as possibilidades tecnológicas atuais. Essa forma de desenvolver softwares educativos é característica de produções em ambientes de pesquisa, visto que possui caráter experimental, bem como preocupa-se com as teorias atuais sobre ensino, aprendizagem, cognição, entre outras áreas, para melhoria das relações de ensinar e de aprender.

De acordo com Tiburcio e Bellemain (2018), para desenvolver software educativo, havia uma lacuna de um referencial teórico-metodológico que aliasse as contribuições teóricas das áreas de ensino e de aprendizagem conectadas aos processos da Engenharia de Software. A percepção desses autores gerou investigações na área da Engenharia de Softwares Educativos, sendo concebida a Engenharia Didático-Informática: um aporte teórico e metodológico relevante para o desenvolvimento de recursos tecnológicos para o ensino e a aprendizagem.

A articulação entre teorias educacionais e possibilidades tecnológicas, para o desenvolvimento de software educativo, ainda não é uma realidade sistêmica e definida. Existe uma carência de uma metodologia robusta com as devidas indicações de procedimentos para criar os softwares. Assim, a Engenharia Didático-Informática vem sendo desenvolvida a fim de propiciar um processo de desenvolvimento de software educativo, que alie contribuições teóricas e potencialidades tecnológicas referentes ao ensino e a aprendizagem com tecnologias digitais. Nas reflexões deste cerne, o quadro teórico-metodológico desta investigação foi constituído pelas Engenharias Didática e de Software, observando as contribuições de ambas para a construção desse referencial.

O início da busca pela articulação

A primeira tentativa em articular a Engenharia Didática à Engenharia de Software foi a investigação de Ramos (2014): os resultados de sua pesquisa de Mestrado foram expressivos para a estruturação da EDI. O trabalho teve o objetivo de investigar os aportes dos princípios teórico-metodológicos da Engenharia Didática à Engenharia de Software para o desenvolvimento de uma versão digital de um jogo existente: o Bingo dos Números Racionais, criado no contexto do Projeto Rede (GITIRANA, et al, 2013).

O Bingo dos Números Racionais (em sua versão em papel) foi construído utilizando-se os princípios da Engenharia Didática. Ramos (2014), com o objetivo de construir uma versão digital para esse jogo. Aproveitou a Engenharia Didática já realizada na versão em papel e reformulou essa engenharia, considerando os “circunscritores informáticos” para desenvolver a versão digital. A ideia de “Engenharia Didático-Informática” foi inspirada na existência da ED do bingo, reformulando-a, considerando os aspectos tecnológicos a serem implementados para desenvolver a versão digital.

Ramos (2014) considerou como questão central de pesquisa investigar os aportes da ED à ESE no caso da criação de situações didáticas digitais em Matemática (2014, p. 23). De

importância tal que motivou o desenvolvimento de uma articulação formal entre ED e ESE, dando início ao aprofundamento da relação entre essas engenharias, culminando na concepção da EDI (TIBURCIO et al, 2015; TIBURCIO, 2016; TIBURCIO; BELLEMAIN, 2018).

A percepção de Ramos (2014) sobre a Engenharia de Softwares Educativos nos conduziu a considerar a necessidade da modelização de um processo de SE ao perceber a importância de aliar teorias educativas aos princípios de desenvolvimento de tecnologias. Em seu estudo, a autora observou, nas primeiras utilizações de computadores como ferramentas de ensino e aprendizagem, a preocupação em relacionar conhecimentos: “[...] veremos no ensino programado dos anos 1950-1960, fortes interações entre teorias educativas da época, a concepção e o desenvolvimento de tecnologias computacionais” (p. 25).

Ao final, para o sucesso de um projeto, é importante considerar as três dimensões: tecnológica, educativa e do contexto de uso. Nesse sentido, para poder contemplar todas as dimensões da concepção e desenvolvimento de um software educativo, uma solução é de reunir profissionais das diversas áreas de conhecimento envolvidas nesse processo (RAMOS, 2014, p. 29).

Analisando as articulações propostas e a fundamentação teórica de sua investigação, Ramos (2014) utiliza o que classificou como “esquema” para desenvolver o protótipo do software proposto, em que a engenharia utilizada tem fases não explicitadas e algumas características não facilitam a sua utilização por usuários que queiram desenvolver produtos de software. Considerando uma abordagem cíclica, não está tão claro onde iniciar o processo e onde concluí-lo, visto que há a necessidade de uma sequência nesse processo. A intenção de relacionar a ED com a ESE não apresentou os elementos comuns e, também, os complementares em ambas as engenharias, além disso, o desenvolvimento do software foi claramente pautado na Engenharia Didática, o que não ficou compreensível quais são as fases do esquema proposto, bem como as etapas de desenvolvimento do Bingo dos Racionais nos resultados da pesquisa, o que nos motivou a dar continuidade ao que fora sugerido por Ramos (2014).

Contudo, Ramos (2014) conclui sua investigação observando que o processo de integração entre as duas Engenharias (Didática e de Softwares) permite evitar que os softwares educativos, de forma geral, não sejam construídos sem o olhar das contribuições teóricas dentro do campo de Pesquisa da Educação Matemática e que também não sejam desconsideradas as contribuições dos estudos da Engenharia de Softwares.

Com isso, demos início à formalização da articulação da Engenharia Didático-Informática: leituras, discussões, pesquisas e sistematizações, a fim de lançar a primeira versão desse referencial de desenvolvimento de software educativo.

Fundamentos Teóricos

Para compreender a articulação entre as engenharias, faz-se necessário definir alguns conceitos provenientes da área de produção de tecnologias e da Educação Matemática. Apresentamos nos tópicos a seguir alguns conceitos que devem fundamentar uma pesquisa de desenvolvimento de software educativo.

Considerações sobre a Engenharia de Software

Bourque e Fairley (2014) definem a Engenharia de Software como a aplicação de uma abordagem sistemática, disciplinada e quantificável para o desenvolvimento, a operação e a manutenção de software. Com essa definição, compreendemos a criação de um software observando que os mesmos são desenvolvidos por etapas/fases, são processos metódicos que antecedem o trabalho da criação de fato: o planejamento, a organização, a delimitação de objetivos, como também a análise do produto final após o mesmo entrar em operação. Com isso, consideramos a Engenharia de Software como:

[...] uma disciplina de engenharia cujo foco está em todos os aspectos da produção de software, desde os estágios iniciais da especificação do sistema até sua manutenção, quando o sistema já está sendo usado [...] A engenharia de software não se preocupa apenas com os processos técnicos do desenvolvimento de software. Ela também inclui atividades, tais como: gerenciamento de projeto de software e desenvolvimento de ferramentas, métodos e teorias para apoiar a produção de software (SOMMERVILLE, 2011, p. 5).

As definições apresentadas estão vinculadas à ideia de procedimento, método ou percurso para a construção de software. Essa ideia é definida na literatura como “processo de software”, que é uma abstração, em forma sequencial, de toda a estrutura da engenharia, indicando os procedimentos, do início ao fim do projeto de desenvolvimento dos produtos. Bourque e Fairley (2014) acreditam que os processos de software devem ser especificados: para facilitar a compreensão, comunicação e coordenação humana; auxiliar o gerenciamento de projetos de software; medir e melhorar a qualidade dos produtos de software de maneira

eficiente; apoiar a melhoria e fornecer uma base para o suporte automatizado da execução dos processos.

Em resumo, os processos de software possuem quatro atividades fundamentais, de acordo com Sommerville (2011): 1. Especificação de software, em que clientes e engenheiros definem o software a ser produzido e as restrições de sua operação. 2. Desenvolvimento de software, em que o software é projetado e programado. 3. Validação de software, em que o software é verificado, para garantir que é o que o cliente quer. 4. Evolução de software, em que o software é modificado para refletir a mudança de requisitos do cliente e do mercado (p. 5-6).

Esses procedimentos para o desenvolvimento de softwares poderiam ser generalizados e utilizados para a criação de softwares educativos, contudo, devemos considerar as características específicas e particulares dos recursos tecnológicos produzidos para auxiliar o ensino e a aprendizagem de conhecimentos. De acordo com Sommerville (2011), existem diversos tipos de sistemas de software, “não faz sentido procurar notações, métodos ou técnicas universais para a engenharia de software, porque diferentes tipos de software exigem abordagens diferentes” (p. 2). Ainda, de acordo com o autor, os diversos softwares necessitam ser construídos com uma engenharia base, embora não necessitem das mesmas técnicas de produção.

Com isso, acreditamos que, para desenvolver software educativo, é necessária uma engenharia específica, visto as finalidades, características e particularidades das ações de ensinar e aprender. Desse modo, escolhemos o referencial da Engenharia Didática para auxiliar nas questões educativas da produção de recursos tecnológicos que contribuem para o ensino e a aprendizagem de conhecimentos.

A Engenharia Didática e a criação de sequências de ensino

Uma primeira investigação, que traz à tona a associação entre teorias de ensino e aprendizagem com conhecimentos da área da tecnologia, considera que, em sua época, não havia uma articulação formal fazendo com que a Engenharia de Software Educativo fosse considerada como uma disciplina, possuindo seus métodos e conceitos específicos (BELLEMAIN, 2000).

Segundo Bellemain (2000) “havia um consenso entre os profissionais das várias áreas envolvidas na criação de tecnologias para a aprendizagem, quanto à necessidade de uma interação estreita entre educação, didática, psicologia cognitiva, ciência da computação, etc”

(p. 198). Contudo, ressaltava-se a necessidade de engajar reflexões sobre a especificidade da Engenharia de Softwares Educativos com a finalidade de criar os próprios métodos e conceitos dessa área.

Ao perceber as contribuições efetivas da Transposição Didática (CHEVALLARD, 1991) para a o ensino e a aprendizagem de conhecimentos, Balacheff (1994 apud BELLEMAIN 2000) propõe uma adaptação e extensão desse referencial teórico inserindo uma dimensão informática, sendo assim classificado como Transposição Informática (BALACHEFF, 1991, 1994).

A transposição didática analisa os fenômenos de transformação do saber de referência em saber a ensinar. A introdução da dimensão informática no estudo destes processos não pode preocupar-se apenas com a encenação do saber a ensinar, uma vez que a introdução do computador participa dessa transformação do saber de referência. Portanto, consideramos primordial investigar a questão da transposição informática, não só do ponto de vista da integração das novas tecnologias no ensino (questão que já vem sendo analisada), mas também do ponto de vista da produção dessas novas tecnologias (BELLEMAIN, 2000, p. 198).

De modo análogo, observamos na Engenharia Didática Clássica (ARTIGUE, 1996, 2009) a necessidade de um aprofundamento teórico e metodológico específico para tratar das questões de desenvolvimento de tecnologias digitais para o ensino e a aprendizagem de conhecimentos. A Engenharia Didática, segundo Artigue (1996), tem por finalidade analisar e propor situações didáticas: “vista como metodologia de investigação, caracteriza-se por um esquema experimental baseado em realizações didáticas na sala de aula, isto é, na concepção, na realização, na observação e na análise de sequências de ensino” (p. 196, grifo do autor).

Com isso, percebemos a possibilidade de desenvolver softwares educativos utilizando o levantamento teórico e os procedimentos metodológicos da Engenharia Didática: adotamos como hipótese que o embasamento para construção de situações de ensino poderia ser útil para levantar os requisitos e orientar na produção dos recursos tecnológicos a serem desenvolvidos.

A criação de sequências de ensino com o referencial da Engenharia Didática Clássica é norteadas pelas seguintes fases: análises prévias - realiza-se uma análise aprofundada sobre o conhecimento alvo da sequência de ensino a ser criada; construção e análise a priori - levantam-se as hipóteses de alternativas ao que foi observado nas análises prévias, com isso a sequência é construída objetivando-se superar os problemas descritos na fase anterior;

experimentação - consiste em testar a sequência didática que foi produzida. Seja para um grupo de estudantes da Educação Básica, superior, entre outros, essa etapa consiste em pôr em prática o que foi construído; por último a análise a posteriori e a validação - têm-se a finalidade de confrontar a análise a priori e validar a sequência criada. Essa fase permite tratar os dados colhidos na experimentação verificando as contribuições da sequência aplicada. Apoiase no conjunto dos dados recolhidos durante a experimentação: observações realizadas nas sessões de ensino, mas também produções dos alunos na sala de aula ou fora dela (ARTIGUE, 1996).

As fases descritas são processos metódicos que devem ser utilizados para construir sequências de ensino. Em específico, um aspecto importante das análises prévias é a delimitação de “dimensões” em que, em nossa investigação de Mestrado (TIBURCIO, 2016) percebeu-se a carência de tratar com detalhes questões tecnológicas. As dimensões da Engenharia Didática Clássica têm por objetivo direcionar o levantamento teórico que será realizado para fundamentar a sequência didática a ser construída, são elas: dimensão epistemológica – associada às características do saber em questão, cognitiva – relacionada às características cognitivas do público ao qual se dirige o ensino; e didática – referente às características do funcionamento do sistema de ensino (ARTIGUE, 1996, p. 200).

Assim, vislumbramos a possibilidade da criação da dimensão informática, bem como toda fundamentação necessária para que o referencial teórico da Engenharia Didática Clássica pudesse ser utilizado para auxiliar o desenvolvimento de softwares educativos que considerassem teorias de ensino, aprendizagem, cognição, entre outras, em sua fundamentação.

Na sessão a seguir, apresentamos as relações estabelecidas entre a Engenharia de Software e a Engenharia Didática, dando origem a Engenharia Didático-Informática.

Articulação entre as engenharias

A justificativa da associação entre a Engenharia Didática e de Softwares ocorre ao passo em que se percebe a ausência de elementos para o desenvolvimento de produtos que contemplem necessidades do ensino e da aprendizagem da Matemática e das tecnologias (aspectos didáticos, cognitivos, epistemológicos, tecnológicos, entre outros). Observamos a necessidade de reunir os elementos pertinentes das duas engenharias: a ED, com os elementos de investigação teóricos e experimentais sobre o ensino e a aprendizagem; e a ES, com a padronização do desenvolvimento de softwares e métodos de obtenção de requisitos, com a

finalidade de construir uma metodologia de desenvolvimento de softwares educativos que observe os avanços tecnológicos, mas que não despreze os estudos teóricos realizados para o ensino e a aprendizagem dos conhecimentos.

Bellemain et al (2014) adotam como posicionamento epistemológico considerar que “a concepção e o desenvolvimento de softwares educativos exigem a mobilização de uma engenharia didática específica que deve integrar conceitos e métodos da informática” (p. 6). Acrescentam ainda que o desenvolvimento de um software educativo tem especificidades que o diferenciam de outros softwares, ratificando a necessidade de uma engenharia peculiar para construir software educativo.

Desse modo, construímos o embasamento teórico que fundamenta a Engenharia Didático-Informática, considerando os aportes da Engenharia de Software e Engenharia Didática.

Percurso Metodológico

A metodologia deste estudo compôs-se da concepção, criação e análise de uma metodologia de desenvolvimento de software educativo. Partindo da problemática de desenvolvimento de SE, observamos a necessidade de criar uma metodologia que fosse útil para desenvolver recursos tecnológicos que considerassem os avanços digitais bem como as teorias de ensino e aprendizagem.

O estudo tomou como hipótese que a articulação entre os conhecimentos da Educação Matemática, por meio da Engenharia Didática, e a Engenharia de Softwares pode fornecer produtos tecnológicos que atendam as especificidades dos conhecimentos a serem ensinados e aprendidos. A revisão de literatura constatou que havia uma carência de uma metodologia padronizada para desenvolver recursos tecnológicos com as especificações educativas. Com isso, demos início a articulação teórica, levantando elementos indispensáveis para desenvolver esses recursos.

Por conseguinte, percebemos a necessidade de articular diversos conhecimentos e esses foram viabilizados com a montagem de uma equipe classificada de pluridisciplinar – no sentido de envolver profissionais de diversas áreas para colaborar com a criação dessa metodologia de desenvolvimento de software. Após a versão inicial da modelização ter sido idealizada e criada iniciamos um estudo de caso para que a Engenharia Didático-Informática pudesse ser analisada em uso.

É válido ressaltar o caráter colaborativo do estudo de caso realizado visto que foi possível colocar em prática nossa estrutura de desenvolvimento de software ainda quando a mesma estava sendo criada. Em parceria com Silva (2016), que possuía foco no estudo do desenvolvimento de um recurso tecnológico para o ensino e a aprendizagem de funções matemáticas, criamos situações de colaboração em que uma pesquisa utilizava os resultados imediatos da outra, funcionando como uma retroalimentação.

Silva (2016) pretendia criar um software para ser utilizado com estudantes de cursos de Cálculo, especificamente no conteúdo de taxa de variação, com a finalidade de verificar as contribuições do construto tecnológico para a aprendizagem dos conhecimentos em questão. O autor tinha por objetivo geral “a prototipação e validação de um software para a abordagem da taxa de variação de funções, orientado pelo quadro teórico-metodológico da Engenharia Didático-Informática” (p. 16). Com isso, além de contribuir com o desenvolvimento do modelo de processo baseado na EDI, o pesquisador utilizou os resultados iniciais da construção da EDI proporcionando a possibilidade de retornos imediatos.

Desse modo, ao percebermos convergências entre os objetivos das duas pesquisas: Tiburcio (2016) com a finalidade de desenvolver uma metodologia de desenvolvimento de software educativo para conhecimentos matemáticos e Silva (2016) com a intenção de conceber um software para a aprendizagem de funções, foi possível direcionar os estudos para um ponto comum. Criando interações, a fim de verificar as possibilidades de relacionar os objetivos das pesquisas, inúmeras reuniões (online e presenciais), formulários, entrevistas, questionamentos etc., chegamos à percepção de que era possível obter resultados com a associação das pesquisas.

Foram criadas situações com as quais fossem possíveis promover as articulações necessárias para as investigações. Definimos a necessidade de reuniões periódicas, utilizamos ferramentas colaborativas de edição de textos, planilhas, desenhos etc. (viabilizadas pelo Google Drive) para, em um momento inicial, compreender a proposta do software que se pretendia desenvolver. Enquanto Tiburcio (2016) fornecia subsídios para o desenvolvimento do software pretendido, obtendo da outra investigação os requisitos iniciais, as situações de uso, entre outras situações; Silva (2016) utilizava a estrutura da Engenharia Didático-Informática (ainda em formação) para idealizar e desenvolver o produto de sua pesquisa.

Dessa forma, o software Function Studium (SILVA, 2016) foi desenvolvido utilizando as premissas da Engenharia Didático-Informática sendo percebida a importância desse referencial para a construção de softwares educativos. Ao apresentar o Function Studium e

suas principais características, definimos o software e o seu desenvolvimento da seguinte forma:

Para o desenvolvimento do Function Studium, utilizamos uma metodologia que integra princípios teórico-metodológicos da Didática da Matemática à Engenharia de Software, por meio da integração das primeiras etapas da Engenharia Didática [Artigue 1996] à Engenharia de Software. O Processo de Desenvolvimento de Softwares Educativos observa os métodos da engenharia de requisitos na Engenharia de Softwares integrados com uma Engenharia Didática, podendo ser definida como uma Engenharia Didático-Informática, na qual contemplamos especificamente as potencialidades teóricas (do ensino e aprendizagem de conhecimentos) e tecnológicas (da computação) (SILVA, GITIRANA, TIBURCIO e BELLEMAIN, 2017).

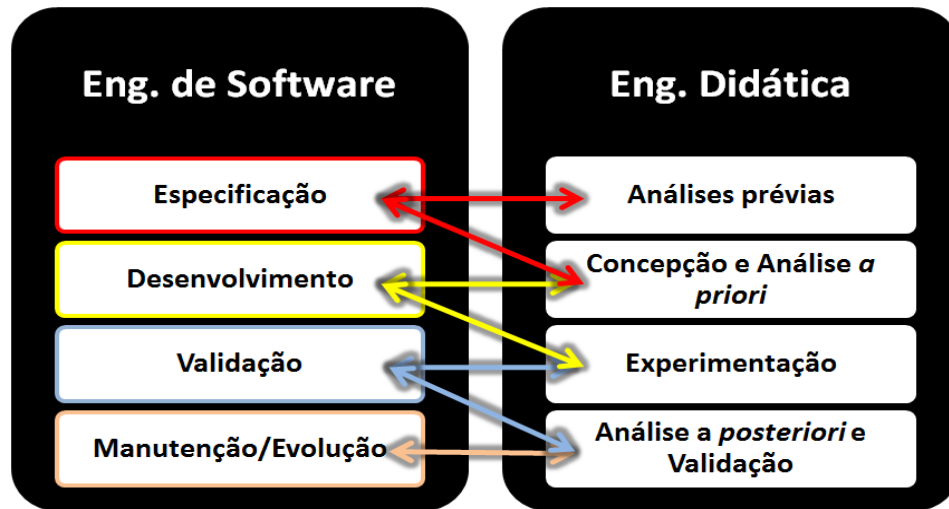
Com a análise dos elementos levantados na experiência de Silva (2016), lançamos a primeira versão do modelo de processo de software da Engenharia Didático-Informática, ao mesmo tempo em que foi possível perceber solicitações de evolução do modelo, o que justificou a continuidade da investigação.

Resultados do Estudo

Os resultados deste estudo mostram que a integração de conhecimentos teóricos sobre o ensino e a aprendizagem de conhecimentos matemáticos com as potencialidades tecnológicas atuais é um fator a ser considerado para o desenvolvimento de softwares educativos, visto que o produto desenvolvido com essa perspectiva de articulação pode contribuir para especificação e implementação dos requisitos, alcançando as especificidades dos conhecimentos em questão.

Nessa perspectiva, a conexão idealizada entre a Engenharia Didática Clássica (ARTIGUE, 1996, 2009) e a Engenharia de Softwares (SOMMERVILLE, 2011; BOURQUE E FAIRLEY, 2014) foi realizada. A Figura 1, a seguir, apresenta a articulação entre as engenharias demonstrando a aproximação de cada uma das fases e como as conectamos.

Figura 1 – Comparativo entre as engenharias



Fonte: Autoria própria, adaptado de Tiburcio (2016, p. 46)

Os processos de desenvolvimento de softwares, de modo resumido, podem ser designados nas quatro etapas ilustradas na primeira parte da Figura 1: 1. Especificação, 2. Desenvolvimento, 3. Validação e, 4. Manutenção/Evolução. As etapas definidas na Engenharia Didática Clássica são: Análises prévias, Concepção e Análise a priori, Experimentação e Análise a posteriori e Validação. As etapas da Engenharia de Software servem de fundamentação metodológica para desenvolver os produtos, enquanto as etapas da Engenharia Didática proporcionam reflexões sobre o ensino e a aprendizagem com o recurso tecnológico em desenvolvimento. Eis a proposta de complementação entre as engenharias (TIBURCIO, 2016).

As relações apresentadas na Figura 1, entre a Engenharia de Software e a Engenharia Didática Clássica, foram definidas e apresentadas nos resultados de pesquisa de Mestrado. As relações entre as engenharias são observadas por etapas: a especificação é o momento em que se define a tipologia do software, os requisitos e as características, observando que a metodologia da Engenharia Didática tem por objetivo delimitar variáveis para a compreensão das dificuldades dos estudantes, dos professores, as características dos conhecimentos, entre outras, a serem trabalhadas na sequência didática, chamados de circunscritores;

A etapa de desenvolvimento considera o levantamento teórico realizado na Concepção e Análise a priori e, nesse momento, é iniciado o processo de experimentação do software: os requisitos levantados, tanto teóricos quanto tecnológicos, são organizados e articulados para que sejam postos em funcionamento no software em desenvolvimento. A etapa de validação

relaciona-se com a experimentação da ED e com a última fase, a análise a posteriori e validação. Na ED, a validação das sequências didáticas consiste em confrontar as hipóteses levantadas na análise a priori com a experimentação da sequência.

Por último, a etapa de manutenção/evolução relaciona-se com a análise a posteriori e validação, no momento em que a análise comparativa fornecida pela ED traz à tona elementos que servem para o aperfeiçoamento e evolução do software desenvolvido. Nesta fase, incluem-se os testes, a criação de situações de utilização, bem como a verificação se o software está em pleno funcionamento. São corrigidos erros de funcionamento e se todas as características idealizadas na construção estão em plena performance.

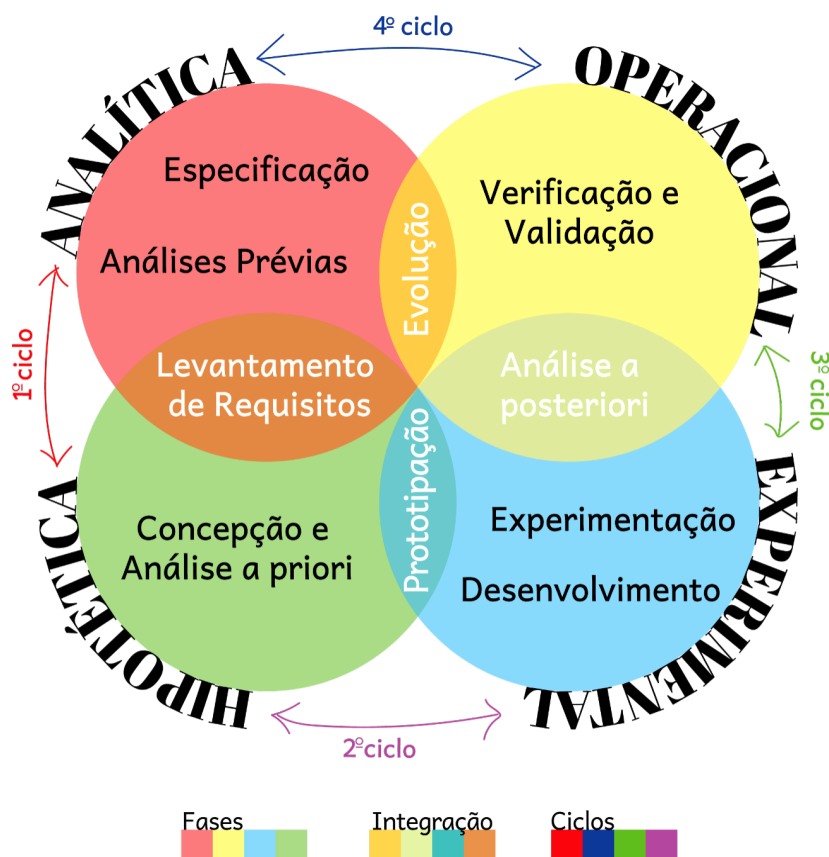
Com o exposto, o modelo de processo da Engenharia Didático-Informática foi criado, apresentando os processos, as etapas, e cada fase para desenvolver os softwares educativos. Com a evolução da pesquisa, o modelo criado foi aperfeiçoado e o apresentamos na sessão seguinte.

Premissas da engenharia didático-informática

A EDI possui um modelo de processo de desenvolvimento de software que orienta os desenvolvedores e interessados nos projetos em como proceder, de forma metódica, para criar tecnologias digitais educativas. Um modelo de processo é uma descrição abstrata em forma de diagrama, texto, ou imagem, em que se exibem as etapas para o desenvolvimento.

O modelo da EDI é composto por quatro fases e quatro ciclos. As fases são: analítica, hipotética, experimental e operacional, e os ciclos são formados pela união das fases agrupadas. De acordo com Tiburcio (2020), cada ciclo reflete um avanço no processo, porém, de forma simultânea, as fases são consideradas e revisitadas. Os ciclos são: analítico-hipotético; hipotético-experimental; experimental-operacional; operacional-analítico, conforme pode ser observado na Figura a seguir,

Figura 2 – Modelo de Processo de Software: Engenharia Didático-Informática



Fonte: Tiburcio (2020, p. 168)

Descrevemos aqui, em síntese, como interpretar e utilizar o modelo de processo da Engenharia Didático-Informática, suas fases, etapas e encaminhamentos. Para fins de compreensão, chamaremos de “líderes do projeto” os responsáveis e interessados por desenvolver software educativo considerando essa metodologia.

Ciclo analítico-hipotético

Realiza-se nesta etapa a especificação inicial – neste momento, delimitam-se os problemas existentes que o software a ser desenvolvido auxiliará em como solucioná-los. Os problemas podem ser diversos: “dificuldades de aprendizagem; ensino pautado em situações que não contribuem para a compreensão do conhecimento; percepção da contribuição tecnológica para auxiliar na aprendizagem e ou ensino, entre outros” (TIBURCIO, 2020, p. 170). Neste momento também se definem os potenciais usuários e os conhecimentos abordados.

O próximo momento é a composição da equipe. A EDI, em sua fundamentação teórica, orienta que a equipe seja transdisciplinar para que novos conhecimentos sejam gerados com as experiências e saberes dos membros que a compõe, e que também sejam inseridos nas equipes os potenciais usuários, visto as contribuições dos mesmos para o processo de idealização e criação.

Além disso, alguns profissionais são indispensáveis: um pesquisador/professor que atue na área de conhecimentos que o software abordará; e um profissional da Ciência da Computação (engenheiro de software/programador/outros) para viabilizar a transposição dos requisitos para os meios tecnológicos. Profissionais como designers, ilustradores, psicólogos e outros que podem contribuir para a criação de software também são importantes, contudo, visto o cenário de recursos e de viabilidade da participação, percebemos que não é algo simples envolver tantos profissionais (TIBURCIO, 2020).

Quanto à participação efetiva de profissionais de diversas áreas, é possível utilizar a EDI reduzindo a complexidade da concepção do software, focando em elementos específicos em detrimento de outros. Considerando que o desenvolvimento de tecnologias educativas são, em sua grande maioria, instrumentos de pesquisa, na ausência de profissionais de design, por exemplo, é possível construir a tecnologia sem ter o foco no estudo das características e situações voltadas para os aspectos de design. Assim, a validação não terá ênfase em analisar características relativas a esse domínio, entretanto, questões de design deverão ser abordadas para o produto final e em estudos futuros, considerando a importância desta área para as tecnologias digitais.

Com a equipe definida, iniciam-se as análises prévias. Realiza-se, nesta etapa, um levantamento analítico com o objetivo de entender os encaminhamentos das dimensões Didática, Epistemológica, Cognitiva e Informática do conhecimento delimitado a ser trabalhado no software. Essas análises devem contemplar os resultados de pesquisas, estudos e investigações sobre os saberes que serão abordados. As análises geram uma série de elementos que se transformarão em características do software, configurando, assim, os primeiros requisitos.

Na etapa de levantamento de requisitos é observado o que fora verificado na fase analítica. A obtenção desses requisitos deve ser guiada pelas contribuições percebidas nas análises realizadas com o objetivo de expor o maior número de subsídios sobre os conhecimentos que o software a ser produzido abordará. Fazem parte da atividade de levantamento: obter, analisar, especificar e validar os requisitos. Ao final do levantamento

obtem-se o documento de requisitos do software. Além disso, considerando um processo cíclico, a Engenharia Didático-Informática orienta que o gerenciamento dos requisitos deve ser realizado durante todo o ciclo de vida do software.

Ciclo hipotético-experimental

Neste momento, idealizam-se as situações de utilização, as hipóteses de interações dos usuários com o sistema, os problemas que podem surgir na utilização e o desenvolvimento do protótipo para iniciar a fase de testes. A etapa de concepção e análise a priori tem como objetivo criar as situações de uso do software, decidir quais serão os referenciais teóricos e metodológicos que, em hipótese, auxiliarão nas práticas docentes e discentes.

Orienta-se que as situações sejam criadas ao mesmo tempo em que o protótipo é idealizado, visto que a prototipação é útil para que os profissionais de engenharia de software compreendam os requisitos e funcionalidades e os interpretem com o que se dispõe de hardware e software. As situações de utilização e o protótipo devem ser construídos com o objetivo de superar os problemas educativos que foram levantados e considerando também os dados analisados nas dimensões da EDI presentes no documento de requisitos.

Observados os diversos meios de construir um protótipo de software, a EDI orienta que a prototipação em telas é a mais indicada visto a facilidade de apresentar os requisitos e funcionalidades desejadas, considerando os referenciais estudados. Podem ser utilizados editores de textos ou imagens com a finalidade de simular o layout inicial, exibindo funcionalidades, botões, menus, entre outros.

Com o protótipo concebido, inicia-se o design e arquitetura do software, que é a parte prática: o desenvolvimento. A EDI indica que, nesta etapa, os componentes e demais características do sistema devem ser elaborados. O desenvolvimento resulta na criação dos componentes internos – especificações de sistemas operacionais, linguagem de programação, definição de hardware, entre outras, e, também, os externos – interfaces, layout, objetos, relações entre objetos, funcionalidades etc.

A etapa do desenvolvimento ocorre de forma simultânea à experimentação, ao passo que a utilização do protótipo nas situações idealizadas em fases anteriores trará subsídios para nortear as características que o produto deve conter. Desse modo, o protótipo desenvolvido deve ser posto à prova pela equipe e possíveis usuários, e as experiências de utilização devem ser registradas com o objetivo de obter elementos de análise e implementação.

Para que a experimentação ocorra de forma que sejam obtidos elementos de análise, a EDI indica que seja criado um “manual do usuário” ou algum documento (em formato de vídeo, gráfico, diagrama etc.) que possa instruir os usuários, exibindo as funcionalidades básicas e o que será utilizado na experimentação: as instruções devem ser claras e concisas para evitar incompreensões nos usuários.

Ciclo experimental-operacional

Este ciclo contempla a experimentação considerada clássica, pois o produto em desenvolvimento deve ser colocado em situações reais de utilização. Caso haja exigências de alterações, modificações e diversas outras implementações, essas devem ser realizadas ao passo que se experimenta o software. Considerando os usuários e a equipe, a EDI orienta que seja realizada uma oficina ou um curso de curta duração em que os objetivos da pesquisa associada (se houver) sejam detalhados aos participantes, bem como o que se almeja com o software. É necessário que a equipe exiba as funcionalidades básicas do produto para que essas sejam compreendidas e utilizadas.

Diretamente relacionado com a análise a posteriori, o processo de experimentação auxilia na testagem da validade das situações propostas. A equipe tem a incumbência de observar quais os auxílios do software para construir os conhecimentos esperados e, também, observar as suas limitações.

Na medida em que a experimentação ocorre, informações importantes são levantadas: os requisitos levantados são analisados – se atendem as expectativas, se auxiliam o ensino e a aprendizagem, se ajudam na compreensão dos saberes; é verificado se os recursos tecnológicos são eficazes para atender o que foi planejado; também são verificados os erros e incompreensões, as sugestões de implementações e outras inconsistências no uso. Conclui-se, assim, que as etapas de análise a posteriori e validação têm início ainda na experimentação e acontecem de forma simultânea.

Ciclo operacional-analítico

Neste ciclo, realizam-se análises conclusivas e evolutivas visto que os experimentos com o software foram realizados. A análise a posteriori consiste em confrontar as hipóteses com o que foi observado na experimentação. Faz-se necessário que se confronte o estudo teórico com os dados obtidos através da experimentação a fim de que novos requisitos e complementos sejam implementados. Essa análise deve se fundamentar na observação do

conjunto dos dados coletados durante a experimentação bem como todo o ciclo de desenvolvimento software.

A EDI orienta que a validação seja realizada de duas formas: Teórica – que consiste na verificação das teorias e das hipóteses que devem ser refutadas ou aprovadas; Experimental – deve ser observado se a utilização do software contribui de forma efetiva para solucionar os problemas elencados na fase de concepção. Ao ser realizada a validação, obtém-se características a serem modificadas no software. Essas, devem ser implementadas caracterizando a evolução do software. A evolução faz o ciclo da Engenharia Didático-Informática recomeçar, caso seja necessário, visto que existe a possibilidade de novas especificação, no sentido de considerar outros objetivos, novos membros participar da equipe, podem surgir outros requisitos e necessidades, entre outros fatores que podem fazer o processo ser iniciado novamente.

Ainda sobre a evolução, temos como uma das premissas o processo de manutenção, a finalidade dessa etapa é a de gerir o funcionamento do software com as sugestões dos usuários ou da equipe de desenvolvedores que continua estudando o software no sentido de aprimorá-lo. Assim, a equipe deve considerar os quatro tipos de manutenção: Corretiva; Adaptativa; Perfectiva e Preventiva.

A manutenção e a evolução estão relacionadas e se configuram como uma etapa contínua, visto que a utilização do software, bem como as pesquisas e as investigações de suas potencialidades, apresentará novas possibilidades de implementação. Assim, os desenvolvedores podem iniciar novamente o ciclo com novas especificações e novas análises.

Desse modo, concluímos a apresentação do modelo de processo de software educativo da Engenharia Didático-Informática apresentando suas fases e orientações de utilização.

Considerações Finais

Como objetivo geral, pretendíamos desenvolver uma metodologia para a produção de software educativo, sendo observados os métodos da Engenharia de Softwares integrados com a Engenharia Didática clássica, articulados em uma análise de potencialidades teóricas e tecnológicas. Acreditamos que esse objetivo foi alcançado, no momento em que o processo desenvolvido foi validado no estudo de caso proposto. Com o desenvolvimento do software Function Studium, conseguimos aprimorar o que fora proposto no processo de software que estava em construção chegando a uma primeira versão desse processo.

No processo de utilização do referencial da EDI (SIQUEIRA, 2019; SILVA, 2019; SILVA, 2016), situações foram observadas a fim de fazer evoluir e aprimorar a metodologia proposta. Estes estudos expuseram dificuldades de compreender cada etapa do referencial, revelando que a descrição de como realizar os procedimentos não foram tão precisas, gerando algumas dificuldades em desenvolver os produtos. Uma primeira situação que justificou a continuidade do estudo foi a explanação de forma objetiva de cada fase do modelo, bem como delimitações mais explícitas do que deve ser realizado em cada etapa, que foi considerado justificando assim os ciclos criados e as etapas apresentadas neste texto.

Outra situação considerada quanto à metodologia de desenvolvimento de software da Engenharia Didático-Informática foi a atualização dos referenciais teóricos. Discute-se atualmente a Engenharia Didática de Segunda Geração (ARTIGUE, 2009; ALMOULOU e COUTINHO, 2008; ALMOULOU e SILVA, 2012; PERRIN-GLORIAN, 2009). São novas perspectivas e preocupações com a utilização de recursos tecnológicos por professores e alunos.

Por sua vez, a Engenharia de Software (BOURQUE; FAIRLEY, 2014; SOMMERVILLE, 2011) também está passando por reformulações para atender às necessidades tecnológicas atuais. Na medida em que as tecnologias de informação e comunicação evoluem, as equipes também percebem novas técnicas e métodos para desenvolver os programas, os sites, os jogos, entre outros.

Tais atualizações, tanto da Engenharia Didática quanto da Engenharia de Software, foram realizadas e os resultados aqui apresentados demonstram que a articulação entre elas é relevante para desenvolver software educativo que contribua para o ensino e a aprendizagem.

Assim, consideramos que a Engenharia Didático-Informática se constitui de um referencial teórico-metodológico considerável para o desenvolvimento de softwares educativos, aliando aspectos teóricos sobre o ensino e a aprendizagem com as potencialidades tecnológicas oriundas da Engenharia de Softwares.

Referências

ALMOULOU, S. A.; COUTINHO, C. Q. S. Engenharia Didática: características e seus usos em trabalhos apresentados no GT-19 / ANPEd. **Revista Eletrônica de Educação Matemática**, Florianópolis, v. 3, n. 1, p. 62-77, jan. 2008. ISSN 1981-1322.

ALMOULOUD, S. A.; DA SILVA, M. J. F. Engenharia didática: evolução e diversidade
Didactic engineering: evolution and diversity. **Revista Eletrônica de Educação Matemática**, Florianópolis, v. 7, n. 2, p. 22-52, dez. 2012. ISSN 1981-1322.

ARTIGUE, M. Engenharia Didática. In: BRUN, J. **Didática das Matemáticas**. 1. ed. Lisboa: Instituto Piaget, 1996. V. 1. Cap. 4, p. 193 -217.

ARTIGUE, M. Ingénierie didactique: quel rôle dans la recherche didactique aujourd'hui? In: **Les dossiers des sciences de l'éducation**, N° 8, 2002. Didactique des disciplines scientifiques et technologiques: concepts et méthodes. pp. 59-72.

ARTIGUE, M. L'ingénierie didactique: un essai de synthèse. in Margolinas et all.(org.): Enamont et en aval des ingénieries didactiques, XV^a École d'Été de Didactique des Mathématiques – Clermont-Ferrand (PUY-de-Dôme). **Recherches em Didactique des Mathématiques**. Grenoble : La Pensée Sauvage, v. 1, p. 225-237, 2009.

BALACHEFF, N., Contribution de la didactique et de l'épistémologie aux recherches em EIAO, **actes des 13ème Journées Francophones sur l'Informatique**, Formation Intelligemment Assistée par Ordinateur, Genève, paginas 9-38. 1991.

BALACHEFF, N., La transposition informatique. Note sur un nouveau problème pour la didactique. **Vingt ans de didactique des mathématiques en France**, La pensée sauvage éditions, Grenoble. 364-370. 1994.

BELLEMAIN, F., A transposição informática na engenharia de softwares educativos. **Anais... I SIPEM**, 22 a 25 de novembro, Serra Negra-SP. 2000.

BELLEMAIN, F., SILVA, A. D. P. R.; TIBURCIO, R. Validação do conhecimento da engenharia de software educativo. In: VII EPEM-Encontro Pernambucano de Educação Matemática, 2017, Garanhuns-PE. **Anais... VII EPEM-Encontro Pernambucano de Educação Matemática-Anais**. Recife: Sociedade Brasileira de Educação Matemática - Diretoria Regional Pernambuco 2017, 2017. v. 1. p. 1-3.

BENITTI, F. B. V., SEARA, E. F. R., SCHLINDWEIN, L. M. Processo de Desenvolvimento de Software Educacional: Proposta e Experimentação. **Revista Novas Tecnologias na Educação** – CINTED UFRGS, Rio Grande do Sul, v. 3, n. 1, p. 1 – 10, Maio, 2005.

BOURQUE, P., FAIRLEY, R.E. **Guide to the Software Engineering Body of Knowledge**, Version 3.0, IEEE Computer Society, 2014.

GITIRANA, V., TELES, R. A. M., BELLEMAIN, P. M. B., CASTRO, A.T., ALMEIDA, I.A.C., LIMA, P. F., BELLEMAIN, F. **Jogos com Sucata na Educação Matemática**. 1a. ed. Recife: Editora Universitária da Universidade Federal de Pernambuco, 2013. v. 1. 180p.

PERRIN-GLORIAN, M. J. L'ingénierie didactique a l'interface de la recherche avec l'enseignement. Développement des ressources et formação des enseignants. in Margolinas et

all.(org.): En amont et en aval des ingénieries didactiques, **XV^a École d'Été de Didactique des Mathématiques** – Clermont-Ferrand (PUY-de-Dôme). Recherches em Didactique des Mathématiques. Grenoble : La Pensée Sauvage, v. 1, p. 57-78, 2009.

RAMOS, C. S. **Princípios da Engenharia de Software Educativo com base na Engenharia Didática: uma prototipação do bingo dos racionais**. 2014. f. 111. Dissertação de Mestrado (Programa de Pós-Graduação em Educação Matemática e Tecnológica – Edumatec). Recife, UFPE. 2014.

SANTOS, G. L. Alguns princípios para situações de engenharia de softwares educativos. **Inter-ação**, Goiás, v. 34, n. 1, 2009. Disponível em: <http://www.revistas.ufg.br/index.php/interacao/article/view/6540/4801>. Acesso em: 06 set. 2010.

SILVA, C. T. J. **A Engenharia Didático-Informática na prototipação de um software para abordar o conceito de taxa de variação**. 2016. f 163. Dissertação de Mestrado (Programa de Pós-Graduação em Educação Matemática e Tecnológica – Edumatec). Recife, UFPE. 2016.

SILVA, C. T. J.; TIBÚRCIO, R S.; GITIRANA, V.; BELLEMAIN, F. Function Studium: um software para o desenvolvimento do raciocínio covariacional. In: II Congresso sobre Tecnologias na Educação - Ctrl+E 2017, 2017, Mamanguape. **Anais do II Congresso sobre Tecnologias na Educação (Ctrl+E 2017)**. Mamanguape: UNIVERSIDADE FEDERAL DA PARAÍBA, 2017.

SOMMERVILLE, I. **Engenharia de Software**. 8. ed. São Paulo: Pearson Education do Brasil, 2011.

TIBÚRCIO, R S. Método de Elicitação de Requisitos para Software Educativo: um estudo a partir da prototipação de um software para função em plataformas móveis. In: XVIII EBRAPEM - ENCONTRO BRASILEIRO DE ESTUDANTES DE PÓS-GRADUAÇÃO EM EDUCAÇÃO MATEMÁTICA, 18., 2014, Recife. **Anais...** Recife: LEMATEC, 2014.

TIBÚRCIO, R. S. **A Engenharia Didático-Informática: uma metodologia para a produção de software educativo**. 2020. f 194. Tese de Doutorado (Programa de Pós-Graduação em Educação Matemática e Tecnológica – Edumatec). Recife, UFPE. 2020.

TIBÚRCIO, R. S. **Processo de desenvolvimento de software educativo: um estudo da prototipação de um software para o ensino de função**. 2016. f. 112. Dissertação de Mestrado (Programa de Pós-Graduação em Educação Matemática e Tecnológica – Edumatec). Recife, UFPE. 2016.

TIBÚRCIO, R. S.; BELLEMAIN, F. Processo de desenvolvimento de software Educativo: análise teórica e experimental com recurso a Teoria da Orquestração Instrumental. In: I Simpósio Latino-Americano de Didática da Matemática, 2016, Bonito. **ANAIS - TRABALHOS 1º LADIMA**, 2016.

TIBÚRCIO, R. S.; BELLEMAIN, F.; RAMOS, C. Engenharia de software educativos, o caso do bingo dos racionais. In: VI SEMINÁRIO INTERNACIONAL DE PESQUISA EM EDUCAÇÃO MATEMÁTICA - SIPEM, 2015, GOIÁS. **ANAIS DO VI SIPEM**, 2015.

TIBÚRCIO, R.; BELLEMAIN, F. Process of educational software development: epistemological and experimental analysis in the creation environment Lematec-Studium. In: Re(s)ources 2018 International Conference, 2018, Lyon. **Proceedings of the Re(s)ources 2018 International Conference**, 2018.

*Recebido em 11 de março de 2020.
Aprovado em 05 de setembro de 2021*