

## Aperfeiçoamento da Engenharia Didático-Informática com Contribuições da Metodologia de Desenvolvimento do Software Modellus

### Improvement of Didactical-Computational Engineering with Contributions from the Modellus Software Development Methodology

Ricardo Tiburcio<sup>1</sup>

Franck Bellemain<sup>2</sup>

#### RESUMO

O presente artigo difunde alguns dos resultados de uma investigação de doutorado com foco na Engenharia de *Software* Educativo. Discutimos no recorte aqui veiculado, os resultados da análise da engenharia do *software Modellus* e como esta contribuiu para (re)pensar os métodos e técnicas da Engenharia Didático-Informática. Assim, a pesquisa aqui apresentada está inserida na problemática de metodologias de desenvolvimento de *software* educativo bem como nas formas de construção de tecnologias digitais que auxiliem, efetivamente, as atividades de ensinar e aprender. Os procedimentos metodológicos foram iniciados com um levantamento bibliográfico a fim de atualizar os referenciais e posteriormente a construção de uma entrevista para obter informações precisas sobre a construção do *software Modellus*. Os resultados deste estudo indicam que: 1. É possível classificar a engenharia a qual o *Modellus* foi submetido como "Didático-Informática" e 2. a Engenharia Didático-Informática se configura como uma metodologia robusta para o desenvolvimento de *software* educativo que alie contribuições teóricas e metodológicas oriundas da Engenharia Didática e da Engenharia de *Software*.

**PALAVRAS-CHAVE:** Engenharia Didático-Informática. Engenharia de *Software*. *Software* Educativo. Engenharia Didática.

---

<sup>1</sup>Doutor em Educação Matemática e Tecnológica pela Universidade Federal de Pernambuco (UFPE). Professor da Prefeitura da Cidade do Recife. Membro do Grupo de Pesquisa Atelier Digitas. E-mail: [rico.tiburcio@gmail.com](mailto:rico.tiburcio@gmail.com). ORCID: <https://orcid.org/0000-0001-8601-3993>.

<sup>2</sup> Doutor em Didactique des mathématiques pela Université Joseph Fourier (Grenoble-I). Professor da Universidade Federal de Pernambuco - Centro de Artes e Comunicação. Membro do Grupo de Pesquisa Atelier Digitas. E-mail: [franck.bellemain@ufpe.br](mailto:franck.bellemain@ufpe.br). ORCID: <https://orcid.org/0000-0001-5358-2057>.



## ABSTRACT

This article disseminates some of the results of a doctoral research focused on Educational Software Engineering. We discuss in the clipping presented here, the results of the analysis of the Modellus software engineering and how it contributed to (re)think the methods and techniques of Didactical-Computational Engineering. Thus, the research presented here is inserted in the problematic of educational software development methodologies as well as in the forms of construction of digital technologies that effectively help teaching and learning activities. The methodological procedures started with a bibliographic survey in order to update the references and later the construction of an interview to obtain precise information about the construction of the Modellus software. The results of this study indicate that: 1. It is possible to classify the engineering to which the Modellus was submitted as "Didactical-Computational" and 2. Didactical-Computational Engineering is configured as a robust methodology for the development of educational software that combines theoretical and theoretical contributions from Didactic Engineering and Software Engineering.

**KEYWORDS:** Didactical-Computational Engineering. Software Engineering. Educational Software. Didactic Engineering.

## Introdução

Inserida na problemática do desenvolvimento de *software* para auxiliar o ensino e a aprendizagem da Matemática, a pesquisa aqui apresentada teve por objetivo aperfeiçoar uma metodologia para produção de *software* educativo a partir de estudos teóricos e da análise de procedimentos metodológicos da engenharia de alguns produtos com pesquisas consolidadas na área. Dentro dessa problemática, uma primeira resposta para criar *softwares* pautados no campo educativo, aliado aos conhecimentos tecnológicos, foi a Engenharia Didático-Informática - EDI (TIBURCIO, 2016; SILVA, 2016; TIBURCIO; BELLEMAIN, 2018; TIBURCIO, 2020).

A Engenharia de *Software*, como área de pesquisa, está preocupada com as atividades de conceber, desenvolver, operar e manter *software*; como requisitos, design, construção, testes, gerenciamento de configurações e outros processos (BOURQUE; FARLEY, 2014). Essa definição também pode ser utilizada para a criação de tecnologias educativas, contudo, carece de especificações que a Educação exige. Assim, a Engenharia de *Software* Educativo foi definida como área de pesquisa interessada na idealização e criação de interfaces, artefatos e tecnologias digitais, com a finalidade de contribuir para a melhoria das relações de ensino e aprendizagem de áreas distintas do conhecimento (TIBURCIO, 2020).

De acordo com Tiburcio (2020), resultados de pesquisas e investigações indicam que muitos recursos tecnológicos digitais, voltados para a Educação, ora são desenvolvidos centrados nas possibilidades oriundas da tecnologia, ora 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” (p. 27). Havia, segundo Tiburcio e Bellemain (2018), uma lacuna quanto às metodologias para desenvolver *software* educativo. A ausência de

articulação entre teorias educacionais e possibilidades tecnológicas não era uma realidade sistêmica e definida. No tocante à criação de *softwares*, os autores observaram a carência de uma metodologia robusta que pudesse aliar as contribuições das áreas de ensino e de aprendizagem aos processos da Engenharia de *Software*, com as devidas indicações de procedimentos. Assim, as observações e investigações desses autores geraram inquietações e questionamentos na área da Engenharia de *Software* Educativo, sendo assim concebida a Engenharia Didático-Informática.

A EDI foi criada em um cenário de tentativas de modelizar processos de desenvolvimento de *software* educativo em que fossem considerados aspectos teóricos e práticos sobre o desenvolvimento de tecnologias digitais para o ensino e a aprendizagem da Matemática. O termo “Engenharia Didático-Informática” constitui-se na percepção de utilizar os procedimentos metodológicos e reflexões teóricas da Engenharia Didática (ARTIGUE, 1996, 2002, 2009; PERRIN-GLORIAN, 2009) e da Engenharia de *Software* (SOMMERVILLE, 2011; BOURQUE; FARLEY, 2014).

Alguns *softwares* foram desenvolvidos utilizando as premissas e encaminhamentos da EDI, nos quais, com o uso, foram verificadas incompreensões por parte dos desenvolvedores dos projetos e necessidades de evoluções em seus referenciais. Ao serem observados *softwares* educativos com pesquisas consolidadas em diversas áreas da Matemática, concluiu-se que seria interessante realizar uma análise histórica. Verificaram-se, então, as engenharias às quais esses produtos foram submetidos, com a finalidade de compreender semelhanças e distanciamentos, reunindo elementos e agregando-os à EDI.

Com isso, na pesquisa aqui discutida, foram organizados, de forma metódica, os procedimentos para a realização de uma "análise histórica". Objetivou-se resgatar a engenharia do *software Modellus* (TEODORO, 2002), por ser um ambiente de simulação/micromundo e por configurar a mesma tipologia dos *softwares* que foram construídos com a EDI; além de ter o contexto acadêmico de investigação e desenvolvimento. Neste artigo, temos como objetivo apresentar uma compreensão da engenharia de desenvolvimento do *Modellus* à luz da EDI e a contribuição dos procedimentos, métodos e técnicas para a evolução da Engenharia Didático-Informática.

Justificamos a escolha desse *software*, além dos critérios mencionados, considerando sua tipologia, visto que é o principal produto da Engenharia Didático-Informática: os ambientes de simulação/micromundos, o que Teodoro (2002)

classificou como “ambientes exploratórios”. Segundo Tiburcio (2020), o termo "micromundo" surgiu no início dos anos 70, oriundo dos estudos sobre Inteligência Artificial, a partir dos trabalhos de Marvin Lee Minsky e Seymour Papert. No âmbito dessas pesquisas, o termo "micromundo" foi, inicialmente, utilizado para definir “um sistema que permite simular ou reproduzir um domínio do mundo real, e que tem como objetivo abordar e resolver uma classe de problemas” (p. 64).

O *Modellus* foi concebido por um grupo de pesquisadores, sob a liderança de Vitor Duarte Teodoro, Professor da Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa – Portugal. O objetivo inicial do programa era auxiliar o ensino dos conhecimentos da Física, porém, ao longo da utilização, foram percebidas possibilidades de envolver conhecimentos da Matemática e o *software* também se consolidou nesta ciência (TIBURCIO, 2020).

Com o exposto, fora considerada, na investigação aqui apresentada, a hipótese de que compreender a engenharia utilizada para desenvolver o *software Modellus* colaboraria para o aperfeiçoamento da EDI. A análise histórica realizada apresentou elementos da engenharia do referido programa utilizando como métodos de obtenção de dados: entrevista on-line (via *Skype*) e estudo das literaturas produzidas pelos desenvolvedores e por pesquisadores que utilizaram o *software*.

Apresenta-se, nesse texto, inicialmente, o atual paradigma de desenvolvimento de *software*: como são criados, quais são as estratégias e quais são as consequências dessas formas de produzir *software*. Exibe-se, em sequência, a engenharia do *software Modellus*. Por conseguinte, é exposta a metodologia da pesquisa aqui discutida, exibindo como os dados foram coletados e analisados e se discute, em síntese, a proposta da Engenharia Didático-Infomática e seus fundamentos teóricos e metodológicos. Por último, apresentam-se as conclusões do estudo realizado e as suas referências.

### **Os métodos de desenvolvimento de software educativo e suas consequências**

A utilização de tecnologias digitais, com a finalidade de favorecer as ações de ensinar e aprender, é uma realidade eminente. A sociedade, ao longo dos últimos anos, vem recebendo forte influência dos avanços tecnológicos, configurando novas formas de trabalho, de relacionamento, de ensino e de aprendizagem, entre outras situações fortemente influenciadas pela evolução tecnológica atual (TIBURCIO, 2020).

Ademais, alguns aspectos relevantes quanto à utilização desses recursos, nas práticas docentes e discentes, são as características peculiares dos *softwares*

educativos disponíveis. Estudos apresentam (BENITTI; SEARA; SCHLINDWEIN, 2005; SANTOS, 2009; COSTA; COSTA, 2013) a problemática da qualidade da maioria dos recursos tecnológicos, indicando a carência de uma engenharia específica para o desenvolvimento de *software* educativo que auxilie efetivamente a construção dos programas.

Foi observado, em alguns estudos, que a grande maioria dos *softwares* educacionais apresentam problemas que dificultam sua utilização. Dentre esses, Benitti *et al* (2005) expõem a falta de uma base pedagógica para melhor fundamentar a construção dessas tecnologias. A referida base apresenta-se como uma premissa de considerar, no desenvolvimento de tecnologias voltadas para a Educação, a inserção de fundamentos (teóricos e metodológicos) sobre o ensino e a aprendizagem dos conhecimentos que serão trabalhados com o uso da tecnologia.

No entanto, mesmo sendo essencial, a articulação entre potencialidades tecnológicas e conhecimentos da área educacional nem sempre é realizada nos desenvolvimentos de *softwares* educativos. Com isso, observando a problemática quanto ao desenvolvimento, Tiburcio (2020) apresentou uma das questões, inserida na problemática deste cerne: “como desenvolver recursos tecnológicos educativos que atendam às demandas da Educação Matemática e Tecnológica?” (p. 18). Em seguida, também apresentou uma resposta hipotética para tal questionamento, trazendo à tona a necessidade da existência de uma metodologia específica para a concepção e a criação de tecnologias voltadas às atividades docentes e discentes.

Desse modo, ao questionar as formas de desenvolver tecnologias digitais educativas, houve a necessidade de investigar, em caráter de revisão de literatura, quais eram os paradigmas de desenvolvimento. Assim, segundo Tiburcio (2016), de forma geral, os *softwares* educativos são desenvolvidos de três formas: utilizando metodologias padronizadas, realizando uma adaptação de metodologias padronizadas e integrando áreas de conhecimentos. Essas vertentes foram observadas em pesquisas sobre produção de *software* educativo.

As metodologias padronizadas são provenientes da Engenharia de *Software* (não educativos). Quando se pretende desenvolver tecnologias empregando esse modo, são utilizados: o Método em Cascata; o Desenvolvimento Iterativo e Incremental, o Método de Prototipagem; o Método em Espiral, as Metodologias Ágeis, entre outros. As equipes de desenvolvedores utilizam as metodologias citadas, sem realizar alterações, para a concepção dos recursos (TIBURCIO, 2016).

Na segunda vertente, adaptação de metodologias, as equipes percebem a necessidade de modificar alguns encaminhamentos propostos, verificadas as características dos *softwares* educativos. Justificam que as metodologias padronizadas auxiliam na criação desses produtos, porém reconhecem que são necessárias adaptações, visto que existem diferenças entre produtos tecnológicos para fins educativos e produtos para outros fins.

Por último, quando os *softwares* educativos são criados na perspectiva da integração de áreas, as equipes observam as contribuições de pesquisas e estudos sobre o ensino e a aprendizagem, verificam o que se discute no âmbito acadêmico, analisam orientações de documentos oficiais e tentam conectar tudo isso às possibilidades tecnológicas, com todo o rigor dos métodos de desenvolvimento de *software*. Segundo Tiburcio (2020), essa vertente de desenvolvimento é comum 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 aprender” (p. 27).

Observando a integração de áreas para o desenvolvimento de tecnologias digitais educativas, o *software Modellus* teve sua engenharia pautada nessa vertente. A equipe de desenvolvedores fundamentou a construção do *software* baseada, principalmente, em contribuições de pesquisas e estudos sobre o ensino e a aprendizagem de conhecimentos da Física e da Matemática, aliadas aos aportes que as tecnologias digitais podem trazer. Ao se analisar a principal motivação para a criação do *software*, percebeu-se que a experiência docente dos desenvolvedores, bem como as dificuldades nestas práticas foram consideradas (TEODORO, 2002).

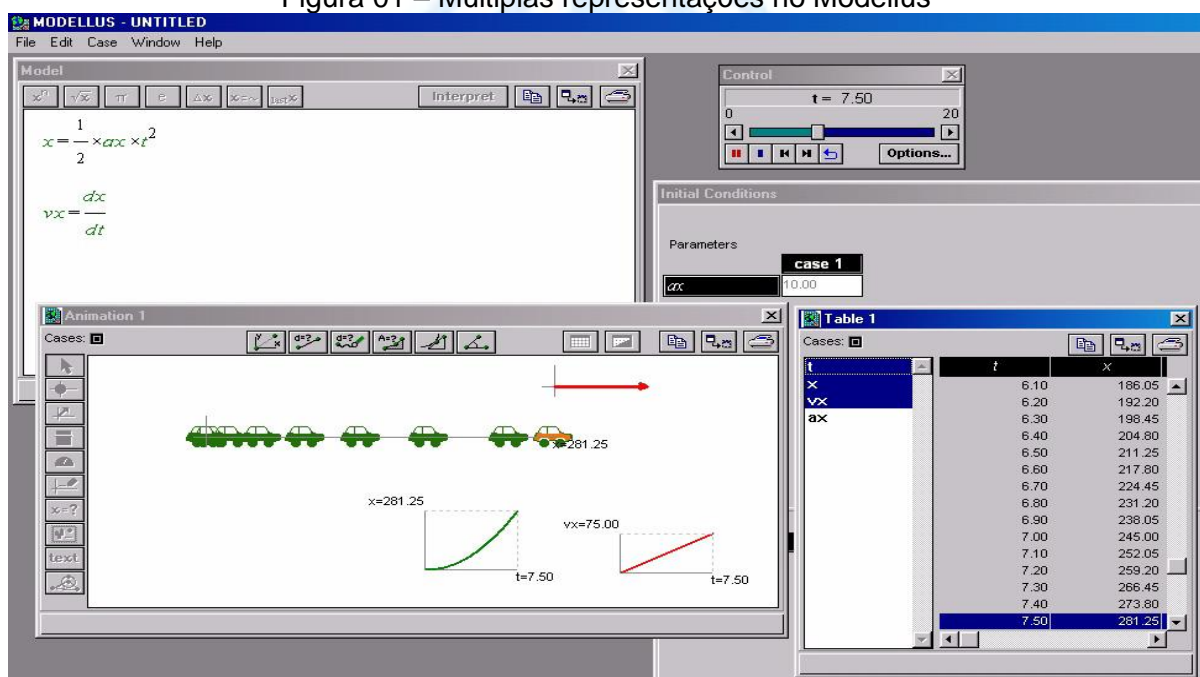
O *Modellus* é uma ferramenta que permite realizar experimentos conceituais usando modelos matemáticos expressos em funções, derivadas, taxas de variação, entre outros conceitos. De acordo com Teodoro (2002), este *software* pode ser considerado um micromundo de computador com base na metáfora "janela modelo": em que os usuários podem escrever modelos matemáticos, quase sempre da mesma maneira que escreveriam com papel e lápis. Ainda segundo o autor, não há um novo idioma a ser aprendido, apenas algumas regras de sintaxe sobre como escrever conceitos matemáticos.

As características e funcionalidades do *Modellus*, em síntese, podem ser assim descritas: 1. Ferramenta de software para criar e explorar várias representações de modelos matemáticos usando funções, equações diferenciais e equações iterativas. 2. Ambiente de múltiplas janelas. Em uma das janelas, o usuário pode escrever um

modelo, escrevendo as equações como estão escritas no papel; em outras janelas, o usuário pode criar e interagir com animações dos modelos, usando objetos abstratos, como vetores e gráficos, ou objetos mais concretos, como vídeos e fotos. 3. A comunicação com o usuário é baseada no conceito de "espelho intelectual" – o *software* atua como um espelho do que o usuário pensa (TEODORO, 2002).

Referente às diversas representações e às várias janelas, ilustra-se na Figura 01 o conceito de “múltiplas representações” de um movimento acelerado no *Modellus*. De acordo com Teodoro (2002), o software consegue representar, simultaneamente, equações, tabelas, gráficos, trajetórias, entre outros).

Figura 01 – Múltiplas representações no Modellus



Fonte: Teodoro (2002, p. 24)

Com o exposto, é perceptível que o desenvolvimento do *Modellus* considerou, em sua engenharia, as características do ensino e da aprendizagem, as dificuldades docentes e discentes com os conhecimentos, bem como as questões cognitivas e epistemológicas aliadas às potencialidades tecnológicas da época. Assim, a aproximação dessa engenharia com a Engenharia Didático-Infomática torna-se eminente, visto que a EDI também considera em seus pressupostos muitas das premissas da engenharia do *Modellus*.

Apresenta-se, na sessão a seguir, a metodologia utilizada na pesquisa de doutorado para que se compreenda quais foram os instrumentos de coleta de dados e como esses foram utilizados para analisar a engenharia do *Modellus* e aperfeiçoar a EDI.

## Delineamento metodológico e foco da investigação

A obtenção e a análise dos dados da engenharia do *software Modellus*, bem como as implementações para aperfeiçoar a Engenharia Didático-Infomática foram realizadas utilizando uma “análise histórica”. Essa abordagem consistiu em estudar o desenvolvimento do referido *software*, observadas a relevância, as pesquisas reconhecidas e consolidadas na área da Educação Matemática e a constatação, no âmbito acadêmico, da importância dessa tecnologia.

Foi realizada, inicialmente, uma entrevista on-line (via *Skype*) com o líder do projeto – Professor Doutor Vitor Duarte Teodoro, que culminou na indicação de se estudar as literaturas produzidas sobre esse *software*. Com isso, a análise histórica proposta pretendia levantar elementos e características pertinentes da engenharia do *Modellus*. O instrumento de coleta de dados, em formato de entrevista, tinha por finalidade acessar as particularidades dessa engenharia em contato direto com o desenvolvedor. Expõem-se, a seguir, os questionamentos que foram realizados e os objetivos de cada interação.

Tabela 1 – Primeiros questionamentos

QUESTIONAMENTOS INICIAIS (PERGUNTAS GERAIS)
Como surgiu a necessidade do desenvolvimento do <i>software</i> ?
O que se pretendia com o <i>software</i> ?
Quais problemas existiam para que o <i>software</i> se apresentasse como solução?

Fonte: Tiburcio (2020)

Esses questionamentos tiveram a finalidade de entender o contexto da concepção do *software* e compreender quais eram as demandas e os objetivos com o uso do produto. Na Tabela 2, estão os questionamentos quanto à compreensão da composição da equipe de desenvolvedores. Pretendia-se conhecer as áreas de formação dos membros e se a pluralidade de áreas havia sido considerada (ou não) na elaboração do programa.

Tabela 2 – Equipe de desenvolvimento

SOBRE A EQUIPE DE DESENVOLVIMENTO
Quais eram as características da equipe?
Foram considerados profissionais de diversas áreas para compor a equipe?
Existiam especialistas em quais áreas?

Fonte: Tiburcio (2020)



Com os questionamentos expostos na Tabela 3, procurou-se verificar os objetivos do *software* quanto ao ensino e à aprendizagem: se haviam sido observadas dificuldades nas atividades de ensinar e aprender e até que ponto o produto poderia ser concebido como uma possível solução para superar os entraves verificados.

Tabela 3 – Auxílio ao ensino e à aprendizagem

OBJETIVOS DE ENSINO E APRENDIZAGEM
O <i>software</i> possuía fins de auxiliar o ensino e a aprendizagem?
Foram observadas as etapas na construção de um conceito, pelo sujeito, para desenvolver o <i>software</i> ?
Foram elencadas dificuldades de ensino e aprendizagem e estas foram consideradas no desenvolvimento do <i>software</i> ?
Quais recursos disponíveis no <i>software</i> foram desenvolvidos para atender as especificidades do ensino e da aprendizagem?

Fonte: Tiburcio (2020)

Os questionamentos presentes na Tabela 4 tinham por finalidade compreender aspectos de programação e prototipação. Buscou-se conhecer qual linguagem de programação foi utilizada e como o protótipo foi desenvolvido, bem como as implementações que surgiram até a primeira versão. Procurou-se entender, também, se os usuários finais tiveram participação no processo de desenvolvimento do *Modellus*.

Tabela 4 – Sobre o desenvolvimento dos *softwares*

QUESTÕES TECNOLÓGICAS
Em que linguagem foi desenvolvido o <i>software</i> ?
Que tipo de prototipação foi utilizada para o seu desenvolvimento?
Quais características das versões iniciais foram implementadas até a primeira versão ser lançada?
Na prototipação do <i>software</i> , os usuários participaram dos testes e da experimentação?

Fonte: Tiburcio (2020)

De forma geral, com o instrumento de coleta de dados aqui exposto, tinha-se o objetivo de interpretar e relacionar as etapas da engenharia do *software Modellus* aos pressupostos que a Engenharia Didático-Infomática apresenta em seus fundamentos. As questões, em síntese, pretendiam verificar se a equipe de desenvolvedores considerou estudos teóricos sobre o ensino, a cognição, a epistemologia, os avanços tecnológicos e tudo o que pode contribuir para a aprendizagem e o ensino dos saberes a serem trabalhados com as tecnologias digitais.

Assim, apresenta-se, na sessão a seguir, o resultado da coleta dos dados, além da análise deles e suas contribuições para a evolução da Engenharia Didático-Informática.

### **A engenharia de *software* do *Modellus* e suas contribuições para aperfeiçoar a EDI**

Veiculam-se, aqui, os resultados da interação com o líder do projeto do *software Modellus*. A partir da entrevista realizada, obtivemos informações que nos fizeram buscar, nas principais literaturas, os dados necessários para compreender a engenharia do referido programa e, retomando um dos objetivos, aperfeiçoar a EDI a partir da compreensão da engenharia desta tecnologia.

#### a) Contexto da concepção do *Modellus*: demandas e finalidades com seu uso

Inicialmente, ao realizar os primeiros questionamentos, foi observado que o conceito do *Modellus* surgiu de múltiplas influências: experiência de ensino e em *design* de ambientes de aprendizado de computador e análise de literaturas sobre dificuldades de aprendizado. Vitor Teodoro destacou que, por muitos anos, em sua experiência docente, teve contato direto com os alunos intrigados com o significado, formas e implicações dos modelos matemáticos. Como supervisor e formador de professores, o autor expôs que os docentes (de Física e Matemática) têm muitas dificuldades com esses modelos (TEODORO, 2002).

A proposta de Teodoro (2002), quanto à utilização do *software Modellus*, apresentava uma nova perspectiva na aprendizagem e na definição dos currículos de Física, em que o uso do computador como ferramenta de modelação é considerado um instrumento chave no processo de aprendizagem.

Essa perspectiva assumia que a aprendizagem: (1) é um processo ativo de criação de significados a partir de representações; (2) decorre de uma comunidade prática, em que os estudantes aprendem a partir do seu próprio esforço e a partir de orientação externa; (3) é um processo de familiarização com conceitos, com ligações entre conceitos e com representações; (4) pode ser facilitada com interfaces baseadas na manipulação direta, permitindo aos estudantes explorar conceitos concreto-abstratos (TEODORO, 2002).

Com isso, ao confrontar tal visão de aprendizagem considerada na engenharia do *software Modellus*, ficou evidente que a Engenharia Didático-Informática deveria conter, em seus pressupostos, preocupações com as etapas da construção do conhecimento por parte dos estudantes como usuários das tecnologias digitais. Isso faria com que os desenvolvedores tivessem a diligência de considerar como esses

constroem os conhecimentos esperados, com auxílio dos artefatos tecnológicos que estão sendo idealizados. Assim, ao elaborar nova versão da EDI, foram considerados tais pressupostos observados na engenharia do *Modellus*.

#### b) Equipe de desenvolvedores e métodos

O *software Modellus* foi desenvolvido no final dos anos 90. Segundo Teodoro (2000), a metodologia de desenvolvimento desse programa considerou cinco fundamentos.

De acordo com o primeiro fundamento, o desenvolvimento é um projeto de equipe, envolvendo diferentes especialistas: pelo menos, designers de *software*, programadores e professores experientes. O segundo destaca que a criação de um *software* só deve ser iniciada após a identificação das experiências de aprendizagem mais relevantes em um determinado domínio. Já o terceiro fundamento considera as várias formas do aprendizado, visto que ocorrem em configurações diferentes: o *software* deve ser validado nas diferentes configurações, não pode ser projetado apenas para salas de aula, por exemplo. Tendo em vista a crescente difusão de computadores, os ambientes de aprendizado são múltiplos: em casa, em centros de recursos, em bibliotecas etc.

Além disso, Teodoro (2002) afirma que o *software* deve ser baseado em interfaces gráficas e de manipulação direta, nas quais o usuário controla suas ações diretamente, não mediado por comandos escritos/digitados; configurando, assim, o quarto fundamento. Por último, o quinto fundamento orienta que o *design* de ambientes exploratórios deve ser um processo iterativo, com melhorias sucessivas baseadas em observação do usuário ou estudos de usabilidade e *feedback* de alunos, professores e desenvolvedores de currículo.

Ao compreender os cinco fundamentos aqui discutidos, foi possível notar a relevância deles para qualquer engenharia de *software* educativo que preze pela efetiva contribuição das tecnologias digitais nas atividades de ensinar e aprender. A multiplicidade de profissionais a serem considerados na composição das equipes é uma premissa considerada também na Engenharia Didático-Informática. Contudo, verificadas as limitações de recursos (financeiros, temporais, de disponibilidade, entre outros), nem sempre as equipes conseguem articular tantos membros para seus projetos (TIBURCIO, 2020).

Uma solução para a ausência desses profissionais, apresentada em Tiburcio (2020), é o acesso às literaturas, a estudos e a pesquisas que podem orientar em como proceder para obter os encaminhamentos dessas áreas. Todavia, é necessário

que as equipes contenham, no mínimo, pesquisadores, professores e especialistas na área de conhecimento em que o *software* irá contemplar e um profissional da Ciência da Computação, para que os requisitos, as funcionalidades e as situações de ensino e aprendizagem sejam interpretados e transpostos para os meios digitais.

### c) Objetivos de ensino e aprendizagem

De acordo com Teodoro (2002), as ferramentas de computador podem ser meios cognitivos eficientes para auxiliar as relações educativas. Para que isso ocorra, é necessário considerar que o *software* deve lidar com conceitos errados; promover aprendizado ativo e com descobertas para os usuários; usar representações dinâmicas e interativas; permitir simulações desenvolvidas pelos próprios usuários/estudantes; e fornecer ambientes de suporte.

Referente a esses recursos, duas características do *Modellus* apresentam os auxílios mencionados: as múltiplas representações – significa que o usuário pode criar, ver e explorar diferentes representações do mesmo modelo; e a manipulação direta – em que o usuário pode interagir diretamente com as representações, usando o mouse e uma interface gráfica comum, sem a mediação da linguagem escrita.

Teodoro (2002) destaca a importância das várias representações quando enfatiza a sua visão quanto à construção de conhecimentos,

movendo-se “entre várias representações mentais”, incluindo “formas simbólicas, matemáticas, descritivas, experimentais, fenomenais e conceituais”, por meio de uma “rede complexa de interações” onde desenvolvem ideias por meio de diálogos para esclarecer conceitos e fazer conexões, com a ajuda de um professor que assume a função de “orientador do aluno” para apoiar o “aprendizado cognitivo”. (TEODORO, 2002, p. 149, tradução nossa<sup>3</sup>).

Além disso, o *Modellus* incorpora modos expressivos e exploratórios de atividades de aprendizagem. Com os modos expressivos, os alunos têm a possibilidade de criar seus modelos e maneiras de representar os objetos. Em um modo exploratório, os alunos podem usar modelos e representações propostos pelo *software*, analisando as relações entre os objetos abstratos. Quanto à atividade docente, Teodoro (2002) afirma que “os professores estão de acordo sobre a importância da modelação na aprendizagem da Física (e da Matemática) e com os

---

<sup>3</sup> Texto original: moving “between various mental representations”, including “symbolic mathematical, descriptive, experimental, phenomenal and conceptual forms”, through a “complex web of interactions” where they develop ideas through conversations in order to clarify concepts and make connections, assisted by a teacher that assume a role of “graduate student advisor” to support “cognitive apprenticeship”.

aspectos mais importantes das propostas sobre integração da modelação como um componente essencial do currículo” (p. 16).

Segundo o autor, as escolas mudam muito lentamente, os currículos tradicionais, com ênfase na aprendizagem mecânica e de fatos pontuais, apenas podem ser modificados se as escolas tiverem acesso a novas e poderosas visões sobre a aprendizagem e a novas ferramentas, que suportem a aprendizagem conceitual significativa e que sejam tão comuns e fáceis de utilizar como o papel e o lápis.

Dessa forma, considerando as questões cognitivas no processo de criação de um *software* educativo, percebeu-se a necessidade de uma maior orientação para que os desenvolvedores escolham fundamentos teóricos e práticos que possam respaldar os produtos a serem criados, considerando esses aportes para uma aprendizagem eficaz. De igual modo, isso deve acontecer em conjunto com as questões didáticas. Os desenvolvedores precisam observar como é realizado o ensino atual dos saberes que o *software* versará, analisar quais são as consequências desse ensino e propor, com o uso do *software* em construção, situações que possam auxiliar o ensino.

A Engenharia Didático-Informática traz, em sua metodologia, além da fundamentação didática e cognitiva, para desenvolver tecnologias digitais educativas, princípios epistemológicos que norteiam as equipes. Dessa forma, facilita a percepção das características dos conhecimentos, dos facilitadores e dificultadores, para que isso seja considerado na utilização do *software*, contribuindo com as atividades discentes e docentes.

#### d) Questões técnicas/metodológicas

Considerando as etapas para criar um *software*, a engenharia de desenvolvimento do *Modellus* tem uma proposta metódica e bem definida para projetar recursos tecnológicos digitais do tipo micromundo/simuladores. Essa estrutura é exibida na Figura 02, desenvolvida por Teodoro (2002) utilizando o Diagrama em V, com duas linhas de abordagem, uma metodológica e a outra teórica.

Figura 02 – Um modelo para orientar o design de ambientes exploratórios de aprendizado de computador para Ciências (e Matemática)



***Design de ambientes exploratórios de computador***

- 1 Integração com outros "recursos" (livros, colegas, professores)
- 2 Equilíbrio entre exploração e instrução
- 3 Progressão conceitual
- 4 Objetos abstratos-concretos
- 5 Múltiplas representações ligando a fidelidade perceptiva à fidelidade conceitual
- 6 Pensamento semiquantitativo
- 7 Interface autoexplicativa

Fonte: Adaptado de Teodoro (2002, p. 121)

A linha de abordagem teórica, quanto à fundamentação do *software* em pesquisas, visão de aprendizagem e visão do processo científico de criação, já foi discutida nas sessões anteriores deste texto. De igual modo, a linha metodológica, que contempla a composição de uma equipe multidisciplinar, identificação de configurações relevantes, metáfora da manipulação direta e melhorias iterativas. Com isso, explicitaremos, aqui, o que Teodoro (2002) classifica como "saída" do Diagrama em V. Eis as sete questões consideradas para o design de ambientes exploratórios (micromundos/simuladores):

**TABELA 05 – Design de ambientes exploratórios de computador  
QUESTÕES CONSIDERADAS PARA O DESIGN DE SOFTWARE**

1. Esse tipo de *software*, por si só, tem uso limitado: é necessário que seu uso seja considerado como parte dos pacotes de aprendizagem; servindo como complemento aos livros e materiais didáticos utilizados em sala de aula.
2. Equilibrar aprendizado exploratório e instrução direta é uma questão fundamental no design de pacotes de aprendizado e na criação de bons ambientes de aprendizado. Os professores devem compreender que os alunos terão algumas dificuldades em explorar adequadamente, sem orientação docente, o que ainda não sabem.
3. O *software* exploratório deve suportar a progressão conceitual, com base na natureza do assunto e na pesquisa didática. Os usuários devem ser capazes de fazer explorações simples com ou mais complexas, sem grandes dificuldades, à medida em que seus conhecimentos aumentam.
4. Os objetos típicos apresentados ao usuário em um ambiente exploratório por computador para aprender ciências são objetos abstratos-concretos. Os usuários

---

podem manipular diretamente objetos dos diversos conhecimentos científicos, porém eles são concretos apenas no computador, contudo, na realidade, são abstratos, pois são construções e representações mentais de propriedades de objetos reais ou imaginários.

---

5. Múltiplas representações são um dos recursos mais importantes do *software* exploratório. Esse recurso oferece aos alunos a possibilidade de interagir com diferentes representações coordenadas de um fenômeno, como equações, gráficos, animações, vídeos etc.

---

6. O *software* exploratório deve permitir que os alunos concentrem o raciocínio em descrições qualitativas, não em algoritmos e cálculos.

---

7. O *software* exploratório deve ter uma interface autoexplicativa: o domínio exploratório deve ser evidente para o usuário iniciante e as funcionalidades do *software* devem ser facilmente acessíveis sem treinamento formal sobre o uso do *software*.

---

Fonte: Adaptado de Teodoro, 2002

Os sete itens apresentados na Tabela 1 refletem premissas importantes a serem consideradas no desenvolvimento de tecnologias digitais para o ensino e a aprendizagem de conhecimentos diversos. Contudo, ao se pensar na variedade de conhecimentos da Matemática e de outras áreas do conhecimento que se interessem em desenvolver tecnologias para auxiliar as atividades docentes e discentes, vislumbra-se, na Engenharia Didático-Informática, que algumas abordagens de ensino e de aprendizagem, são eficazes para determinados conhecimentos e podem não ser para outros.

Desse modo, a metodologia da EDI não determina por antecipação quais serão os modos de explorar os conhecimentos que serão abordados nas tecnologias a serem construídas, deixando a critério dos pesquisadores, professores e engenheiros decidir, a partir das análises teóricas nas dimensões didática, cognitiva, epistemológica, informática e de outras naturezas, quais serão as melhores situações, formas, referenciais, que darão conta das demandas dos conhecimentos escolhidos.

Ainda sobre as questões técnicas e metodológicas da engenharia de desenvolvimento do *software Modellus*, vimos que a prototipação foi uma etapa que não ocorreu, já que o programa se tratava de uma evolução. Sendo assim, não foram utilizados protótipos para a construção da versão aqui discutida. Enquanto a experimentação para validar o *software* foi realizada com dois grupos de estudantes: um da Educação Básica e outro do Ensino Superior, ambos de Portugal.

O primeiro experimento, de acordo com Teodoro (2002), mostrou que os alunos do Ensino Médio podem começar a criar modelos com o *software*, após uma breve introdução a seus recursos, se tiverem conhecimentos prévios da Física e da Matemática necessários para criá-los. Os resultados do segundo experimento mostraram que estudantes de graduação com pouca formação em Física podem usar

facilmente o *Modellus* para criar seus próprios modelos com funções lineares, quadráticas e paramétricas. Ambas as experimentações constataram que os estudantes reconhecem que o *software* pode ser uma ferramenta importante para o aprendizado da Física e da Matemática.

A experimentação do *software Modellus* é uma forma considerada “clássica” para a testagem e a validação de *softwares* educativos. No entanto, além de obter dados a partir da experiência do usuário, na EDI, considera-se como de grande valia que também seja realizada uma validação teórica, no sentido de confrontar as análises realizadas inicialmente, as hipóteses e as possíveis contribuições pensadas pela equipe de desenvolvedores. Dessa forma, sabe-se, de fato, o que o *software* consegue propor com suas características e funcionalidades. A EDI sugere, em sua metodologia, que sejam feitas duas validações de *software*: uma experimental, nos moldes que se realizou a do *Modellus* e uma Teórica, conforme o mencionado.

Com as informações aqui discutidas sobre a engenharia de desenvolvimento do *software Modellus*, foram consideradas algumas das premissas e outras foram refutadas, com a finalidade de aperfeiçoar a Engenharia Didático-Informática. Na sessão seguinte, apresenta-se a metodologia de produção de *software* educativo (SE), eficaz para desenvolver tecnologias digitais educativas que considerem contribuições sobre o ensino e a aprendizagem, aliadas às potencialidades tecnológicas e à EDI.

### **Princípios da Engenharia Didático-Informática presentes na engenharia do *software Modellus***

As demandas do processo de criação de *software* educativo exigem que se articule fundamentação teórica sobre ensino, aprendizagem, epistemologia, entre outros fatores de diversas naturezas que auxiliem as atividades de ensinar e aprender, sem desconsiderar os métodos, técnicas e potencialidades da produção de tecnologias digitais. Com essa percepção, a Engenharia Didático-Informática foi concebida articulando premissas da Engenharia Didática e da Engenharia de *Software*, viabilizando um caminho metodológico para a criação de recursos tecnológicos (TIBURCIO, 2020).

A articulação dessas engenharias foi formalizada observando as seguintes características: a Engenharia Didática trata da construção de sequências de ensino a partir da utilização de conceitos e resultados de pesquisas, enquanto a Engenharia de *Software* coordena os conhecimentos sobre como as ferramentas digitais devem ser



desenvolvidas, atendendo aos requisitos levantados. A EDI é estruturada da seguinte forma:

da Engenharia de Software consideramos os avanços tecnológicos, as técnicas de levantamento de requisitos, a estrutura e organização das equipes de desenvolvimento – inserindo o usuário no processo, e as etapas organizacionais para a concepção e construção do software. A Engenharia Didática auxilia na compreensão de como as tecnologias podem ser utilizadas para contribuir com o ensino e a aprendizagem, trazendo a reflexão sobre aspectos didáticos, cognitivos, epistemológicos, e de outras naturezas, que possam auxiliar no levantamento dos requisitos do software, na criação de situações de uso, bem como na análise da utilização, e na validação do produto em elaboração (TIBURCIO, 2020, p. 75).

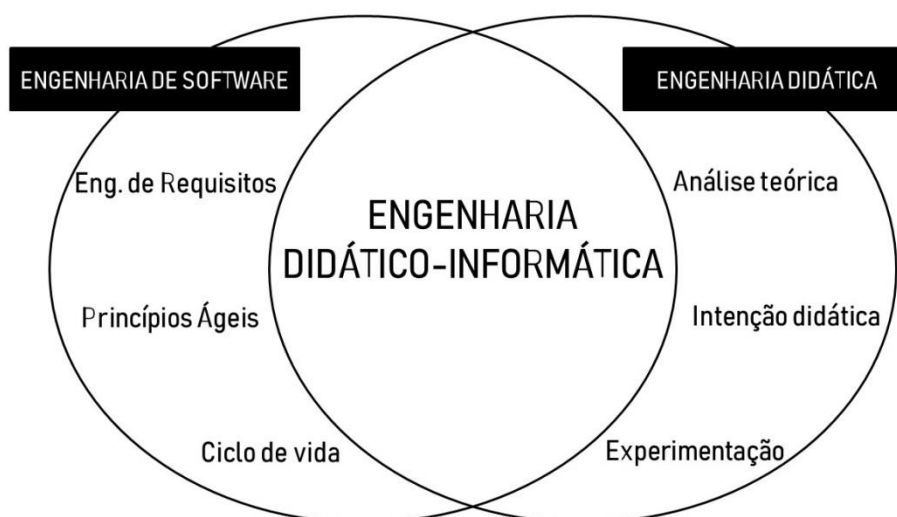
Com isso, a necessidade de articular as engenharias Didática e de *Software* foi percebida quando foram analisadas suas limitações para a criação de *software* educativo. A Engenharia Didática não contempla, em suas contribuições, a totalidade das exigências para a concepção de SE, ela não foi criada para o desenvolvimento de tecnologias e, sim, para desenvolver e investigar sequências de ensino. Ao passo que a Engenharia de *Software* não reúne especificidades que as tecnologias digitais educativas necessitam, visto que os modelos padronizados de desenvolvimento foram criados para produtos comerciais, bancários, domésticos, entre outros; desconsiderando as finalidades de ensinar e aprender que são características de tecnologias educativas (TIBURCIO, 2020).

Sendo assim, em Tiburcio (2020), observamos as finalidades da Engenharia de *Software* Educativo, a saber:

desenvolver software educativo é um processo de união entre os métodos, técnicas e análises da Engenharia de Software com as teorias, orientações e referenciais da Educação. Dentre os diversos referenciais metodológicos e teóricos no campo educacional, sobre a construção de situações de ensino e aprendizagem, sobre pesquisa, técnicas de elaboração de aulas e sequências de ensino, escolhemos a Engenharia Didática visto sua estrutura que alia pesquisa e ensino para compreender as relações educativas (TIBURCIO, 2020, p. 75).

Destarte, a Engenharia Didático-Informática foi construída como a interseção dos seguintes elementos, conforme apresenta-se na Figura 03,

Figura 03 – Aportes da EDI



Fonte: Tiburcio (2020, p. 74)

De acordo com Tiburcio (2020), a EDI é definida como metodologia para a produção de *software* educativo considerando como métodos os princípios da Engenharia de *Software* e da Engenharia Didática, assim como os processos de criação de situações de utilização e situações didáticas propostos nessas engenharias. Com os estudos e investigações realizados, observou-se que a EDI auxilia o desenvolvimento de *software* educativo, considerando as pesquisas e os encaminhamentos sobre o ensino e a aprendizagem, aliada às potencialidades das tecnologias digitais atuais.

### Considerações finais sobre o estudo realizado

O presente texto teve por objetivo apresentar a engenharia de desenvolvimento do *software Modellus* e como suas características contribuíram para o aperfeiçoamento da Engenharia Didático-Informática. Considerando a entrevista realizada com o desenvolvedor do referido programa, bem como as literaturas sobre sua criação e contribuições, foi possível obter dados e, com estes, aprimorar os encaminhamentos da EDI. No tocante à problemática de criação de tecnologias digitais para o ensino da Matemática que contribuam efetivamente para as atividades docentes e discentes, a investigação aqui veiculada apresentou as premissas de uma metodologia para criação de *softwares* educativos.

Os métodos utilizados para desenvolver *softwares* educativos são contestados vista a qualidade de muitos produtos disponíveis. Verificou-se que, rotineiramente, essas tecnologias são desenvolvidas utilizando metodologias (ou adaptações dessas) oriundas da Engenharia de *Software* (não educativos) e realizando-se a integração de conhecimentos educacionais com procedimentos da informática. Assim, na revisão de

literatura da pesquisa aqui exibida, notou-se que havia a carência de uma metodologia bem definida e específica para a criação de tecnologias digitais voltadas para o ensino e a aprendizagem.

Desse modo, uma primeira resposta para essa carência foi a construção da Engenharia Didático-Informática: metodologia que contempla, em sua fundamentação, elementos da Engenharia Didática, dando suporte teórico e prático quanto à utilização de tecnologias para auxiliar o ensino e a aprendizagem nas dimensões Cognitiva, Didática, Epistemológica, Tecnológica e de outras naturezas. Aliada à Engenharia de Software, a EDI traz contribuições metódicas sobre o processo de levantamento e interpretação de requisitos, gestão e organização das equipes de desenvolvimento e ciclo de vida de *software*.

A fim de aperfeiçoar e fazer evoluir o referencial da Engenharia Didático-Informática, em investigação de doutorado, resolveu-se analisar a engenharia dos softwares *Casyopée*, *Function Probe* e *Modellus*, para compreender as características desses projetos de desenvolvimento, tendo em vista as inúmeras contribuições e pesquisas consolidadas na área com esses produtos. Neste texto, apresentamos apenas a engenharia do *Modellus* e as características que foram verificadas para aprimorar a EDI.

Uma conclusão do estudo aqui exposto é que a metodologia de desenvolvimento do *software Modellus* pode ser classificada como uma engenharia Didático-Informática, visto que considerou os cuidados de desenvolver situações de ensino e aprendizagem, analisando referenciais teóricos, situações hipotéticas e prevendo as interações com os usuários – elementos da Engenharia Didática. Além disso, foram considerados, também, os métodos de levantamento de requisitos, as potencialidades tecnológicas e as características de *layout* e *design* – elementos da Engenharia de *Software*.

Conclui-se, pois, que a Engenharia Didático-Informática é uma metodologia robusta para criação de recursos tecnológicos digitais que auxiliam, de fato, nas relações de ensinar e de aprender. Assim, essa discussão constitui-se com base em fundamentações teóricas ricas e relevantes, a partir de conhecimentos e aportes tecnológicos substanciais.

## Referências

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

ARTIGUE, Michèle. 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. p. 59-72.

ARTIGUE, Michèle. L'ingénierie didactique: un essai de synthèse. in Margolinas et all.(org.): En amont et en aval des ingénieries didactiques, XV<sup>a</sup> 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.

BENITTI, Fabiane Barreto Vavassori, SEARA, Everton Flávio Rufino, SCHLINDWEIN, Luciane Maria. 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, Pierre .; FAIRLEY, Richard E. (Dick). **Guide to the Software Engineering Body of Knowledge**, Version 3.0. Disponível em: <https://www.computer.org/education/bodies-of-knowledge/software-engineering>. Acesso em: 10 de jan. 2020.

CONFREY, Jere; SMITH, Erick. Function Probe: Multi-Representational Software for Learning about Functions. New York State Association for Computers and Technology in Education 6, pp. 60-64. Using computers to promote students' inventions on the function concept. In Malcom, S., Roberts, L., & Sheingold, K. (Eds.). **This year in school science 1991**. (pp. 141-174). Washington, DC: American Association for the Advancement of Science. 1992.

COSTA, Antonioedro., COSTA, Estela Barreto. Contributos para o Desenvolvimento de Software Educativo tendo por base Processos Centrados no Utilizador. **EM TEIA: Revista de Educação Matemática e Tecnológica Iberoamericana**, v. 4, n. 2, p. 1–15, 2013.

LAGRANGE, Jean-Baptiste. Curriculum, classroom practices and tool design in the learning of functions through technology-aided experimental approaches. **International Journal of Computers in Mathematics Learning**10: 143–189. 2005.

PERRIN-GLORIAN, . 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<sup>a</sup> 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.

SANTOS, Gilberto Lacerda. Alguns princípios para situações de engenharia de *software* educativos. **Inter-ação**, Goiás, v. 34, n. 1, 2009. Disponível em: <https://www.revistas.ufg.br/index.php/interacao/article/view/6540/4801>. Acesso em: 06 de set. 2020.

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

SOMMERVILLE, Ian. **Engenharia de Software**. 9. ed. São Paulo: Pearson Education, 2011.

TEODORO, Vitor Duarte. **Modellus: Learning Physics with Mathematical Modelling**. 2002. f 248. Tese de Doutorado. Faculdade de Ciências e Tecnologia. Lisboa. Universidade Nova de Lisboa, 2002.

TIBURCIO, Ricardo dos Santos. **Engenharia Didático-Informática: uma metodologia para a produção de software educativo**. 190f. Tese de Doutorado (Programa de Pós-Graduação em Educação Matemática e Tecnológica – Edumatec). Recife, UFPE, 2020.

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

TIBURCIO, Ricardo dos Santos.; BELLEMAIN, Franck Gilbert René. **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...** Re(s)ources 2018 International Conference, 2018.

Submetido em janeiro de 2021.

Aceito em julho de 2021.