

# UM FRAMEWORK MDE PARA A CRIAÇÃO DE CENÁRIOS PEDAGÓGICOS AUTOMATIZADOS

**Wandemberg Rodrigues GOMES (1); Tatieures gomes PIRES (2); Gisele Pereira dos SANTOS (3); César Olavo de MOURA FILHO(4)**

Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE), Av. Treze de Maio, 2081 Fortaleza-CE

(1) wandemberg.rodriques@gmail.com

(2) tatieures.gomes@gmail.com

(3) gijeleps@gmail.com

(4) cesar\_olavo@yahoo.fr

## RESUMO

Este artigo apresenta técnicas para o desenvolvimento de aplicações voltadas à educação, com ênfase no protótipo MDEduc, que permite a criação de cenários pedagógicos pelos próprios educadores, mesmo que estes não possuam nenhuma competência na área de desenvolvimento de software. Estes cenários serão expansíveis de acordo com as concepções e necessidades de cada educador e, para isto, eles próprios serão modeladores da linguagem com o qual irão construir as aplicações. Será feita uma análise detalhada da arquitetura deste protótipo, propondo aperfeiçoamentos, a fim de torná-la mais robusta e extensível.

**Palavras-chave:** Learning design, Design Instrucional (Instruction Systems Design – ISD), Acceleo, JET (Java Emitter Templates), Eclipse Modeling Framework (EMF), Meta-modelo (Ecore).

## 1 INTRODUÇÃO

Os avanços tecnológicos têm grandes reflexos em todos os campos de atuação na nossa sociedade. Os computadores, a telecomunicação, os produtos eletrônicos e as tecnologias de automação desafiam as organizações a abandonar métodos operacionais antiquados, produtos obsoletos e serviços desatualizados em prol de iniciativas que acompanhem esse desenvolvimento tecnológico.

Com a educação não seria diferente. Os recentes avanços, principalmente em relação à melhoria e popularização do uso da internet, possibilitaram mudanças drásticas neste cenário ao difundir a ideia de software de apoio pedagógico e educação à distância. Isso permite que mesmo estudantes de áreas remotas com mobilidade reduzida, entre outros casos específicos, tirem vantagem do andamento assíncrono dos cursos à distância.

Nos últimos anos, estudos têm sido realizados com o intuito de aperfeiçoar o desenvolvimento de aplicações voltadas à educação. Algumas técnicas foram propostas, inicialmente bastante semelhantes a um ciclo de software comum, apenas adaptado a softwares educativos, como foi o caso do ISD (Reigeluth, 1983). Outras trazem consigo avanços mais recentes, que permitem, por exemplo, que o educador possa ser o próprio modelador de tais aplicativos, como foi o caso das linguagens de modelagem educacionais – EML (Koper e Tattersall, 2005). Estas técnicas serão melhores explicadas nos tópicos seguintes.

Recentemente, com o intuito de oferecer uma alternativa às anteriores, foi desenvolvido o MDEduc (De Moura Filho, 2007), que utiliza técnicas de Engenharia Orientada a Modelo e que, assim como o EML, contempla o fato de que os educadores possam ser os próprios desenvolvedores dos programas de apoio pedagógico. Caberia aos programadores, então, a tarefa de criar um ambiente onde estes profissionais possam construir aplicações flexíveis de acordo com as suas necessidades.

Com o objetivo de estender a abrangência destas aplicações para as diferentes orientações pedagógicas (por exemplo, behavioristas, construtivistas, sócio-construtivistas, etc.), visto que, pelo fato delas serem muito diferentes entre si, dificilmente uma aplicação que satisfaça educadores de uma determinada corrente satisfará os de outra, o MDEduc propõe que os próprios educadores criem linguagens personalizadas através

das quais eles irão construir as aplicações. Como parte desta proposta, foi implementado um protótipo de framework de maneira a validar essas ideias.

O intuito deste artigo consiste em reavaliar a arquitetura do MDEduc, de modo a torná-la mais robusta e extensível. Isto será feito implementando-se um framework de design instrucional baseado nos conceitos da Engenharia Orientada a Modelos, com o fim de simplificar e melhorar a qualidade dos cenários pedagógicos por meio da extensão do protótipo.

Este artigo está disposto da seguinte forma: a seção 2 trata dos trabalhos relacionados; a seção 3 contextualiza o leitor ao cenário no qual o MDEduc foi proposto e analisa o problema em questão: o desenvolvimento de aplicações por profissionais da educação, não necessariamente possuem competência de desenvolvimento de software; a seção 4 apresenta a análise da proposta e desenvolvimento do protótipo; a seção 5 mostrará quais as potencialidades da nova versão do Framework MDEduc; a seção 6 apresenta os artifícios utilizados para aperfeiçoamento do protótipo. A seção 7 expõe a conclusão e trabalhos futuros.

## **2 CONCEPÇÃO DE APLICAÇÕES VOLTADAS PARA A EDUCAÇÃO**

Durante anos, os programas de computador de auxílio ao ensino têm sido desenvolvidos utilizando-se técnicas instrucionais - Instructional System Design (ISD). Esta técnica seguia um ciclo de desenvolvimento de software comum no qual o desenvolvedor entregava a aplicação completamente desenvolvida ao usuário final, feita com base na análise das necessidades do público alvo e das metas de aprendizado do projeto.

Mais recentemente, outras técnicas têm surgido com o objetivo de oferecer uma alternativa ao ISD e tornar os softwares voltados à educação cada vez mais aprimorados, principalmente em relação à área da automação de metodologias pedagógicas. Foi com base nisso que surgiu o conceito de EML (Educational Modeling Languages), que propõe a criação de um ambiente onde os próprios profissionais da educação possam construir aplicações flexíveis de acordo com as suas concepções e objetivos numa fração de tempo e custo bem menor que a do ISD.

Isso foi possível através de um meta-modelo, normalmente na forma de um XML Schema, e um conjunto de ferramentas, dentre as quais, editores e motores de execução. Assim os cenários tornaram-se configuráveis, podendo ser modificados e configurados conforme a necessidade do educador.

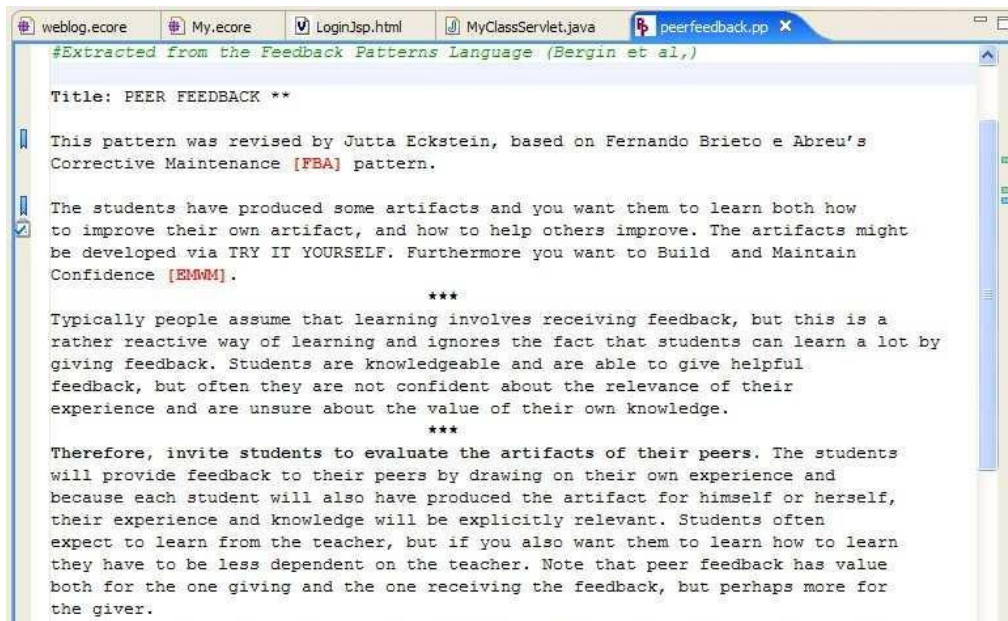
O grande problema do EML, é que mesmo dando ao usuário a possibilidade de criar sua própria aplicação, ela acabava ficando bastante limitada, pois era concebida por um meta-modelo generalista, a ser utilizado para construir todo e qualquer tipo de cenário. Este universo de discurso não seria, portanto, capaz de suprir todas as necessidades dos profissionais, mormente de correntes distintas na educação. E foi com o intuito de abordar este problema que foi proposta a metodologia multi-EML (De Moura Filho, 2007), bem como o protótipo MDEduc.

O MDEduc permite que os próprios profissionais da educação, mesmo sem conhecimentos na área do desenvolvimento de software passem a criar suas próprias linguagens genéricas, a partir de notações informais, através das quais irão gerar as suas próprias aplicações, de acordo com a sua concepção. No tópico seguinte será mostrado como isso foi possível.

## **3 O framework MDEduc para criação de aplicativos para a educação**

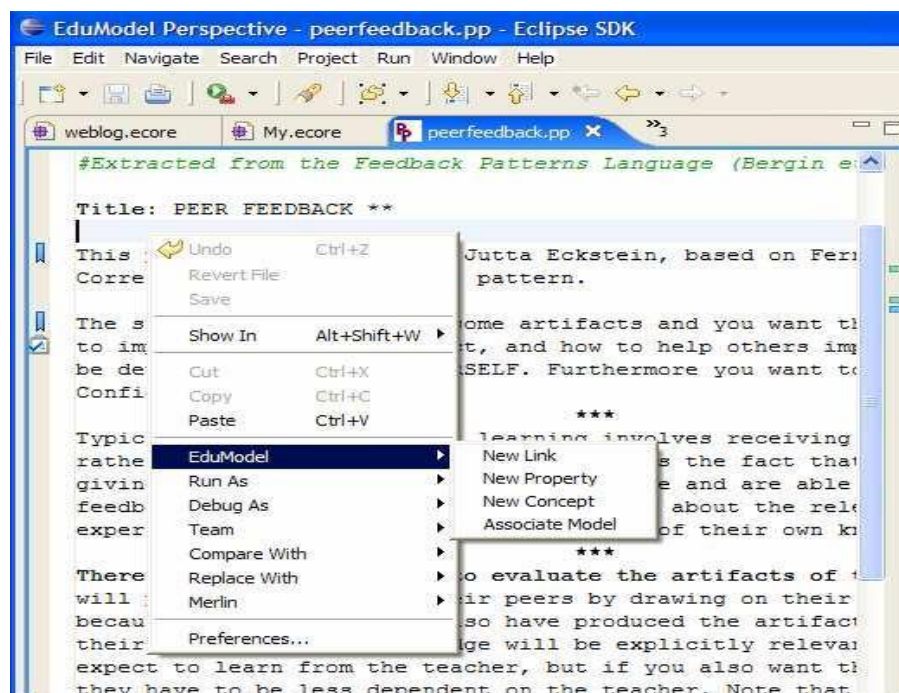
MDEduc é um ambiente onde profissionais da educação podem descrever as suas estratégias pedagógicas utilizando qualquer notação textual. Ele oferece facilidades adicionais para a criação de estratégias, respeitando a notação padrão de projeto pedagógico (Ppp, 2001) – no qual os designers de aprendizagem podem assumir a partir deles para criar um modelo formal e para gerar automaticamente um aplicativo - em vez de escrevê-lo à mão. Este protótipo foi construído na plataforma Eclipse, mais precisamente usando Eclipse Modeling Framework (EMF) (Steinberg, 2008).

O framework MDEduc é constituído por quatro plug-ins. Destes, dois plug-ins são usados para a criação do editor de padrões de projeto pedagógicos (PPP), uma notação bem definida - mas informal do ponto de vista informático – no qual os especialistas (professores) descrevem suas praticas pedagógicas.



**Figura 1: Mostra o editor do Framework MDEduc**

Um terceiro plug-in, o EduModel, é usado no framework para transformar os cenários anteriormente especificados informalmente na notação de padrão de projeto pedagógico em uma linguagem formal que servirá de base posteriormente para geração automática de aplicativos. Tecnicamente, esta linguagem formal é uma meta-linguagem (ou meta-modelo) especificada na meta-meta-linguagem Ecore (uma linguagem para especificar linguagens).



**Figura 2: Ações do EduModel**

O EduGen é quarto e último plug-in utilizado pelo framework para geração do código das aplicações correspondentes ao que está descrito no meta-modelo especificado anteriormente com o EduModel. Para isso, o MDEduc utiliza o framework JET (Java Emitter Templates) (Popma, 2004) para a geração das aplicações. A tecnologia JET (Java Emitter Templates) necessita que, para cada plataforma a ser gerada, um conjunto de templates seja implementado, correspondentes a cada tipo de arquivo a ser gerado. Por exemplo, o MDEduc fornece templates para uma aplicação Java para web, composto de servlets, JSPs, páginas HTML, descritores de configuração, DAOs, DTOs e scripts ANT. Assim, atualmente, o protótipo está limitado a gerar código somente para uma plataforma. Novos templates JET devem ser implementados, se for preciso.

expandir o MDEduc para outras plataformas. Como a programação de templates não é algo trivial, fica difícil estender o MDEduc para que especialistas consigam gerar automaticamente código para outras plataformas. Outra desvantagem do JET é a falta de uma comunidade mais atuante junto à qual poderiam ser tiradas dúvidas dos usuários e desenvolvidas templates de distintas plataformas.

#### 4 Usando o framework Acceleo para a geração de código

Em decorrência das deficiências apontadas na seção anterior, decidiu-se avaliar a substituição da ferramenta de geração de código JET pelo Acceleo (Musset, Juliot *et al.*, 2006), é um Framework usado para gerações de código a partir de um meta-modelo também especificado via meta-modelo Ecore. Uma das vantagens de usar o Acceleo é que há uma comunidade muito ativa, desenvolvendo e disponibilizando novos templates de geração de código para diversas plataformas (como o JET, o Acceleo também utiliza tecnologia de templates para gerar códigos). Assim, em vez de implementar novos templates, pode-se simplesmente reutilizar os templates já existentes. Com isto, torna mais fácil expandir o MDEduc para gerar aplicativos para diferentes plataformas. Além disso, o Acceleo fornece um editor com várias ferramentas que simplificam o desenvolvimento de novos templates por possuir ferramenta de auto-completação, sintaxe simplificada (ao contrario do JET, ele disponibiliza uma *linguagem de expressão*, o que diminui a implementação de templates em uma linguagem de baixo nível). A figura 3 ilustra um template editado no Acceleo.

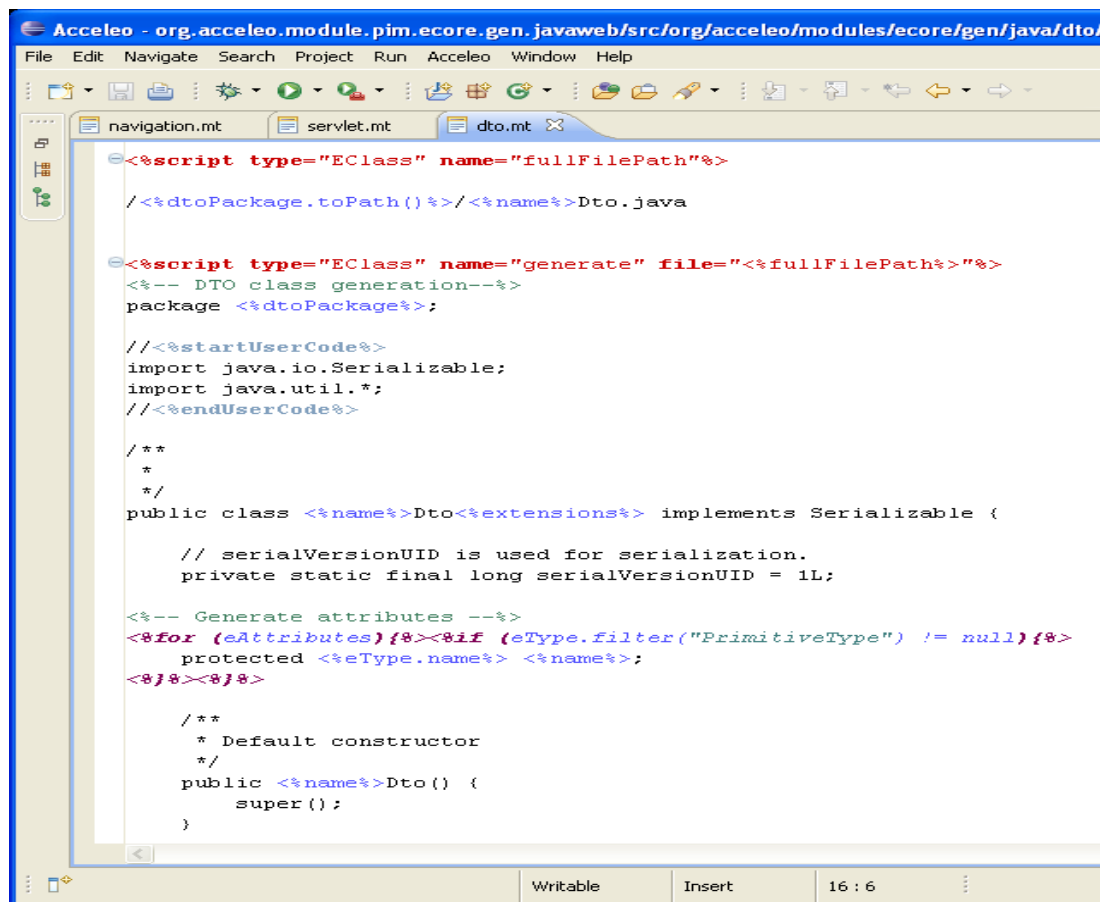
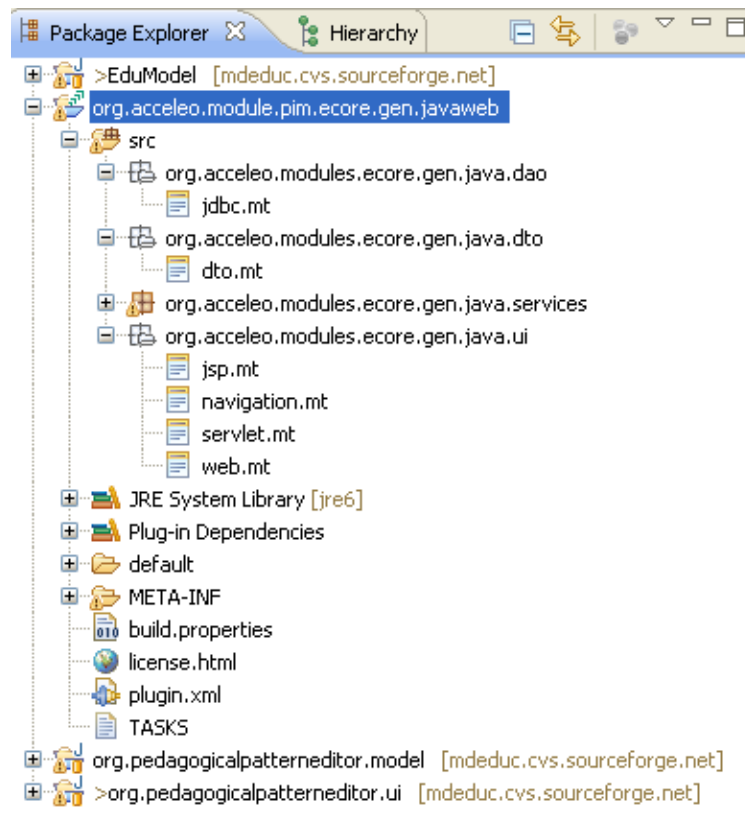


Figura 3: Editor de templates do Acceleo

Uma das desvantagens de usar o Acceleo no projeto é o aumento de um novo framework que terá de ser adicionado ao projeto para fazer a geração de código. Enquanto que o EduGen, que utilizava a tecnologia JET, estava bem amalgamado com o resto dos plugins -ele era utilizado a partir de menus do Eclipse, o que fazia o usuário acreditar que se tratava de uma única aplicação - o Acceleo é uma ferramenta externa, o que obriga o usuário a percorrer passos adicionais para gerar código. Outra desvantagem, decorrida da substituição da tecnologia JET pelo Acceleo, foi que a comunidade do Acceleo ainda não havia disponibilizado um módulo para a geração de código justamente para a plataforma interessada, isto é, Java para Web a partir de um meta-modelo (Ecore). Por isto foi desenvolvido tal módulo, já que esta era a plataforma para a qual havia desenvolvido os templates JET. Se isso, por um lado dificultou o trabalho de adaptação à tecnologia Acceleo, por outro lado, o trabalho extra será recompensado pelo fato de contribuir

com a comunidade, com ela compartilhando os novos templates desenvolvidos. A figura 4 mostra o módulo que foi implementado para gerar aplicações Java para Web a partir de um meta-modelo (Ecore).



**Figura 4: Módulo de geração de arquivos Java para Web usando o Acceleio**

A nova versão do MDEduc que utiliza o Framework Acceleio dá a possibilidade aos usuários gerar código em várias plataformas. Assim, pode-se facilmente gerar arquivos, por exemplo, nas linguagens de programação PHP e Python, pois a comunidade já disponibilizou módulos que conseguem gerar os arquivos para estas plataformas. Com a expansão da comunidade do Acceleio, futuramente estarão disponíveis vários outros módulos de geração para outras linguagens de programação.

## 5 Potencialidades da nova versão do MDEduc

A grande vantagem adquirida com as mudanças realizadas no MDEduc foi, sem dúvida, a possibilidade do projeto ser facilmente ampliado à medida que o Framework Acceleio também se desenvolve.

A versão anterior estava limitada a gerar somente aplicações para a plataforma JAVA para Web e com a necessidade de gerar aplicações para outras plataformas, foi preciso desenvolver uma nova versão desse Framework que conseguisse preencher esse requisito, ou seja, gerar aplicações para outras tecnologias.

A nova versão é capaz de capturar informações de uma linguagem informal e gerar aplicações em várias plataformas como, por exemplo, JAVA para Web, PHP e Python. Isso faz com que o Framework se torne mais extensível e mais robusto.

Além disso, tendo em vista que o Acceleio possui uma comunidade bastante atuante, estima-se que em um curto período de tempo, sejam adicionados ao MDEduc novos módulos de geração para outras plataformas além das quais o mesmo já consegue gerar. Desse modo, a linguagem de programação na qual o software será gerado que ficará a critério do usuário, de acordo com as suas necessidades. Por exemplo, dependendo do servidor de aplicações que o mesmo irá utilizar para alocar o aplicativo.

## 6 Criando de um cenário pedagógico com o MDEduc e utilizando o Acceleo para a geração de código

Com o MDEduc, o processo de criação de um cenário pedagógico ficou muito simples. Seguindo estes poucos passos que estão descritos abaixo, os especialistas sem o conhecimento da área de desenvolvimento de software poderão criar suas aplicações pedagógicas em um tempo de desenvolvimento menor e com uma qualidade melhor que os processos de criação dos cenários convencionais nas quais os especialistas passam a função de programar a suas aplicações a outras pessoas com o conhecimento da área de informática.

Passo 1- Antes de tudo precisa-se fazer o download da nova versão do framework MDEduc que está no repositório sourceforge<sup>1</sup> e do framework Acceleo com o módulo de geração para a plataforma desejada que está disponível do site do fabricante<sup>2</sup>;

Passo 2- Já com todos os frameworks e plug-ins instalados no Eclipse, deve-se criar um projeto no Eclipse com um editor pedagógico, um meta-modelo (Ecore) e uma “chain” (corrente de geração), responsável pela geração da aplicação pedagógica;

Passo 3- O professor descreve suas estratégias pedagógicas no editor do MDEduc, o que é simples, pois se trata de uma linguagem natural (português, por exemplo);

Passo 4- Um especialista em modelagem captura as palavras-chave da metodologia escrita no passo anterior, estabelecendo o relacionamento entre as mesmas em um processo ao final do qual, uma linguagem formal da mesma estratégia pedagógica do passo 3 estará disponível.

Passo 5- Depois de descrever as notações, utiliza-se o Acceleo para gerar a aplicação final na plataforma desejada, utilizando, para tal, a corrente de geração (chain) apropriada, como mostra a figura 5. Finalmente, os arquivos gerados automaticamente podem ser vistos na figura 6.

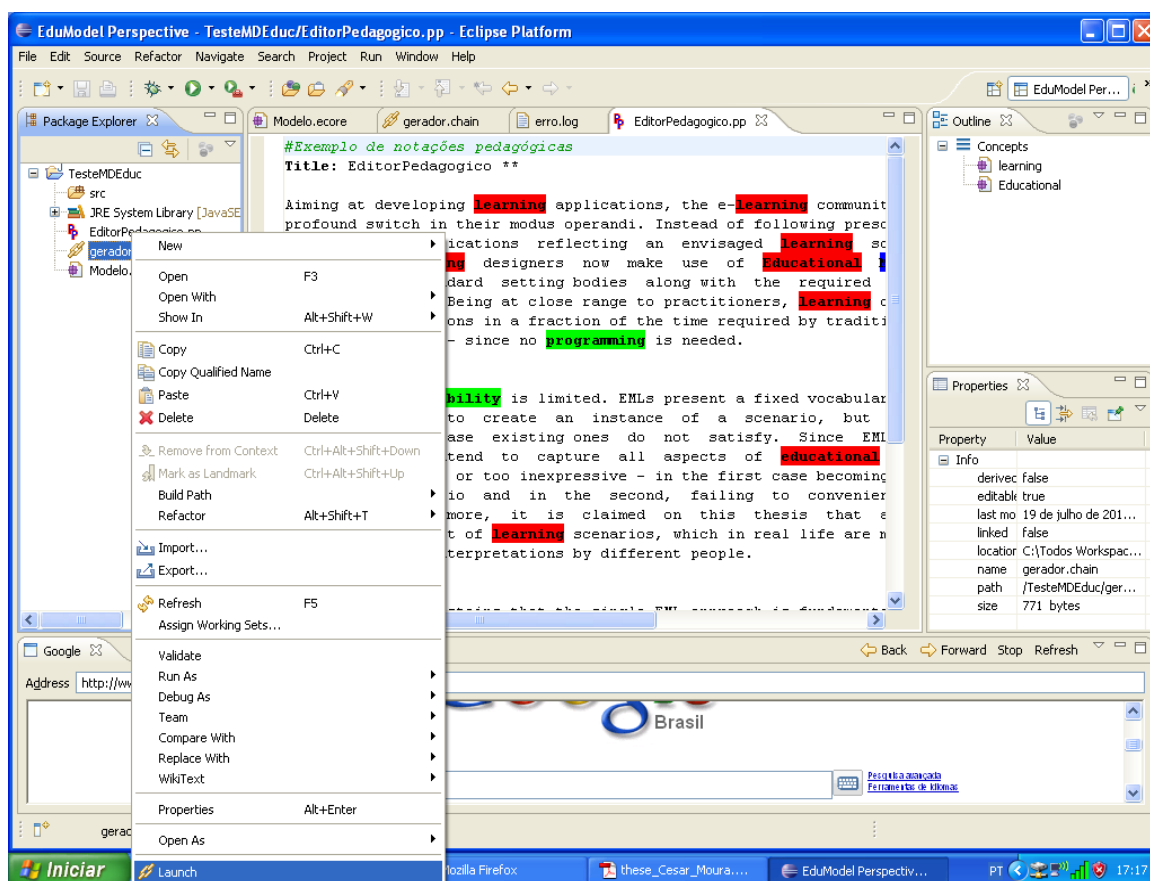


Figura 5: Mostra como é simples gerar código usando a nova versão do MDEduc

<sup>1</sup> <http://mdeduc.sourceforge.net>

<sup>2</sup> <http://www.acceleo.org/update/>



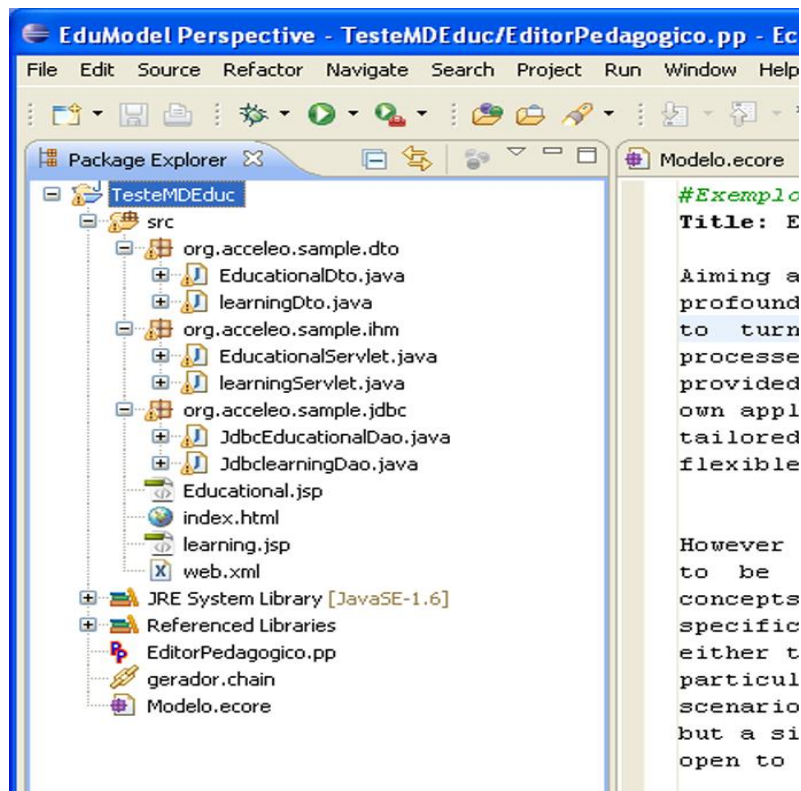


Figura 6: Mostra os arquivos gerados pela nova versão do MDEduc

## 7 Conclusão e trabalhos futuros

O protótipo MDEduc propõe contribuir para a automatização do desenvolvimento de aplicações pedagógicas, ou seja, os especialistas poderão criar seus cenários pedagógicos a partir de descrições das notações pedagógicas em uma linguagem informal. Neste trabalho, propõe-se uma nova versão do MDEduc, que utiliza o framework Acceleio para geração de código, o que vai facilitar sobremaneira a geração de aplicações para diversas plataformas sem ter de passar pela trabalhosa tarefa de desenvolver novas templates de geração para cada nova plataforma desejada.

Nos trabalhos futuros, será modificado o MDEduc para deixar sua interface de criação de cenários pedagógicos do framework mais amigável para o desenvolvimento das aplicações. Isto aumentará o nível de abstração, ou seja, o usuário precisará de cada vez menos conhecimento da área de desenvolvimento de software. Futuramente, estará disponível novos módulos de geração, ajudando a estender o MDEduc para outras plataformas além das quais já existem. Assim, a possibilidade do especialista escolher a plataforma da qual a aplicação será gerada será maior.

## REFERÊNCIAS

BERGIN, J.; ECKSTEIN, J.; MANNS, M. e SHARP, H. **The Pedagogical Patterns Project (PPP)**. Disponível em: <<http://www.pedagogicalpatterns.org>> Acesso em: 5 nov 2009.

BUDINSKY, F.; STEINBERG, D.; MERKS, E.; ELLERSICK, R. e GROSE, T.J. **Eclipse modeling framework : a developer's guide**. Boston, MA: Addison-Wesley. 2003 (The eclipse series)

DE MOURA FILHO, C. O. **MDEduc: conceiving and implementing a language-oriented approach fo the design of automated learning scenarios**. Computer Science, University of Lille, Lille, 2007. 266 p.

Koper, R. e C. Tattersall. **Learning design : a handbook on modelling and delivering networked education and training**. Berlin ; New York: Springer. 2005. xxviii, 412 p. p.

MUSSET, J.; JULIOT, É. e LACRAMPE, S. **Architect Tutorial**. Disponível em:  
<<http://www.acceleo.org/doc/otheo/en/acceleo-2.6-architect-tutorial.pdf>> Acesso em: 15 jan 2010.

MUSSET, J.; JULIOT, É. e LACRAMPE, S. **Reference**. Disponível em:  
<<http://www.acceleo.org/doc/otheo/en/acceleo-2.6-reference.pdf>> Acesso em: 15 jan 2010.

MUSSET, J.; JULIOT, É. e LACRAMPE, S. **User Guide**. Disponível em:  
<<http://acceleo.org/doc/otheo/en/acceleo-2.6-user-guide.pdf>> Acesso em: 15 jan 2010.

POPMA, R. **JET Tutorial Part 1 (Introduction to JET) 2004A**. Disponível em: <<http://www.eclipse.org>>  
Acesso em: 20 out 2009.

POPMA, R. **JET Tutorial Part 2 (Write code that writes code) 2004B**. Disponível em:  
<<http://www.eclipse.org>> Acesso em: 20 out 2009.

REIGELUTH, C. M. **Instructional-design theories and models**. Hillsdale, N.J.: Lawrence Erlbaum Associates. 1983

STEINBERG, D.; BUDINSKY, F.; PATERNOSTRO, M. e MERKS, E. **EMF: Eclipse Modeling Framework**. Upper Saddle River, N.J.: Addison-Wesley. 2008. xxix, 704 s. p.