

ROODA DEVEL: Uma proposta de Framework para Construção de Plataformas de Educação à Distância

Profa. Dra. Patricia Alejandra Behar¹, Juliano de Vargas Bittencourt,
Maicon Brauwiers, Silvia de Oliveira Kist

¹Faculdade de Educação - Universidade Federal do Rio Grande do Sul
Av. Paulo Gama s/n - Prédio 12201 - Centro - CEP 90046-900 - Porto Alegre, RS, Brasil

bpehar@terra.com.br, {juliano,maicon,silvia}@edu.ufrgs.br

Abstract. *ROODA Devel is a framework that helps in the construction process of platforms that support Distance Learning. The principal purpose is to diminish the time needed for programming this kind of software and simultaneously to add flexibility and robustness to them. The present work describes the main features of this framework, and its development process.*

Resumo. *ROODA Devel é um framework que auxilia no processo de construção de plataformas que apoiam a Educação à Distância. Este tem como objetivo diminuir o tempo de programação deste tipo de software e simultaneamente agregar-lhes flexibilidade e robustez. O presente trabalho descreve as principais características deste framework e seu processo de desenvolvimento.*

1. Introdução

Nos últimos tempos a Educação à Distância (EAD) vem sendo vista como uma das soluções encontradas para sanar vários problemas de ensino no país. Esta é uma das razões da crescente demanda de pesquisas na área, com o intuito de tornar as propostas de EAD cada vez mais personalizadas e transparentes para o usuário, respeitando o seu processo de construção de conhecimento.

É neste contexto que se insere o Projeto ROODA que vem sendo desenvolvido dentro no NUTED¹, na Faculdade de Educação da UFRGS, que faz parte do CINTED²/UFRGS. Este tem como objetivo realizar estudos sobre os processos de ensino/aprendizagem/cooperação em EAD, bem como a construção de ambientes que propiciem isto.

Portanto, o primeiro passo foi o desenvolvimento de uma plataforma de educação à distância, denominada ROODA, que significa Rede cOOperativa De Aprendizagem. Desde a sua primeira versão (março de 2000) até a atual (3ª versão - março de 2002), este software está servindo como uma ferramenta de pesquisa do NUTED. Sobre ela, estão sendo realizadas experiências, tanto para a análise dos processos de aprendizagem dos alunos, quanto para a verificação da efetividade da plataforma. Através do levantamento de problemas e hipóteses, são coletados os dados nos diferentes cursos à distância oferecidos. Estes servem de base para o estudo de possíveis alterações a serem realizadas na plataforma, com o fim de aprimorá-la cada vez mais, de acordo com as necessidades dos usuários.

Entretanto, realizar alterações significativas em uma plataforma é uma tarefa árdua que exige um grande esforço de programação, e conseqüentemente, muito tempo. Esta atividade se

¹NUTED- Núcleo de Tecnologia Digital aplicada à Educação, disponível em <http://www.nurted.edu.ufrgs.br>.

²CINTED- Centro de Estudos Interdisciplinares em Novas Tecnologias na Educação.

torna mais difícil ainda, quando tais alterações são parte importante da metodologia de pesquisa e devem acontecer rapidamente, por causa dos cursos e disciplinas que estão sendo ministrados.

Outro aspecto importante do projeto a ser destacado foi a necessidade de pesquisar e desenvolver outras plataformas que possuíssem um desenho metodológico diferente do ROODA, e/ou que se destinavam a públicos alvo diferentes.

A partir destas constatações, ficou claro que era necessário desenvolver uma arquitetura robusta que permitisse aos desenvolvedores alterar ou desenvolver novas plataformas para EAD, de forma mais simples e rápida. Logo, surgiu o ROODA Devel, um *framework* para o desenvolvimento de ambientes para educação à distância.

Portanto, neste estudo é apresentado o ROODA Devel, descrevendo os principais aspectos de sua arquitetura.

2. Necessidades de Plataformas para EAD

Durante o desenvolvimento do ROODA e de outras plataformas pela equipe do NUTED, constatou-se uma série de funções comuns entre estes sistemas, e com modelos encontrados na literatura [Britain and Liber, 2001]. Estas funções normalmente representavam partes importantes do ambiente cujo bom funcionamento refletia nas demais partes do software. Para facilitar o processo de desenvolvimento eram utilizadas fatias de código do ROODA, o que diminuía o tempo necessário para a programação do ambiente. Entretanto o código era muito específico, e eram necessárias alterações para adaptá-lo a cada plataforma. Este processo, muitas vezes, inseria erros nos sistemas que eram de difícil detecção.

Durante estes processos pode-se destacar as principais características comuns destas plataformas:

- Controle de Logon: todas as plataformas de EAD disponíveis, em algum momento, solicitam ao aluno e/ou professor uma autenticação no sistema. Esta função é uma das mais importantes sobre o aspecto da segurança do sistema, pois um sistema de má qualidade de *Logon/logout* permite que usuários não autorizados tenham acesso aos *scripts* da plataforma.
- Organização e Controle dos Cursos: a maior parte das plataformas existentes permite que um mesmo site possua vários cursos em andamento simultaneamente. Cada um destes possui seus próprios alunos e professores, que podem até mesmo ocuparem papéis diferentes em cursos diferentes. Normalmente, existe um processo de matrícula no qual os alunos solicitam vagas, que posteriormente são concedidas pelos professores.
- Ferramentas: todas as plataformas oferecem ferramentas, tais como fóruns, *chats*, e-mails, etc.; com as quais os usuários interagem uns com os outros. Estas ferramentas são muitas vezes confundidas com a própria plataforma, mas na maior parte dos casos constituem entidades externas que se integram a interface. Por exemplo, um curso de matemática pode exigir uma ferramenta de fórum na qual possa ser possível digitar equações matemáticas, mas sem necessitar de uma nova plataforma. Apenas a inclusão desta alteração no fórum seria suficiente. Muitas plataformas optam por tentar oferecer tudo a todos, o que complica sensivelmente a interface das ferramentas.
- Design de Interface Padrão: um dos maiores esforços de programação é o desenvolvimento de interfaces. Muitas vezes perde-se muito tempo criando páginas *Web* que servirão como modelo e replicando-as em todos os *scripts* do sistema.
- Envio de materiais e publicação de páginas: a maior parte das plataformas utiliza páginas HTML como meio de disponibilizar materiais, tanto pelos professores quanto pelos

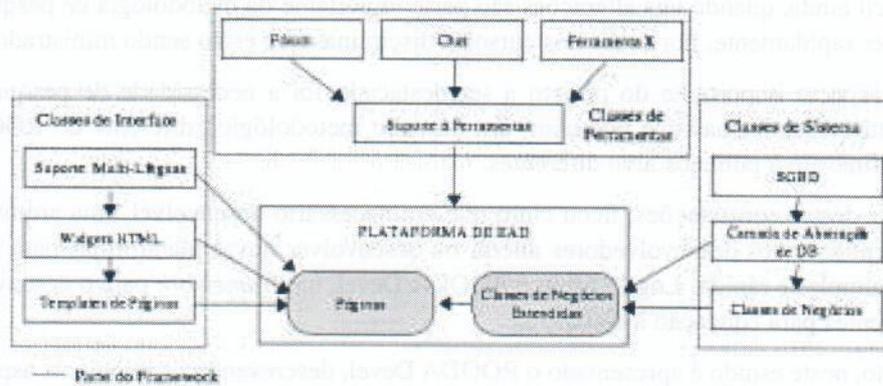


Figura 1: Arquitetura do framework

alunos. Na maior parte dos casos estas páginas são construídas nas estações de trabalho dos participantes do curso e posteriormente enviadas para o servidor. Assim, todos os ambientes que utilizam essa forma de disponibilização de material provém um mecanismo de *upload*, o qual permite aos usuários enviar, apagar, copiar, visualizar os arquivos.

3. Descrição do Sistema

Com base nas constatações vistas na seção anterior optou-se pelo desenvolvimento de um *framework* que contemplasse tais necessidades e que fosse disponibilizado através da GPL [Free Software Foundation, 1989]. Este *framework* foi desenvolvido como uma série de classes em PHP rodando em um servidor RedHat Linux; entretanto ele pode ser utilizando em outros sistemas operacionais. A opção pela orientação a objetos se deve ao fato de permitir a expansão das classes originais. Por exemplo, para criar uma classe de usuário específica para uma determinada plataforma de EAD, basta criar uma nova sub-classe que herda a classe original de usuários. Como podemos observar na Figura 1, o *framework* é dividido em três grandes conjuntos de classes: as classes de sistema, as classes de interface e as classes de ferramentas.

3.1. Classes de sistema

As classes de sistema constituem parte importante do *framework*, pois elas representam entidades como usuários, cursos, matrículas, projetos, etc. Cada classe possui as regras necessárias para recuperar e gravar seus dados junto ao Banco de Dados, bem como toda a parte de verificações de persistência dos dados. Junto às classes de negócios, também está presente uma camada de abstração de SGBD. Esta camada provê métodos genéricos de acesso ao banco de dados para as classes superiores, de forma que qualquer sistema de banco de dados possa ser utilizado, desde que os métodos de acesso sejam adaptados. Dessa forma, é possível uma mudança de SGBD sem a necessidade de se alterar grande parte do código do sistema.

3.2. Classes de Interface

Como já foi apresentado no presente estudo, a implementação de interfaces costuma depender muito tempo. Para diminuir este tempo foi criada uma solução composta de classes de *Widgets*, *Templates* de Páginas e *Suporte Multi-Líguas*:

- *Templates* são modelos de páginas que já vem com um estrutura padrão definida. Todos os *scripts* da plataforma instanciam um *template*, e a ele vão adicionando código HTML ou *Widgets*. Ao final do *script* chama-se um método do *widget* que envia ao cliente a saída

HTML adequada. Esta metodologia é muito útil, pois desta forma, para alterar a aparência de todo ambiente, é necessário apenas alterar as classes dos *templates* correspondentes, não precisando fazer a alteração em cada página do software.

- *widgets* são classes que funcionam como objetos HTML que implementam uma funcionalidade específica como, por exemplo, formulários que conferem os dados. Estas classes surgiram da necessidade de implementar comportamentos complexos na interação como usuário em diversas partes da plataforma.
- O Suporte Multi-Línguas consiste de uma centralização de toda e qualquer mensagem que o usuário visualiza em um arquivo único. Nenhuma mensagem é escrita diretamente no código, o que facilita muito a tradução da plataforma para outros idiomas. O *framework* disponibiliza classes que reconhecem automaticamente a língua padrão do *browser* do usuário e escolhe o arquivo de linguagem apropriado a ser carregado.

3.3. Classes de Ferramentas

Como já foi citado anteriormente, a maior parte das plataformas considera as ferramentas de interação como parte integrante e imutável das plataformas. Entretanto o ROODA Devel considera as ferramentas como entidades separadas do sistema, que podem variar dependendo da instalação. Para tanto, ele oferece uma classe de ferramenta que possui as características principais de uma ferramenta. Cada nova ferramenta instancia esta classe e insere suas próprias funcionalidades e características. Logo depois, esta ferramenta é integrada automaticamente pelo ambiente no sistema através de uma série de arquivos de configuração. No ambiente ROODA, o próprio ambiente oferece um *script* que lê a especificação de uma ferramenta e automaticamente a agrega ao sistema.

4. Considerações Finais

O ROODA Devel surgiu como uma ferramenta para ajudar a equipe do NUTED no desenvolvimento de plataformas para EAD. Além disso, ele se constitui no retorno que o projeto ROODA oferece à comunidade de Software Livre. Este mesmo projeto, que surgiu dentro das idéias de liberdade do projeto GNU e se forjou nos exemplos de cooperação entre programadores em sites, como o Source Forge, só foi possível devido à existência de software livre.

Assim como o ROODA se beneficiou da comunidade, hoje ele espera retribuir compartilhando suas experiências de modo a ajudar outros projetos a desenvolverem suas próprias soluções.

Referências

- Bassett, P. G. (1997). *Framing Software Reuse - Lessons from the Real World*. Computation Series. Yourdon Press.
- Britain, S. and Liber, O. (2001). A framework for pedagogical evaluation of virtual learning environments. University of Wales - Bangor. Disponível em <http://www.jisc.ac.uk/jtap/html/jtap-041.html>.
- Free Software Foundation (1989). *GNU General Public License*. Disponível em <http://www.gnu.org/copyleft/gpl.html>. Acesso em 11 de Nov. de 2000.
- Stallman, R. (99). *Open Sources: Voices from the Open Source Revolution*, chapter The GNU Operating System and The Free Software Movement, page 53. O'Reilly Open Source.