

Incrementando a produtividade de software através de ferramenta RAD e framework

Everton Flávio Rufino Seára¹, Fabiane Barreto Vavassori Benitti^{1,2}

¹Centro de Ciências Tecnológicas da Terra e do Mar (CTTMar) – Universidade do Vale do Itajaí (UNIVALI), 88.302-202 – Itajaí – SC – Brasil

²Departamento de Sistemas e Computação
Universidade Regional de Blumenau (FURB) – Blumenau, SC – Brazil

{everton.flavio, fabiane.benitti}@univali.br

Abstract. *The current software development scenario is increasingly competitive demanding rapid and continuous changes in order to its survival and growth. Once the software development process is complex and demands several resources such as time, there is a trend on developing information systems using fast development methods and tools. Thus Rapid Application Development (RAD) and frameworks are considered powerful alternatives for the practice and evolution of such process. It is recognized that both methods work without integration. In such case, it is believed that the union of both RAD and framework can provide not only a fast software development but also high quality product and easy maintenance. This paper proposes a RAD tool to automate the framework compatible code generation, thus automating web applications developments that use PHP technology.*

Resumo. *O cenário cada vez mais competitivo, também no que diz respeito a desenvolvimento de software, vem exigindo das instituições rápidas e contínuas adaptações para a sua sobrevivência e crescimento. Tendo em vista que o processo de desenvolvimento de software é complexo e consumidor de inúmeros recursos, dentre eles tempo, existe uma grande tendência em desenvolver sistemas de informação utilizando ferramentas e métodos de desenvolvimento rápido. Neste contexto, considera-se RAD (Rapid Application Development) e framework como poderosas alternativas para prática e evolução deste processo. Visto que ambos trabalham sem integração intrínseca, acredita-se que a junção de RAD e framework, possa proporcionar mais do que um desenvolvimento rápido de software, mas, também, um produto de qualidade e de fácil manutenção. Portanto, este artigo propõe uma ferramenta RAD capaz de automatizar a geração de código compatível com framework, a fim auxiliar no desenvolvimento de aplicações web que utilizam tecnologia PHP.*

1. Introdução

O desenvolvimento de software envolve um processo complexo e consumidor de recursos como tempo, dinheiro e pessoal, o que evidenciá-se na medida em que os sistemas tornam-se mais complexos. Tendo em vista as profundas mudanças ocorridas

na última década, Malcolm (2001) afirma que existe uma grande tendência em desenvolver sistemas utilizando ferramentas e métodos de desenvolvimento rápido.

No contexto do desenvolvimento rápido de software, vale destacar dois conceitos importantes abordados neste artigo: RAD (*Rapid Application Development – Desenvolvimento Rápido de Aplicações*) e *frameworks*. RAD, segundo Agarwal *et al.* (2000), refere-se a um ciclo de vida projetado para fornecer um desenvolvimento muito mais rápido que os tradicionais processos de software, além de prover resultados de maior qualidade. Já *frameworks*, de acordo com Gamma *et al.* (2000), são um conjunto de classes cooperantes que constroem um projeto reutilizável para uma específica classe de software e, segundo Johnson e Foote (1988 *apud* BOSCH *et al.*, 1999) representam um projeto abstrato para soluções de uma família de problemas relacionados.

Tendo em vista que ambas abordagens prometem acelerar a produtividade no desenvolvimento de software, acredita-se que a junção de ambas as soluções, RAD e *framework*, proporcione mais do que um desenvolvimento rápido de software, mas também um produto de qualidade e de fácil manutenção por parte dos desenvolvedores.

Sendo assim, este artigo apresenta uma ferramenta RAD, intitulada RADPHP (Rapid Application Development Hypertext Preprocessor), que tem como objetivo acelerar o desenvolvimento de sistemas *web*, a partir da geração automática de código-fonte padronizado, ou seja, baseado em *framework* (neste caso, o *framework* PHPLib-UNIVALI). A escolha por este *framework* baseia-se em sua ampla utilização (e conseqüente validação) nos sistemas *web* da Univali, constituindo um padrão para o desenvolvimento de todas as aplicações da Intranet. Portanto, a ferramenta apresentada teve como motivação inicial incrementar a produtividade do departamento de TI da Univali.

Deste modo, a Seção 2 apresenta o *framework* PHPLib-UNIVALI, a Seção 3 aborda os aspectos pertinentes a estrutura da ferramenta RADPHP, a Seção 4 apresenta o seu funcionamento e as considerações finais são expostas na Seção 5.

2. Framework PHPLib-UNIVALI

Para Peters e Pedrycz (2001), um *framework* é uma combinação de componentes reutilizáveis que simplifica a construção de aplicações. Neste contexto, a proposta do *framework* PHPLib-UNIVALI é diminuir a complexidade do desenvolvimento de aplicações *web* que utilizam tecnologia PHP no âmbito da Universidade do Vale do Itajaí.

De forma geral, o *framework* PHPLib-UNIVALI é composto por duas camadas distintas, apresentadas na Figura 1. O pacote “PHPLib-UNIVALI” é o seu alicerce e contém um conjunto de 8 classes co-relacionadas que ditam a arquitetura, definem a estrutura geral e as responsabilidades de cada um dos objetos dentro da aplicação. O pacote “PHPLib-UNIVALI-Func” compreende um grupo de arquivos de funções que se agregam ao *framework* e podem, eventualmente, compor as aplicações que o utilizam como base.

Além das duas camadas citadas também se observa na Figura 1 o pacote “Aplicação”, que não compõe o *framework*, mas representa um sistema desenvolvido utilizando-o e que herda todas as suas características. Assim, o desenvolvedor pode

centrar seus esforços nas regras de negócio da aplicação, uma vez que a estrutura básica do aplicativo foi definida previamente, o que acarretará na redução do tempo de desenvolvimento do projeto.

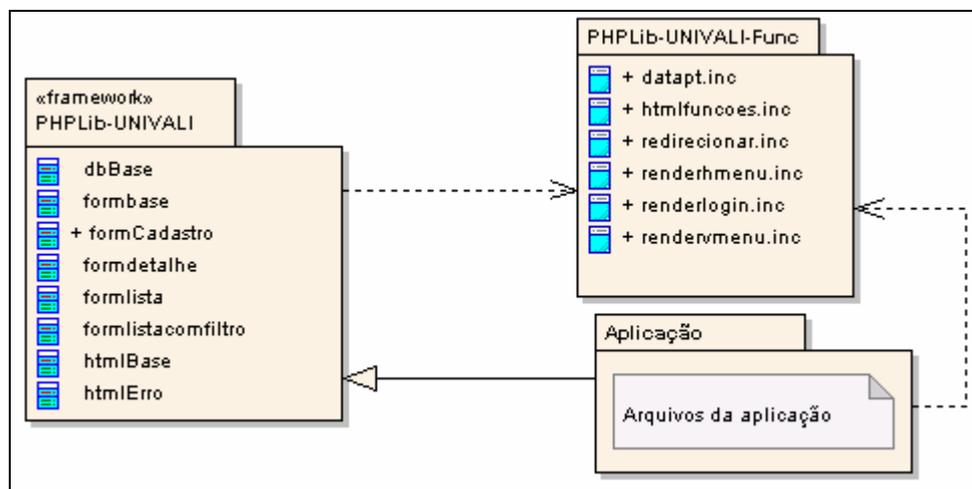


Figura 1. Modelo de pacotes do framework PHPLib-UNIVALI

O grande enfoque do *framework* PHPLib-UNIVALI destina-se às classes de formulários¹ (listagem, detalhe e cadastro) onde ocorre maior parte do trabalho por parte dos desenvolvedores. Assim, o *framework* demarca a estrutura e pré-define os componentes que farão parte desses formulários, além de encapsular o tratamento de erros e métodos de conexão com banco de dados.

3. Arquitetura da Ferramenta

A arquitetura da ferramenta RADPHP foi projetada em duas camadas distintas, (i) camada de interface, que constitui o módulo de interação com o usuário (apresentado na Seção 4, juntamente com um exemplo de uso da ferramenta); e (ii) camada de aplicação, que contém a implementação de cada um dos módulos necessários para o seu funcionamento. É conveniente, antes de detalhar a estrutura de cada um dos módulos, apresentar as principais funções desempenhadas pela ferramenta. Dessa forma, a Figura 2 exibe o diagrama de casos de uso da ferramenta RADPHP.

¹ **Listagem:** retorno de uma consulta SQL envolvendo vários registros. **Detalhe:** apresenta a informação detalhada de uma tupla selecionada no formulário de listagem. **Cadastro:** permite efetuar inclusão, alteração e exclusão de registros no banco de dados.

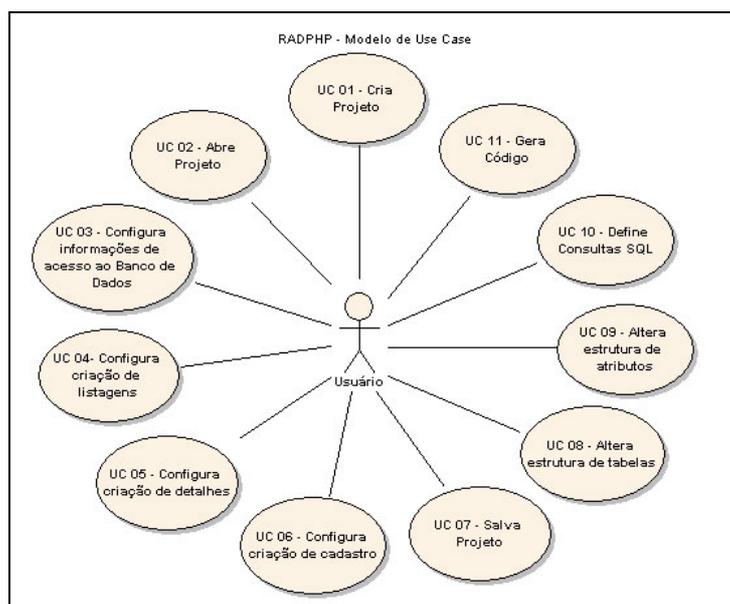


Figura 2. Diagrama de Casos de Uso da ferramenta RADPHP

A utilização da ferramenta está diretamente relacionada à criação de um projeto (UC 01) que irá fornecer a estrutura necessária para a execução das demais funções. Uma vez que o projeto foi criado, este pode ser manipulado (UC 03, UC 04, UC 05, UC 06, UC 08, UC 09 e UC 10), ou seja, uma aplicação pode ser desenvolvida através da ferramenta. O caso de uso UC 07 é responsável por salvar as informações do projeto (em formato XML) e o caso de uso UC 02 em abrir um projeto salvo anteriormente. Por fim, o caso de uso UC 11 descreve o processo de geração de código da aplicação.

Referente a camada de aplicação (mencionada acima) que contém a implementação dos módulos necessários para o funcionamento da ferramenta RADPHP, estão divididos em (i) Analisador de Linguagem SQL (ALS); (ii) Estrutura de Armazenamento (EA); e (iii) Gerador de Código (GC).

O módulo ALS é incumbido de converter a estrutura do banco de dados da aplicação – inicialmente em linguagem SQL (*Structured Query Language*) – para uma forma entendida pela ferramenta RADPHP. Este módulo foi implementado em conformidade com o banco de dados Oracle 8i e possui funções de reconhecimento de tabelas, atributos, tipos e tamanho de campos, comentários e *constraints* (reconhecidos através de analisadores léxico, sintático e semântico). É importante destacar que o desenvolvimento do ALS foi amparado pela ferramenta GALS, desenvolvida por Gesser (2003), que forneceu auxílio na geração dos analisadores léxico e sintático a partir de uma gramática independente de contexto.

O módulo EA é responsável pela estruturação e armazenamento das informações dos projetos criados através da ferramenta. Para tal, esse módulo compreende funções de leitura e escrita em documentos XML (*eXtensible Markup Language*) através do método DOM (*Document Object Model*) e de estruturação/validação de informação baseando-se em XML *Schema*. Tendo em vista que as informações do projeto são armazenadas em documentos XML, não existe necessidade de utilização de banco de dados, o que facilita a portabilidade da informação.

Por fim, o módulo GC é o responsável pela geração do código da aplicação respeitando a padronização definida pelo *framework* PHPLib-UNIVALI. Seu funcionamento dá-se através de *templates* os quais contém *tags* pré-definidas pela ferramenta (*tags* iniciadas por “<%rhp”) que são substituídas através de um *parser*, observando as configurações realizadas pelo usuário, gerando assim o código PHP correspondente. A Figura 3 apresenta como exemplo parte de um arquivo de *template* utilizado na ferramenta.

```
// SQL
$sql = "
    SELECT DISTINCT
        <%rhpSelectedFields%>
    FROM
        <%rhpSelectedTables%>
        <%rhpWhereCondition%>
        <%rhpOrderBy%>";
```

Figura 3. Parte do *template* para geração de código

Em sua totalidade, a ferramenta RADPHP é composta por 20 arquivos de modelo, divididos em três grupos: (i) elementos, formado pelos componentes do *framework*, como campo de edição de textos, lista de valores, etc.; (ii) ações, que contem os modelos responsáveis pela interação com o banco de dados; e (iii) formulários, constituído dos modelos para definição dos formulários de listagem, detalhe, cadastro e consulta.

4. Ferramenta RADPHP

A ferramenta RADPHP foi concebida com a intenção de fornecer a funcionalidade necessária para a criação de projetos que empreguem o *framework* PHPLib-UNIVALI. Sendo assim, a ferramenta não define um processo de desenvolvimento específico para sua utilização, apenas considera como requisito que o processo envolva a elaboração de um artefato de projeto contendo Diagrama Entidade-Relacionamento, pois o esquema relacional será entrada para a implementação (etapa onde a ferramenta atua diretamente).

Sendo assim, a utilização da ferramenta está diretamente vinculada à criação de um projeto, o qual armazena as informações referentes ao banco de dados da aplicação (obtidas através da análise de arquivo SQL pelo módulo ALS), dos formulários e consultas geradas a partir da ferramenta. Uma vez que um projeto foi criado, inicia-se o desenvolvimento da aplicação, que consiste basicamente na configuração das informações de conexão com o banco de dados e configurações dos formulários.

A configuração dos três tipos de formulários fornecidos pelo *framework* (listagem, detalhe e cadastro) ocorre de forma similar. Para tal, o usuário deve selecionar a tabela desejada, conforme apresenta a Figura 4, e manipular as informações do formulário e seus atributos, como: nomes, rótulos e tamanhos de campos, preenchimento obrigatório de campos e tipo componente a ser utilizado no formulário de cadastro, etc.

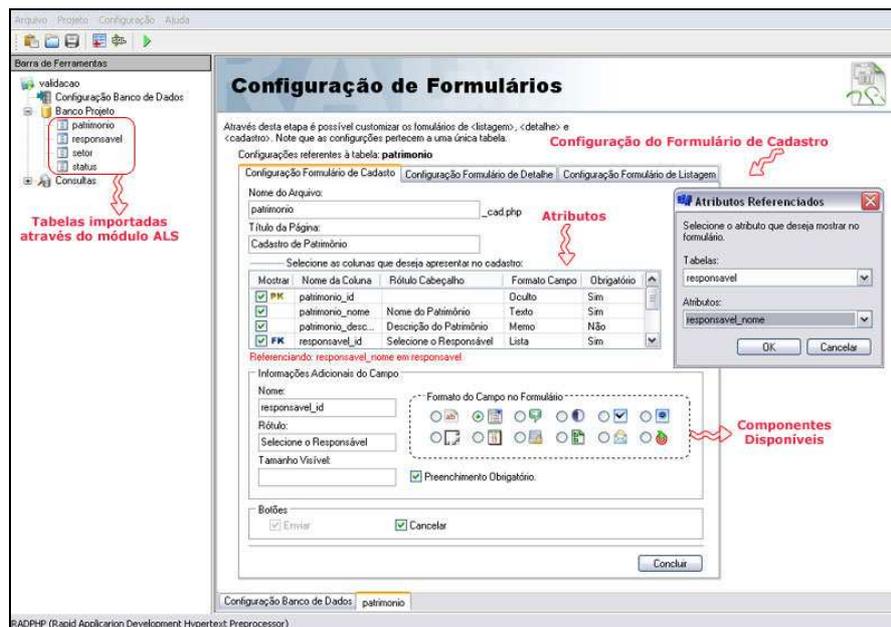


Figura 4. Exemplo de modelo para geração de código

A última etapa refere-se a geração do código fonte observando uma estrutura de diretórios que incorpora o nome do projeto, e sub-pastas contendo os arquivos do *framework* (caso desejado), as imagens e o arquivo de estilos.

Além destas funções já descritas, o usuário pode criar listagens a partir de consultas SQL e modificar a estrutura interna (tabelas e atributos) do banco de dados da aplicação. É importante destacar que toda a estrutura do banco de dados da aplicação é obtida através de um arquivo SQL e mapeada para uma estrutura XML contendo as definições do usuário. Assim, o funcionamento da ferramenta é completamente *off-line*, não necessitando de nenhum tipo de conectividade com banco de dados ou internet.

5. Considerações Finais

É evidente a necessidade de desenvolver software de maneira rápida e eficiente, assim, torna-se cada vez mais viável a utilização de ferramentas RAD para acelerar este processo. Neste contexto, considera-se este trabalho uma possível solução para a necessidade apresentada, uma vez que a abrange conceitos referentes a RAD e *framework*, o que amplia, consideravelmente, a velocidade e padronização no desenvolvimento de software. Além disso, considera-se que a ferramenta estende seus benefícios para outras 2 etapas, além da implementação: (i) etapa de testes, onde tem-se a diminuição dos erros de sintaxe com a geração automatizada do código, e (ii) na manutenção do sistema desenvolvido, pois assegura-se a utilização de uma estrutura de projeto padronizada.

Um aspecto importante a ser destacado é que após a configuração da aplicação, todos os dados são armazenados em formato XML, descrito através de um *Schema*. Sendo assim, com pequenas alterações na ferramenta, torna-se possível a geração de código-fonte padronizado para diferentes linguagens de programação. Para tal, se faz necessário o desenvolvimento de um *framework* (baseado no *framework* PHPLib-UNIVALI) na linguagem desejada, bem como a criação e configuração dos arquivos de modelo, apresentados na Seção 3.

Visando identificar a produtividade da ferramenta, um projeto piloto teve sua produtividade medida através da Análise de Pontos de Função, a qual permite determinar o tamanho de uma aplicação a partir de sua funcionalidade (VAZQUEZ, 2003). Com o emprego desta técnica detectou-se que o projeto piloto contendo 81 pontos por função ajustados (103 pontos por função brutos) obteve o tempo de desenvolvimento de 13 minutos, tendo-se gerado uma aplicação de aproximadamente 2000 linhas de código. Sendo assim, estima-se que a produtividade obtida com o uso da ferramenta RADPHP em relação aos pontos por função ajustados seja de 373 pontos por hora.

Embora os resultados obtidos até o momento com a ferramenta sejam satisfatórios, melhorias podem ser incorporadas, como por exemplo, abranger diferentes bancos de dados (atualmente é restrito ao Oracle), permitir a utilização de *views* e possibilidade de definir o *layout* da aplicação a partir da ferramenta.

6. Referências

- AGARWAL, Ritu. et al. "Risks of Rapid Application Development". **Communications of ACM**, New York, v. 43, n.11, p.177-188, Nov. 2000.
- BOSCH, Jan. et al. "Object-oriented frameworks - problems & experiences". **Building Application Frameworks**. New York: John Wiley & Sons, 1999. p.55-82.
- GAMMA, Erich. et al. "Padrões de projeto: soluções reutilizáveis de software orientado a abjetos". Porto Alegre: Bookman, 2000.
- GESSER, Eduardo Carlos. "GALS: gerador de analisadores léxicos e sintáticos". 2002. 150 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação), Centro Tecnológico, Universidade Federal de Santa Catarina, Florianópolis, 2003.
- MALCOLM, Eva. "Requirements acquisition for Rapid Applications Development". In: *Information & Management*, Northampton, v. 39, n. 2, p. 101-107, Dez. 2001.
- PETERS, J. F.; PEDRYCZ, W. "Engenharia de software". Rio de Janeiro: Campus, 2001.
- VAZQUEZ, Carlos Eduardo et al. "Análise de pontos de função : medição, estimativas e gerência de projetos". São Paulo: Érica, 2003.