

UNIVERSIDADE FEEVALE

CRISTIAN CARDOSO

GERADOR DE CÓDIGO PARA APLICAÇÕES WEB
(Título Provisório)

Anteprojeto de Trabalho de Conclusão

Novo Hamburgo, Setembro de 2010.

CRISTIAN CARDOSO

ctncardoso@ctncardoso.com.br

GERADOR DE CÓDIGO PARA APLICAÇÕES WEB
(Título Provisório)

Universidade Feevale
Instituto de Ciências Exatas e Tecnológicas
Curso de Sistemas de Informação
Anteprojeto de Trabalho de Conclusão

Professor orientador: Juliano Varella de Carvalho

Novo Hamburgo, Setembro de 2010

RESUMO

Com a popularização do uso da Internet e o grande crescimento da utilização de sistemas *on-line*, juntamente com os constantes avanços na área de Tecnologia da Informação, as tecnologias inovadoras viabilizaram o desenvolvimento de aplicações e ferramentas complexas sob o ambiente web. Uma destas tecnologias que tem ganhado grande ênfase é os geradores de códigos em linguagens de programação orientadas a objetos, voltadas à programação web, que tem como principal objetivo, realizar a construção de funcionalidades das aplicações de maneira mais eficiente e segura e de forma padronizada, garantindo também além da diminuição do tempo de desenvolvimento, também a redução dos custos do projeto. Neste trabalho será realizado um estudo e comparação entre sistemas geradores de código existentes para a linguagem de programação PHP, que é uma das principais linguagens existentes para Internet, esta linguagem é gratuita e muito conhecida por sua simplicidade, praticidade e portabilidade. Será proposto também o desenvolvimento de um novo gerador de código que reúna os conceitos mais atuais em programação para a Internet, tendo como base os *frameworks* Zend Framework e JQuery, existentes no mercado atual, a fim de agilizar o desenvolvimento de sistemas web, utilizando a linguagem de programação PHP.

Palavras-Chave: Gerador de código, CRUD, Framework, Zend Framework, PHP.

SUMÁRIO

MOTIVAÇÃO	4
OBJETIVOS	8
METODOLOGIA	9
CRONOGRAMA	10
BIBLIOGRAFIA	11

MOTIVAÇÃO

No início da *World Wide Web* (entre 1990 e 1995), os sites web eram formados por pouco mais do que um conjunto de arquivos de hipertextos ligados, que apresentavam informações usando texto e alguns gráficos. Com o passar do tempo, a popularização do uso da Internet, não somente para páginas web, mas também para a utilização de sistemas *on-line*, fez com que as indústrias de *software* tenham como uma de suas principais metas a construção de sistemas e sites de alta qualidade e em um curto espaço de tempo. Na grande maioria das vezes estas metas não são alcançadas devido ao aumento da complexidade dos sistemas, fazendo com que muitos projetos ultrapassem seus prazos, custos ou que o software apresente baixa qualidade.

Na construção de sistemas, normalmente se observa a reutilização de códigos desenvolvidos anteriormente. Na maioria das vezes esta reutilização é feita através de "copiar/colar/modificar". A qualidade do código produzido varia conforme a quantidade destas operações de reuso. Quanto maior a quantidade de adaptações, pior a qualidade, devido a estas operações frequentemente levarem a erros, conforme (PRESSMAN, 1995) toda vez que uma mudança é introduzida num procedimento lógico complexo, o potencial de erros cresce.

Uma forma de reduzir este tipo de reutilização seria através da utilização de programas de geração de código. Um gerador de código é uma ferramenta de auxílio ao processo de desenvolvimento de sistemas que atua na fase de implementação do projeto, gerando o código-fonte que seria criado pelo programador. De forma geral, esses geradores criam o código-fonte com base nas informações existentes no modelo de dados. Isso significa ganho de produtividade, redução de tempo e qualidade no código gerado.

A qualidade do código varia durante o ciclo de vida de um projeto, podendo começar no alto e decair ou vice-versa. De acordo com (PAULA FILHO, 2003) uma fração significativa das modificações de manutenção introduz, por sua vez, novos defeitos. O código gerado aumenta sua qualidade com o tempo, pois os erros encontrados podem ser uniformemente reparados na base do código.

Outra maneira de agilizar o processo de desenvolvimento, a fim de evitar erros, seria com a utilização de *framework*. Conforme (LISBOA, 2009), utilizar um *framework* de desenvolvimento significa reaproveitar o trabalho e o conhecimento de outra(s) pessoa(s) e seguir suas orientações. A palavra "reaproveitar" nos remete à orientação a objetos. O reaproveitamento de código não é privilégio nem invenção da orientação a objetos, mas esse paradigma possui técnicas eficazes para implementá-lo.

A decisão por usar um *framework* de desenvolvimento reside na necessidade de estruturar os projetos de software, devido à grande complexidade que os mesmos alcançaram. Um projeto estruturado reduz custos, aumenta a qualidade da aplicação e reduz o tempo de desenvolvimento (LISBOA, 2009).

Ao utilizar um *framework*, muitas vezes nos deparamos com uma nova forma de programar ou até mesmo de se pensar sobre um sistema. Conforme (MINETTO, 2007) muitas vezes surge a sensação de estar "engessado", pois é preciso fazer as coisas da forma que o *framework* trabalha, de modo que qualquer coisa diferente requer um empenho melhor, devido a necessidade de aprender uma nova sintaxe, utilizar convenções de nomes de arquivos, variáveis e tabelas de banco de dados. Contudo, as vantagens a médio e longo prazo fazem valer esse pequeno esforço inicial.

Como todos os desenvolvedores que usam determinado *framework* programam usando as mesmas convenções, classes e bibliotecas, a manutenção de um programa torna-se muito mais fácil, mesmo que determinado *script* tenha sido escrito por outra equipe há vários meses. Isto contribui para que novos desenvolvedores ingressantes em uma equipe possam rapidamente se inteirar da forma como o *framework* é trabalhado, reduzindo custos e tempo para treinamento (MINETTO, 2007).

Atualmente existem diversos *frameworks* PHP disponíveis, entre os mais conhecidos podem ser citados: Zend Framework, CakePHP, Symfony, CodIgniter e Yii. A escolha de um *framework* para desenvolvimento envolve alguns fatores que podem variar bastante

dependendo do projeto. Mas existem pontos em comum, tais como a quantidade e a qualidade da documentação, a facilidade de encontrar desenvolvedores, o envolvimento da comunidade em relação ao *framework*, a curva de aprendizado, entre outros (FREIRE, 2009).

Outra vantagem dos *frameworks* de acordo com (SCHMITZ, 2010) é que eles facilitam a relação com bancos de dados e a camada externa de exibição. Isso acontece devido ao fato de os *frameworks* implementarem padrões de projeto que otimizam a comunicação com o banco de dados, e também deixarem a aplicação mais segura.

Com estes *frameworks* é possível trabalhar com a metodologia RAD, *Rapid Application Development* (Desenvolvimento Rápido de Aplicações), com isso é possível fazer tarefas triviais do desenvolvimento de uma aplicação, tais como criar um novo projeto, criar *controllers*, *view*, *models*, de forma muito simples e rápida.

Para este trabalho será utilizado o Zend Framework, pois algo que deve ser levado em consideração além da quantidade e qualidade da documentação, conforme (FREIRE, 2009) é a existência de grandes empresas apoiando o desenvolvimento, tais como Zend, Adobe, IBM, entre outras. Outro fator decisivo para a escolha foi o processo de desenvolvimento do framework ser bem definido e controlado pela Zend. As alterações são feitas tomando-se cuidado para manter compatibilidade com versões anteriores, o que é muito importante. Empresas como Google, Microsoft e StrikeIron mantêm parcerias com a Zend para fornecer interfaces de *web services* e outras tecnologias que pretendam disponibilizar aos desenvolvedores do Zend Framework (ZEND, 2010).

Neste primeiro momento este trabalho será desenvolvido para os SGBDs (Sistema Gerenciador de Banco de Dados) PostgreSQL e MySQL por serem uma das alternativas de software livre, além de serem ferramentas *open source*:

- O PostgreSQL é otimizado para aplicações complexas, isto é, que envolvem grandes volumes de dados ou que tratam de informações críticas. Assim, para um sistema de comércio eletrônico de porte médio/alto, por exemplo, o PostgreSQL é mais interessante, já que esse SGBD é capaz de lidar de maneira satisfatória com o volume de dados gerado pelas operações de consulta e venda (HOSTWEB, 2010).

- O MySQL, por sua vez, é focado na agilidade. Assim, se sua aplicação necessita de retornos rápidos e não envolve operações complexas, o MySQL é a opção mais adequada, pois é otimizado para proporcionar processamento rápido dos dados e tempo curto de resposta sem exigir muito do hardware (HOSTWEB, 2010).

Atualmente o Zend Framework não possui um gerador de código capaz de gerar tanto código *front-end* quanto *back-end*. Neste trabalho será proposto o desenvolvimento de uma ferramenta com o objetivo de agilizar o processo de desenvolvimento de software para aplicações Web, onde o usuário configurará a conexão com o banco de dados e informará os dados de acesso. Será realizada uma leitura do banco de dados, exibindo todas as tabelas existentes, e permitindo ao usuário selecionar quais ele utilizará. Então o usuário poderá selecionar os campos das tabelas que comporão o formulário, podendo definir algumas configurações como: título do formulário, título dos campos, se o(s) campo(s) é obrigatório, se existirá alguma máscara a ser aplicada em um determinado campo, etc.

Após o processo de configuração, todas as informações definidas pelo usuário serão enviadas para o módulo PHP que passará a gerar os arquivos de código-fonte, de acordo com a estrutura MVC do Zend Framework.

Além dos arquivos gerados para manipulação do formulário, será criado outro para armazenar as configurações definidas pelo usuário, para que o mesmo possa em um segundo momento editar e gerar uma nova versão sem a necessidade de iniciar o trabalho do zero.

OBJETIVOS

Objetivo Geral

Desenvolver um gerador de código com a linguagem de programação PHP, que crie classes e interfaces a partir de um esquema relacional, integrado à estrutura do Zend Framework.

Objetivos Específicos

- Explorar e caracterizar geradores de código disponíveis no mercado;
- Investigar a estrutura do Zend Framework;
- Utilizar o Zend Framework como estrutura da aplicação;
- Configurar as páginas geradas a partir dos atributos de cada tabela;
- Gerar código automaticamente a partir das configurações definidas pelo usuário;

METODOLOGIA

Para atingir os objetivos propostos serão realizadas as seguintes atividades.

1. Pesquisa sobre geradores de código;
2. Pesquisa sobre Zend Framework e JQuery;
3. Redação do anteprojeto;
4. Entrega do anteprojeto;
5. Pesquisa sobre geradores de códigos existentes;
6. Pesquisa sobre Zend Framework e outros frameworks;
7. Buscar exemplos de geração de código em outros frameworks
8. Desenvolvimento do protótipo da ferramenta proposta;
9. Redação do relatório do Trabalho de Conclusão I;
10. Entrega do texto final do Trabalho de Conclusão I;
11. Desenvolvimento da interface do usuário;
12. Desenvolvimento do gerador de interface;
13. Desenvolvimento do gerador de CRUD (Create, Retrieve, Update e Delete);
14. Avaliação da ferramenta proposta junto a outros desenvolvedores;
15. Redação do relatório do Trabalho de Conclusão II;
16. Entrega do texto final do Trabalho de Conclusão II;
17. Apresentação do trabalho realizado à banca avaliadora.

CRONOGRAMA

Trabalho de Conclusão I

Etapa	Meses			
	Ago	Set	Out	Nov
1	■			
2	■	■		
3	■	■		
4		■		
5		■	■	
6		■	■	
7		■	■	
8			■	■
9			■	■
10				■

Trabalho de Conclusão II

Etapa	Meses			
	Mar	Abr	Mai	Jun
11	■	■		
12		■	■	
13		■	■	
14			■	
15	■	■	■	■
16				■
17				■

BIBLIOGRAFIA

FREIRE, Flávia. **CakePHP Receita simples para quem quer agilidade**. TI Digital. Rio de Janeiro, n° 01, Ano 01, p. 26-36, Abril 2009.

HOSTWEB. **MySQL ou PostgreSQL, qual usar?**. Disponível em:
<<http://blog.hostweb.com.br/mysql-ou-postgresql-qual-usar/hostweb>>. Acesso em: 5 Set. 2010.

LISBOA, Flávio Gomes da Silva. **Zend Framework Componentes Poderosos para PHP**. São Paulo: Novatec. 2009. 352 p.

SCHMITZ, Daniel; STEPHANOU, Lucas; TELLES, Marcos; JUNIOR, Oberaldo Büll; WILHELM, Ivan. **Frameworks PHP: compare os cinco mais famosos e agilize seus projetos**. TI Digital. Rio de Janeiro, n° 15, Ano 02, p. 26-36, Maio 2010.

MINETTO, Elton Luís. **Frameworks para Desenvolvimento em PHP**. São Paulo: Novatec. 2007. 192p.

PAULA FILHO, Wilson de Pádua. **Engenharia de software: Fundamentos, Métodos e Padrões**. 2. ed. Rio de Janeiro: LTC. 2003. 602 p.

PRESSMAN, Roger S. **Engenharia de Software**. São Paulo. Makron Books, 1995. 1056 p.

ZEND. **Manual do ZEND Framework**. Disponível em:
<<http://www.zendframework.com/manual/>>. Acesso em: 20 Jun. 2010.