JHONY MAIKI MASETO

ANÁLISE AVALIATIVA ENTRE FRAMEWORKS DE PHP

Monografia apresentada à UNOCHAPECÓ como parte dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.
Orientador: Elton Luís Minetto

Chapecó - SC, Dezembro de 2006.

ANÁLISE AVALIATIVA ENTRE FRAMEWORKS DE PHP

JHONY MAIKI MASETO

Esta Monografia foi julgada para obtenção do título de Bacharel em Ciência da Computação, na área de concentração e aprovada pelo curso de Ciência da Computação

ORIENTADOR(A): Prof. Elton Luís Minetto

COORDENADOR(A) DO CURSO: Prof.º Mônica Tissiani de Toni Pereira

BANCA EXAMINADORA

PRESIDENTE: Prof. Mônica Tissiani de Toni Pereira

Prof. Valdemar Lorenzon Junior

Prof. Jean Carlos Hennrichs

AGRADECIMENTOS

Agradeço ao meu orientador Elton Luís Minetto pela paciência, incentivo e a disposição em tornar este projeto possível.

Agradeço aos meus pais pela oportunidade de lhes causar orgulho. E a todos que tiveram paciência para aturar meu mau humor constante.

Sumário

Índice de ilustrações

Índice de Tabelas / Quadros

Abreviaturas

API	Application Programming Interface
ASP	Application Service Provider
CRUD	Create, Read, Update, Delete
HTML	HiperText Markup Language
HTTP	HyperText Transfer Protocol
MVC	Modelo-Visão-Controlador
PEAR	PHP Extension and Application Repository
PHP	Hypertext Preprocessor.
SQL	Structured Query Language
WEB	World Wide Web
XML	Extensible Markup Language
GTK	Gimp ToolKit
SO	Sistema Operacional
ID	Digito identificador
TCC	Trabalho de conclusão de curso

RESUMO

Desde a criação dos primeiros produtos de softwares os desenvolvedores se preocupam com o melhoramento e otimização das suas ferramentas de trabalho. Tendo em vista a grande quantidade de ferramentas disponíveis e o pouco tempo que o desenvolvedor possui para analisar qual é a melhor ferramenta para seu tipo de aplicação, pretende-se mostrar através desta análise avaliativa novas ferramentas de auxilio aos desenvolvedores de php, e assim também trazer mais adeptos para os projetos de frameworks existentes.

Com o intuito de melhora das ferramentas pretende-se também apontar aos desenvolvedores, algumas características que seus frameworks não estão contemplando e que seria de interesse da comunidade desenvolvedora que estive-se embutido em seu projeto.

Para facilitar esta definição de qual é a estrutura de framework mais adequada para cada tipo de caso foi desenvolvido um protótipo utilizando os frameworks para melhorar a qualidade desta análise.

ABSTRACT

Since the creation of the first software products, the developers have been worried with the improvements of their tools of work. Having in view the big quantity of tools available and the short time that the developer has to analyze which is the best one for the kind of application, it is intended to show through this evaluative analysis new tools to auxiliary the php developers, and this way, also conquest more followers of the existing projects of framework. With the intention of improvement of the tools also it is going to be pointed to the developers everything that the frameworks are not contemplating and that would be of the developers community's interest that it was part of their project. To facilitate this definition of which is the structure of framework more appropriated to each kind of case, it was developed a prototype using the frameworks to improve the quality of this analysis.

1. INTRODUÇÃO

Desde quando foi criado o conceito de softwares os desenvolvedores se preocupam em criar soluções, que venham a auxiliar quem trabalha com desenvolvimento de produtos e serviços para o mercado consumidor. Por meio desta análise avaliativa venho propor a divulgação das novas soluções e auxiliar a comunidade desenvolvedora colhendo não somente mais membros para os projetos que estão surgindo, mas também auxiliando os desenvolvedores na escolha da melhor solução para cada tipo de caso proposto.

Pretende-se também apontar aos desenvolvedores dos projetos quais foram as dificuldades encontradas ao se trabalhar com as soluções propostas, bem como expô-las para a comunidade. Exibir também o que cada uma das tecnologias contemplam e descrever sobre as soluções propostas.

1.1 ORGANIZAÇÃO DO PROJETO

Este projeto está dividido em vários capítulos para facilitar o melhor entendimento do conteúdo proposto. Segue abaixo uma pequena descrição do que vai ser abordado em maiores detalhes no decorrer deste projeto.

- No capítulo um temos a parte introdutória do projeto que serve para explicar ao usuário de uma forma mais sucinta o que vai ser retratado ao decorrer do projeto.
- No capítulo dois temos uma apresentação breve através de uma pequena introdução sobre frameworks de desenvolvimento, sua conceitualização, suas vantagens e desvantagens onde eles se aplicam e também um pouco da sua metodologia de trabalho.
- No terceiro capítulo temos a apresentação do PHP, como surgiu, suas vantagens e desvantagens, características principais e alguns exemplos de aplicação orientado ao objeto.
- No capítulo quatro abordaremos os detalhes e características dos frameworks escolhidos para a análise avaliativa descrita neste projeto. E também serão abordados alguns requisitos de uso dos frameworks descritos.
- No capítulo cinco temos todo o desenvolvimento dos protótipos, a modelagem utilizada, as adaptações para cada framework, os requisitos da aplicação, detalhes da instalação dos aplicativos, as dificuldades encontradas, algumas telas da aplicação e descrição detalhada de como foram construídos os protótipos. Ainda temos descrito neste capítulo toda a parte que trata da comparação entre os frameworks bem como a explicação de cada item comparado.
- No capítulo seis está presente a descrição das conclusões obtidas com o projeto. E também está descrito a relação de trabalhos futuros, incluindo

alguns temas que poderiam ter sido abordados neste projeto, mas que fugiriam ao escopo inicial do mesmo.

- Descreve-se no capítulo sete toda a referência utilizada para compor este projeto.

2. FRAMEWORKS

Com a necessidade de reutilização de código pelos programadores obteve-se o conceito de criação de sistemas a partir de códigos e objetos já escritos. Com a reunião deste grupo de códigos e objetos obteve-se então o conceito de framework. Desta forma, o principio de todo framework é ser uma solução reusável, estável e bem documentada.

De acordo com (FAYAD, W.E.; SCHMIDT, D,C.; JOHNSON, R.E,1999 apud DE SOUZA,Marcos Vinícius Bittencourt ,2004), “Um framework consiste em um conjunto de classes que se relacionam e representam uma solução incompleta. Um framework é o esqueleto de uma aplicação que pode ser customizado por um desenvolvedor da aplicação".

Ainda de acordo com (JOHNSON, R.; FOOTE,1998 apud DE SOUZA,Marcos Vinícius Bittencourt ,2004) , “Um framework orientado a objetos é um projeto re-utilizável de software definido por um conjunto de classes abstratas e pela maneira pela qual as instâncias dessas classes colaboram entre si”.

No desenvolvimento do software, um framework pode ser considerado uma estrutura de suporte definida em que um outro projeto de software pode ser organizado e desenvolvido. Tipicamente, um framework pode incluir programas de apoio, bibliotecas de código, linguagens de script e outros softwares para ajudar a

desenvolver e juntar diferentes componentes do seu projeto. (WIKIPÉDIA.2006).

Especificamente em orientação a objeto, framework é um conjunto de classes com objetivo de reutilização de um design, provendo um guia para uma solução de arquitetura em um domínio específico de software (WIKIPÉDIA.2006).

Framework se diferencia de uma simples biblioteca (toolkit)[1], pois esta se concentra apenas em oferecer implementação de funcionalidades, sem definir a reutilização de uma solução de arquitetura (design). (WIKIPÉDIA.2006).

Existem várias definições para frameworks e segundo (Sauvé Jacques Philippe.), parte delas abordam fortemente as seguintes características:

- Um framework provê uma solução para uma família de problemas semelhantes.
- Observe que um framework é uma aplicação *quase* completa, mas com pedaços faltando.
- Ao utilizar um framework, seu trabalho consiste em prover os pedaços que são específicos para sua aplicação.

Em resumo, um framework captura funcionalidades comuns em várias aplicações e as disponibiliza em uma estrutura que tende a ser de fácil manuseio e entendimento.

Ainda segundo SAUVÉ Jacques Philippe(SD), existem grandes diferenças em um framework e uma biblioteca de classes orientada a objeto.

Em uma biblioteca de classes, cada classe é única e independente das

1 **Toolkit** é um conjunto de elementos básicos para construção de software. Normalmente são implementados como uma biblioteca de rotinas ou uma plataforma para aplicativos.

outras, em contrapartida em um framework, as dependências ou colaborações já estão embutidas no projeto.

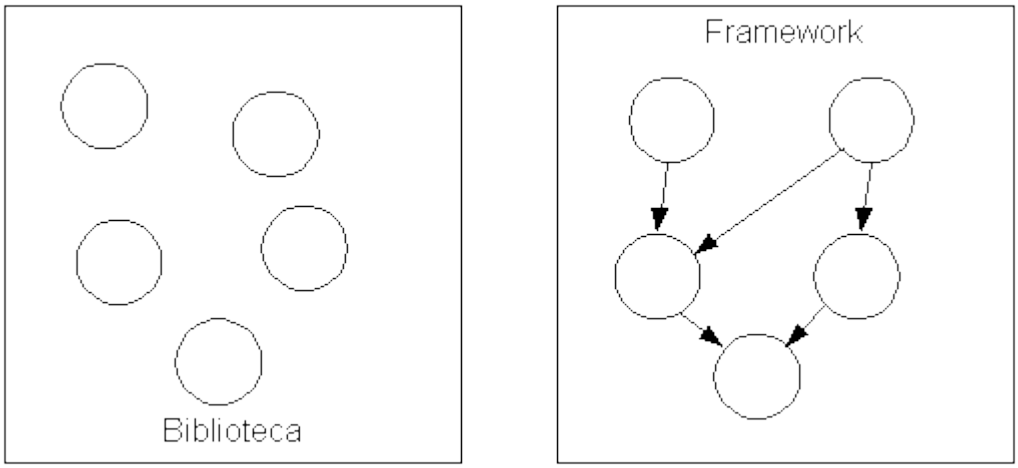

Ilustração 1: Exemplo da dirença entre uma biblioteca de classes e um framework(Sauvé Jacques Philippe)

Ainda segundo SAUVÉ Jacques Philippe(SD), já que a comunicação entre objetos já está definida, o projetista de aplicações não precisa saber quando chamar cada método: é o framework que faz isso.

Ainda segundo SAUVÉ Jacques Philippe(SD), um framework deve ser reusável, mas para ser reusável primeiro ele tem que ser usável, bem documentado e fácil de usar.

- Deve ser extensível:
    - O framework contém funcionalidade abstrata (sem implementação) que deve ser completada

- Deve ser de uso seguro:
    - O desenvolvedor de aplicações não pode destruir o framework

- Deve ser eficiente:
    - Devido a seu uso em muitas situações, algumas das quais poderão necessitar de eficiência

- Deve ser completo:
    - Para endereçar o domínio do problema pretendido

2.1 Vantagens e desvantagens

Segundo CARNEIRO,Rafael(SD), se o framework estiver pronto, os benefícios são claros em termos de:

- Redução de custos;
- Redução de time-to-market[2;]

Motivos:

- Maximização de reuso (análise, design, código, testes);
- Desenvolvedores se concentram em adicionar valor em vez de reinventar a roda;
- Menos manutenção;
    - Fatoração de aspectos comuns a várias aplicações;
    - Uso de herança permite corrigir todas as aplicações com a troca de uma classe-mãe;
    - Estabilização melhor do código (menos defeitos) devido ao uso em várias aplicações;

2 **TIME-TO-MARKET** Tempo que leva entre a análise a sua disponibilidade para a venda.

- Melhor consistência e compatibilidade entre aplicações;

Ainda segundo CARNEIRO,Rafael(SD), os frameworks possuem algumas desvantagens tais como:

- Construir um framework é complexo;
- O Reuso não vem sozinho: deve ser planejado;
- É mais complexo e demora mais fazer uma aplicação tendo que construir um framework em vez de fazer a aplicação do zero;
- Benefícios são realizados em longo prazo, quem pode pensar em longo prazo quando se está competindo "On Internet time"? Poucas empresas;
- Precisa modificar o processo de desenvolvimento e criar novos incentivos;

Para que se torne possível entender o conceito de frameworks devemos estudar um pouco da linguagem ao qual o nosso estudo foi baseado. Estudando o PHP, seus conceitos e suas vantagens e desvantagens podemos ter uma idéia do objetivo principal do uso dos frameworks.

3. PHP

PHP (um acrônimo[3] recursivo para "PHP: Hypertext Preprocessor") é uma linguagem de script Open Source de uso geral, muito utilizada e especialmente guarnecida para o desenvolvimento de aplicações Web embútivel dentro do HTML. (PHP.NET.2005).

O PHP é uma linguagem de programação de computadores interpretada, livre e muito utilizada para gerar conteúdo dinâmico na Web. Apesar de ser uma linguagem de fácil aprendizado e de uso para pequenos scripts dinâmicos e simples, o PHP é uma linguagem poderosa orientada à objetos. (WIKIPÉDIA.2006).

> A linguagem surgiu por volta de 1994, como um subconjunto de scripts Perl criados por Ramus Lerdof. Com as adições de Zeev Suraski e Andi Gutmans, dois programadores israelitas pertencentes ao Technion[4] o instituto israelita de tecnologia, que reescreveram o parser[5], era lançada em 1997 a PHP 3, primeira versão estável e parecida com a linguagem atual. Ao reescrever o parser, foi criado o Zend Engine, que é mantido oficialmente pela empresa Zend em conjunto com a comunidade PHP. Em maio de 2000 veio a público a versão 4, e em julho de 2004, a versão 5, onde a principal mudança foi uma nova API[6] para orientação a objetos provida pelo Zend Engine2.(WIKIPÉDIA.2006).
>
> Trata-se de uma linguagem extremamente modularizada, o que a

3 ***U***m **acrónimo** ou sigla é um agrupamento das iniciais de várias palavras. Acrônimos recursivos são acrônimos onde a expansão inclui o próprio termo, como na definição de funções recursivas. PHP: Hypertext Pre-Processor (Originalmente, PHP significava Personal HomePage)

4 O **Technion** - Instituto Israelita de Tecnologia - é uma universidade em Haifa, Israel. Fundada em 1924, é a universidade mais antiga de Israel. O Technion é famoso pela ciência e engenharia, mas também oferece bons cursos de arquitetura e medicina, entre outros.

5 Um **Parser** é um programa de computador (ou apenas um componente de um programa) que serve para analisar a estrutura gramatical de uma entrada

6 **API**, de Application Programming Interface (ou Interface de Programação de Aplicativos) é um conjunto de rotinas e padrões estabelecidos por um software para utilização de suas funcionalidades por programas aplicativos -- isto é: programas que não querem envolver-se em detalhes da implementação do software, mas apenas usar seus serviços.

> torna ideal para instalação e uso em servidores web. Diversos módulos são criados no repositório de extensões PECL (PHP Extension Community Library) e alguns destes módulos são introduzidos como padrão em novas versões da linguagem. É muito parecida, em tipos de dados, sintaxe e mesmo funções, com a linguagem C e com a C++. Pode ser, dependendo da configuração do servidor, embutida no código HTML. Além disso, destaca-se a extrema facilidade com que PHP lida com servidores de base de dados, como MYSQL, PostgreSQL,Microsoft SQL Server e Oracle. (WIKIPÉDIA. 2006).

Quanto ao modo de execução dos scripts PHP, destaca-se:

> O que distingui o PHP de algo como Javascript no lado do cliente é que o código é executado no servidor. Se você tivesse um script similar ao do exemplo descrito abaixo em seu servidor, o cliente receberia os resultados da execução desse script, sem nenhum modo de determinar como é o código fonte. Você pode inclusive configurar seu servidor para processar todos os seus arquivos HTML como PHP, e então não haverá nenhum modo dos usuários descobrirem que se você usa essa linguagem ou não. (The PHP Group.2006).

Um exemplo introdutório:

```
<html>
  <head>
    <title>Exemplo</title>
  </head>
   <body>
     <?php
       echo "Olá, Eu sou um script PHP!"; //O servidor executa este comando
      ?>                                  // no momento do carregamento da página.
   </body>
</html>
```

Tabela 1: Um exemplo básico de código PHP embutido no HTML.(the PHP Group.2006)

Existem iniciativas para utilizar o PHP como linguagem de programação de sistemas cliente-servidor. A mais notável é a PHP-GTK. Define-se como um conjunto do PHP com a biblioteca, portada do C++ , GTK fazendo assim softwares inter-operacionais entre Windows e Linux.(WIKIPÉDIA.2006).

Vantagens em utilizar o PHP:

Segundo SOUSA.,Sandro(2005), as principais vantagens do PHP são:

- Licença gratuita;
- Plataforma (SO) gratuita para se rodar ele (GNU/Linux) ;
- Velocidade de processamento;
- Eficiência de processamento;
- Métodos de segurança eficientes;
- Roda em qualquer tipo de plataforma (SO);
- Código fonte livre;
- Exceptions (para controle de fluxo);
- Orientação a objetos;
- É a linguagem Web mais popular e que mais cresce (em ritmo bem acelerado) no mercado segundo netcraft[7];
- Possibilita a utilização dos maiores e mais utilizados Bancos de dados no mercado (Adabas D, InterBase, PostgreSQL, dBase, FrontBase, SQLite, Empress, mSQL, Solid, FilePro, Direct MS-SQL Sybase, Hyperwave, MySQL, Velocis, IBM DB2, ODBC, Unix dbm, Informix, Oracle (OCI7 e OCI8), Ingres, Ovrimos, Firebird) sem necessitar de configuração externa;
- Está sempre em atualização e tendo corrigidas falhas e adicionados novos recursos;
- É mais estável consumindo menos recursos de hardware do servidor;
- Flexibilidade;

7 **Netcraft** é uma companhia de serviços muito conceituada, pode ser visto em mais detalhes em: http://news.netcraft.com/about-netcraft

- Componentes nativos, não dependendo de componentes externos para algumas funcionalidades básicas;
- Documentação, controle e reportamento de erros;
- Comunidade de desenvolvimento super participativa e prestativa;
- Planos de hospedagem Web (na grande maioria dos casos) mais baratos e sem nenhum custo extra para a utilização do MySQL em conjunto com o PHP;

Desvantagens no uso do PHP:

Mesmo com uma grande lista de vantagens de PHP como a acima citada alguns autores consideram como desvantagens:

- Segundo CANAL Html.(2006), há uma centralização incômoda das variáveis, e propenso a muitos bugs, nos quais um programador desatento pode deixar uma brecha para uma invasão;
- Segundo ARSYS Internet S.L.(2006), Poucos são os inconvenientes que podemos encontrar com esta linguagem de script. Tratam-se mais bem de pontos menos favoráveis, mas em nenhum caso, em desvantagem face ao uso de scripts em Perl[8];

Um destes pontos é:

- A depuração de erros, ainda que comum em todas as linguagens de script (e mais destacado no próprio Perl);

O outro é:

8 **Perl** é uma linguagem de programação estável e multiplataforma, usada em aplicações de missão crítica em todos os setores, e é bastante usada para desenvolver aplicações web de todos os tipos.

- O fato de o PHP ser uma linguagem especificamente concebida para a criação de scripts web, faz que esteja em desvantagem para realizar outras tarefas, em relação às linguagens de propósito gerais como Perl. No entanto, no desenvolvimento de aplicações web,o PHP possui um ótimo desempenho;

Uma das grandes desvantagens do php é realmente a questão da mistura dos códigos de php com as tags do html, sendo separados apenas por blocos php. Em contrapartida disto, pode-se utilizar uma série de técnicas e padrões de projetos, dentre estes o mais conhecido é o "MVC[9] ". E uso dos templates.

3.1 PHP4

Foi oficialmente lançada em Maio de 2000, quase dois anos após o seu predecessor, o PHP 3.0. Além do melhoramento da performance, o PHP 4.0 incluiu outras características chaves como o suporte para varios servidores WEB, sessões HTTP, buffer de saída, maneiras mais seguras de manipular input de usuários e muitas construções novas na linguagem.

Implementou principiamente as características de Orientação Objeto.(The PHP Group.2005).

9 **MVC** está descrito em maiores detalhes no capítulo que trata do comparativo entre frameworks.

3.2 PHP5

Uma das principais características do PHP5 certamente é a implementação do conceito de programação orientada ao objeto em seu projeto. O que com certeza fez com que muitos programadores vissem o PHP com outros olhos. Até a versão 4 o PHP não suportava todas as características que uma linguagem de programação orientada à objeto deve conter. Com a reescrita do núcleo da linguagem ocorrida na versão 5 isso mudou.

3.2.1 Orientação à Objeto e conceitos

Neste tópico são apresentados alguns conceitos básicos de orientação a objeto e sua relação com PHP.

"**Objeto**, representa uma coisa física, tangível, uma idéia ou conceito. Possui um estado (o que ele sabe) e um comportamento (o que ele é capaz de fazer, como ele reage a estímulos externos)".(LOZANO,Fernando.2002).

Também pode-se afirmar que um objeto é qualquer coisa, real ou abstrata, a respeito do qual armazenamos dados e os métodos de manipulação destes dados.(MARTIN, James.1995).

Exemplo para instanciar objetos em PHP:

```
<?php
	// criar instancia
	$objeto = new NomeDaClasse;
	// chamada de método
	$objeto->NomeDoMetodo();
?>
```

Tabela 2: Instanciação de objetos em php (DALL´OGLIO, Pablo.2005).

Classe: Uma classe é uma implementação de um tipo de objeto. Ela especifica uma estrutura de dados ou métodos operacionais permissíveis que se aplicam a cada um de seus objetos.(MARTIN, James.1995).

“Classe é um "molde" para a criação de objetos, fornecendo o seu comportamento padrão e a definição de todos os seus estados possíveis”.(LOZANO,Fernando.2002).

Exemplo de uma classe em PHP:

```
<?
# Classe Pessoa
Class Pessoa
	{
	function Pessoa($nome)
		{
			$this->nome = $nome;
		}
		function GetNome()
		{
			return $this->nome;
		}
	}

# Função CriaPessoa
function CriaPessoa($nome)
	{
	return new Pessoa($nome);
	}
?>
```

Tabela 3: Exemplo de classes em PHP (DALL´OGLIO, Pablo.2005).

Herança: Ocorre quando uma classe filho herda atributos de uma classe pai.

Especialização. Uma nova classe pode ser definida em termos de uma classe pai, herdando o seu comportamento. A nova classe especializa a classe pai, definindo apenas onde o seu comportamento deve ser diferente. (LOZANO,Fernando.2002).

Segue abaixo um exemplo de especialização:

```
<?php
	include "ContaCorrente.php";

	$conta = new ContaCorrente (1000.0);
	echo ("Saldo inicial: {$conta->saldo}<BR>");
	$conta->saque (150.0);
	echo ("Novo saldo: {$conta->saldo}<BR>");
?>

Include_once "ContaCorrente.php";

class ContaEspecial extends ContaCorrente
{
	var $limite;
	function ContaEspecial ($valor, $limite) {
		$this-> ContaCorrente ($valor);
		$super ($valor);
	}
	function saque ($valor) {
		if ($this->saldo + $this->limite ->= $valor)
			$saldo = $valor;
	}
}
```

Tabela 4: Exemplo do uso de herança em PHP (DALL´OGLIO, Pablo.2005).

Encapsulamento: Encapsulamento é o ato de ocultar do usuário os detalhes de implementação de um objeto. .(MARTIN, James.1995).

Exemplo de encapsulamento:

```
<?php
	class Predio {
		private $cor;
		private $coresPossiveis = array("red", "green", "blue");

		function mudaCor($cor) {
			if (in_array($cor, $this->coresPossiveis)) {
				$this->cor = $cor; # muda o valor da propriedade
			}		# deste objeto.
		}
	}
	$umPredio = new Predio();
	$umPredio->mudaCor("yellow"); //não altera o objeto
	$umPredio->mudaCor("red"); //altera o objeto
?>
```

Tabela 5: Exemplo de encapsulamento (MORAIS, João Cruz.2004).

"**Instância** é uma ocorrência particular, identificada, de um objeto de uma determinada classe, com seu estado particular, independente de outras instâncias da mesma classe".(LOZANO,Fernando.2002).

Segue abaixo exemplo de Instância:

```
<?php
        $instancia  = new UmaClasseQualquer();
        $atribuido  = $instancia;
        $referencia =& $instancia;

        $instancia->var = '$atribuido vai ter este valor.';
        $instancia = null; // $instancia e $referencia ficam null

        var_dump($instancia);
        var_dump($referencia);
        var_dump($atribuido);
?>
NULL
NULL
object(UmaClasseQualquer)#1 (1) { ["var"]=> string(30) "$atribuido vai ter este
valor." }
```

Tabela 6: Exemplo de instância em PHP (MORAIS, João Cruz.2004).

"**Polimorfismo**. A mesma mensagem, quando enviada para objetos de classes diferentes, executa código particular da classe, mesmo que quem enviou a mensagem não tenha conhecimento do tipo específico de objeto sendo referenciado". (LOZANO,Fernando.2002).

Segue abaixo um exemplo que ilustra a utilização do polimorfismo:

```
<?php
        class Relogio {
                function descricao() {
                echo "Indico as horas que são… agora!";
                }
        }
        class Livro {
                function descricao() {
                        echo "Uma série de páginas com frases escritas";
                }
        }
        $objectos = array(new Relogio(), new Livro());
        foreach($objectos as $objecto)
                echo $objecto->descricao();
?>
```

Tabela 7: Exemplo com Polimorfismo (MORAIS, João Cruz.2004).

“Agregação e Composição. Objetos podem conter outros objetos como partes constituintes, imitando o mundo real onde objetos são construídos em função de outros objetos. Podemos ou não expor as partes constituintes como parte da interface de um objeto”. (LOZANO,Fernando.2002).

Após ter tomado conhecimento dos aspectos básicos da linguagem PHP, e conhecida a forma de trabalho da linguagem, faz-se necessário a apresentação dos frameworks que vão ser estudados neste projeto. No próximo capítulo descreve-se os frameworks suas características e forma de utilização.

4. FRAMEWORKS DE PHP

Neste capítulo estão descritos em maiores detalhes os três frameworks que serão abordados neste projeto para o desenvolvimento do protótipo.

4.1 PRADO-PHP5 FRAMEWORK

"O PRADO é um PHP 5 Framework baseado em componentes e eventos, que torna o modelo de programação WEB muito similar ao ASP.NET. Criado por Qiang Xue, o PRADO foi inicialmente inspirado no projeto Apache Tapestry[10]. Durante o design e implementação, o Borland Delphi e o ASP.NET tiveram um papel importante na definição do Framework. Aqueles que já conhecem essas tecnologias vão se sentir mais confortáveis no seu entendimento".(FRAMEWORK, Prado.2005).

"O PRADO ainda possui vários recursos como (viewstate, sessions, caching, validação de formulários, autenticação e autorização)[11]. Além de tornar possível a criação de componentes de forma simples e prática".(Framework, Prado.2005).

Assim como o ASP.NET O Prado Framework utiliza uma técnica que define um campo escondido chamado "VIEWSTATE" e coloca no seu valor toda a informação de estado da página, com isso obtém-se algumas vantagens abaixo

10 **Tapestry** é um framework open-source para criar dinâmicamente aplicações altamente escalável para web em Java.
11 Maiores detalhes no decorrer do capítulo e no capítulo que trata do comparativo.

citadas:

- Não utiliza recursos do servidor, já que os valores ficam na página no cliente;
- É compatível com qualquer navegador, já que os campos escondidos são um recurso antigo do HTML;
- Os valores a serem armazenados no campo VIEWSTATE são codificados em uma string juntamente com um "checksum[12]" para detectar eventuais alterações no conteúdo.(SANT'ANNA, Mauro.2002);

Ainda segundo SANT'ANNA,Mauro.(2002), o uso de VIEWSTATE é comandado por propriedades EnableViewState, presente tanto no formulário como nos componentes individuais. Caso a propriedade EnableViewState do formulário seja desligada, nenhum componente manterá estado;

O Prado Framework ainda oferece o suporte a sessões, através da chamada do ID pelo usuário, a cada acesso é guardado um cookie[13] no lado do usuário contendo o seu ID. O que permite construir aplicações mais personalizadas e atrativas.(THE PHP GROUP.2006);

PRADO fornece uma técnica caching genérica usadas em diversas partes do núcleo da sua estrutura. Por exemplo, quando o caching é permitido, a classe TTemplateManager conservará os moldes analisados gramaticalmente.(Pradosoft Component Framework for PHP 5.2006);

Segundo FRAMEWORK, PRADO.(2005), O PRADO fornece os seguintes benefícios para os desenvolvedores:

12 **CHECKSUM:** um número de bits transmitido com os dados para que o dispositivo receptor possa verificar a precisão dos dados recebidos. Se o número de bits que chega é o mesmo enviado, a transmissão é considerada concluída.

13 **Cookie**s são um mecanismo para guardar dados no navegador remoto possibilitando o acompanhamento ou identificação de usuários que retornam.

- Reusabilidade - os códigos dos componentes de PRADO são altamente reusáveis. Tudo em PRADO é um componente reusável;
- Facilidade de utilização - criar e usar componentes são extremamente fáceis. Geralmente envolvem simplesmente configurar propriedades componentes;
- Robusto – PRADO reduz o esforço empreendido pelos colaboradores na criação de mais código, codificam nos termos dos objetos, métodos e propriedades, em vez de URLs e de parâmetros da pergunta;
- Fornece um mecanismo de relatório de erro mais preciso;
- Desempenho - PRADO usa uma técnica de caching para assegurar o desempenho das aplicações baseadas nele;
- Integração da equipe - PRADO permite a separação do índice e da apresentação. Os componentes, tipicamente páginas, têm seu índice (lógica) e apresentação armazenada em locais diferentes;

Ainda segundo FRAMEWORK, PRADO.(2005),50 a 75% do trabalho de uma aplicação web é realizado para gerar a interface e validar os dados fornecidos pelos usuários.

Entre seus principais recursos estão :

- Html separado do código php;
- Alto nível de reusabilidade por utilizar o conceito de componentes;
- Componentes para validação de formulários;
- Suporte a módulos;
- Arquivos em XML definem a configuração da aplicação dos módulos e dos

componentes;

- Suporte a internacionalização;
- Recursos de cache para aumentar a performance da aplicação;

O PRADO oferece suporte a várias bases de dados através dos drivers ADODB[14]. Ele utiliza uma classe chamada TAdodb, que é uma classe derivada da classe de ADODB.

4.2 SYMFONY PHP FRAMEWORK

"Symfony é uma estrutura PHP5 Orientado a objeto baseada no modelo MVC[15]. Symfony permite a separação de regras de negócio, a lógica do usuário e a visão da apresentação de uma aplicação para web. Também contém ferramentas numerosas wue visam encurtar a fase de desenvolvimento de uma aplicação complexa para web." (H3RALD.2006).

Fatores favoráveis:

- A estrutura inteiramente caracterizada, inclui tudo que o programador da web necessita;
- O suporte completo e nativo a internacionalização;

14 **ADODB** são classes de conexão com o banco pode ser visto em maiores detalhes em : http://www.xisc.com/wiki/index.php/TAdodb ou http://adodb.sourceforge.net/

15 **MVC** Esta arquitetura estabelece uma separação da estrutura em três partes distintas: Modelo, Vista e Controle. Em maiores detalhes no capítulo que trata do comparativo.

- Boa documentação, com: tutoriais, wiki´s[16], livros, os screencasts[17], APIs, e exemplos reais;
- Geradores de códigos;
- Vários módulos e bibliotecas "pré-construídas" para as tarefas mais comuns;
- Foi inspirado pelos melhores conceitos e práticas de vários outros frameworks;
- Bom suporte da comunidade;

Fatores desfavoráveis a sua utilização:

- Parece ser muito grande comparado aos outros, e com muitos recursos que não são úteis a todos os colaboradores.(H3RALD.2006);
- Contempla somente o PHP5 .(H3RALD.2006);
- Não é recomendado para projetos simples. (H3RALD.2006);

Symfony implementa a execução fácil de AJAX[18] e inclui o suite inteiro de scripts .”script aculo.us”[19] de efeitos do Javascript. Symfony tem também a habilidade de gerar “ CRUD e scaffolding”[20] da aplicação de uma base de dados já

16 **Wiki** é um sistema de site que permite a edição do seu conteúdo por qualquer um. Não só se adiciona conteúdo, como se complementa o conteúdo existente, editando-o, acrescentando ou corrigindo.

17 **Screencasts** : É um processo de criar demonstrações e simulações interativas de softwares através de uma série de telas de uma aplicação.

18 **AJAX** é um termo que descreve uma técnica de desenvolvimento web para criar softwares de interação. Em mais detalhes no decorrer do capítulo.

19 **Script.aculo.us** fornece consigo bibliotecas de fácil utilização, e compatíveis com o Javascript. São bibliotecas para desenvolver interfaces mais ricas.

20 Um **scaffold** é um meio de criar código para um determinado modelo (que indicamos) através de um determinado controlador (que indicamos também). É um meio de começarmos rapidamente a ver os resultados no nosso navegador web e um método muito rápido de implementar o CRUD (Create, Retrieve, Update, Delete) na sua aplicação. Lembrando que scaffold cria código que, fatalmente, vai ser alterado depois, a não ser que você deseje manter o nível muito básico de interface e controle padrão de campos que ele proporciona.(Rangel, Eustáquio).

construída em SQL. Isso significa que ele incorpora toda SQL.(PULIDO,Nick.2006).

Bases de dados suportadas: MySQL, PostgreSQL, SQLite, Oracle, MS SQL e qualquer outra suportada pela camada de abstração da base de dados Creole[21].(H3RALD.2006).

4.3 CAKE PHP FRAMEWORK

O Cake é um framework afiado no desenvolvimento rápido de aplicações. E na fácil execução de AJAX.(PULIDO,Nick.2006).

Cake é um Framework rápido de desenvolvimento para PHP que usa padrões como ActiveRecord[22].

Este modelo fornece uma estrutura que permite aos usuários de PHP em todos os níveis o poder de desenvolver rapidamente aplicações robustas para web, sem nenhuma perda à flexibilidade. (H3RALD.2006).

Fatores favoráveis:

- Contém somente o código essencial;
- Funciona em PHP4 e PHP5;
- É necessária somente a execução de uma configuração simples e curta da base de dados e em algumas constantes que podem ser modificadas. Você pode literalmente começar a programar em menos de cinco minutos.

21 **Creole** é uma camada de abstração de bases de dados para php5.

22 **Active Record** efetua automaticamente a correspondência entre tabelas e classes, linhas a objetos (exemplos das classes modelos), e colunas para atributos de objetos.

- Permite a criação de arquiteturas complexas da base de dados;
- Estrutura extremamente lógica e funcional de diretórios;
- Fácil uso de AJAX através dos ajudantes que auxiliam na criação de AJAX e de Javascript;
- Possui um script de linha de comando para gerar automaticamente partes do código, chamado BAKE;
- Comunidades muito ativas;
- Apropriado para qualquer tipo de website, da aplicação pessoal de pequeno porte à aplicação avançada de e-business;

Segundo H3RALD.(2006), podemos citar alguns fatores desfavoráveis em relação ao Cake:

- Nenhuma sustentação "oficial" de internacionalização para a versão atual, mas será incluída no passo seguinte;
- Não faz uso inteiramente das vantagens oferecidas pelo PHP5;
- A documentação oficial necessita ainda alguma melhoria, embora agora pareça consideravelmente completa e exaustiva;

Bases de dados suportadas: MySQL, PostgreSQL, SQlite, MS SQL + e outras que suportem a camada de abstração das bases ADOdb ou PEAR::DB[23].

23 **PEAR db** é uma biblioteca de PHP que possibilita o acesso a várias bases de dados.

4.4 REQUISITOS DE USO DOS FRAMEWORKS

Para fazer uso de um framework de PHP você precisa de algumas ferramentas tais como um servidor HTTP rodando e o PHP instalado. Com exceção do Cake Framework que roda também com o php4 as outros dois frameworks analisados necessitam que esteja rodando o PHP5. Tendo então estes aplicativos corretamente instalados e configurados, basta ir até o site do seu framework preferido e baixar o pacote do framework para sua maquina, e então descompactá-los no diretório root do seu servidor web, No caso do Ubuntu linux em: /var/www/ e fazer o seu acesso via browser no endereço http://localhost/Nome_do_framework/.

Em geral todos os frameworks oferecem suporte as mais variadas bases de dados existentes. Resta então ao usuário definir-se por alguma delas e proceder a instalação, sempre lembrando o mesmo de verificar a compatibilidade do framework com a base de dados escolhida, para evitar posteriores problemas. Nos tópicos no decorrer do trabalho será detalhada a instalação e configuração dos frameworks usados na comparação.

4.5 COMPARAÇÃO DAS FUNCIONALIDADES AGREGADAS

Descreve- se aqui uma série de sub-tópicos que podem auxiliar o desenvolvedor na escolha do framework mais adequado para o seu tipo de aplicação.

4.5.1 Tópicos

Neste item estão descritas algumas especificações das ferramentas de desenvolvimento de aplicações com frameworks PHP, e algumas de suas principais características bem como o que cada uma contempla.

Dentre todas as características dos frameworks foram escolhidas algumas para servir de base no auxilio da escolha de qual ferramenta de framework se enquadra melhor ao seu tipo de aplicação.

Os principais tópicos utilizados para comparação entre os frameworks estão descritos abaixo:

4.5.1.1 MVC

A arquitetura MVC foi desenvolvida para ser usada no projeto de interface visual em Smalltalk[24]. Esta arquitetura estabelece uma separação da estrutura da aplicação em três partes distintas: Modelo, Vista e Controle.

- Vista: gerencia a saída gráfica e textual da parte da aplicação visível ao usuário.
- Controle: interpreta as entradas de mouse e teclado do usuário, comandando a Vista e o Modelo para se alterarem de forma apropriada.
- Modelo: gerencia o comportamento e os dados do domínio da aplicação, responde as requisições sobre o seu estado (geralmente

24 **Smalltalk** é uma linguagem de programação Orientada a Objetos criada na década de 70 pelo Centro de Pesquisa de Palo Alto da Xerox.

vindas da Vista), e responde às instruções para mudança de estado (geralmente vindas do Controle). (BURBECH,2002).

Na arquitetura MVC o Controlador é responsável por receber e tratar os eventos gerados pelo usuário, No entanto o Controlador está diretamente ligado com a Visão. A interação entre os objetos torna-se extremamente necessária devido ao Controlador enviar notificações para o objeto Visão, o qual pode ser atualizado e modificado dependendo da ação tomada pelo controlador. (STRAPAZZON,Itanor José.2004).

Segue abaixo figura exemplificando o funcionamento do MVC:

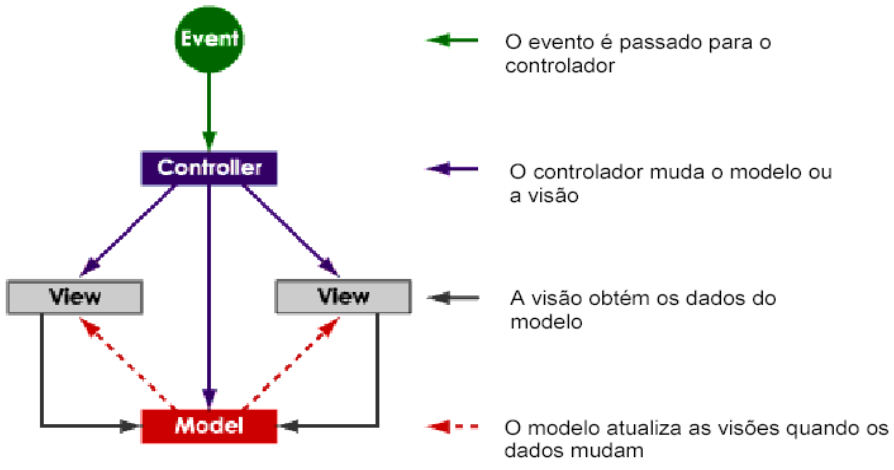

Ilustração 2: Exemplo da arquitetura MVC

4.5.1.2 MULTIPLES DB'S

Todos os frameworks pesquisados oferecem suporte a múltiplas DB´s como descrito no capítulo no início deste capítulo.

4.5.1.3 ORM

Object Relational Mapping ou "Mapeamento Objeto Relacional" é uma técnica de desenvolvimento utilizada para reduzir a impedância do desenvolvimento orientado ao objeto utilizando bancos de dados relacionais. As tabelas do banco de dados são representadas como objetos. Com esta técnica, é possível associar tabelas dos bancos de dados relacionais com classes de uma linguagem de programação orientada a objeto, utilizando tecnologia XML de maneira que o programador não precise se preocupar com os comandos em linguagem SQL, mas simplesmente utilize uma Interface de Programação simples (API) que faça todo o trabalho de persistência. (OBJECT-RELATIONAL MAPPÍNG.2006).

4.5.1.4 VALIDATION

Indica se o framework tem alguma estrutura de validação embutida em sua estrutura.

4.5.1.5 AJAX

AJAX(Asynchronous JavaScript and XML) é um termo que descreve uma

técnica de desenvolvimento web para criar softwares de interação. A tecnologia usa combinações do HTML, XML e CSS para apresentar as informações, trocando dados assincronamente com o servidor web.

Entre algumas vantagens de seu uso estão:

- Mais acessível.
- Maior controle sobre o consumo da memória da máquina do usuário.
- Não precisa de plugin.

A grande diferença entre o AJAX e os modelos de aplicações até então usadas é que no modelo convencional o usuário envia uma requisição ao servidor que por sua vez recebe e processa o código, conversa com outros sistemas paralelos e retorna ao servidor de internet que publica o código em forma de HTML. Com o AJAX isso acontece de forma assíncrona ou seja enquanto o usuário esta visualizando ou utilizando algo no site, são feitas inúmeras requisições ao servidor sem prévio aviso. Fazendo com que a exibição do site se torne uma visualização em tempo real. (VIEIRA,Victor.2006).

4.5.1.6 AUTH MODULE

Indica se a estrutura tem um módulo embutido para assegurar a autenticação do usuário.

4.5.1.7 MODULES

Indica se a estrutura do framework é modularizada por exemplo, possui um parser[25] da alimentação de RSS[26], o módulo de separação de pdf´s ou etc;

4.5.1.8 TABLELESS

> O nome Tableless é um nome mais "publicitário" para se referir a sites que seguem os Padrões. Os sites Tableless não são construídos usando as famigeradas tables. Elas usam XHTML para apresentar a informação e as Folhas de Estilo (CSS) para formatar essa informação. Pelo motivo de as tables não serem usadas para a estruturação, essa metodologia se chama Tableless.(TABLELESS.2006).

4.5.1.9 DB OBJECTS

Indica se a estrutura inclui outros objetos da base de dados, como classes de linguagem de programação orientada a objetos.

4.5.1.10 TEMPLATES

> Segundo MINETTO,Elton.(2006),Qualquer webmaster que enfrentou a missão de manter um site de médio a grande porte, sabe a dificuldade que isso significa, principalmente no que diz respeito a relação entre o design e a programação. Alterar dezenas de páginas cada vez que um detalhe de design, como uma cor de fundo ou uma

25 **parsing** é o processo de analisar uma sequência de entrada (lida de um arquivo ou do teclado, por exemplo) para determinar sua estrutura gramatical segundo uma determinada gramática formal. Este processo é formalmente chamado de análise sintática. Um **parser** é um programa de computador que executa essa tarefa.

26 **RSS** é um sub-conjunto de "dialetos" XML que servem para agregar conteúdo . É usado principalmente em sites de notícias e blogs.

> imagem, é alterado é uma tarefa desgastante. Mas, para alívio de todos, existe uma solução para o problema e esta solução está nos templates.
> Os templates, ou modelos, têm por finalidade separar o design, a parte HTML, da programação, no nosso caso, o PHP.

4.5.1.11 CACHING

Caching é o melhor recurso para a redução de processamento do servidor envolvendo a renderização e lógica da camada de acesso a dados.

Podemos dividir o caching em três tipos básicos:

- Cache de Respostas (OutputCache)

O cache de respostas é a forma mais simples de se tirar proveito do sistema de cache, sem necessidade de redesenho ou alterações de código - o conteúdo a ser enviado ao cliente (resposta) é armazenado em memória e disponibilizado para as próximas requisições. De fato, todo o conteúdo dinâmico pode ser armazenado em mecanismos que suportem o HTTP 1.1 (servidor web, navegadores e proxies) de forma que requisições subseqüentes sejam servidas direto do cache sem execução de código.(FERREIRA, Miguel.2006).

- Cache de Fragmento

> Existem situações onde o “cacheamento” de toda uma página de resposta não é adequado. Um caso óbvio, seria àquele no qual o desenho da aplicação requerer que algumas partes/seções de uma página específica tenham versões específicas para um determinado grupo de usuários (e.g. menu baseado no perfil do usuário) e que outras partes sejam comuns à todos (e.g. conteúdo genérico c/ notícias gerais). Este é o cenário ideal para o uso do fragment caching.(FERREIRA, Miguel.2006).

- Cache de Objetos

O objeto de Cache é recriado a cada reinício da aplicação, o que nos lembra uma certa similaridade com o objeto Application. A principal diferença entre os dois reside no fato de que o objeto de Cache proporciona funcionalidade específicas, tais como dependências e políticas de expiração.(FERREIRA, Miguel.2006).

4.5.1.12 INTERNACIONALIZAÇÃO

Suporte a internacionalização, ou seja, criar sites em vários idiomas apenas trocando os arquivos de propriedades, sem ter que mexer no fonte.

4.5.1.13 INTEGRAÇÃO PEAR

Pear é um sistema de repositório e de distribuição para componentes reusáveis de php.

Pear é composto por:

- Repositório organizado e padronizado de código aberto para programadores de PHP Sistema para a distribuição e manutenção de pacotes.
- PFC (PHP Foundation Classes).
- Estilo padronizado de código.
- Sites e listas de discussão para a comunidade de colaboradores.(PEAR THE PHP GROUP.2005).

O código PEAR é dividido em “pacotes”. Cada pacote é um projeto separado

com sua própria equipe de desenvolvimento, número de versão, ciclo da liberação, documentação e uma relação definida a outros pacotes(dependências including).(PEAR THE PHP GROUP.2005).

4.5.1.14 DOCUMENTAÇÃO DETALHADA

Uma boa documentação não é somente uma descrição do que é a ferramenta e para que ela serve. Mas sim toda descrição histórica, todo um apoio para os desenvolvedores como exemplos práticos de sua aplicação, detalhes da sua criação e implementação, descrição do comportamento e funcionamento da mesma, fóruns de discussão, e demais conceitos abordados acima.

4.5.1.15 TABELA COMPARATIVA

Segue abaixo uma tabela comparativa ilustrando cada um dos frameworks a serem analisados e suas principais características.

Frameworks	PHP4	PHP5	MVC	MULTIPLES DB's	ORM	
CAKE PHP	Sim	Sim	Sim	Sim	Sim	
SYMFONY	Não	Sim	Sim	Sim	Sim	
PRADO	Não	Sim	Não	Sim	Não	
Frameworks	Validation	Ajax	Auth Modules	Modules	Tableless	
CAKE PHP	Sim	Sim	Sim	Sim	Sim	
SYMFONY	Sim	Sim	Sim	Sim	Sim	
PRADO	Sim	Sim	Sim	Sim	Sim	
Frameworks	DB Objects	Templates	Caching	Internacionalização	Integração Pear	Documentação detalhada
CAKE PHP	Sim	Não	Sim	Não	Não	Parcialmente
SYMFONY	Sim	Sim	Sim	Sim	Sim	Sim
PRADO	Sim	Sim	Sim	Sim	Não	Parcialmente

Tabela 8: Tabela comparativa entre os frameworks (COMUNIDADE CYANEUS.2006).(THE PALLETT GROUP.2005).

5. DESENVOLVIMENTO DO PROTÓTIPO

Para o desenvolvimento do protótipo foram utilizadas somente ferramentas livres. Segue abaixo uma lista.

- Sistema operacional Ubuntu 6.06 LTS;
- Para modelagem do banco foi utilizado o DBDesigner4.0.5.4;
- PHP 5;
- Apache 2;
- Mysql;
- Open office 2.0;
- Prado Framework;
- Symfony Framework;
- Cake Framework;

5.1 APLICAÇÃO

Para o desenvolvimento da aplicação proposta foi utilizado com parâmetro um caso de uso que já foi aplicado sobre alguns frameworks de PYTHON[27],O

27 **Python:** é uma línguagem de programação orientada a objetos, dinâmica que pode ser usada para muitos tipos do desenvolvimento do software. Pode ser visto em maiores detalhes em :http://www.python.org/

PyWebOff[28].

Escolhemos este caso pelo sucesso no comparativo e também para facilitar quem sabe uma futura comparação entre frameworks de Python e PHP.

O PyWebOff é uma adaptação de um exercício do WARMUP[29], que foi escrito originalmente com o objetivo de familiarizar os estudantes com a escrita de aplicações WEB e as camadas persistentes em java. (PyWebOff.SD).

Foi aplicado a alguns frameworks de PYTHON para auxiliar os usuários a decidirem qual é a estrutura de frameworks que melhor se encaixa em suas aplicações. Consiste na descrição de uma aplicação com alguns requisitos que é aplicado as estruturas a serem analisadas para verificar qual é a melhor para cada tipo de caso.(PyWebOff.SD).

A aplicação é simples o bastante para ser implementada e complexa o suficiente para englobar alguns dos principais requisitos de uma aplicação web. Trata-se de um Professor que gostaria de uma aplicação WEB simples para manter-se informado sobre os livros que ele empresta.

Quando Bhargan[30] entra no sistema, deve poder visualizar e também editar quem tem registro no sistema. Quando os estudantes fazem logon ao sistema, devem ver que livros estão no seu empréstimo; mas não devem editar nenhuma informação.

O sistema de Bhargan contém três funcionalidades:

1. Um livro tem um título, um ano da publicação, e um ou mais autores/editores.

28 **PyWebOff:**: Foi desenvolvido por Michelle Levesque com propósitos estudantis.

29 **Warmup:** é um exercício pode ser visto em mais detalhes em https://wwwcgi.cdf.utoronto.ca/~cs494hf/cgi-bin/argon.cgi/uilib/wiki/WarmupExerciseWinter2006

30 **Bhargan:** nome do professor que deseja o sistema, segundo (PyWebOff).

2. Um empréstimo grava que usuário pediu um livro particular em um dia particular.
3. Um usuário contém um nome, endereço de email , senha, e a que classe eles pertencem e se sua classificação no sistema, exemplo : se eles são usuários ou são administradores.

Bhargan gostaria do seu sistema de registros com uma tela similar a da figura abaixo quando ele fizer logon ao sistema:

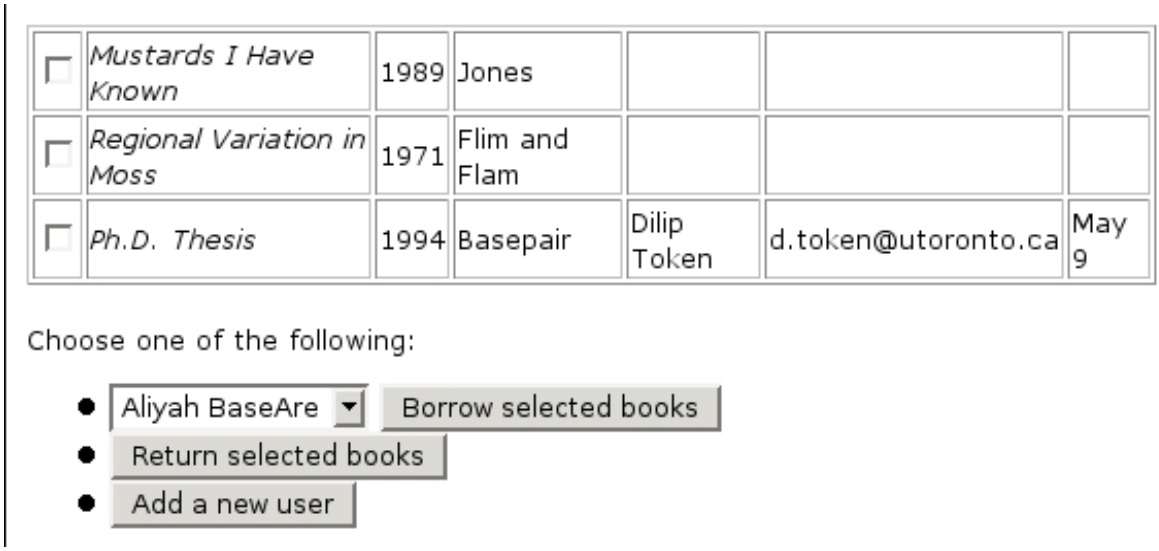

Ilustração 3: Tela Inicial do sistema de Bhargan (PYWEBOFF.SD).

Bhargan gostaria também de uma tela de início de sessão, em que os usuários que não são Admin possam visualizar uma tabela. (sem poder editar qualquer coisa), mas que esta tabela contenha somente os endereços de E-mail dos usuários que pertencem à mesma classe que eles, se não deve apenas dizer o Email do usuário em questão.

Se Bhargan verificar que um ou mais livros não estão mais emprestados,

clicando em “Retornar livros selecionados”, os empréstimos associados a aqueles livros estão apagados da base de dados. Se Bhargan verificar que um ou mais livro que não estão emprestados e não pertencem a um usuário clicando em “Emprestar livros selecionados”, o sistema adiciona um ou mais empréstimo grava à base de dados. Se Bhargan tentar retornar os livros que não estão no empréstimo, ou tentar emprestar os livros que já estão emprestados, o sistema mostra novamente a página com uma mensagem de erro, mas não muda a base de dados. Se Bhargan clicar sobre “adicionar novo usuário”, deve-se então ter acesso a um formulário que o deixe adicionar usuários novos à base de dados.

5.1.1 MODELAGEM

Esta figura representa a modelagem a ser utilizada nos testes com as três aplicações de frameworks diferentes. Esta modelagem não consta no PyWebPOff, e foi desenvolvida neste projeto para poder refletir a aplicação escrita.

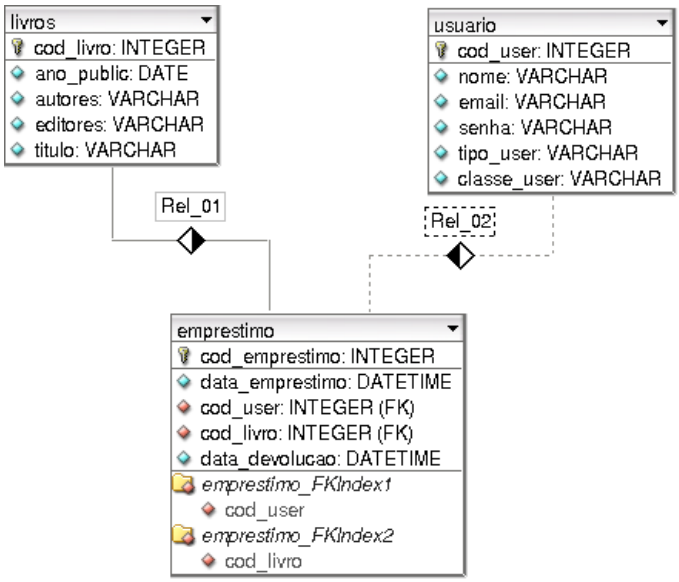

Ilustração 4: Modelagem Padrão do sistema para os testes

5.1.2 REQUISITOS DA APLICAÇÃO

Para elaboração do protótipo foram levados em consideração alguns requisitos funcionais que abordam aspectos de relativa importância na aplicação. Esses requisitos também não são encontrados no modelo PyWebOff e foram escritos neste projeto com o intuito de enriquece-lo ainda mais. Segue abaixo a descrição dos requisitos.

5.1.2.1 Requisitos Funcionais

- A aplicação deve ser capaz de validar o usuário e senha e distinguir se eles são simples usuários ou pertencem ao grupo de administrador do sistema;
- Deve oferecer uma interface de administração da aplicação e outra para acesso dos clientes (usuários);
- Deve direcionar para a interface correta de acordo com o usuário que efetuar login;
- A interface de administração da aplicação deve oferecer:
    - Opção de menu para inserção de novos usuários;
    - Opção de menu para alteração de usuários;
    - Opção de menu para exclusão de usuários;
    - Opção de menu para exibir todos os usuários cadastrados;
    - Opção de menu para efetuar logout da aplicação;

- A interface de usuário deverá conter:

    - Opção de menu para retornar ao momento inicial da aplicação;

- Opção de menu para efetuar logout da aplicação;
- Opção para ver os livros emprestados pelo usuário;

5.1.2.2 Requisitos Não-Funcionais

- A interface deve ser simples e objetiva;
- A aplicação não deve conter cores carregadas que prejudiquem a visualização dos dados, usar preferencialmente cores leves e de tom claro;
- Ser desenvolvido usando uma linguagem que evite o uso de programas auxiliares, como máquinas virtuais por exemplo, facilitando o acesso a qualquer tipo de usuário;
- Utilizar somente ferramentas fornecidas como código aberto;
- Ser leve e de fácil adaptação para que possa ser alocada em qualquer máquina ou servidor;

5.2 INSTALAÇÃO DOS APLICATIVOS BÁSICOS

Para que o desenvolvimento com os Frameworks fosse possível no Ubuntu foi necessário fazer algumas configurações e instalar alguns pacotes, pois o sistema operacional não traz isso nativo com a sua instalação.

Começamos com a instalação dos serviços essenciais como o servidor web, o banco de dados e o PHP . No Ubuntu bastou apenas um comando para que isso

fosse feito:

"*sudo apt-get install apache2 mysql-server php5 php5-mysql*"

Com este comando acima citado ele vai buscar em seus repositórios os programas solicitados e os instala na máquina.

Depois de instalado foi preciso configurar os serviços para que atuem conforme a nossa necessidade.

Dentro do diretório */etc/apache2/mods-enabled/* foram criados os seguintes links para os módulos a serem configurados com os comandos abaixo listados:

sudo ln -ns ../mods-available/php5.conf php5.conf

sudo ln -ns ../mods-available/php5.load php5.load

sudo ln -s /etc/apache2/mods-available/rewrite.load

Depois de criados os links no diretório */etc/apache2/sites-available* foi editado o arquivo chamado default, e dentro dele foi localizado a linha:

<Directory /var/www/>

AllowOverride none

Foi alterado o estado de *AllowOverride*[31] *none* para *AllowOverride all*. Feito isto foi só reiniciar o serviço que já esta funcionando, pode ser utilizado o comando:

sudo /etc/init.d/apache2 restart

31 **AllowOverride:** é uma diretiva do Apache que determina se os arquivos .htaccess podem alterar as configurações do servidor em um determinado diretório.

5.3 DESENVOLVIMENTO NO PRADO

Para desenvolver com o framework Prado foi necessário acessar ao site do framework e baixar o pacote da última versão estável. Após descompactá-la foi só copiar o seu conteúdo para o diretório root do servidor, no caso do Ubuntu */var/www*.

Feito estes passos, adicionei a pasta Prado as permissões de execução e acessei pelo navegador : *http://localhost/prado*

Para me certificar que estava tudo certo, acessei o link requirement checker e pude visualizar os componentes que podiam ser utilizados todos em verde, isso significa que já podia começar a trabalhar com o Prado.

Para criação da estrutura do projeto utilizei-me do script Prado-cli, que esta no diretório */var/www/prado/framework/*

Estando no diretório raiz do framework executei:

- *php framework/prado-cli.php -C nome_do_projeto*

Com isso obtivemos a estrutura de diretórios para começar a aplicação:

Segue abaixo tabela que representa a estrutura do framework:

prado	Raiz do framework
assets	diretório que armazena arquivos privados publicados
protected	Caminho base que armazena dados da aplicação e arquivos de scripts privados.
pages	Caminho base do armazenamento de todas as páginas PRADO.
Home.page	Página padrão retornada quando o usuário não especifica a página requisitada
runtime	caminho base que armazena arquivos em tempo de execução
index.php	Script de entrada do aplicativo.

Tabela 9: Estrutura de diretórios da aplicação com Prado

Na construção de uma aplicação com o Prado recomenda-se utilizar uma padronização, para o bom funcionamento e utilização dos recursos do framework.

Esta descrição não foi encontrada formalmente escrita no manual do framework, mas todos os exemplos de código do próprio framework utilizam este formato. Segue a descrição.

- Todas as tabelas do banco de dados devem estar no plural.
- Utilizar como chave primária o campo “id”.
- Para fazer uso de chave estrangeira deve-se utilizar o nome da tabela no singular acrescido do campo id.

Segue abaixo a modelagem que foi desenvolvida segundo as indicações para gerar aplicação com o Prado Framework.

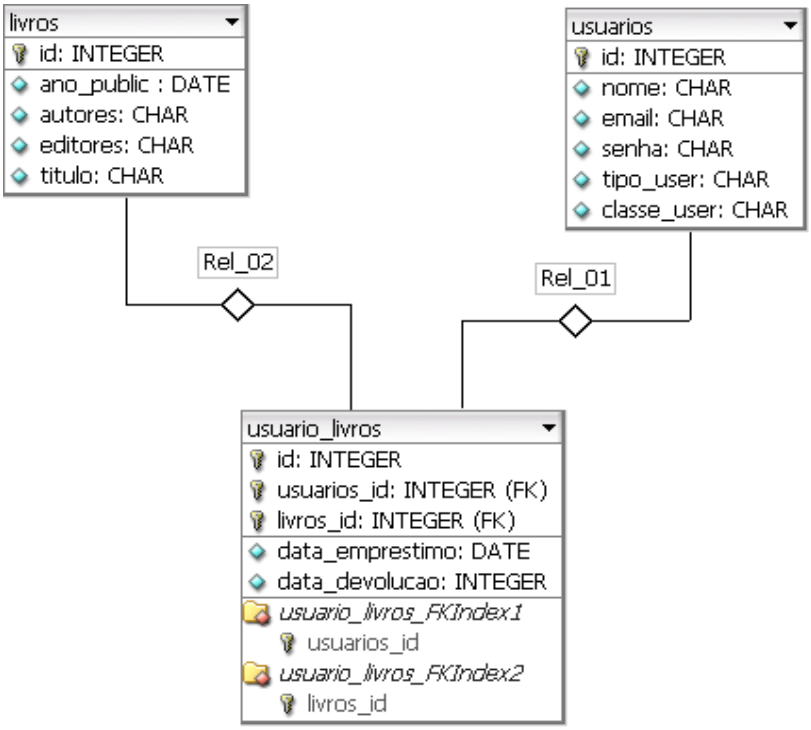

Ilustração 5: Base de dados utilizada no Prado

Depois de pronta a base de dados precisamos seguir alguns passos para poder conectar com a base criada:

- Baixar o pacote adodb.
- Criar uma pasta na raiz do framework chamada adodb e colar o conteudo dentro.
- Dentro da pasta *protected/modules* temos que colar o arquivo *adodb.php* para que o Prado entenda a chamada do adodb.
- Acessar a pasta da aplicação que você criou, e dentro dela a pasta protected.
- Criar o arquivo *application.xml.* O mesmo deverá conter as seguintes linhas:

```
<?xml version="1.0" encoding="utf-8"?>
<application id="myAppId" Mode="Debug">
	<paths>
		<using namespace="Application.modules.*" />
	</paths>

	<modules>
		<module id="adodb" class="adodb"
			Driver="mysql" Host="localhost" Username="root"
Password="******" Database="Nome_da_base_criada" />
	</modules>


</application>
```

Tabela 10: Exemplo do arquivo Application.xml

E a base de dados já pode ser acessada..

Para criar a lista de empréstimos foi necessário utilizar um componente chamado datagrid. Segue abaixo o exemplo do código dos arquivos emprestimos.page e emprestimos.php que geram a lista de empréstimos:

```
<?php

class Emprestimos extends TPage{

        protected function getData() //função que busca os dados do banco
        {                                // e joga no data grid
                $db = $this->Application->getModule('adodb');
                $sql = '
       SELECT usuario_livros.*, usuarios.nome, livros.titulo
       from usuario_livros, usuarios, livros
       where usuario_livros.usuario_id = usuarios.id and
            usuario_livros.livro_id = livros.id';
                $result = $db->Execute($sql);
                $ret = array();
                $i=0;
                while(!$result->EOF) { //enquanto o resultado for diferente de fim de
arquivo
                        $ret[$i] = array('id'=>$result-
>fields["id"],'data_emprestimo'=>$result-
>fields["data_emprestimo"],'usuario_id'=>$result-
>fields["usuario_id"],'usuario_nome'=>$result->fields["nome"],'livro_id'=>$result-
>fields["livro_id"],'livro_titulo'=>$result->fields["titulo"],'data_devolucao'=>$result-
>fields["data_devolucao"]);
                        $i++;
                        $result->MoveNext(); //move para o próximo registro retornado
pela consulta sql
                }
                return $ret;

        }

        public function onLoad($param)
        {
                parent::onLoad($param);
                if(!$this->IsPostBack)
                {
                        $this->DataGrid->DataSource=$this->Data;
                        $this->DataGrid->dataBind();
                }
        }
        public function toggleColumnVisibility($sender,$param) //torna o data grid visivel
        {
                foreach($this->DataGrid->Columns as $index=>$column)
                        $column->Visible=$sender->Items[$index]->Selected;
                $this->DataGrid->DataSource=$this->Data;
                $this->DataGrid->dataBind();
        }

        public function deleteitem($sender,$param) { //deleta dados do data grid
                $id = $this->DataGrid->DataKeys[$param->Item->ItemIndex];
                $db = $this->Application->getModule('adodb');
                $sql = "delete from usuario_livros where id=$id";
                $result = $db->Execute($sql);
                $this->DataGrid->DataSource=$this->Data;
```

```
                $this->DataGrid->dataBind();
        }

        public function editItem($sender,$param)
        {
                $this->DataGrid->EditItemIndex=$param->Item->ItemIndex;
                $this->DataGrid->DataSource=$this->Data;
                $this->DataGrid->dataBind();
        }

        public function saveItem($sender,$param) // salva os dados editados
        {
                $item = $param->Item;

       $id = $this->DataGrid->DataKeys[$item->ItemIndex];

        $data_emprestimo = $item->colunadataemprestimo->TextBox->Text;
                $usuario_id = $item->colunausuarioid->TextBox->Text;
                $livro_id = $item->colunalivroid->TextBox->Text;
                $data_devolucao = $item->colunadatadevolucao->TextBox->Text;
                 //echo "$data_devolucao";
                $db = $this->Application->getModule('adodb');
           $sql = "update usuario_livros set data_emprestimo='$data_emprestimo',
usuario_id='$usuario_id', livro_id='$livro_id', data_devolucao='$data_devolucao' where
id=2";
                $result = $db->Execute($sql);
       $this->DataGrid->EditItemIndex=-1;
                $this->DataGrid->DataSource=$this->Data;
                $this->DataGrid->dataBind();
        }

}
?>
```

Tabela 11: Exemplo do arquivo emprestimos.php

```
<html>
<head>
<title> Prado Framework</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<h2>Lista de Emprestimos</h2>
<body>

<com:TForm>
<com:TDataGrid  //cria o data grid
        Width="700px"
        CellPadding="2"
        ID="DataGrid"
        DataKeyField="id"
        AutoGenerateColumns="false"
        HeaderStyle.BackColor="red"
        HeaderStyle.ForeColor="white"
        ItemStyle.BackColor="BFCFFF"
        ItemStyle.Font.Italic="false"
        AlternatingItemStyle.BackColor="#E6ECFF"
   OnDeleteCommand="deleteItem"
        OnEditCommand="edititem"
        OnUpdateCommand="saveitem"
        OnCancelCommand="cancelitem"
   >
        <com:TBoundColumn  //cria a coluna ID
      ItemStyle.HorizontalAlign="Center"
                ItemStyle.Wrap="false"
                ItemStyle.Font.Italic="false"
                ItemStyle.ForeColor="black"
                HeaderText="Codigo"
                DataField="id"
                ID="colunaid"
        />
        <com:TBoundColumn //cria a coluna data de emprestimo
                ItemStyle.HorizontalAlign="Center"
                ItemStyle.Wrap="false"
                ItemStyle.Font.Italic="false"
                ItemStyle.ForeColor="black"
                HeaderText="data do emprestimo"
                DataField="data_emprestimo"
                ID="colunadataemprestimo"
        />
        <com:TBoundColumn
      ItemStyle.HorizontalAlign="Center"
                ItemStyle.Wrap="false"
                ItemStyle.Font.Italic="false"
                ItemStyle.ForeColor="black"
                HeaderText="Codigo do usuario"
                DataField="usuario_id"
                ID="colunausuarioid"
        />
   <com:TBoundColumn
      ItemStyle.HorizontalAlign="Center"
```

```
                ItemStyle.Wrap="false"
                ItemStyle.Font.Italic="false"
                ItemStyle.ForeColor="black"
                HeaderText="Nome do usuario"
                DataField="usuario_nome"
                ID="colunausuarionome"
        />
        <com:TBoundColumn
      ItemStyle.HorizontalAlign="Center"
                ItemStyle.Wrap="false"
                ItemStyle.Font.Italic="false"
                ItemStyle.ForeColor="black"
                HeaderText="Codigo do livro"
                DataField="livro_id"
                ID="colunalivroid"
        />
        <com:TBoundColumn
      ItemStyle.HorizontalAlign="Center"
                ItemStyle.Wrap="false"
                ItemStyle.Font.Italic="false"
                ItemStyle.ForeColor="black"
                HeaderText="Titulo do livro"
                DataField="livro_titulo"
                ID="colunalivrotitulo"
        />
        <com:TBoundColumn
      ItemStyle.HorizontalAlign="Center"
                ItemStyle.Wrap="false"
                ItemStyle.Font.Italic="false"
                ItemStyle.ForeColor="black"
                HeaderText="data da devolucao"
                DataField="data_devolucao"
                ID="colunadatadevolucao"
        />
 <com:TEditCommandColumn  //cria o campo para edição
                HeaderText="Edit"
                HeaderStyle.Width="100px"
                UpdateText="Save"
                ItemStyle.HorizontalAlign="Center"
                ItemStyle.Font.Italic="false"
        />
        <com:TButtonColumn  //cria o campo para deleção
                ID="DeleteColumn"
                HeaderText="Delete"
                HeaderStyle.Width="50px"
                ItemStyle.HorizontalAlign="Center"
                ItemStyle.Font.Italic="false"
                Text="Delete"
                CommandName="delete"
        />
</com:TDataGrid>
</com:TForm>
<h3>
<ul>
```

```
   <li><a href="?page=Admin.Novoemp">Novo Emprestimo</a></li>
   <li><a href="?page=Admin.Home">Lista dos Usuarios</a></li>
   <li><a href="?page=Admin.Livros">Lista dos Livros</a></li>
   <li><a href="?page=Home">Sair</a></li>
</ul>
</h3>
</body>
</html>
```

Tabela 12: Exemplo do arquivo emprestimos.page

Segue abaixo também a tela da aplicação gerada com o código acima exibido.

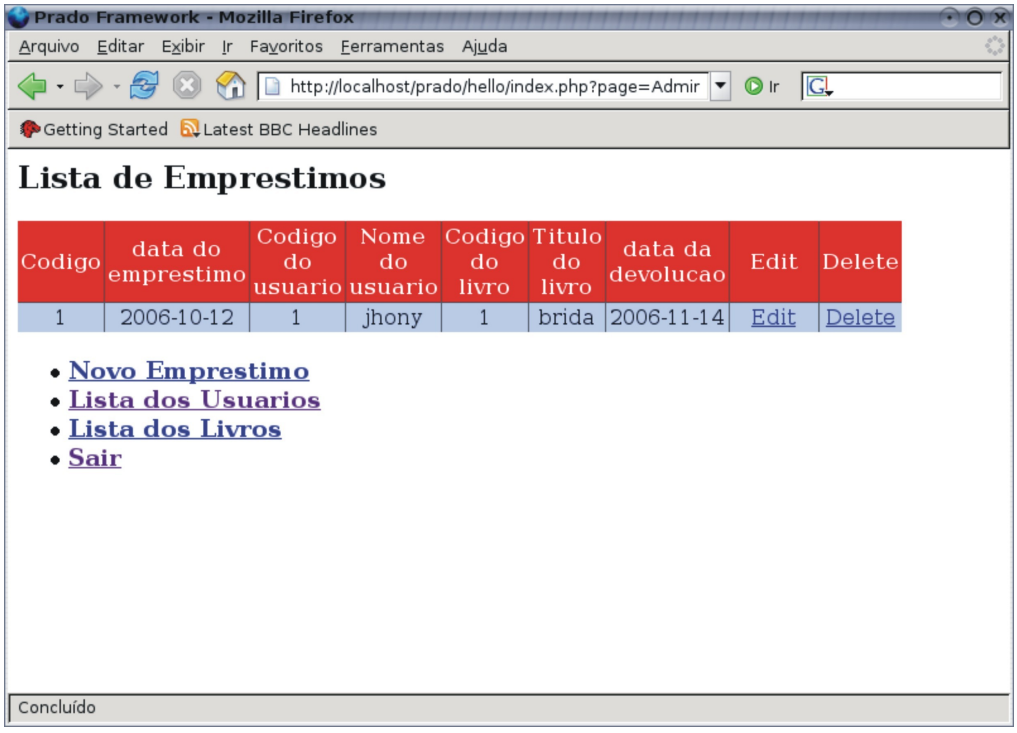

Ilustração 6: Tela de empréstimos exemplo da iteração com os data grids

Uma das coisas mais interessantes do Prado é esta iteração com os data grids, Pois o campos de ação dos objetos, interagem na própria linha do objeto sem alterar as demais e assim provendo bom uso da estrutura. Segue abaixo uma tela exemplificando.

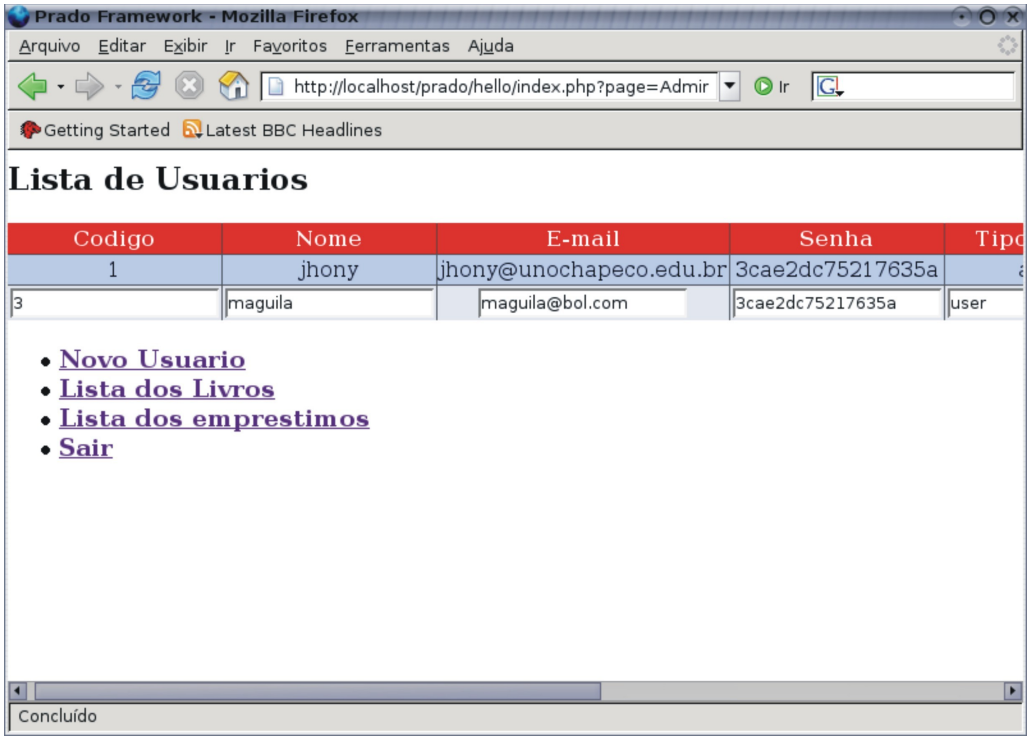

Ilustração 7: Exemplo da edição no data grid

Segue abaixo mais algumas telas da aplicação gerada.

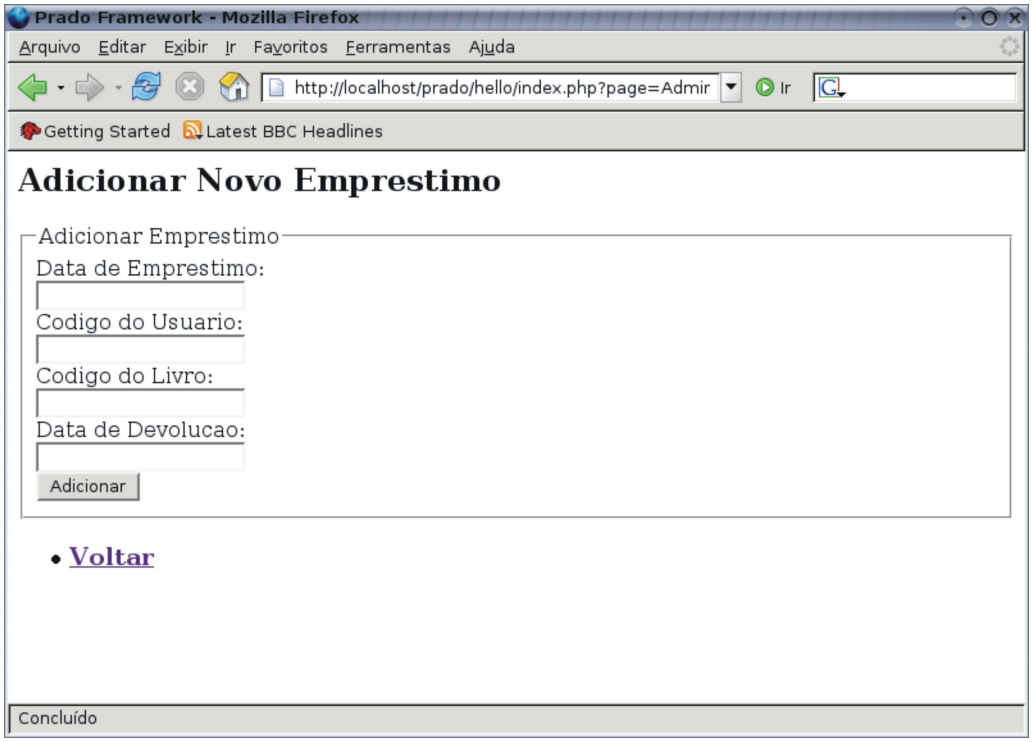

Ilustração 8: Tabela de inserção de empréstimos

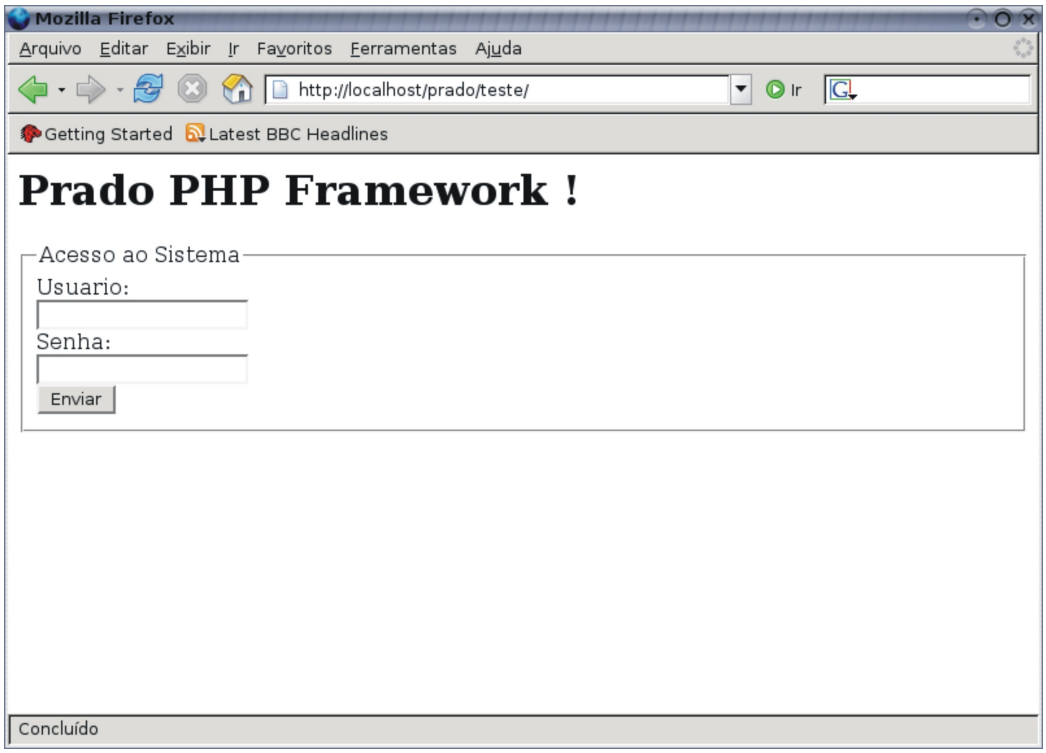

Ilustração 9: Tela de login da aplicação prado

5.4 DESENVOLVIMENTO NO SYMFONY

Para se desenvolver com o symfony framework faz-se necessário escolher o tipo de instalação que seja mais adequada para sua utilização.

A instalação do symfony pode ser feita de três formas distintas:

- Através do pacote SANDBOX. É um projeto symfony vazio onde todas as bibliotecas requeridas já estão incluídas, e onde a configuração básica já está feita.
- Através do uso da integração PEAR. A instalação PEAR é recomendada a àqueles que querem trabalhar com diversos projetos symfony , com uma maneira fácil de se integrar. Requer a versão 1.4.0 do PEAR ou mais atualizada, que é embutida na maioria das distribuições de PHP.
- Ou ainda instalado manualmente. A instalação manual é indicada a ser usada somente pelos colaboradores avançados de PHP, que querem fazer uso das vantagens ou adicionar características do seus próprios projetos, ou ainda corrigir defeitos da distribuição.

Neste projeto optamos por utilizar a instalação do pacote SANDBOX por ser uma aplicação de porte simples, e não precisarmos atualizações constantes. Seguem abaixo os detalhes de instalação do pacote SANDBOX , e também para não criarmos tendências quanto ao modo de instalação, segue abaixo também a

descrição da instalação via pacotes PEAR.

Como nota dos colaboradores do projeto, quando for necessário desenvolver algo de porte maior que um simples site, a versão sandbox não é a mais indicada devido ao grande número de atualizações feitas no projeto symfony e levando em conta a dificuldade de atualização usando o sandbox , Em resumo a instalação sandbox é indicada para quem ainda esta aprendendo sobre os frameworks e não quer se incomodar com detalhes na instalação e aprimoração do projeto.

O sandbox é um simples pacote de arquivos. Que contém um projeto symfony vazio onde todas as bibliotecas requeridas para seu uso (symfony, o pake, creole, propele e phing) já estão incluídas. Para instalá-lo, basta descompactar apenas o arquivo sob o diretório root do seu servidor web e configura-lo para seu usuário.

O sandbox foi desenvolvido para que você pratique com symfony em um computador local, para desenvolver aplicações mais simples, que podem terminar na web. Entretanto, a versão de symfony enviado com o sandbox é inteiramente funcional e equivalente ao pacote que você pode instalar através da integração PEAR. E por este motivo foi escolhido neste projeto. A única desvantagem é que o SANDBOX não é facilmente atualizado, mas isto não é problema neste caso.

O pacote sandbox pode ser obtido neste endereço: *http://www.symfony-project.com/get/sf_sandbox.tgz.*

Após baixá-lo, foi descompactado em */var/www*

É necessário dar permissões de execução à pasta do SANDBOX a fim de que o apache consiga fazer as execuções. Para visualizar a instalação: *http://localhost/sf_sandbox/web/* . Pode-se visualizar uma página de felicitações.

Instalação do Symfony utilizando os pacotes PEAR

Para utilizar a instalação PEAR no Ubuntu faz-se necessário instalar primeiramente o Pacote PEAR que pode ser feito através da seguinte linha de comando:

- *sudo apt-get install php-pear.*

Após a instalação do pear, o arquivo *php.ini* que está dentro do diretório */etc/php5/cli/* deve ser editado e realizada a alteração do conteúdo relacionado a linha *memory_limit*[32] *= 8m*. Alterar para *memory_limit = 16m*.

Adicionar então o Pear ao projeto Symfony para que torne-se possível baixar a sua última versão estável. Pode ser feito com o seguinte comando:

- *sudo pear upgrade PEAR && sudo pear channel-discover pear.symfony-project.com*

Podemos então baixar a última versão estável dos repositórios PEAR com o seguinte comando:

- *sudo pear install symfony/symfony*

O manual de instalação pede que seja instalado junto também o pacote phing que inclui alguns pacotes de controle de versão, pode ser feito com o comando:

- *sudo pear install --alldeps http://phing.info/pear/phing-current.tgz.*

Feita a instalação alguns passos se fazem necessários a fim de inicializar o

32 **memory_limit :** Essa diretiva configura a quantidade máxima de memória (bytes) que o script pode alocar. Isso ajuda a prevenir que scripts mal feitos consumam toda a memória disponível em um servidor.

projeto com o framework symfony. Os seguintes passos foram executados.

- Criar dentro de /var/www/ a pasta do projeto.
- Acessando a pasta e executar a criação do projeto: *symfony init-project tcc.*
- Para inicializar a aplicação: *symfony init-app hora.*
- Para habilitar o modulo re-write executar via linha de comando *a2enmod rewrite.*
- Dentro de /etc/apache2/sites available editar o arquivo default e adicionar as seguintes linhas:

```
<Directory "/usr/share/php/data/symfony/web/sf">
    AllowOverride All
    Allow from All
</Directory>

<Directory "/var/www/tcc/web">
      AllowOverride All
      Allow from All
</Directory>
```

Tabela 13: Configuração do apache

- Depois de salvar o arquivo é só executar *sudo a2ensite default* para o apache2 reler a configuração.
- E por fim executar *apache2ctl restart.*
- Editar também o arquivo php.ini dentro do diretório */etc/php5/apache2/php.ini*

e substituir o valor do campo *magic_quotes_gpc*[33] *=on por magic_quotes_gpc =off*

Após a instalação foi possível visualizar em *http://localhost/tcc/web/* uma mensagem de congratulações.

O symfony assim como outros frameworks trabalha com uma estrutura de diretórios que serve para melhor organização do projeto e torna possível o modelo MVC , segue abaixo uma tabela com a descrição desta estrutura:

apps/	Onde se encontra os diretórios de cada aplicação.
tcc/	Subdiretório onde fica a aplicação.
batch/	Arquivos php chamados pela linha de comando ou agendados para rodar em processos de carga.
cache/	È o cache do projeto, cada aplicação terá seu subdiretório contendo HTML pré-processados e arquivos de configuração.
config/	Local onde ficam armazenados as configurações gerais do projeto.
data/	Local onde ficam armazenados os arquivos de dados do projeto, como o esquema do banco de dados.
sql/	Onde ficam os arquivos com instruções SQL.
doc/	Onde ficam armazenada a documentação do projeto.
api/	Onde ficam os documentos gerados pelo phpdoc
lib/	Onde ficam armazenadas as classes e bibliotecas de terceiros
model/	Armazena o modelo de objetos do projeto
log/	Armazena os arquivos de log gerados pelo symfony
test/	Contém unidades de testes utilizadas pelo framework
web/	Torna-se o diretório raiz do servidor web, onde se encontram disponíveis os arquivos.
css/	Onde se armazena arquivos de css
images/	Onde se armazena as imagens da aplicação
js/	Onde se armazena os javascripts da aplicação
uploads/	Diretório de upload para a aplicação

Tabela 14: Estrutura raiz da apicação Symfony (symfony-projetc.2006).

33 **magic_quotes_gpc:** Define o estado para as aspas mágicas para operações GPC (Get/Post/Cookie). Quando as aspas mágicas estiverem em on, todas ' (aspas simples), " (aspas duplas), \ (barras invertidas) e NULL's são escapados com uma barra invertida automaticamente. Maiores detalhes em: http://www.php.net/manual/pt_BR/ref.info.php#ini.magic-quotes-gpc

Ainda segundo SYMFONY-PROJECT(2006). A estrutura em árvore para todos os diretórios de aplicação é a mesma:

tcc/	Diretório da aplicação
Config/	Arquivos de configuração específicos da aplicação
i18n/	Diretório utilizado para os arquivos de internacionalização da aplicação, este pode ser desprezado no caso da utilização de banco de dados
lib/	Contém as bibliotecas especificas da aplicação
modules/	Armazena os módulos que contém as funcionalidades da aplicação
templates/	Armazena os templates globais da aplicação, que vão ser utilizados por todos os módulos
layout.php	Este arquivo é gerado por padrão a cada nova aplicação, contém somente um layout básico, que pode ser editado.
error.php	Responsável pela saída de erros gerados pela aplicação
error.txt	Saída de erro quando a aplicação é chamada sem um navegador web

Tabela 15: Estrutura em árvore para cada aplicação (symfony-project.2006).

Por padrão os diretórios i18n, lib, modules estarão vazios a cada nova aplicação gerada.

No desenvolvimento do protótipo com o Framework Symfony vamos utilizar a seguinte modelagem de dados que foi baseada na inicial porém foram feitas algumas pequenas alterações devido a algumas particularidades do próprio framework.

Na construção de uma aplicação com o symfony recomenda-se utilizar uma padronização, para o bom funcionamento e utilização dos recursos do framework.

Segue a descrição.

- Todas as tabelas do banco de dados devem estar no plural.
- Utilizar como chave primária o campo “id”.
- Para fazer uso de chave estrangeira deve-se utilizar o nome da tabela no singular acrescido do campo id.

  Segue abaixo a modelagem que foi desenvolvida segundo as indicações para gerar aplicação com o framework symfony.

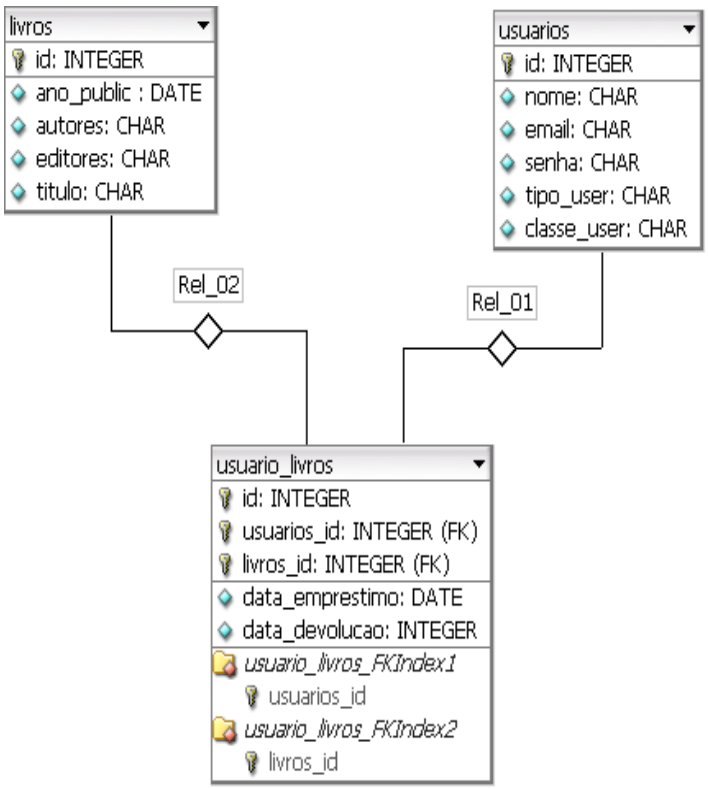

Ilustração 10: Base de dados utilizada no Symfony

Como o Symfony trabalha com modelos XML para o banco você pode estar utilizando uma ferramenta de modelagem qualquer como o DBDESIGNER e gerar o arquivo *schema.xml* para posteriormente conectar com o banco via symfony.

Você pode também escrever sua base neste arquivo XML ou se você preferir utilizar ferramentas de interação com o banco como PHPMyAdmin para construir sua base de dados.

Para trabalhar com uma base de dados existente foi preciso configurar o arquivo *propel.ini* que fica no diretório *config* dentro do pacote *sf_sandbox*.

Após esta configuração, foi só rodar o script abaixo:

- *./symfony.sh propel-build-schema*

Com isto foi gerado o XML da base corrente. Abaixo o exemplo de parte do XML que foi gerado:

```
<?xml version="1.0" encoding="utf-8"?>
<!--Autogenerated by CreoleToXMLSchema!-->
<database name="propel">
  <table name="livros" idMethod="native">
    <vendor type="mysql">
      <parameter name="Name" value="livros"/>  // tabela livros
      <parameter name="Engine" value="MyISAM"/>
      <parameter name="Version" value="10"/>
      <parameter name="Row_format" value="Dynamic"/>
      <parameter name="Rows" value="3"/>
      <parameter name="Avg_row_length" value="44"/>
      <parameter name="Data_length" value="132"/>
      <parameter name="Max_data_length" value="281474976710655"/>
      <parameter name="Index_length" value="2048"/>
      <parameter name="Data_free" value="0"/>
      <parameter name="Auto_increment" value="4"/>
      <parameter name="Create_time" value="2006-10-12 17:35:00"/>
      <parameter name="Update_time" value="2006-10-18 10:44:03"/>
      <parameter name="Check_time" value=""/>
      <parameter name="Collation" value="latin1_swedish_ci"/>
      <parameter name="Checksum" value=""/>
      <parameter name="Create_options" value=""/>
      <parameter name="Comment" value=""/>
    </vendor>
    <column name="id" type="INTEGER" required="true" autoIncrement="true"
primaryKey="true">
      <vendor type="mysql">
        <parameter name="Field" value="id"/> //campo ID
        <parameter name="Type" value="int(10) unsigned"/>
        <parameter name="Null" value="NO"/>
        <parameter name="Key" value="PRI"/>
        <parameter name="Default" value=""/>
        <parameter name="Extra" value="auto_increment"/>
      </vendor>
    </column>
    <column name="ano_public" type="DATE">
      <vendor type="mysql">
        <parameter name="Field" value="ano_public"/>
        <parameter name="Type" value="date"/>
        <parameter name="Null" value="YES"/>
        <parameter name="Key" value=""/>
        <parameter name="Default" value=""/>
        <parameter name="Extra" value=""/>
      </vendor>
    </column>
    <column name="autores" type="VARCHAR" size="50">
      <vendor type="mysql">
        <parameter name="Field" value="autores"/>
        <parameter name="Type" value="varchar(50)"/>
        <parameter name="Null" value="YES"/>
        <parameter name="Key" value=""/>
        <parameter name="Default" value=""/>
        <parameter name="Extra" value=""/>
      </vendor>
```

```
  </column>
  <column name="editores" type="VARCHAR" size="50">
   <vendor type="mysql">
    <parameter name="Field" value="editores"/>
    <parameter name="Type" value="varchar(50)"/>
    <parameter name="Null" value="YES"/>
    <parameter name="Key" value=""/>
    <parameter name="Default" value=""/>
    <parameter name="Extra" value=""/>
   </vendor>
  </column>
  <column name="titulo" type="VARCHAR" size="50">
   <vendor type="mysql">
    <parameter name="Field" value="titulo"/>
    <parameter name="Type" value="varchar(50)"/>
    <parameter name="Null" value="YES"/>
    <parameter name="Key" value=""/>
    <parameter name="Default" value=""/>
    <parameter name="Extra" value=""/>
   </vendor>
  </column>
 </table>
```

Tabela 16: Exemplo do arquivo schema.xml tabela livros

Houve um pequeno problema com a ordem em que o XML foi gerado, ele gerou uma tabela associativa antes da associada e quando eu tentei gerar o modelo não funcionou até eu corrigir a ordem no arquivo .

Para gerar os modelos foi preciso executar o script abaixo:

- *./symfony.sh propel-build-model*

Para gerar os códigos com as funções básicas foi utilizado:

- *./symfony.sh propel-generate-crud frontend livros Livros*.
- *./symfony.sh propel-generate-crud frontend usuarios Usuarios*
- *./symfony.sh propel-generate-crud frontend usuariolivros UsuarioLivros*

Depois executei também o script abaixo como recomenda o manual:

- *./symfony.sh cc-frontend-config.*

  Em um teste que eu fiz com uma máquina sem ter instalado a versão PEAR mas baixado somente o pacote sf_sandbox, este comando acima citado não respondeu, então utilizei o comando abaixo e a aplicação funcionou.

- *./symfony.sh clear-cache*

Como resultado disso obteve-se o acesso ao menu dos livros, dos usuários e da tabela dos empréstimos com as funções básicas já implementadas, segue abaixo a ilustração com o resultado:

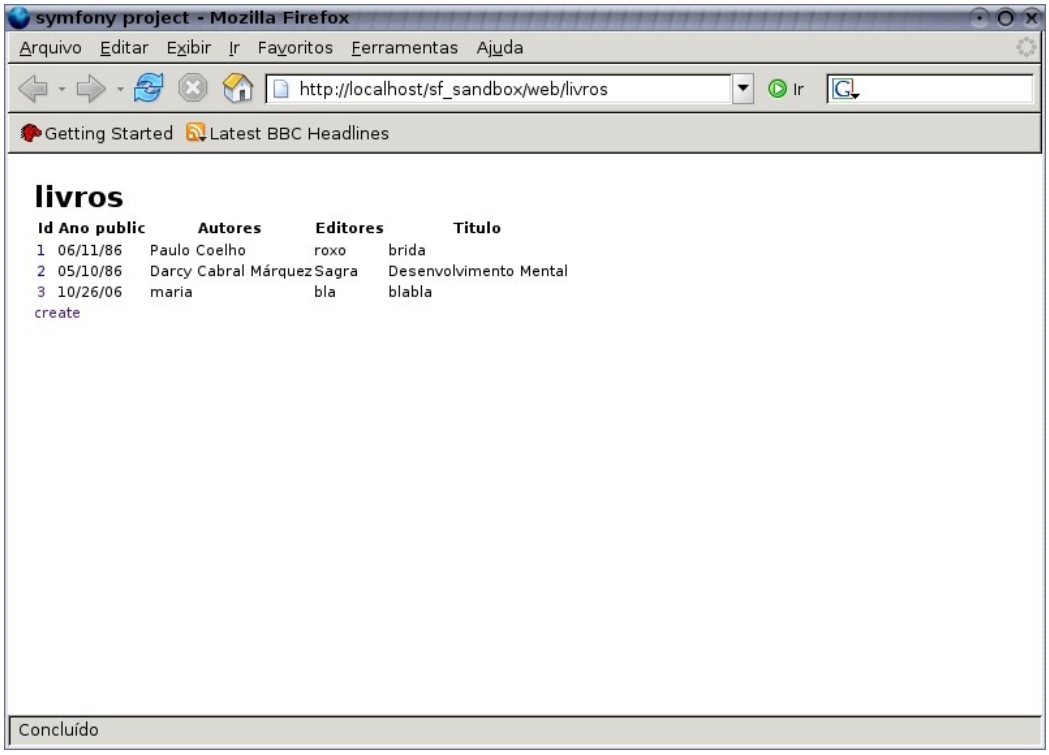

Ilustração 11: Exemplo tela dos livros, aplicação gerada com o symfony

Depois de gerado a aplicação base, foi editado o arquivo layout.php localizado em: */var/www/sf_sandbox/apps/frontend/templates* , para customizar um pouco a aplicação: ele é o arquivo de layout geral da aplicação symfony. Segue tela do layout alterado:

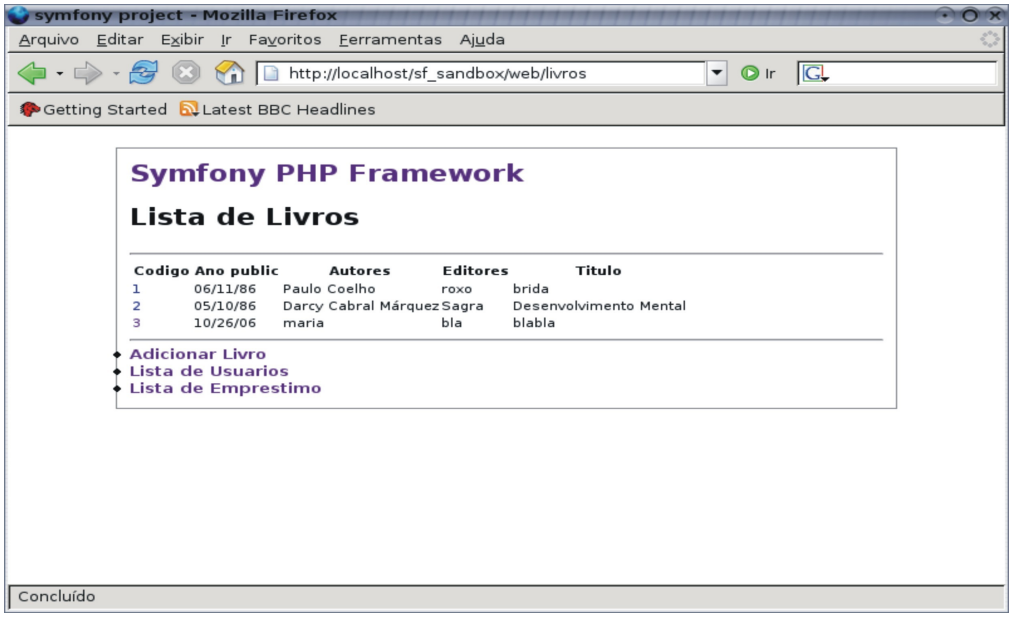

Ilustração 12: Tela dos livros com alteração de layout

Segue abaixo mais alguns exemplos das telas da aplicação:

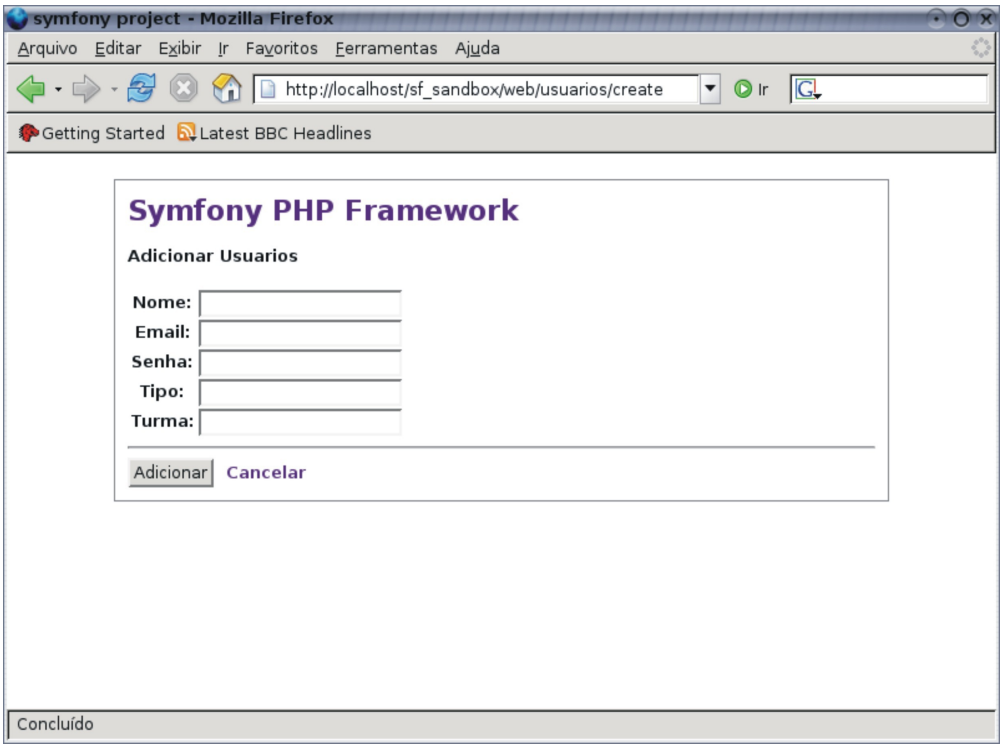

Ilustração 13: Exemplo adicionar novo usuário

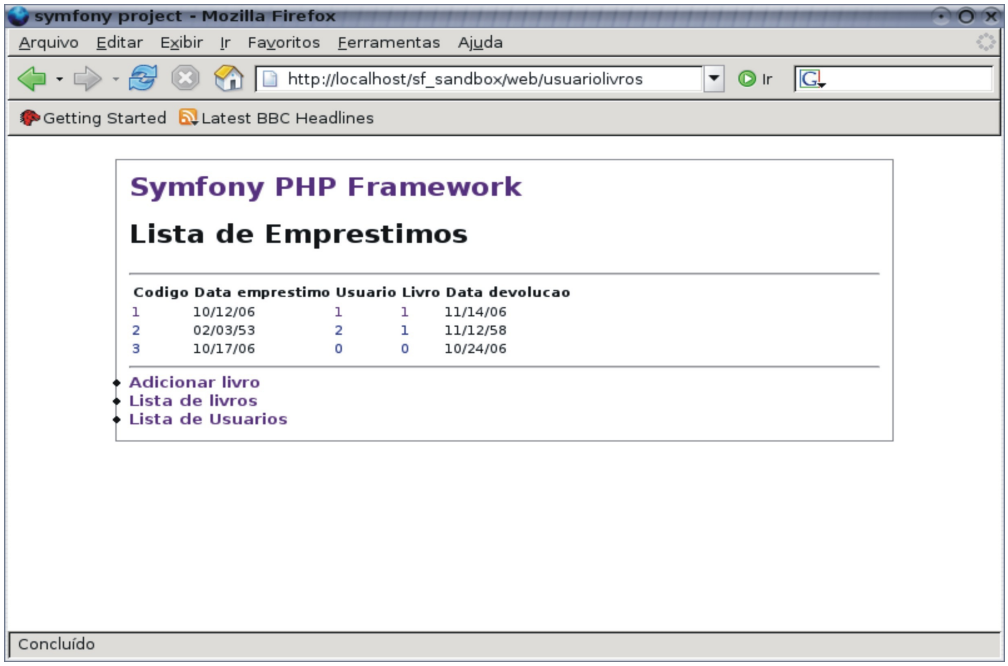

Ilustração 14: Tabela de empréstimos

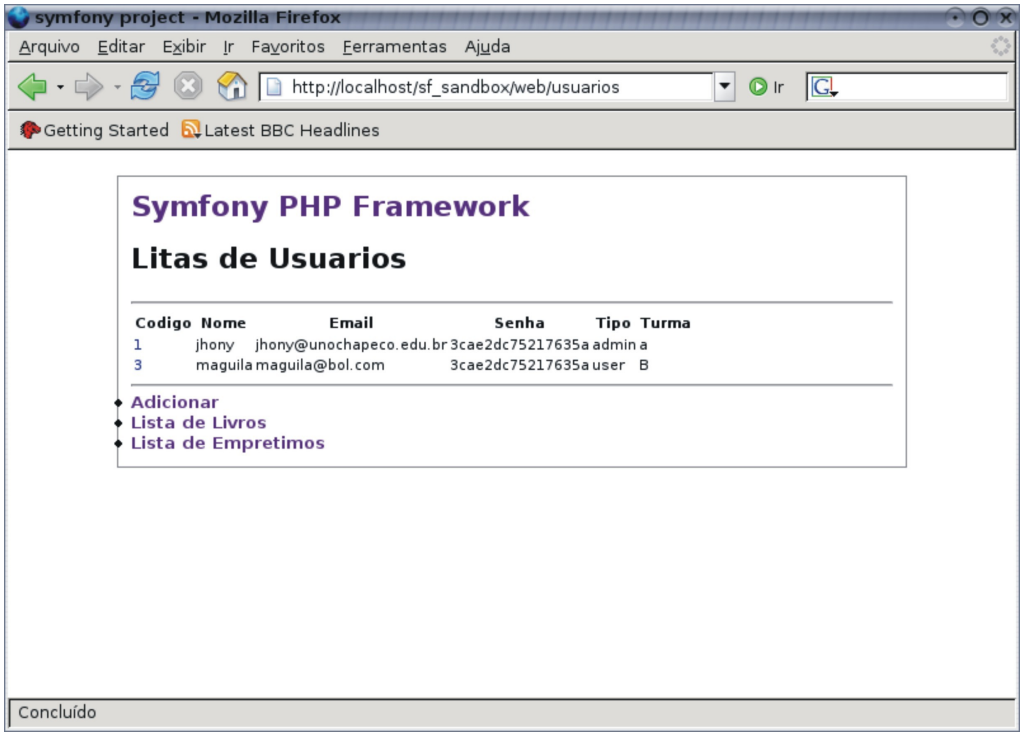

Ilustração 15: Exemplo lista de usuários

Segue abaixo alguns exemplos de códigos gerados pelo script do symfony, já com algumas alterações para melhorar a usabilidade do protótipo.

```
<?php
// auto-generated by sfPropelCrud
// date: 2006/10/19 10:51:50
?>
<br>
<h3> Detalhes Livros</h3> // HTML da aplicação
<br>
<table>
<tbody>
<tr>
<th>Id: </th>
<td><?php echo $livros->getId() ?></td>
</tr>
<tr>
<th>Ano public: </th>
<td><?php echo $livros->getAnoPublic() ?></td>
</tr>
<tr>
<th>Autores: </th>
<td><?php echo $livros->getAutores() ?></td>
</tr>
<tr>
<th>Editores: </th>
<td><?php echo $livros->getEditores() ?></td>
</tr>
<tr>
<th>Titulo: </th>
<td><?php echo $livros->getTitulo() ?></td>
</tr>
</tbody>
</table>
<hr />
<?php echo link_to('Editar', 'livros/edit?id='.$livros->getId()) ?>
 <?php echo link_to('Voltar', 'livros/list') ?>
```

Tabela 17: Exemplo do arquivo showsuccess.php

Segue abaixo exemplo do arquivo que contempla as classes e as funções de ações do symfony para a tabela livros.

```
<?php
// auto-generated by sfPropelCrud
// date: 2006/10/19 10:51:50
?>
<?php

/**
 * livros actions.
 *
 * @package    ##TCC II##
 * @subpackage livros
 * @author     Jhony Maseto
 * @version    SVN: $Id: actions.class.php 1415 2006-06-11 08:33:51Z fabien $
 */
class livrosActions extends sfActions
{
  public function executeIndex ()
  {
    return $this->forward('livros', 'list');
  }

  public function executeList ()
  {
    $this->livross = LivrosPeer::doSelect(new Criteria());
  }

  public function executeShow ()
  {
    $this->livros = LivrosPeer::retrieveByPk($this->getRequestParameter('id'));
    $this->forward404Unless($this->livros);
  }

  public function executeCreate ()
  {
    $this->livros = new Livros();

    $this->setTemplate('edit');
  }

  public function executeEdit ()  //função de edição
  {
    $this->livros = LivrosPeer::retrieveByPk($this->getRequestParameter('id'));
    $this->forward404Unless($this->livros);
  }

  public function executeUpdate ()  // função de update no banco
  {
    if (!$this->getRequestParameter('id', 0))
    {
      $livros = new Livros();
    }
    else
    {
      $livros = LivrosPeer::retrieveByPk($this->getRequestParameter('id'));
```

```
    $this->forward404Unless($livros);
   }

   $livros->setId($this->getRequestParameter('id'));
   if ($this->getRequestParameter('ano_public'))
   {
    list($d, $m, $y) = sfI18N::getDateForCulture($this-
>getRequestParameter('ano_public'), $this->getUser()->getCulture());
    $livros->setAnoPublic("$y-$m-$d");
   }
   $livros->setAutores($this->getRequestParameter('autores'));
   $livros->setEditores($this->getRequestParameter('editores'));
   $livros->setTitulo($this->getRequestParameter('titulo'));

   $livros->save();

   return $this->redirect('livros/show?id='.$livros->getId());
  }

  public function executeDelete ()
  {
   $livros = LivrosPeer::retrieveByPk($this->getRequestParameter('id'));

   $this->forward404Unless($livros);

   $livros->delete();

   return $this->redirect('livros/list');
  }
}
```

Tabela 18: Exemplo do arquivo actions.class.php

5.5 DESENVOLVIMENTO NO CAKE

Para se desenvolver com CAKE-PHP faz-se necessário estudar e aplicar alguns passos relativos a sua instalação e configuração.

- O primeiro é acessar o site do Framework e baixar a ultima versão estável.
- Depois de salvar o pacote de arquivos do Cake na máquina, descompactei o mesmo e copiei todo o seu conteúdo para o diretório root do servidor web.
- Adicionei a pasta cake as permissões para que ela pude-se ser executada

pelo apache.

Ao acessar no navegador “http://l127.0.0.1/cake” visualizei uma mensagem informando sobre o status da conexão com o banco de dados ("Your database configuration file is not present.").

Para melhor entendimento do funcionamento do cake descreve-se na figura abaixo sua árvore de diretórios e a explicação do que cada um contém.

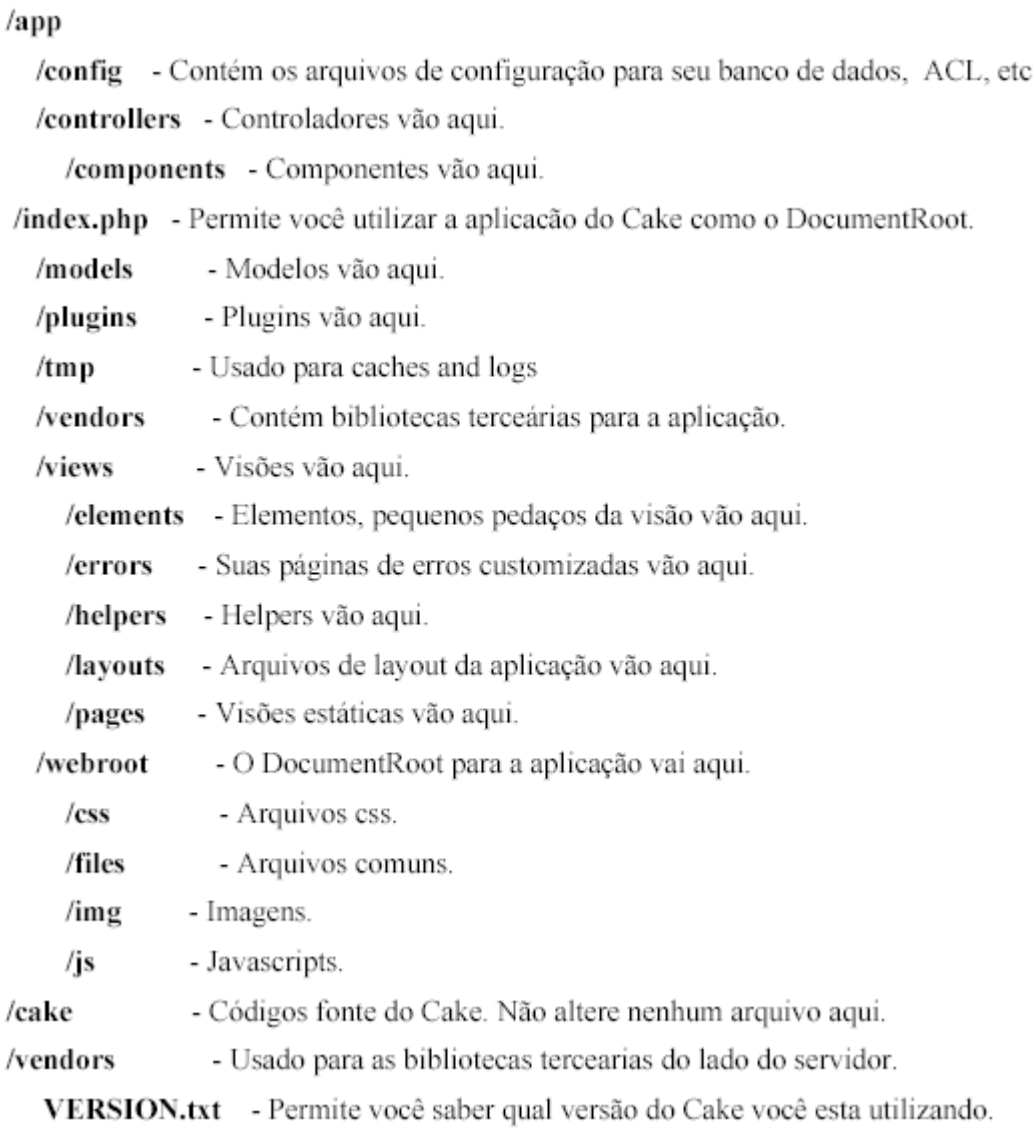

Ilustração 16: Árvore dos diretórios do Cake-PHP (Cake Software Foundation.2006).

Como próximo passo concentre-se na sua base de dados, pois o Cake tem algumas particularidades relativas a isso. Para que sua aplicação funcione corretamente o cake recomenda utilizar uma padronização.

- Todas as tabelas devem estar no plural.
- A chave primária para todas deve ser o campo “id”.
- Para utilizar chave estrangeira, deve possuir nome da tabela no singular acrescido do campo id.

  Ex: “usuario_id”.
- O relacionamento “n – n”, deve ser organizado em ordem alfabética, e o primeiro nome no singular acrescido de um “_” nome segunda tabela plural. Ex: usuario_livros;

Segue abaixo a modelagem que foi adaptada da padrão e utilizada no protótipo do cake:

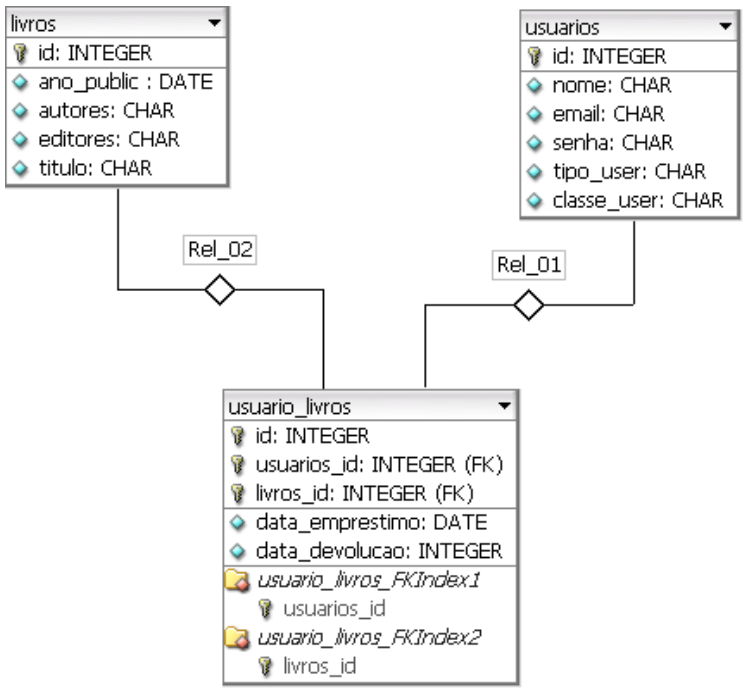

Ilustração 17: Modelo dos dados para utilização do Cake-PHP

A base de dados foi criada manualmente de acordo com a modelagem acima.

Existem alguns artifícios para desenvolver aplicações com o Framework Cake dentre eles foi abordado no protótipo um gerador de código chamado BAKE , que interpreta as instruções passadas a ele pelo shell e gera códigos PHP. Para que fosse possível utilizá-lo via shell precisamos instalar o php-cli, que é o interpretador de php via linha de comando. No Ubuntu nos basta um simples comando:

- *sudo apt-get install php-cli.*

Depois da instalação entrei na pasta cake e rodei o utilitário BAKE com o seguinte comando:

- *php cake/scripts/bake.php -project /var/www/cake/projeto_cake*

O resultado foi uma tela semelhante a está:

```
jhony@bravo:/var/www/cake$ php cake/scripts/bake.php -project
/var/www/cake/projeto_cake


 ___  __ _  _ ___  __ _  _ __    __  __ _  _ ___
|   |__| |_/  |__  |__] |__| |__]   |__] |__| |_/  |__
|___ |  | | \_ |___ |    |  | |      |__] |  | | \_ |___
---------------------------------------------------------------


---------------------------------------------------------------
Skel Directory: /var/www/cake/cake/scripts/templates/skel
Will be copied to:
New App Directory: /var/www/cake/projeto_cake
---------------------------------------------------------------

Look okay? (y/n/q)
[y] >
```

Tabela 19: Exemplo para criação de projetos com o script bake

Com isso adicionei um novo projeto ao cake, os arquivos foram gerados com o comando a seguir:

- *php cake/scripts/bake.php -app projeto_cake*

```
jhony@bravo:/var/www/cake$ php cake/scripts/bake.php -app projeto_cake


 ___  __  _  _  ___  __  _  _  __    __  __ _  _  ___
|   |__| |_/  |__  |__] |__| |__]   |__] |__| |_/  |__
|___ |  | | \_ |___ |    |  | |      |__] |  | | \_ |___
---------------------------------------------------------------



Your database configuration was not found. Take a moment to create one:


---------------------------------------------------------------
Database Configuration Bake:
---------------------------------------------------------------

What is the hostname for the database server?
[localhost] >
```

Tabela 20: Exemplo de Geração dos arquivos pelo Bake

Então forneci as informações relativas a minha base de dados como:

- Nome do servidor : 127.0.0.1
- Nome do usuário do banco: root
- A senha do usuário do banco:******
- Nome da database: hora

Após ter sido gerado os códigos com as estruturas Livros e Usuários e sua associação você poderá alterar os códigos conforme sua intenção.

Segue abaixo algumas telas gerada pelo script bake :

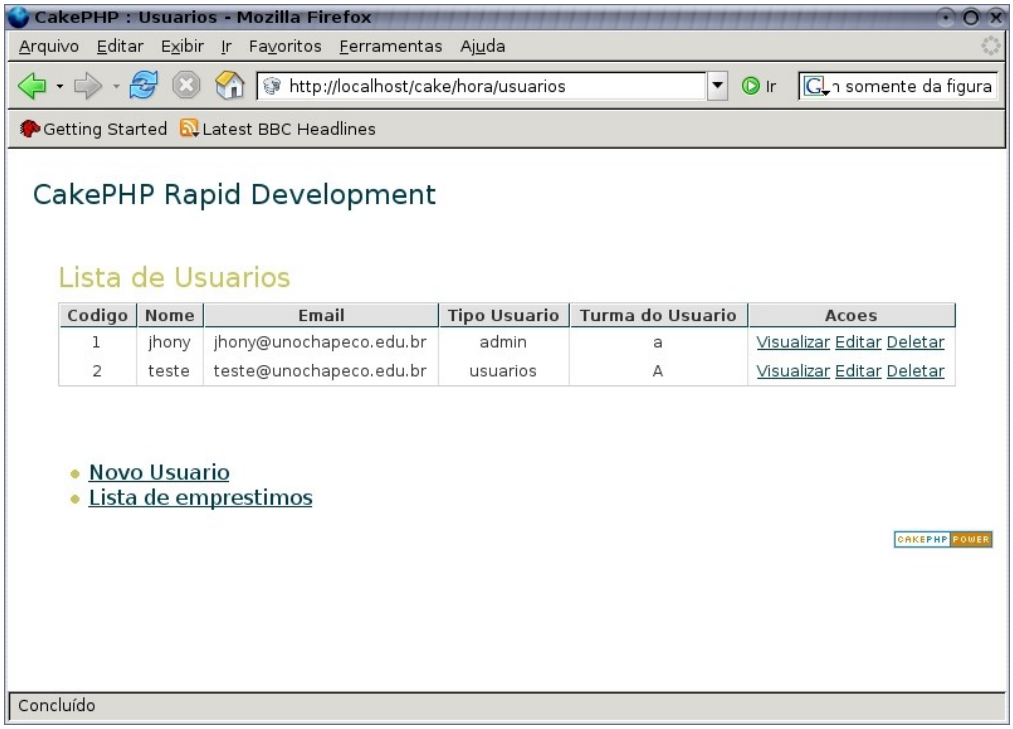

Ilustração 18: Lista de usuários cadastrados no sistema.

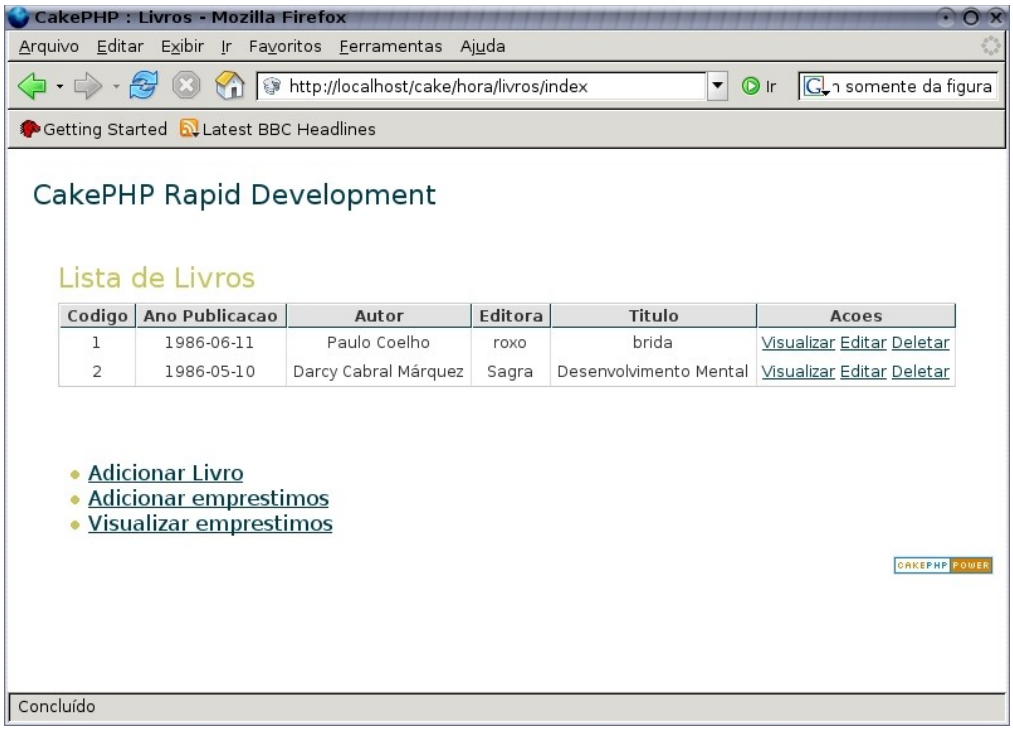

Ilustração 19: Lista dos livros cadastrados no sistema.

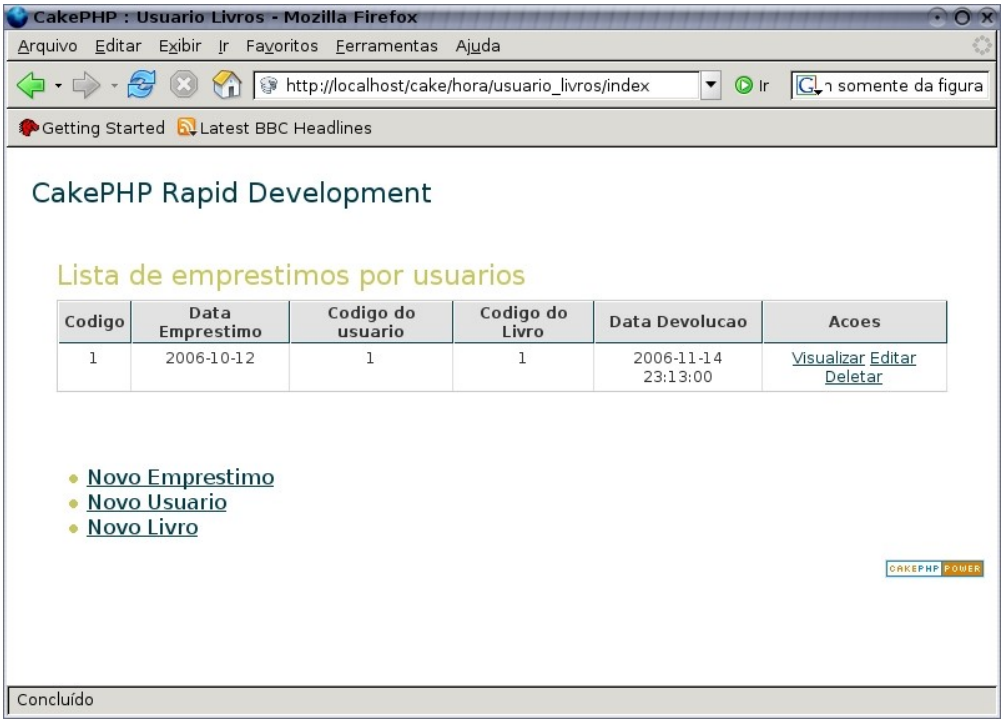

Ilustração 20: Lista de empréstimos gerados no sistema

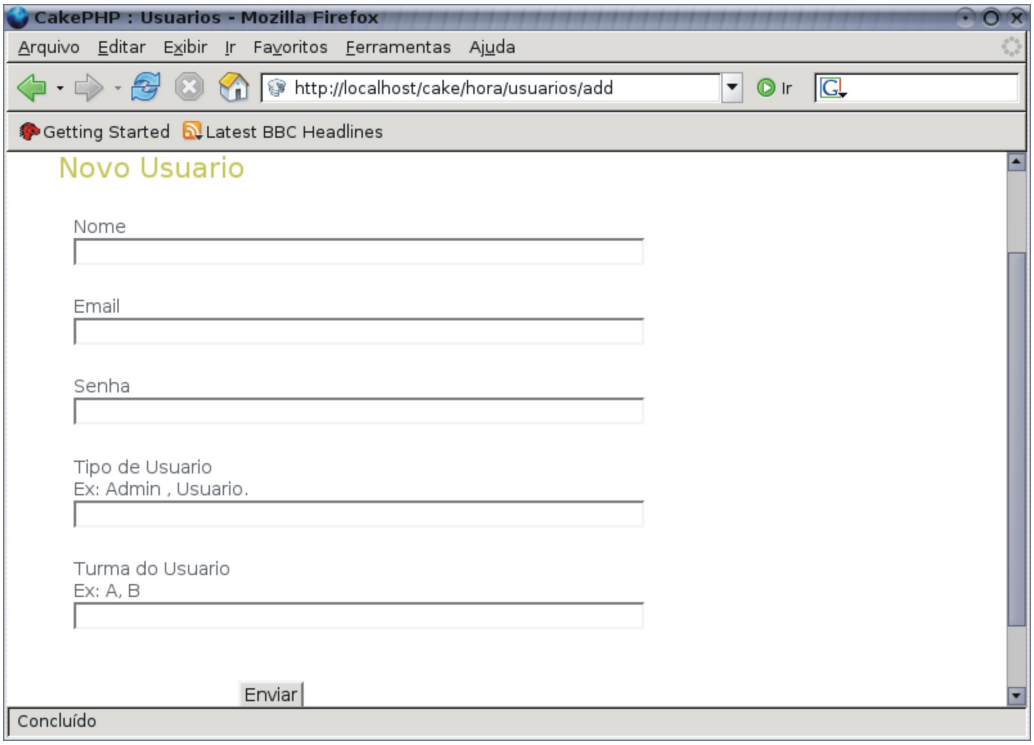

Ilustração 21: Adicionar novo usuário

O script Bake gera todas as aplicações em inglês, send, que, para visualizá-las como na ilustração acima, foi necessário alterá-las manualmente.

A aplicação de login não foi gerada com o script bake mas foi feita segundo exemplos da documentação do prórpío framework. Segue abaixo exemplo:

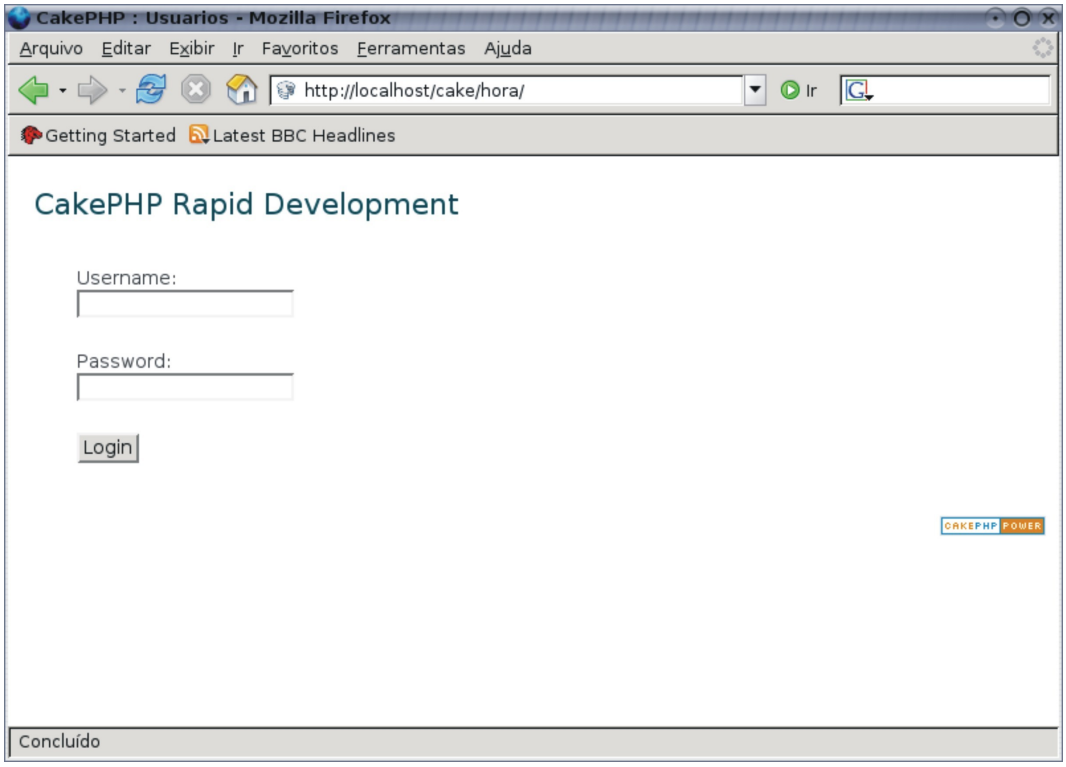

Ilustração 22: Tela de login da aplicação com cake

Segue abaixo exemplos dos códigos da aplicação com cake, já customizados para a melhora da utilização.

```
<h2>Visualizar Livros</h2>

<dl>
	<dt>Id</dt>
	<dd> <?php echo $livro['Livro']['id']?></dd>
	<dt>Ano Public</dt>
	<dd> <?php echo $livro['Livro']['ano_public']?></dd>
	<dt>Autores</dt>
	<dd> <?php echo $livro['Livro']['autores']?></dd>
	<dt>Editores</dt>
	<dd> <?php echo $livro['Livro']['editores']?></dd>
	<dt>Titulo</dt>
	<dd> <?php echo $livro['Livro']['titulo']?></dd>
</dl>
<ul class="actions">
	<li><?php echo $html->link('Editar Livro',   '/livros/edit/' . $livro['Livro']['id']) ?
> </li>
	<li><?php echo $html->link('Deletar Livro', '/livros/delete/' . $livro['Livro']['id'],
null, 'Você esta prestes a deletar o livro de id ' . $livro['Livro']['id'] . '?') ?> </li>
	<li><?php echo $html->link('Listar Livros',   '/livros/index') ?> </li>
	<li><?php echo $html->link('Novo Livro','/livros/add') ?> </li>
</ul>
```

Tabela 21: Exemplo do arquivo view.thtml responsável pela exibição dos dados da tabela livros

```
<?php
class Livro extends AppModel
{
	var $name = 'Livro';

}
?>
```

Tabela 22: Exemplo do arquivo livro.php que corresponde a implementação da classe model

```
<?php
class LivrosController extends AppController
{
	//var $scaffold;
	var $name = 'Livros';
	var $helpers = array('Html', 'Form' );

	function index() {  //função que mostra todos os dados em lista
		$this->Livro->recursive = 0;
		$this->set('livros', $this->Livro->findAll());
	}

	function add() {
		if(empty($this->data)) {
			$this->render();
		} else {
			$this->cleanUpFields();
			if($this->Livro->save($this->data)) {
				$this->flash('Livro saved.', '/livros/index');
			} else {
			}
		}
	}

	function edit($id) {
		if(empty($this->data)) {
			$this->data = $this->Livro->read(null, $id);
		} else {
			$this->cleanUpFields();
			if($this->Livro->save($this->data)) {
				$this->flash('Livro saved.', '/livros/index');
			} else {
			}
		}
	}

	function view($id) { //função de visualização
		$this->set('livro', $this->Livro->read(null, $id));
	}

	function delete($id) { //função de deleção
		if($this->Livro->del($id)) {
			$this->flash('Livro deleted: id '.$id.'.', '/livros/index');
		}
	}

}
?>
```

Tabela 23: Exemplo do arquivo de controller da aplicação livros_controllers.php, responsável pelas funções da tabela livros.

6. CONCLUSÕES E TRABALHOS FUTUROS

O presente trabalho teve como objetivo avaliar e divulgar algumas das novas ferramentas de auxílio aos desenvolvedores de PHP, os frameworks. Sendo que não foi encontrado nenhuma norma que pudesse auxiliar na escolha do melhor framework, mas apenas alguns pequenos comparativos dispostos em sites distintos, retemos o que tinha de mais interessante nos comparativos e mesclamos ao projeto.

Procurou-se pesquisar e expor três frameworks diferentes: o Cake por ser muito comentado no momento e que demonstrava estar crescendo de forma acelerada, o Prado por ser diferenciado dos demais tendo grande similaridade com o Delphi e com ASP, e o Symfony que possuía quase todas as características pesquisadas embutida no seu projeto.

Após eleitos os três frameworks a serem analisados buscou-se obter todas as informações que fossem relevantes a respeito dos mesmos para retratar no projeto. Feito toda esta descrição das características e funcionalidades dos frameworks, aplicou-se no projeto as comparações obtidas pelos sites pesquisados.

Mas uma dúvida restava, Como fazer um protótipo simples e produtivo utilizando os frameworks. Tomou-se conhecimento então de uma comparação similar que já tinha sido aplicado com sucesso a alguns frameworks de python com o

intuito de avaliar as suas funcionalidades.

Tentou-se contato com a autora da comparação, mas não obtivemos êxito. Como o comparativo da autora é muito interessante, foi adaptado o exemplo disponível no site do PyWebOff e o descrito para ser aplicado no protótipo.

A experiência de desenvolvimento com o cake muito interessante, tendo como principal destaque os geradores de código e a forma como ele estrutura os diretórios da aplicação, colocando cada coisa no seu devido lugar. O cake ainda possui alguns fatores desfavoráveis, como a sua documentação que ainda deixa a desejar, pois ele não contempla o conceito de internacionalização, não faz nenhuma referencia a integração PEAR, não existe um mecanismo que faça a atualização do framework. Porém, como ele ainda esta em fase de crescimento acredita-se que logo vai comportar todas as necessidades do desenvolvedor.

Como sugestão aos desenvolvedores pode-se citar:

- Melhoria na documentação.
- Embutir no projeto a integração PEAR.
- Pensar em uma forma mais prática para fazer a atualização das versões.
- Aprimorar ainda mais o script de gerar código, talvez usando PHP-GTK[34].

O Symfony é um framework bem completo, pode se dizer que ele contempla praticamente todos os itens pesquisados e mais algumas funcionalidades como

34 Toolkit para geração de interfaces gráficas que pode ser integrado ao PHP

gerador de código, controle de versões, documentação vasta , sua estrutura é disponibilizada em três formas diferentes, como já explicado no decorrer do projeto. È realmente muito interassante a integração com o XML proporcionada pelo framework, que permite utilizar um programa para modelagem de dados e exportar o arquivo xml da base de dados direto para o framework, e então gerar a aplicação com os scripts do Symfony. Um ponto negativo é a forma com que o Symfony emprega o MVC, pois me pareceu bastante confusa a organização dos arquivos.

Como sugestão aos desenvolvedores pode-se citar:

- Melhorias na estrutura.
- Aprimorar ainda mais o gerador de código, também vislumbrando a possibilidade de utilizar PHP-GTK.

O prado framework é um framework bem diferente dos outros dois já descritos pois não implementa MVC e o conceito de orientado a eventos é bem presente nesta estrutura. Para quem está familiarizado com o processo de desenvolvimento no Delphi realmente deve se sentir muito confortável em trabalhar com este framework. Por não utilizar o MVC a estrutura de arquivos fica praticamente a cargo do desenvolvedor. Para as proximas versões acredita-se que os desenvolvedores já anexem ao projeto pois presenciei uma grande diferença na organização da estrutura das primeiras versões para a atual, apesar de não terem sido muitas as versões. Outro ponto de destaque é o funcionamento dos “data grid´s”. Realmente é muito interessante a maneira como são construídos e alterados.

Como sugestão aos desenvolvedores pode-se citar:

- Um gerador de código.
- A criação de uma interface GTK, imitando a idéia do Delphi com botões e eventos prontos.
- Uma melhora na documentação.
- Embutir no projeto o uso de MVC para ajustar a estrutura de arquivos.
- Uma melhora na documentação do projeto.
- Embutir a conexão com o banco de dados sem ter que utilizar ADODB.

Após uma boa pesquisa posso dizer que os frameworks são boas ferramentas para programadores experientes, que sabem como maximizar o reuso da sua aplicação e trazer melhores resultados com menores esforços.

Depois de todo este estudo pode -se afirmar que não existe o melhor framework, mas sim o mais indicado para cada tipo de caso. Para iniciantes, certamente o Cake é a melhor opção, por ser o mais fácil de aprender e o mais rápido para começar a trabalhar. Para programadores experientes ou organizações que visam começar um grande projeto e querem apostar em frameworks, certamente o Symfony é a melhor opção, e para quem está familiarizado com Delphi e não quer trabalhar com MVC o Prado é uma boa opção.

6.1 Tabela explicativa

Segue abaixo uma tabela com as avaliações dos frameworks para melhor visualização das conclusões.

Frameworks	
	Pontos Fortes
PRADO	Uso dos DATAGRID´S
SYMFONY	Contempla todos os itens analisados, integração com o XML
CAKE	Estrutura de diretórios, facilidade na utilização
	Facilidade na utilização
PRADO	Moderado
SYMFONY	Moderado a difícil
CAKE	Fácil
	Pontos Fracos
PRADO	Não contempla MVC, não possui gerador de códigos
SYMFONY	Muito complexo
CAKE	Problemas na atualização, não contempla a internacionalização
	Produtividade
PRADO	Média
SYMFONY	Boa
CAKE	Muito boa

Tabela 24: Tabela das conclusões

6.2 TRABALHOS FUTUROS

Com relação aos trabalhos futuros poderia indicar:

- Ampliar o estudo estendendo a mais frameworks.
- Completar este projeto aplicando mais quesitos aos frameworks
- Fazer uma avaliação similar, com frameworks de PHP, Phyton e Java.

7. REFERÊNCIAS

Arsys Internet S.L.2006. Acessado em 15/05/2006 disponível em: http://www.arsys.pt/suporte/programacao/linux.htm

BURBECH.2002. Acessado em 08/05/2006 disponível em: http://hercules.nce.ufrj.br/arq-mvc.html

Cake Software Foundation.2006 acessado em 01/05/2006 disponível em : http://www.cakephp.org/

Canal Html.2006. Acessado em 15/05/2006 disponível em: http://www.htmlstaff.org/entrevistas/entrevistas24.php

Carneiro,Rafael. Acessado em 26/05/2006 Disponível em: http://www.univel.br/roberta/FRAMEWORKS/frameworks%5B1%5D.pdf

Cavalcanti,Éric, PRADO - PHP 5 Framework.2004 Acessado em 01/05/2006 disponível em: http://www.linhadecodigo.com.br/artigos/videos/eric_cavalcanti/prado/prado.html

Comunidade Cyaneus.2006. Acessado em 14/05/2006 disponível em: http://4h17.cyaneus.net/frameworksphp

Dall'Oglio, Pablo.2005. As novidades do PHP5 Acessado em 23/05/2006 Disponível em: http://www.dalloglio.net/events/php5.pdf

De Souza,Marcos Vinícius Bittencourt. Estudo Comparativo entre Frameworks Java para Construção de Aplicações Web.2004. Acessado em 27/04/2006 disponível em: http://www-usr.inf.ufsm.br/~marvin/monografia.pdf

DMITRUK, Hilda Beatriz (Org). Cadernos Metodológicos: Diretrizes do trabalho científico . 6. ed. rev. ampl. atual. Chapecó: Argos, 2004.

Ferreira, Miguel. ASP.NET Caching - Escalabilidade e desempenho à mão Acessado em 22/05/2006 Disponível em: http://www.microsoft.com/brasil/msdn/tecnologias/aspnet/aspnetcache.mspx?mfr=true

Framework Prado.2005 Acessado em 02/05/2006 disponível em: http://www.xisc.com/

h3rald.2006 Acessado em 05/05/2006 disponível em: http://www.h3rald.com/articles/view/rails-inspired-php-frameworks

Lozano,Fernando Programação Orientada a Objetos com PHP.2002. Acessado em 18/05/2006 disponível em: http://www.lozano.eti.br/palestras/oo-php.pdf

Manual do PHP Acessado em: 02/05/2006 disponível em: http://ceres.inf.ufsc.br/doc/php-br/function.sqlite-create-aggregate.html

Martin,James. Análise e projeto orientados a objeto/ James Martin, James J. Odell; tradução José Carlos Barbosa dos Santos; revisão técnica Ronald Stevis Cassiolato. -São Paulo: Makron Books, 1995.

Minetto,Elton Luís.2006. O uso de templates em PHP Acessado em 05/06/06 Disponibilizado em: http://200.135.240.21/~elm/docs/php_templates.htm

Morais, João Cruz. PHP Webpages Dinâmicas Acessado em 02/05/2006 Disponibilizado em: http://64.233.187.104/search?q=cache:Iv0hyoMPHV4J:groups.tagus.ist.utl.pt/neiist/web/eventos/ca2/aps/php-mod2-alameda-14-out-2004/php_-_webpages_dinamicas_-_modulo2.pps+exemplos+de+encapsulamento+em+php&hl=pt-BR&gl=br&ct=clnk&cd=1

Object-relational mapping.2006. Acessado em 15/05/2006 disponível em: http://en.wikipedia.org/wiki/Object-relational_mapping

Pear The PHP Group.2005 Acessado em 25/05/2006 disponibilizado em: http://pear.php.net/

Pradosoft Component Framework for PHP 5.2006 Acessado em 02/05/2006 disponível em: http://www.pradosoft.com/

Pulido,Nick The Web 2.0 Dev.2006 Acessado dia 05/05/2006 disponível em : http://www.theweb20dev.com/wordpress/2006/05/03/5-next-generation-php-frameworks/

PyWebOff .Comparativo de Frameworks Python Acessado em 13/09/2006 disponível em : http://www.third-bit.com/pyweb/index.html

Rangel, Eustáquio.Tutorial de Rails. Acessado em 22/05/2006 Disponível em: http://plentz.org/unsorted/ruby/tutorialrails.pdf

Sant'Anna,Mauro,Mantendo estado em páginas ASP.NET.2002 acessado em 02/05/2006 disponível em: http://72.14.203.104/search?q=cache:Gye7UImWOYMJ:www.mas.com.br/Artigos/mantendo.htm+viewstate&hl=pt-BR&gl=br&ct=clnk&cd=1

Sauvé Jacques Philippe. O que é um framework Acessado em 24/10/2006 disponível em: http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/frame/oque.htm

Souza.S.J.Sandro.Vantagens do PHP 5 frente ao ASP.NET.2005 acessado dia 10/05/2006 disponível em : http://phpbrasil.com/articles/article.php/id/1173

Symfony-project Framework de PHP 5.2006 Acessado em 02/05/2006 disponibilizado em : http://www.symfony-project.com

Strapazzon, Itanor José. Um Ambiente para a criação de aplicações baseadas na arquitetura MODEL- VIEW -CONTROLLER,pag 29. Chapecó: Monografia Unochapecó,2004.

Tableless.2006. Acessado em 08/05/2006 disponível em: http://pt.wikipedia.org/wiki/Tableless

The Pallett Group.2005. Acessado em 14/05/2006 disponível em: http://www.phpit.net/article/ten-different-php-frameworks/1/

The PHP Group.2005. Acessado em 03/05/2006 disponível em: http://www.php.net/manual/pt_BR/history.php

The PHP Group.2006 Acessado em 03/05/2006 disponível em: http://www.php.net/manual/pt_BR/introduction.php

(VIEIRA,Victor. Www.com.br. Por dentro do Ajax. Editora Europa. São Paulo, n.63,p.50-52,2006).

Wikipédia, a enciclopédia livre.2006 Acessado em 01/05/2006 disponível em: http://pt.wikipedia.org/wiki/Framework

Wikipédia, a enciclopédia livre.2006 acessado em 11/05/2006 disponível em: http://pt.wikipedia.org/wiki/PHP#column-one

Wikipédia, a enciclopédia livre.2006 Acessado em 19/05/2006 disponível em: http://pt.wikipedia.org/wiki/PHP-GTK