

Documentação Sistema de Estágios

Gabryel Rigol da Silva

Lucas Marques Macedo Navarezi

Victor Gaíva

6 de outubro de 2015

Sumário

1	Introdução	3
2	Configuração URL	3
3	Arquivos Principais	4
3.1	index.php	4
3.2	config.php	4
3.3	loader.php	4
3.4	global-functions.php	4
4	Classes Padrão	5
4.1	class-SistemaMvc.php	5
4.2	class-SistemaBD.php	5
4.2.1	query:	5
4.2.2	insert:	5
4.2.3	update:	6
4.2.4	delete:	6
4.3	class-UserLogin.php	6

1 Introdução

O Sistema de Estágios é um site desenvolvido usando o modelo MVC(Model-View-Controller) com a finalidade de agilizar a aprovação da documentação referente aos estágios dos alunos, inicialmente da FACOM-UFMS.

Este documento possui a finalidade de demonstrar toda a arquitetura do site e o funcionamento das páginas, para uma boa compreensão das pessoas que irão fazer a manutenção deste sistema.

2 Configuração URL

Para definirmos qual controlador e qual ação será usado, nós passamos os nomes como parâmetros, assim temos o seguinte formato:

`http://www.sistema.com/index.php?url=controlador/ação/parametro1/parametro2/etc...`

Mas para uma melhor visualização da url utilizamos o arquivo .htaccess para fazer a reescrita dela, assim ela ficará no seguinte formato:

`http://www.sistema.com/controlador/ação/parametro1/parametro2/etc...`

3 Arquivos Principais

3.1 index.php

O arquivo index.php apenas inclui o arquivo config.php

3.2 config.php

O arquivo config.php é responsável por definir algumas constantes e incluir o arquivo loader.php;

3.3 loader.php

O arquivo loader.php inclui o arquivo global-functions.php, que é responsável por manter todas as funções globais, e também é responsável por instanciar a classe [SistemaMVC](#) que vai controlar todo o início da aplicação.

3.4 global-functions.php

A função mais importante que temos é a `_autoload`, para carregar classes automaticamente.

4 Classes Padrão

Dentro do diretório Classes temos os arquivos que contém as classes bases do sistema. A padronização dos nomes dos arquivos é class-NomeDaClasse.php

4.1 class-SistemaMvc.php

O construtor dessa classe é responsável por ler os parâmetros passados na url, incluir o arquivo que contem o controlador, instanciar a classe do controlador e chamar a ação.

Caso o controlador não seja passado, será incluído o controlador Login. Caso a ação não seja passada, será chamada a ação index.

4.2 class-SistemaBD.php

Classe responsável por fazer a conexão com o banco de dados e realizar consultas, possuímos a seguintes funções:

4.2.1 query:

Parâmetros: \$stmt(String da consulta SQL), \$data_array(array dos valores);

Funcionamento: Prepara e executa a consulta, caso a consulta não seja feita retorna false, senão retorna o resultado da consulta.

Exemplo de uso:

```
db->query('SELECT * FROM tabela WHERE campo = ? AND outro_campo = ?',  
          array( 'valor', 'valor' ))
```

4.2.2 insert:

Parâmetros: \$table, os valores da array são coletadas pela função func_get_args();

Funcionamento: Cria uma string usando a tabela e a array, e então chama a função query()

Exemplo de uso:

```
db->query('tabela',  
          // Insere uma linha  
          array('campo_tabela' => 'valor', 'outro_campo' => 'outro_valor'),  
          // Insere outra linha  
          array('campo_tabela' => 'valor', 'outro_campo' => 'outro_valor'),  
          // Insere outra linha  
          array('campo_tabela' => 'valor', 'outro_campo' => 'outro_valor'))
```

4.2.3 update:

Parâmetros:\$table (tabela onde está o dado),
\$where_field (coluna de comparação),
\$where_field_value(valor a ser comparado),
\$values(campos a serem alterados e seus valores);

Funcionamento: Cria uma string usando a tabela, campo_where, valor_where, e a array, e então chama a função query()

Exemplo de uso:

```
db->query('tabela', 'campo_where', 'valor_where',  
        // Atualiza a linha  
        array('campo_tabela' => 'valor', 'outro_campo' => 'outro_valor'))
```

4.2.4 delete:

Parâmetros:\$table (tabela onde esta o dado),
\$where_field (campo de comparação),
\$where_field_value (valor a ser comparado)

Funcionamento: Cria uma string usando a tabela, campo_where, valor_where e então chama a função query()

Exemplo de uso:

```
db->query('tabela', 'campo_where', 'valor_where')
```

4.3 class-UserLogin.php