

# Manual de Instalação e Configuração do Miolo

# Índice

|       |                           |   |
|-------|---------------------------|---|
| 1     | Introdução.....           | 3 |
| 1.1   | Arquivos importantes..... | 3 |
| 2     | Instalação.....           | 4 |
| 2.1   | Pacote de instalação..... | 4 |
| 2.2   | Repositório.....          | 4 |
| 2.2.1 | Pré-requisitos.....       | 4 |
| 2.2.2 | Miolo.....                | 4 |
| 3     | Configuração.....         | 6 |
| 3.1   | configure.....            | 6 |
| 3.2   | setconfig.....            | 6 |
| 3.3   | getconfig.....            | 6 |
| 3.4   | removeconfig.....         | 6 |
| 3.5   | createmodule.....         | 7 |
| 3.6   | createhandler.....        | 7 |
| 3.7   | createform.....           | 7 |
| 3.8   | createsearchform.....     | 7 |
| 3.9   | creategrid.....           | 8 |
| 3.10  | createbusiness.....       | 8 |
| 3.11  | createmvc.....            | 8 |
| 3.12  | createtheme.....          | 8 |
| 3.13  | start.....                | 8 |
| 3.14  | translate.....            | 9 |
| 3.15  | help.....                 | 9 |

# 1 Introdução

O Miolo é um framework para desenvolvimento de aplicações Web. Desenvolvido na linguagem PHP e utilizando conceitos de programação orientada a objetos, o Miolo visa facilitar a programação de sistemas Web através de classes e da abstração de camadas comuns nesse tipo de sistema, como segurança, geração de HTML e conexão com a base de dados.

Na seção a seguir são explicados alguns arquivos importantes para o entendimento do Miolo.

## 1.1 Arquivos importantes

**Handler:** Arquivo responsável pela chamada aos formulários, grids e outros componentes para exibí-los na página.

**MForm:** Classe que representa um formulário.

**MGrid:** Classe utilizada para exibir os dados de uma tabela.

**MBusiness:** Classe utilizada para representar uma tabela da base de dados.

Para mais informações sobre o Miolo, consulte os documentos presentes no repositório oficial:

<https://svn.solis.coop.br/miolo/branches/2.6/docs/>

No capítulo 2 é explicado como instalar o Miolo e o capítulo 3 explica o utilitário de configuração do Miolo. Os passos descritos nesse manual devem ser executados em um terminal (linha de comando) do Linux.

## 2 Instalação

A instalação do Miolo pode ser feita de duas formas: através do pacote de instalação do Ubuntu ou através do repositório Subversion.

### 2.1 Pacote de instalação

Caso a distribuição utilizada seja Ubuntu, faça o download do pacote através do seguinte link:

[https://svn.solis.coop.br/miolo/branches/2.6/package/ubuntu/miolo\\_2.6\\_i386.deb](https://svn.solis.coop.br/miolo/branches/2.6/package/ubuntu/miolo_2.6_i386.deb)

Em seguida, basta clicar duas vezes no arquivo e seguir os passos, ou então executar o seguinte comando.

```
$ gdeb miolo_2.6_i386.deb
```

Agora o Miolo está instalado e pronto para ser acessado através do seu navegador pela URL a seguir:

<http://miolo.localhost/>

### 2.2 Repositório

Caso utilize outra distribuição, o Miolo pode ser obtido através do repositório Subversion. Dessa forma devem ser realizadas algumas configurações de forma manual.

#### 2.2.1 Pré-requisitos

Para a instalação e utilização do Miolo são necessárias algumas ferramentas, cujas instalações não são foco desse manual. São elas:

- Apache
- PHP
- Subversion

#### 2.2.2 Miolo

Escolha um diretório para armazenar seus projetos. Certifique-se que você se encontra no diretório desejado e faça o download da versão 2.6 do Miolo, utilizando o Subversion.

```
$ cd /home/usuario/projetos
```

```
$ svn co https://svn.solis.coop.br/miolo/branches/2.6 miolo-2.6
```

No servidor Apache deve ser adicionado um *Virtual Host* para o diretório do Miolo, conforme exemplo abaixo.

```
<VirtualHost *:80>
    ServerAdmin admin@miolo.org.br
```

```
ServerName miolo.org.br
DocumentRoot /home/usuario/projetos/miolo-2.6/html
<Directory /home/usuario/projetos/miolo-2.6/html>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
</VirtualHost>
```

Utilize o utilitário do Miolo para realizar sua configuração inicial, informando a URL escolhida, no caso <http://miolo.org.br>. Todas as funcionalidades desse utilitário são explicadas no capítulo 3.

```
$ cd miolo25
```

```
$ bin/miolo.php configure http://miolo.org.br
```

Agora o Miolo está configurado e pronto para ser acessado através do seu navegador pela URL informada.

## 3 Configuração

Para a configuração do Miolo, existe um utilitário em linha de comando, o qual pode ser encontrado no diretório *bin* no diretório raiz do Miolo. Seus comandos e sua interface de ajuda estão em inglês. Neste manual estão descritas suas funcionalidades em português.

Sintaxe de utilização a partir da raiz:

```
$ bin/miolo.php <comando> [<parâmetros>]
```

### 3.1 configure

Cria um arquivo inicial de configuração. Modo de uso:

```
$ bin/miolo.php configure [url]
```

### 3.2 setconfig

Define o valor do parâmetro de configuração (miolo.conf). Espera pelo menos dois parâmetros, o nome do parâmetro de configuração e o novo valor a atribuir a esse parâmetro.

O parâmetro de configuração utiliza o caractere '.' como separador.

O terceiro parâmetro não é obrigatório. Com ele pode ser definido que a configuração a ser alterada é específica do módulo (module.conf).

Modo de uso:

```
$ bin/miolo.php setconfig <configuração> <valor> [módulo]
```

### 3.3 getconfig

Obtém o valor da configuração do Miolo (miolo.conf). Caso um módulo seja informado, obtém o valor da configuração do módulo (module.conf).

Modo de uso:

```
$ bin/miolo.php getconfig <configuração> [módulo]
```

### 3.4 removeconfig

Remove a configuração informada do arquivo de configuração (miolo.conf). Remove a configuração do módulo caso ele seja informado (module.conf).

Modo de uso:

```
$ bin/miolo.php removeconfig <configuração> [módulo]
```

### 3.5 createmodule

Cria um módulo com o nome informado, gerando uma estrutura inicial básica (MVC). Como segundo parâmetro pode ser informado o módulo no qual o novo deve se basear, ou seja, será feita uma cópia do *base\_module*, alterando apenas seu nome.

Modo de uso:

```
$ bin/miolo.php createmodule <módulo> [módulo_base]
```

### 3.6 createhandler

Cria um arquivo handler no módulo informado. O parâmetro *<handler>* é o nome do handler, *<título>* é o texto a ser exibido na barra de navegação e *<formulário>* é o nome do formulário a ser exibido pelo handler.

Modo de uso:

```
$ bin/miolo.php createhandler <módulo> <handler> <título> <formulário>
```

### 3.7 createform

Cria um formulário no módulo informado. O parâmetro *<formulário>* define o nome da classe e *<título>* define o título do formulário.

Modo de uso:

```
$ bin/miolo.php createform <módulo> <formulário> <título>
```

### 3.8 createsearchform

Cria um formulário de busca no módulo informado, ou seja, cria uma tela para exibição dos registros de uma tabela, com possibilidade de filtrá-los.

O parâmetro *<formulário>* define o nome da classe, *<título>* define o título do formulário, *<grid>* é o nome da classe MGrid a ser utilizada para exibição dos dados, *<tabela>* é o nome da tabela na base de dados cujos dados devem ser exibidos. Opcionalmente também podem ser informados os filtros, os quais devem ser os nomes das colunas da tabela separados por espaço.

Modo de uso:

```
$ bin/miolo.php createsearchform <módulo> <formulário> <título> <grid> <tabela> [filtro1 ...  
filtroN]
```

### 3.9 creategrid

Cria uma *grid* no módulo informado. O parâmetro *<grid>* define o nome da classe e *<título>* define o título da *grid*. Os parâmetros opcionais a seguir informam os nomes das colunas da *grid*.

Modo de uso:

```
$ bin/miolo.php creategrid <módulo> <grid> <título> [coluna1 ... colunaN]
```

### 3.10 createbusiness

Cria um objeto no módulo informado, representando uma dada tabela da base dados, a qual contém apenas um método de busca.

Modo de uso:

```
$ bin/miolo.php createbusiness <módulo> <table>
```

### 3.11 createmvc

Cria uma estrutura MVC funcional no módulo informado. A estrutura é composta por um *handler*, um *business*, um formulário de busca; um formulário de gerenciamento para inserir, remover, editar registros, uma *grid* e uma base dados *sqlite* já populada.

Modo de uso:

```
$ bin/miolo.php createmvc <módulo>
```

### 3.12 createtheme

Cria um novo tema baseado em um tema existente. Modo de uso:

```
$ bin/miolo.php createtheme <nome> <tema_base>
```

### 3.13 start

Cria uma nova instalação do Miolo utilizando a mesma cópia base do Miolo. Recebe por parâmetro o diretório onde deve ser criada essa nova instalação. Modo de uso:

```
$ bin/miolo.php start <path>
```



### 3.14 translate

Utilitário de tradução. Espera pelo menos um parâmetro, podendo ser *extract* ou *generate*. O primeiro realiza a extração dos textos não traduzidos e os põe nos arquivos com extensão *.po* no diretório *misc/i18n/* para ser feita a tradução.

```
$ bin/mioLo.php translate extract
```

Em seguida, deve ser utilizado o comando *generate* para efetuar a compilação dos arquivos de tradução. Para visualizar os novos textos na sua aplicação, deve-se reiniciar o Apache após esse comando.

```
$ bin/mioLo.php translate generate
```

### 3.15 help

Exibe os comandos suportados pela ferramenta. Também pode ser obtida a ajuda específica de cada comando, passando-o por parâmetro.

Modo de uso:

```
$ bin/mioLo.php help [comando]
```