



# MANUAL

de Construção  
de Templates



## Sumário

1. Criando um Template.....	3
2. Estrutura.....	5
3. Resgatar dados no Template.....	6
4. Notícias.....	6
5. Páginas.....	8
6. Categorias.....	9
7. Menus.....	10
8. Banners.....	12
9. Páginas Estáticas.....	12
10. Geração de Dados no Banco.....	13
11. Empacotando o Template.....	13

## 1. Criando um Template

Para acessar o painel de controle do Suindara, o usuário deverá digitar o endereço < <http://seudominio.com/login> >. A primeira tela a ser mostrada é a tela de login:

A imagem mostra a interface de login do sistema Suindara. No topo, há o logo do Suindara, que consiste em uma coruja vermelha e o nome "Suindara" em uma fonte preta. Abaixo do logo, o título "Login" é exibido em uma barra branca. O formulário de login contém dois campos de entrada: "E-mail" com o valor "exemplo@exemplo.com" e "Senha" com pontos para ocultar o texto. Abaixo dos campos, há uma mensagem: "Campos com esta marca são obrigatórios." e um link "Esqueceu a senha?". Um botão vermelho com o texto "Acessar" está na parte inferior direita. Uma seta azul aponta para o botão "Acessar".

Imagem 01: Tela inicial do programa.

Seguindo o próprio exemplo da página, deve ser digitados o e-mail e a senha de acesso. Clicando na opção "Acessar", indicada na Imagem 01, você poderá ter acesso ao sistema. Na tela seguinte à essa, mostrada na Imagem 02, selecione a opção "Administração" e clique em "Acessar".

A imagem mostra a interface de gerenciamento da área administrativa do sistema Suindara. No topo, há o logo do Suindara, que consiste em uma coruja vermelha e o nome "Suindara" em uma fonte preta. Abaixo do logo, o título "Gerenciar sites" é exibido em uma barra branca. O formulário de gerenciamento contém um campo de seleção com o texto "Selecione um site para gerenciar" e a opção "Administração" selecionada. Um botão vermelho com o texto "Acessar" está na parte inferior direita. Uma seta azul aponta para o botão "Acessar".

Imagem 02: Gerenciamento da área Administrativa

## Manual de Construção de Templates do CMS Suindara

Na próxima página, clique na opção “Templates”; indicada na Imagem 03, no menu lateral esquerdo.

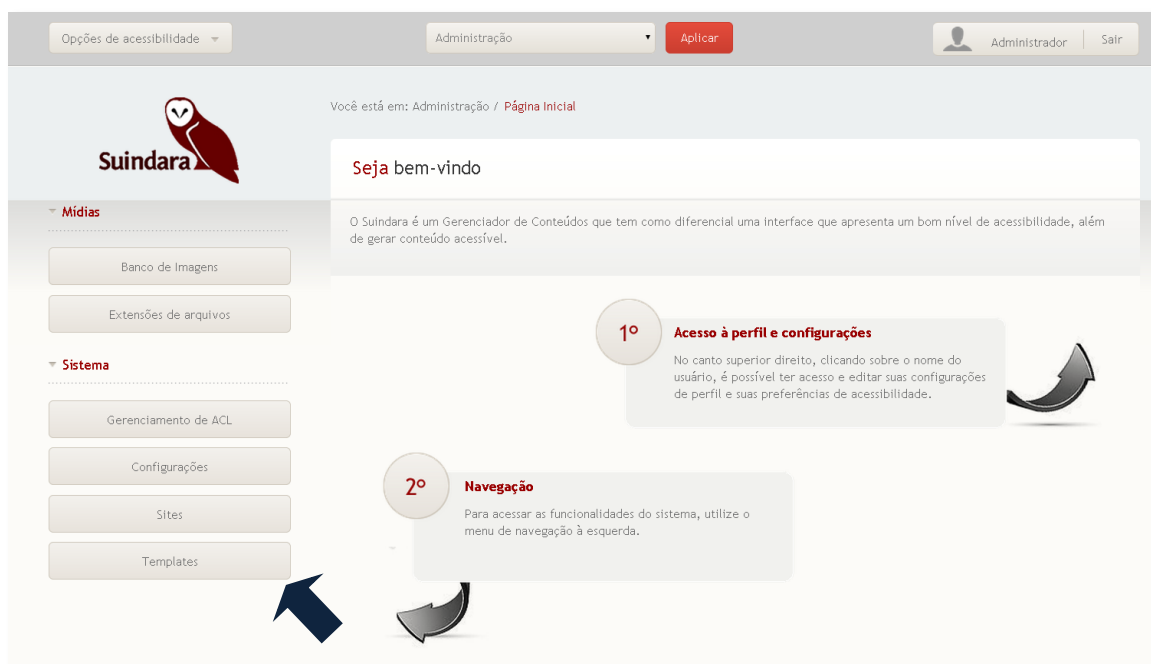


Imagem 03: Painel de Controle do Suindara

Depois, clique na opção “Gerar Novo Template”. Uma mensagem, com o mesmo formato da Imagem 04, deverá aparecer no topo da página, identificando sua localização no disco.

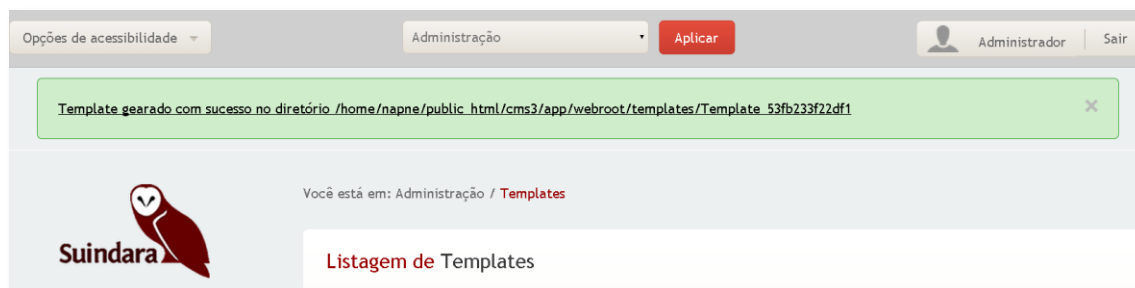


Imagem 04: Caso a operação seja executada com sucesso, esta mensagem deverá aparecer na tela.

Este será adicionado à listagem de Templates, com o nome de “Template\_X”, sendo ‘X’ o mesmo nome identificador do template da mensagem. Padronizadamente, todos os templates são criados no diretório <Local de instalação do CMS3>/app/webroot/templates/ Template\_x >

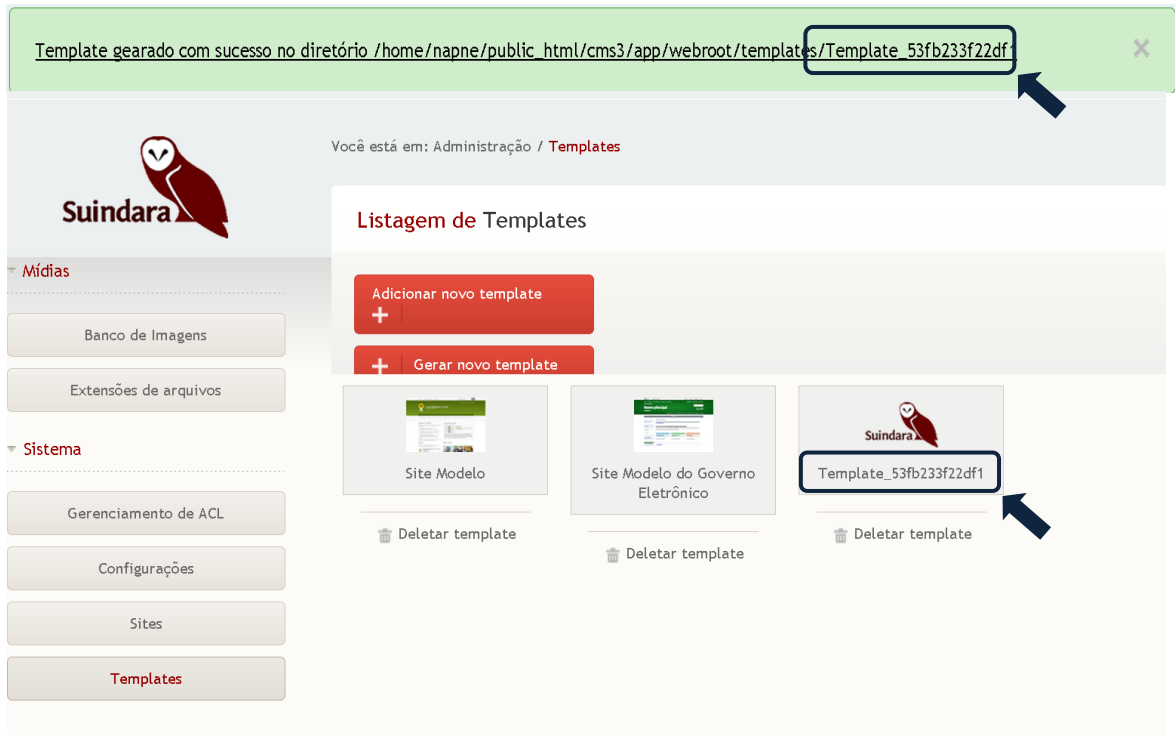


Imagem 05: Na imagem, está indicado o nome ‘x’ do novo Template

## 2. Estrutura

O Template é composto pela seguinte estrutura de pastas:

**\_init:** Contém o script init.sql utilizado para gerar tabelas e inserções no banco de dados na instalação do Template;

**\_end:** Contém o script end.sql utilizado para apagar as tabelas do banco de dados na remoção do Template;

**css:** Contém os arquivos de css;

**js:** Contém os arquivos de script;

**images:** Contém as imagens utilizadas no Template;

**noticias:** Contém os arquivos de visualização e listagem das notícias;

**paginas:** Contêm o arquivo de visualização das páginas;

**Elements:** Arquivos parciais do Template. Normalmente arquivos de visualização que podem ser compartilhados, como o cabeçalho e rodapé das páginas;

**Layouts:** Arquivos para alterar o layout de todas as páginas do Template;

As pastas 'Elements e Layouts' são recursos do CakePHP. Para maiores informações verificar a documentação no link (<http://book.cakephp.org>).

### 3. Resgatar dados no Template

Os exemplos apresentados abaixo foram aplicados no template "Site Modelo" que acompanha o CMS Suindara e seu código pode ser encontrado em:

<Local de instalação do CMS3>/app/webroot/templates/default>

É importante destacar que todos os arquivos de visualização utilizados possuem a extensão ".ctp".

Para resgatar os dados são utilizados subclasses dos ViewHelpers (<http://book.cakephp.org/2.0/en/views/helpers.html>). Cada ViewHelper é responsável por tratar e retornar os dados de uma área do sistema (notícias, páginas, banners ...).

Nas próximas seções serão apresentados alguns trechos de código para resgatar os dados de cada área do sistema. Maiores informações podem ser encontradas nos arquivos fonte do template "Site Modelo" e na documentação que acompanha o CMS Suindara.

### 4. Notícias

O trecho de código mostrado na Imagem 06 carrega uma notícia, além de exibir algumas informações da mesma:

```
1  <?php
2      $noticia_id = $this->CmsTemplate->getParams(0);
3      $noticia = $this->CmsNoticias->getNoticia($noticia_id);
4  ?>
5  <div>
6      <?php echo $noticia->htmlTitulo() ?>
7      <p>Publicado em
8          <?php echo $noticia->htmlDataPublicacao() ?>
9      </p>
10     <p> <?php echo $noticia->htmlTexto() ?> </p>
11 </div>
```

Imagem 06: Trecho de código para  
"Notícias"

Na linha 02, o valor retornado pelo método `getParams` é o primeiro parâmetro recebido pela url que, neste caso, é o identificador da notícia. Na linha 03 é carregada a notícia. Além do `getNoticias`, existem outros métodos que podem simplificar a busca por notícias. É importante observar que, devido a padrões, somente as notícias que possuem o status de "publicada" serão retornadas pelo método.

Nas linhas 06, 08 e 10 é gerado o HTML para a apresentação das informações. Também é possível acessar o campo sem gerar o HTML correspondente; para isso basta chamar pelo nome do campo registrado no banco de dados. Por exemplo, para acessar o título da notícia sem gerar o HTML, pode-se utilizar:

```
1  <p> <?php echo $noticia->titulo ?> </p>
```

Imagem 07: Método de acesso ao título  
de uma notícia, não utilizando HTML

Para acessar imagens, vídeos e arquivos associados à notícia existem os métodos `getImagens`, exemplificado no código da Imagem 08, `getVideos` e `getArquivos`, os quais funcionam de forma similar.

```
1    <?php
2        $imgs = $noticia->getImagens();
3        if ($imgs) {
4            foreach ($imgs as $img) {
5                echo $img->htmlImagem(TIMG_ORIGINAL,
6            }
7        }
8    ?>
```

Imagem 08: Exemplo de código, o qual permite o acesso às imagens da notícia carregada anteriormente.

Na linha 02, todas as imagens relacionadas à notícia são carregadas, e na linha 05 é gerado o HTML necessário para exibir cada imagem. Na função 'htmlImagem' é possível informar se a imagem utilizada será a original ou uma versão menor. Para mais informações sobre os métodos utilizados e outros disponíveis, verifique a documentação.

## 5. Páginas

O acesso aos dados das páginas é semelhante ao de notícias, como podemos verificar no exemplo da Imagem 09, a qual indica o local onde os dados de uma página são carregados e exibidos.

```
1    <?php
2        $pagina_id = $this->CmsTemplate->getParams(0);
3        $pagina = $this->CmsPaginas->getPagina($pagina_id);
4
5        echo $pagina->htmlTitulo()
6        echo $pagina->htmlTexto()
7    ?>
```

Imagem 09: Local onde dados de páginas são carregados e exibidos



As linhas 02 e 03 exemplificam o carregamento da página com o identificador informado pelo primeiro parâmetro passado pela url. Já os títulos e texto da página são gerados e exibidos nas linhas 05 e 06. As imagens são inseridas utilizando-se o método `getImagens`.

## 6. Categorias

O trecho de código, exibido na Imagem 10, especifica o processo de carregamento de uma categoria:

```
1 $categoria = $this->CmsCategorias->getCategoria("Evento");
```

Imagem 10: Carregamento de Categoria

Neste caso a categoria carregada possui o identificador "Evento", sendo este uma string registrada assim que a mesma é criada no CMS. Outra possibilidade é utilizar-se o ID para carregá-la.

A categoria também pode ser utilizada para filtrar notícias que estejam vinculadas a ela. Por exemplo:

```
1 $noticias = $categoria->getNoticias();  
2 foreach ($noticias as $noticia) {  
3     echo '<p>' . $noticia->titulo() . '</p>';  
4 }
```

Imagem 11: Na imagem a cima, a categoria de exemplo está sendo utilizada para filtrar notícias

Na linha 01, todas as notícias vinculadas à categoria "Eventos" são carregadas; nas linhas 02 e 03, o título de cada notícia é exibido.

## 7. Menus

Para carregar um menu, é necessário informar o seu identificador, de forma semelhante ao das categorias, mostrado anteriormente.

```
1 $menu = $this->CmsMenu->getMenu('AcessibilidadeVirtual');
```

Imagem 12: Exemplo de carregamento de um menu

O identificador "AcessibilidadeVirtual" é definido quando o usuário cria o menu pelo CMS.

O menu é composto pela 'raiz' e pelos 'sub-menus', podendo comportar vários níveis desse último. Para exibi-lo, pode-se iterar sobre seus sub-menus, apresentando as informações necessárias:

```
1  foreach ($menu->getItens() as $submenu){  
2      echo "<a href={\$submenu->getLink()}>  
          {\$submenu->nome}  
          </a>";  
3  }
```

Imagem 13: Exemplo de um menu, subdivido em 'raiz' e 'sub-menus'

Para a apresentação do menu, em certos casos, pode ser necessária a criação de formas complexas, ou até mesmo a reutilização destas em outros Templates. Em tais situações, a montagem do menu pode ser encapsulada em um objeto, o qual implemente a interface `CmsMontadorMenu`. Essa interface utiliza o método `htmlMenu(CmsMenuelement $baseMenu)`, a qual deve retornar uma string com o HTML gerado do menu. A imagem 14, apresenta um exemplo de um montador de menus, o qual utiliza apenas um nível.

```
1 App::uses('CmsMontadorMenu', 'View/Helper/Util');
2 App::uses('CmsMenuElement', 'View/Helper/Din');
3
4 class MontadorMenuExemplo implements CmsMontadorMenu
5 {
6     public function htmlMenu(CmsMenuElement $baseMenu)
7     {
8         $result = "";
9         foreach ($baseMenu->getItens() as $submenu) {
10             $result .= "<a href={$submenu->getLink()}>
11         }
12
13         return $result;
14     }
15 }
```

Imagem 14: Exemplo de um montador de menus de um nível

Para apresentar o menu utilizando o MontadorMenuExemplo basta modificá-lo para o método htmlMenu:

```
1 $montador = new MontadorMenuExemplo();
2 echo $menu->htmlMenu($montador);
```

Imagem 15: Para apresentação do menu, modifique-o para o método htmlMenu

O exemplo de um montador de menus com mais níveis pode ser encontrado em:

<Local de instalação do CMS3>/app/View/Helper/Util/CmsMontadorMenuPadrao.php

## 8. Banners

Para carregar um banner, é necessário informar o seu identificador, como especificado na Imagem 16. Para apresentá-lo, siga o exemplo da Imagem 17. Cada banner registrado no CMS pertence a um grupo, sendo possível filtrá-los, como observamos na Imagem 18.

```
1 $banner = $this->CmsBanners->getBanner('Banner01');
```

Imagem 16: Para carregar um banner, informe o seu identificador

```
1 $banner->htmlBanner();
```

Imagem 17: Modelo para apresentar um Banner

```
1 $grupo = $this->CmsBanners->getGrupo('Principal');  
2 foreach ($grupo->getBanners() as $banner) {  
3     echo '<li>' . $banner->htmlBanner() . '</li>';  
4 }
```

Imagem 18: No exemplo acima, todos os banners registrados no grupo "Principal" serão mostrados como itens de uma lista.

## 9. Páginas estáticas

Em determinados casos, pode ser necessária a criação de uma página diferenciada, que não apresente nenhuma similaridade com os padrões já aplicados no projeto. Em tais situações, é possível criar as chamadas 'Páginas Estáticas' que possuem um layout próprio.

Para gerar uma página estática, é necessário criar um arquivo com a extensão '.ctp' no diretório do Template. Sua chamada pela url será feita através do caminho lógico do arquivo, por exemplo se a página 'contato.ctp' for criada no diretório raiz do Template, seu endereço será:

`/contato`

Se estiver dentro de uma pasta chamada 'geral', então seu endereço será:

`/geral/contato`

Exemplos de páginas estáticas podem ser encontrados no Template do Site Modelo, o qual acompanha o Suindara CMS.

### 10. Gerar dados no banco

Conforme especificado na seção Estrutura, o arquivo `_init/init.sql` pode ser executado na instalação do Template, gerando as tabelas necessárias para o mesmo. Já o arquivo `_end/end.sql` será executado na remoção do Template, podendo remover qualquer dado utilizado por ele.

### 11. Empacotando o Template

Para preparar o pacote a ser distribuído é importante descrevê-lo no arquivo `info.json`, o qual deve acompanhar todos os Templates. A estrutura do arquivo `info.json` é apresentada na Imagem 17:

```
1  {  
2    "nome": "Nome do Template",  
3    "print": "Imagem.png",  
4    "autor": "Autor",  
5    "descricao": "Exemplo"  
6  }
```

Imagem 17: Estrutura do arquivo

O campo "print" será uma imagem pequena, utilizada para a preview do Template na listagem de Templates.

O arquivo "info.json" deve ser colocado no diretório raiz do Template. Para finalizar o pacote é necessário compactar os arquivos para um .zip. É importante salientar que os arquivos do Template devem estar, obrigatoriamente, na raiz do arquivo .zip.

Para instalar o pacote do Template, acesse a área de Templates como Administrador e clique na opção "Adicionar novo Template"; selecione o arquivo .zip e clique em "Enviar arquivo". Assim que a instalação do Template for concluída, ele será apresentado na listagem de Templates com o mesmo nome informado no arquivo "info.json".

Depois de instalar o Template, o mesmo pode ser utilizado em um site. Para isso acesse a seção "Sites" como 'Administrador', selecione "Editar" no site que deseja aplicar o Template, no campo de seleção "Template" marque o nome do Template desejado e clique em "Salvar".