

SWOP

MANUAL PARA DESENVOLVIMENTO

SUMÁRIO

ACESSIBILIDADE	3
CAPA - MÓDULOS	3
DIRETRIZES DA ESTRUTURA E ESTILOS	4
TESTES DA CAPA ANTES DE SOLTAR O SITE	8
SOBRE A PROGRAMAÇÃO DO SITE	8
ARQUIVOS QUE NÃO SERÃO ATUALIZADOS AUTOMATICAMENTE POR SEREM DIFERENTES EM CADA PREFEITURA	8
ARQUIVOS INCLUÍDOS PADRÃO	9
ARQUIVOS COMUNS	9
MIGRAÇÃO DE TABELAS DO BANCO DE DADOS	9

ACESSIBILIDADE

De acordo com as normas do governo federal o layout deverá possuir funções e os atalhos:

- Conteúdo [1] Menu [2] Busca [3] Rodapé [4]
- Auto Contraste
- Mapa do site
- Aumento/diminuição de fontes
- Link para a página de acessibilidade
- VLibras

Para esconder o botão padrão do VLibras e necessário adicionar os estilos abaixo no **/css/estrutura.css**:

```
/* VLIBRAS */  
#vlibras_include div[vw-access-button] { display: none; }  
#vlibras_include div[vw].enabled { transform: translateY(5vh); }
```

CAPA - MÓDULOS

1) Todos os arquivos da capa (módulos) padrão deverão ter o nome com a inicial **p**.

- p_noticias.php
- p_agenda.php
- /css/p_noticias.css
- /css/p_agenda.css
- /js/p_noticias.js
- /js/p_agenda.js

2) Todos os arquivos da capa (módulos) exclusivos deverão ter o nome com a inicial **e**.

- e_noticias.php
- e_agenda.php
- /css/e_noticias.css
- /css/e_agenda.css
- /js/e_noticias.js
- /js/e_agenda.js
- *Seguir os nomes dos módulos idênticos aos módulos do sistema. Com exceções aos módulos conjunto, ex: agenda com arquivos.

3) Os estilos (css) dos banners de serviços na maioria das vezes são iguais, sendo assim o arquivo **p_banners_servicos.css** ou **e_banners_servicos.css** será utilizado para os banners de serviços superior, rodapé e central. Caso o banner rodapé precise de estilos diferentes, basta criar o arquivo **e_banners_servicos_rodape.css**.

4) Todo módulo, independente de ser exclusivo ou padrão, deverá ter uma div antes do conteúdo.

```
<div class="swop_capa">
    ...
</div>
```

DIRETRIZES DA ESTRUTURA E ESTILOS

- 1) Em todo site obrigatoriamente vai existir um banner no topo que o usuário pode alterar com um link para "/portal". O cliente poderá também configurar uma imagem para o fundo do site.

As contenções abaixo devem sempre seguir esse padrão e são obrigatórias em todos os layouts. E para que isso funcione dinamicamente temos que já deixar isso definido no **cabecalho.php**:

```
<style>
    #e_banner_topo_dinamico {width: 1200px; height: 187px; float: left; margin-top: 4px;
background: url(/fotos/<?php echo $banner_topo;? >.jpg) no-repeat;}
    #e_banner_topo_dinamico_clique {width: 180px; height: 173px; float: left;}
</style>
```

E no arquivo **topo.php/e_topo.php**:

```
<div id="e_banner_topo_dinamico">
    <a href="/portal/"><div id="e_banner_topo_dinamico_clique"></div></a>
</div>
```

- 2) No layout já deixar todos os scripts funcionando.
- 3) Os estilos exclusivos de uma prefeitura/câmara (capa, rodapé e outros) obrigatoriamente começarão com a letra **e**. Exemplo:

```
.e_area_noticias {...}
```

- 4) As fontes personalizadas da capa deverão ser colocadas no arquivo **/css/style.css**.
- 5) A área de **newsletter** (não é considerado um módulo da capa) deverá ficar dentro do arquivo **rodape.php** e **e_rodape.php**. Pode haver exceções caso o cliente desejar.
- 6) As respostas da enquete deverão obrigatoriamente ter a classe **e_cont_resposta_enquete**. Pois o retorno busca esta classe para adicionar o conteúdo. E o id **e_cont_enquete** no conteúdo principal, pois o retorno da votação é adicionado neste id.

- 7) Divs que sempre devem estar no **/css/estrutura.css** pois influenciam diretamente nas internas portanto, não podendo mexer:

```
.e_centralizar {width: 100%; width: calc(100% - 40px); max-width: 1200px; margin: 0 auto;}
#conteudo_esquerda {width: 100% !important;}
#cont_conteudo_site_interna {width: 100% !important;}
```

- 8) Os arquivos **/css/style_interna_mobile.css** e **/css/style_interna.css** são únicos para cada cliente, é nele que se personaliza as cores e estilos para deixar as internas e página mobile com estilos diferentes para cada cliente.
- 9) O arquivo **/css/estrutura_interna.css** não é mais utilizado, todo o conteúdo dele foi adicionado para o **/css/estrutura_interna_mobile.css**. Ainda não foi removido dos arquivos do sistema por questões de estrutura, logo vamos remover.
- 10) Os arquivos abaixo personalizam as cores e estilos para deixar as internas e capa do mobile com estilos diferentes para cada cliente:

```
/css/style_interna_mobile.css
/css/style_capa_mobile.css (para versão mobile "/portal/mobile" - Não é mais utilizado, mas ainda existe por questões de estrutura e contrato)
/css/menu_mobile.css
/css/style_mobile.css
/css/estrutura_mobile.css
```

- 11) Imagens das internas em que as cores são neutras e que podem aparecer em qualquer cliente (prefeitura ou câmara) devem ser adicionadas na pasta **imgcomum** para que o sistema de auto atualização também atualize essas.

- 12) Sempre deixar na pasta **img** de cada cliente os arquivos e pasta:

```
logo_mobile.png
logo_branca.png
logo-rodape.png
img_tempo (pasta com as imagens da previsão do tempo)
```

- 13) Sempre criar as seguintes imagens para o cliente, que posteriormente serão inseridas no sistema:

```
Logo compartilhar (logo_compartilhar.jpg, não necessariamente precisa ter este nome)
Imagem sem imagem da Secretaria (secretaria_sem_img.jpg, não necessariamente precisa ter este nome)
favicon.ico
```

- 14) Caso o layout tenha imagens de arquivos personalizadas (de preferência não alterar, porque se houver alteração deverá ser levada em consideração a exibição dessas imagens no SG e nas páginas internas do portal) os nomes deverão ser:

```
arq_arquivo.(svg, jpg ou png)
arq _planilha.(svg, jpg ou png)
arq _img.(svg, jpg ou png)
arq _multimedia.(svg, jpg ou png)
arq _pdf.(svg, jpg ou png)
arq _grafico.(svg, jpg ou png)
arq _script.(svg, jpg ou png)
arq _txt.(svg, jpg ou png)
arq _zip.(svg, jpg ou png)
```

- 15) Todos os clientes devem ter em sua pasta os arquivos:

```
cabecalho.php
diretrizes.php
/img/img_tempo (pasta)
/css/font (pasta)
/css/estrutura.css
/css/estrutura_mobile.css
/css/menu_mobile.css
/css/style.css
/css/style_interna.css
/css/style_interna_mobile.css
/css/style_mobile.css
*E todos os arquivos de módulos, caso seja exclusivo. Ex: e_topo.php, e_rodape.php, e_agenda.php,
e_enquete.php. O mesmo se aplica para o css e js.
```

- 16) Quando o site vai para a versão mobile o topo e o rodapé são alterados. Para isso é necessário que o conteúdo dos arquivos **topo.php/e_topo.php** e o **rodape.php/e_rodape.php** tenham as seguintes divs com ids específicos:

```
<div id="e_cont_topo">
...
</div >

<div id="e_cont_rodape">
...
</div >
```

- 17) Se o site tiver topo fixo, deve-se levar em conta:

- a) A página de Transparência obrigatoriamente deve ter **Menu Lateral** (visto o topo já estar fixo);

b) Visto página interna de Legislação conter topo fixo, há a necessidade de estilizar a condição do topo no **css de estrutura** para mídias **acima de 1000px**. Deve-se compensar a posição top da barra de acordo com a altura do topo, como no exemplo:

```
/* Legislação */
@media (min-width: 1000px) {
    .leg_area_barra_ferramentas .leg_barra_ferramentas.fixed { top: 80px; }
}
```

18) Para estilizar a barra de **login do internauta** no topo, deve-se replicar as classes utilizadas e realizar o reset:

```
/* Área login topo internauta */
.itn_area_login_internauta {}
.itn_area_login_internauta .itn_conteudo_interno_login {}
.itn_area_login_internauta .itn_cont_login {}
.itn_area_login_internauta .itn_cont_login:before {}
.itn_area_login_internauta .itn_cont_login:first-child:before {}
.itn_area_login_internauta .itn_cont_login .itn_info_login {}
.itn_area_login_internauta .itn_cont_login .itn_info_login {}
.itn_area_login_internauta .itn_cont_login .itn_info_logado {}
.itn_area_login_internauta .itn_cont_login .itn_info_logado .int_cadastro_validado {}

/* Botão consentimentos */
.itn_area_login_internauta .itn_btn_consentimentos {}
.itn_area_login_internauta .itn_btn_consentimentos:hover {}
.itn_area_login_internauta .itn_btn_consentimentos:after {}

/* Botões login internauta */
.itn_area_login_internauta .itn_btn_login_internauta {}
.itn_area_login_internauta .itn_btn_login_internauta:hover {}
.itn_area_login_internauta .itn_btn_login_internauta .fa {}
.itn_area_login_internauta .itn_btn_login_internauta .itn_btn_sair {}
.itn_area_login_internauta .itn_btn_login_internauta .itn_btn_sair .fa {}
.itn_area_login_internauta .itn_btn_login_internauta .itn_btn_login .fa {}

/* Login internauta */
.itn_area_login_internauta .itn_login_internauta {}
.itn_area_login_internauta .itn_login_internauta .itn_area_botoes_internauta {}
.itn_area_login_internauta .itn_login_internauta .itn_area_botoes_internauta
.itn_cont_botoes_internauta {}
.itn_area_login_internauta:hover .itn_login_internauta .itn_area_botoes_internauta
.itn_cont_botoes_internauta {}
.itn_area_login_internauta .itn_login_internauta .itn_area_img_internauta {}
.itn_area_login_internauta .itn_login_internauta .itn_area_img_internauta img {}
.itn_area_login_internauta:hover .itn_login_internauta .itn_area_img_internauta {}
```

* **Observação:** Em alguns casos há a necessidade de utilizar o atributo **!important** para sobrescrever algum estilo pré definido.

TESTES DA CAPA ANTES DE SOLTAR O SITE

- Retirar as imagens das notícias, agenda, galeria de fotos para ver como fica se serão substituídas pela imagem sem imagem padrão ou configurada.
- Testar textos (título e descrição) grande e pequeno para verificar como o layout se comporta.
- Testar o conteúdo da capa, exemplo: se há 3 notícias, teste com 1, 2, 4, 5 e 6 notícias para ver se dá algum problema. Se tem 2 agendas, teste com 1, 3 e 5 etc...
- Editar o arquivo **diretrizes.php** para todas as dimensões de imagens.
- Verificar a versão mobile.
- Todo módulo deverá se adaptar à quantidade de itens exibidos. Ex: Se o módulo notícias estiver configurado para exibir até 5 registros e o cliente tiver apenas 2 cadastrados, o layout deverá se adaptar para exibir 2 registros (sem quebrar/deixar espaços em branco na página).

SOBRE A PROGRAMAÇÃO DO SITE

- Não poderão ser atendidos isoladamente pedidos para inclusão de um novo campo em cadastros, um novo tipo de cadastro, exceto se após análise a modificação no sistema é pertinente a todas as prefeitura e câmaras. Caso o cliente insista ele deverá pagar o valor justo e concordar em abandonar o benefício de receber novas atualizações.
- Dentro das internas o cliente não pode querer alterar a disposição das estruturas criadas. Exceto se for necessária tal alteração para todos os clientes.
- Todos os sites têm o padrão de 1200px. A capa do site, topo e rodapé são exclusivos do cliente, podem ser alterados com exclusividade para cada um, sem perda de atualização desde que a medida de 1200px do site seja mantida. Exceto os clientes que possuem layout padrão, esses não podem sofrer alterações isoladas, qualquer modificação será aplica para todos os clientes com layout padrão.

ARQUIVOS QUE NÃO SERÃO ATUALIZADOS AUTOMATICAMENTE POR SEREM DIFERENTES EM CADA PREFEITURA

Uma interna em um cliente nunca poderá ser corrigida isoladamente, pois o sistema irá sobrescrever a correção. Exceto para os arquivos abaixo:

```
/css/estrutura.css
/css/style.css
/css/style_interna_mobile.css
/css/style_capa_mobile.css (para versão mobile “/portal/mobile” - Não é mais utilizado, mas ainda
existe por questões de estrutura e contrato)

/css/meu_mobile.css
/css/style_mobile.css
diretrizes.php
url_exclusivas.php
cabecalho.php
img (pasta)
imginternas (pasta)
```

/sg/sistema.class.php

/sg/dbconfig.php

*Todos os arquivos com iniciais **e_** seja ele **php**, **css** ou **js** (estes são exclusivos de cada cliente).

Os demais arquivos serão substituídos sempre que houver atualizações.

ARQUIVOS INCLUÍDOS PADRÃO

O arquivo **topo_todos.php** foi criado com o intuito de conter códigos em comum para todos os clientes para facilitar a manutenção.

O **topo_todos.php** é incluído no **topo.php** de cada cliente.

Quando um cliente possui layout exclusivo o **e_topo.php** é incluído no **topo.php**, e o mesmo acontece com o rodapé.

ARQUIVOS COMUNS

topo_todos.php

- **Função:** Contém códigos em comum para todos os clientes que necessitam estar no topo (php e js).
- **Inclusão:** É chamado no topo.php.

internautas_topo_padrao.php

- **Função:** Contém códigos do topo do internauta (login/cadastro), caso futuramente algum cliente quiser um topo diferenciado, basta alterar este arquivo individualmente.
- **Inclusão:** É chamado no topo.php.

MIGRAÇÃO DE TABELAS DO BANCO DE DADOS

Quando houver mudanças drásticas no banco de dados, alterar os dados de uma tabela para outra, por exemplo, os arquivos que sofrerão alterações deverão conter uma mensagem de “Em manutenção” com um “exit;”, tanto na área pública quanto no SG. Pois não poderá ser feita nenhuma ação nesta página. Após migrar as páginas deverão voltar ao funcionamento normalmente.