**/config.php**

* Arquivo com as configurações gerais da aplicação como o BASE\_URL, ABSPATH, UP\_ABSPATH que aponta o diretório para uploads e as informações do banco de dados.

**/functions/global\_functions.php**

* Arquivo que contém as funções globais da aplicação, utilizadas pela gerência do MVC.

**/functions/atividades\_functions.php**

* Arquivo que deve conter os métodos comuns da aplicação. Caso crie uma função que será utilizado em outros controllers, adicionar neste arquivo como um método static. Para utilizar o método nos controllers, usar **CLASSE::MÉTODO()**

Ex. de utilização: *AtividadesFunctions::init\_menus(null, null);*

**/controllers**

* Adicionar o controller criado neste diretório;
* O nome do controller deve sempre ter o "\_controller" concatenado no fim do nome.
* Dentro de um controller, caso seja necessário trocar de controller, utilizar o método redirect() do arquivo global\_functions. Caso o método que será utilizado é o índex(), não precisa passar o nome do método, exemplo ***redirect(“home”)*** , caso contrário passar o método após o nome do controller e da barra (/), exemplo ***redirect("login/logout")***.
* Quando o controller exibir alguma tela, caso a mesma possua o menu lateral, chamar o método ***init\_menus(null, null)*** da classe ***AtividadesFunctions***. O método ***init\_menus*** possui **dois parâmetros**, o primeiro o usuário logado para que seja verificado o tipo do usuário e o segundo parâmetro é qual o item do menu deve estar destacado.
* Para importar algum arquivo php, utilizar: ***require\_once ABSPATH . '/functions/atividades\_functions.php';***

**OBS**.: o ***ABSPATH*** é configurado no arquivo config.php e indica o diretório raiz da aplicação para não ter que colocar caminhos fixos no código.+

* Sempre que o controller for carregar uma tela, deve importar o template através do comando ***require ABSPATH . '/views/includes/template.php'*** além da tela com o form da transação que se está trabalhando.

**Exemplo de controller**

*class HomeController {*

*public function index() {*

*require\_once ABSPATH . '/functions/atividades\_functions.php';*

*$menus = AtividadesFunctions::init\_menus(null, null);*

*$main\_page = ABSPATH . '/views/home\_page.php';*

*require ABSPATH . '/views/includes/template.php';*

*}*

*}*

**/views**

* Como os views utilizam o bootstrap, para separar os elementos do formulário em linhas, utilizar a tag ***<div class=”row”>*** e dentro desta div pode ser colocadas 12 colunas através da tag ***<div class=”col-lg-2”>***, **col** = coluna, **lg (≥1200px) =** tipo de tela, pode ser ainda **sm (≥768px)**, **md (≥992px)**, **xs (< 768px)** e o **2** é o número de colunas que pode ser de **1** a **12**, **sempre** somando 12.
* Exemplo de form nas telas. Na action do form utilizar BASE\_URL concatenado com o nome do controller sem o “\_controller”, concatenado com o método do controller. Abaixo um exemplo:

***<form class="form-horizontal" action="<?php echo BASE\_URL; ?>login/logar" method="post">***

**/models**

Todas as classes que acessam o banco de dados devem ser coloca neste diretório, e o controller deve acessa estas classes para realizar alterações e consultas ao banco.

**Referências:**

http://www.tutsup.com/2014/09/17/mvc-em-php-parte-1/#indexphp

http://getbootstrap.com/