## **DOCUMENTAÇÃO DO SHOWNET PARA DEV’S**

## 

### [**Shownet Tecnologia**](https://shownet.showtecnologia.com/)

É um sistema web voltado ao gerenciamento interno da empresa, nele pode-se gerenciar toda informação relacionada tanto à empresa como à nossos clientes.

Esta documentação é voltada aos desenvolvedores que irão criar solução para o sistema, sendo assim, terá uma linguagem mais técnica.

## **SUMÁRIO**

1. [Requisitos do sistema (Ferramentas/Tecnologias)](#_jat20uqrg61u)
2. [Metodologia](#_f9oihs359v3y)
3. [Arquitetura do Software](#_f0p2dj1c1iuf)
4. [Regras de Desenvolvimento](#_rqdy7x78ov1g)

## **1. Requisitos do Sistema (Ferramentas/Tecnologias)**

O sistema está construído em cima das seguintes ferramentas:

1. PHP 5.6
2. JavaScript (Jquery 3.4.1)
3. HTML5
4. CSS3
5. [Datatables.net](http://datatables.net/) (plugin JS utilizado para visualização de dados em tabelas).

## 

## **2. Metodologia**

Usamos a metodologia ágil, por motivos de trabalharmos sob a demanda das necessidades dos clientes, onde procuramos ter uma proximidade e a partir do feedback desenvolvemos e disponibilizamos soluções.

## 

## **3. Arquitetura do Software**

Adotamos o modelo MVC(Model-View-Controller) para organizar as camadas de comunicação dentro do sistema.

Ficando os arquivos distribuídos da seguinte forma:

1. views no diretório: application/views
2. controllers no diretório: application/controllers
3. models no diretório: application/models

## 

## **4. Regras de Desenvolvimento**

1. Ao criar arquivos de controller nomea-los no plural e para arquivos de modal nomea-los no singular.
2. Ao pedir para o controller montar a view, chamar o arquivo *head-new* para montar o cabeçalho e o arquivo *footer\_new* para montar o rodapé, pois esses dois arquivos contêm as importações das bibliotecas necessárias para o funcionamento correto do sistema.e  
   Ex.:

$this->load->view('fix/header-new', $dados);

$this->load->view('linhas/index');

$this->load->view('fix/footer\_new');

1. Carregar apenas as bibliotecas que serão utilizadas na página.

* no arquivo head-new escreva:

<?php if (isset($load) && in\_array('nome-biblioteca-aqui', $load)) : ?>

<script type="text/javascript"src="sua-biblioteca-aqui"></script>

<?php endif; ?>

* No arquivo controller usar:

$dados['load'] = array('nome-biblioteca-aqui');

$this->load->view('fix/header-new', $dados);

1. O sistema está habilitado a utilizar internacionalização, ou seja, desenvolvemos para mais de uma língua. Portanto, crie a string nos arquivos ***application/linguage/english/en\_lang.php*** e ***application/linguage/portuguese/pt\_lang.php*** antecipadamente e após isso chame essa string usando o método lang(‘nome\_string’) ou sprintf( lang(‘nome\_string\_print’), ‘João’ ).  
   ex.: no arquivo de linguagem fazer:

$lang['nome\_string'] = 'Meu texto';

$lang['nome\_string\_print'] = "olá senhor: %s";

1. Para importar um arquivo de biblioteca/imagem usar o método versionFile(‘pasta’, ‘nome\_arquivo’), esse método garante que um arquivo que foi alterado substitui-a o arquivo antigo que esta em cache.  
   Ex.:

versionFile('assets/js', 'xlsx.full.min.js')

1. Para internacionalização das legendas da datatable, existe um método que carrega os arquivos json com as traduções, bastando chamar esse método na instanciação da datatable.

language: lang.datatable

1. Para ajustar o tamanho e orientação da página dos arquivos exportados (pdf, excel, csv, imprimir) da datatable, utilizar o método:

customPageExport(id\_tabela, atributo)

Id\_tabela (o id da datatable).

atributo (podendo ser: orientation ou pageSize), se “orientation” o método retornará a orientação apropriada dependendo do número de colunas da tabela, se “pageSize” retornará o tamanho da página segundo o número de colunas da tabela.

1. Para ordernar corretamente uma datatable por uma coluna de datas no formato ‘dd/mm/yyyy’, utilizar o atributo:

columnDefs[ { type: 'date-uk', targets: 6 }]

Onde targets, é a coluna de datas.