

1. **URL**
   * Todos os nossos parâmetros são obtidos por [HTTP GET](https://www.todoespacoonline.com/w/) no seguinte formato:
   * http://www.sistemadeestagios.com/index.php?url=controlador/ação/parametro1/parametro2/etc…
   * Temos apenas um parâmetro na URL acima, o $\_GET[‘url’]. Isso porque é usado o arquivo .htaccess do Apache para reescrever a URL.
   * A mesma URL acima ficará assim:
   * http://www.sistemadeestagios.com/controlador/ação/parametro1/parametro2/etc...
2. **Index.php**
   * Apenas inclui o arquivo config.php
3. **config.php**
   * Faz a definição de algumas constante, e das informações do banco de dados. E inclui o arquivo loader.php.
4. **loader.php**
   * Inicia a seção, inclui o arquivos /functions/global-functions.php e inicia uma instancia da classe SistemaMVC que está declarada no arquivo /classes/class-SistemaMVC.php.
5. **/functions/global-functions.php**
   * Esse arquivo declara funções que serão muito usadas durante a aplicação.
   * **chk\_form\_fields($fields)**
     + Verificar se todos os campos que estão no parâmetro foram enviados pelo método POST.
   * **\_\_autoload($class\_name)**
     + Função crucial para o sistema, pois ele permite que você instancie qualquer classe sem precisar incluir o arquivo onde ela foi declara.
     + Ela identifica onde está o arquivo que contem a classe que está sendo instanciada e o inclui automaticamente.
6. **/classes/class-SistemaMVC.php**
   * Essa é a classe que faz a arquitetura MVC funcionar. Ela pega os parâmetros passados na URL, instancia e cria o arquivo do Controller correspondente e executa a Action.
7. **Controllers**
   * Os Controllers são classes estendidas de MainConroller que está no arquivo /classes/class-MainController.php.
   * O Controller possui métodos declarados nele, onde cada método representa uma Action.
   * As Actions são responsáveis incluir e instanciar o Model que ela ira usar e incluir a View correspondente.
8. **/classes/class-MainController.php**
   * Contem a classe MainController que é a classe base para todos os Controllers.
   * É uma classe estendida da classe UserLogin, localizada no arquivo /classes/class-UserLogin.php
   * Faz a instancia da classe SistemaDB e PasswordHash, localizadas nos arquivos /classes/class-SistemaDB.php e /classes/class-PasswordHash.php respectivamente.
   * Faz a verificação se o usuário está logado.
   * Possui as funções:
   * **load\_model($model\_name = false)**
     + Função responsável por fazer a inclusão e instanciação do Model.
   * **css($css\_name)**
     + Inclui o arquivo css da pasta /views/\_css/$css\_name.css
   * **js($js\_name)**
     + Inclui o arquivo js da pasta /views/\_js/$js\_name.js
9. **/classes/class-UserLogin.php**
   * **check\_userlogin()**
     + Verifica se a requisição atual é um POST tentando fazer o login, caso seja faz os procedimentos para fazer o login do usuário
   * **logout($redirect = false)**
     + Faz o logout do usuário e redireciona
   * **goto\_login()**
     + Redireciona para a página de Login
   * **goto\_page($page\_uri)**
     + Redireciona para a página desejada.
10. **/classes/class-SistemaDB.php**
    * Faz a conexão com o banco de dados
    * Disponibiliza algumas funções para facilitar os CRUD’s e as pesquisas
11. **/classes/class-PasswordHash.php**
    * Responsável por fazer Hash em senhas.
12. **Models**
    * Os Models são responsáveis por fazer consulta no Bando de Dados a passar essas informações para a View
    * São estendidos da classe MainModel.php. localizada no arquivo /classes/class-MainModel.php
13. **/classes/class-MainModel.php**
    * Cria as instancias das classes que representam as Tabelas no banco de dados, e estão localizadas em /table/
14. **Table**
    * Contem arquivos que declaração classes, onde cada classe representa uma tabela no Bando de Dados.
    * Cada classe possuem métodos referentes a operações que podem ser feitas na sua tabela.
15. **VIews**
    * Arquivos com conteúdo HTML e PHP
    * Esses arquivos são a parte visual da página mostrada no momento.