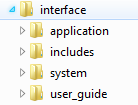
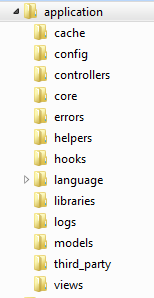
**Interface**

**Estrutura de diretórios:**

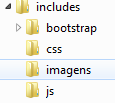


Na pasta interface contém todo o conteúdo da interface:

* Application: Na pasta application estão os arquivos: cache, config, controllers, core, errors, helpers,hooks, language, libraries, logs, models, third\_party e views.

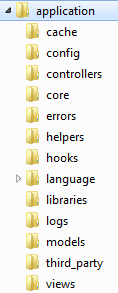


* Includes: Na pasta includes contém a pasta **bootstrap** que está todo o conteúdo do framework (bootstrap), a pasta **css** aonde tem todos os arquivos de estilo ccs da interface, a pasta **imagens** contendo as imagens usadas na interface e a pasta **js** onde tem todas as bibliotecas e arquivos Java Script e jQuery.



* System: Na pasta system toda a “ação” do CI acontece. Para organizar a estrutura presente, há subpastas para segmentar e organizar os arquivos e fluxo de trabalho.
* User Guide: Está a documentação do CodeIgniter 2.

**Application**

****

* cache. Pasta usada quando se está usando caching no CodeIgniter. É aqui que as páginas cacheadas ficam armazenadas e são servidas.
* config. Contém diversos (e importantes) arquivos relacionados a configurações de seu CI. São arquivos de configuração de database, variáveis sobre URL, quais libraries e helpers serão carregados automaticamente e muitas outras coisas.

Autoload: Carrega as bibliotecas utilizadas.

Config: Arquivo de configuração. base\_url: Url base do sistema

Constants: Definida as constantes da interface.

Database: Define as configurações do banco de dados: usuário, senha, database, driver do banco.

* controllers. Armazena os controllers que você cria para o seu software.
* core. Pasta criada no CI 2 para conter algumas bibliotecas consideradas mais “core” que outras. Não é usada em todos os projetos.

Nesta pasta contém um controlador chamado MY\_Controller, aonde temos o controle da linguagem utilizada pela interface e o controle de login, usuário logado == 1.

* errors. Vem com os templates de páginas de erros do CodeIgniter (erros genéricos, 404, conexão ao banco de dados, etc). É conveniente que tudo isso seja alterado para que os erros do aplicativo fiquem personalizados e consonantes com a finalidade deste.
* helpers. Para armazenar todos os helpers que você venha a criar/aprimorar.
* hooks. Para colocar os hooks que você cria. Em artigos futuros será mostrado que hooks são a maneira mais rápida e segura de você extender o core do CodeIgniter (geralmente feito por usuários avançados).
* language. Para aplicativos multi-idioma, esta pasta é bem usada por armazenar as mensagens nas diferentes escritas.

Nesta pasta contém as pastas que definem as linguagens que podem ser utilizadas pela interface, sempre usando o padrão linguagem em 2 letras traço (-) país em maiúsculo, por exemplo: pt-Br, en-US.

* libraries. Aqui ficam as libraries personalizadas, com funcionalidades para o programa a ser criado. Perceba que há diferença entre as pastas system/libraries e system/application/libraries.
* logs. Quando se habilita o salvamento de logs em arquivo em /application/config/config.php, é nesta pasta que eles serão salvos.

migrations. Quando se usa acesso externo ao CodeIgniter via linha de comando (CLI), a pasta “migrations” é usada para algumas operações.

* models. Armazena os models que você cria para seu aplicativo.
* third\_party. É possível distribuir aplicativos inteiros do CodeIgniter de forma fácil, e, para tal, é possível criar estrutura personalizadas nesta pasta, onde constam todos helpers, liberaries, models, etc, destas aplicações.
* views. Armazena os views que você cria para seu programa.