Universidad de Costa Rica
Sede de Occidente, Recinto Grecia
Multimedios
Profesor: Jonathan Rojas
Estudiantes:
Victor Julio Gonzalez Quesada B12948
Brayan Núñez Rojas B24808

Guía para iniciar el uso del framework CodeIgniter.

Índice

Introducción

Los framework son herramientas muy poderosas que nos permiten agilizar nuestro desarrollo de aplicaciones. Estos brindan una base sobre la cual desarrollar nuestros proyectos. CodeIgniter es un ejemplo de un framework que permite agilizar todo tipo de proyectos, ofreciéndonos una estructura de desarrollo adecuada y gran variedad de librerías que nos permiten agregar funcionalidad a nuestras aplicaciones.

En este documento se pretende guiar a los interesados en iniciar el uso de este framework a conocer los conceptos básicos de configuración y funcionalidad que CodeIgniter ofrece a los desarrolladores, para que puedan iniciarse en el uso de esta gran herramienta.

Descarga

Para descargar CodeIgniter simplemente accedemos a http://www.codeigniter.com/ y descargamos la última versión disponible:

Una vez descargado, tendremos un archivo con extensión .zip el cual deberemos descomprimir. Hecho esto tendremos ya todas las carpetas y archivos que CodeIngniter requiere para funcionar.

Luego debemos contar con un servidor para alojar nuestra aplicación y además un intérprete de php. Para este ejemplo se decidió utilizar XAMPP ya que cuenta con los requerimientos necesarios.

Una vez tengamos funcionando nuestro servidor debemos colocar la carpeta de CodeIgniter dentro de la carpeta htdocs con la que cuenta XAMPP. Esto permitirá acceder a nuestra aplicación desde el localhost una vez en el navegador.

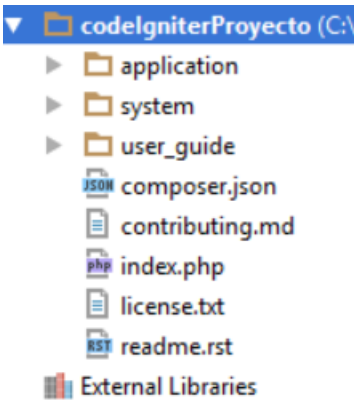
Esta es la página de bienvenida la cual nos indica que ya podemos empezar a usar el framework de CodeIgniter.

Guía de usuario

CodeIgniter cuenta con una guía de usuario a la cual podemos acceder mediante la URL, en ella podremos encontrar todo lo referente al uso adecuado de las herramientas que CodeIgniter ofrece:

La estructura de las carpetas y archivos

A continuación se mostrara la estructura de carpetas y archivos con los que cuenta CodeIgniter:



Application

Como desarrolladores nuestras mayorías de tareas van a estar centradas en la carpeta application.

System

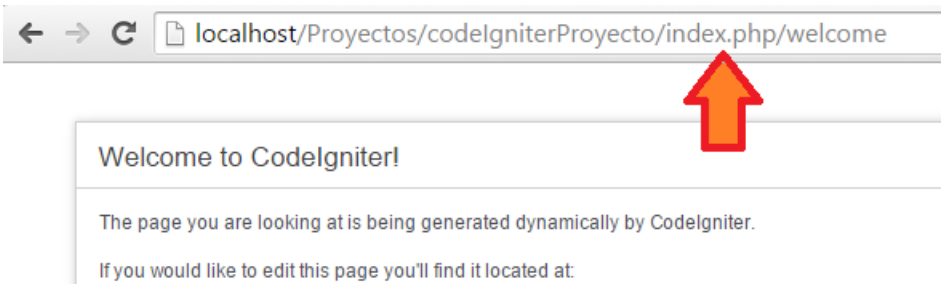
En la carpeta system se encuentran las librerías del framework.

User_guide

En la carpeta user_guide se encuentra la guía de usuario. Esta carpeta puede ser eliminada y no afectara nuestro framework.

Index.php

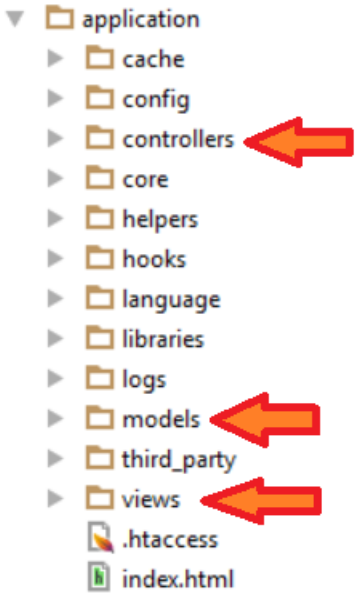
Es el archivo por el cual pasaran todas las solicitudes antes de comunicarse con el controlador:



Existe una manera de mostrar una URL más "amigable" en donde podemos eliminar el "index.php" de nuestra URL, pero lo veremos más adelante en este documento.

Modelo, Vista y Controlador.

Como sabemos CodeIgniter hace uso de patrón conocido como Modelo Vista Controlador (MVC). En la carpeta application podemos encontrar las carpetas que nos ayudan a implementar dicho patrón, las cuales son controllers, models y views.



Además existen otras carpetas de gran importancia como por ejemplo la carpeta config en donde se ingresara las configuraciones de nuestra aplicación tales como la necesaria para utilizar nuestra base de datos. Además en la carpeta libraries podemos crear nuestras propias librerías, al igual que en la carpeta helpers podemos crear nuestros propios helpers.

URL Base.

Una vez abrimos el Proyecto debemos indicar o establecer la variable “base_url” la cual se encuentra en application/config /config.php.

Nos encontraremos algo como esto:

```
$config['base_url'] = "";
```

Una vez modificado deberá quedar:

```
$config['base_url'] = 'http://localhost/Proyectos/codeIgniterProyecto';
```

Según la ubicación de nuestro proyecto. Esta variable será muy útil ya que nos permitirá acceder a la dirección base de nuestra aplicación desde cualquier parte de la aplicación.

Nota: Para poder acceder al método base_url() debemos invocar al helper url de la siguiente manera:

```
$autoload['helper'] = array('url');
```

Más adelante se explicara como invocar helpers.

Controlador.

Como sabemos el controlador es el intermediario entre el modelo y la vista, por eso es importante hacer un análisis de cómo el controlador invoca a los elementos tanto de la vista como del controlador.

Interacción con la vista.

Un ejemplo de como un controlador puede acceder a una vista es el siguiente:

```
public function index()
{
  $this->load->view('welcome_message');
}
```

En este caso se mostrara el archivo dentro de la carpeta views con el nombre wellcome_messagg.

Además desde un controlador podemos enviar por parámetros datos que queramos mostrar en la vista de la siguiente manera:

```
public function index()
{
  $dato['string'] = "Hola mundo";
```

```
   $this->load->view('welcome_message',$dato);
}
```

El controlador también recibe datos desde la vista que por ejemplo vengan de un formulario en donde exista un input con el atributo name = nombre.

```
function recibirDatos(){
   $datoRecibido = $this->input->post('nombre');
}
```

Interacción con el modelo

También el controlador debe interactuar con el modelo. Un ejemplo de cómo se invoca un modelo es:

```
function metodo(){
 $this->load->model('ejemplo_model');

 $data = array('nombre'=>$this->input->post('nombre'),'apellido'=>$this->input-
>post(apellido')
);
$this->micontrolador_model->crearCurso($data);

}
```

En donde ejemplo_model es el nombre del controlador invocado. Una vez invocado podemos acceder a dicho modelo, llamar a sus métodos y enviarle parámetros. Además dichos métodos pueden retornar datos al controlador.

Acceder a librerías y helpers.

Podemos acceder a librerías con el siguiente código dentro de los controladores:

```
$this->load->library('miLibreria');
```

Para acceder a los helpers utilizamos:

```
$this->load->helper('mihelper');
```

Además podemos invocar una librería o helpers desde el archivo application/config /autoload.php de la siguiente manera:

```
$autoload['helper'] = array('url');
```

En el ejemplo anterior se cargó el helper llamado “url”. La ventaja de cargar las librerías o helpers en el archivo autoload.php es que se cargaran cuando inicie la aplicación y podemos acceder a sus métodos desde cualquier parte de la aplicación.

Conexión con la Base de Datos.

Es necesario configurar nuestra aplicación para que se pueda conectar a una base de datos. Para ello vamos al archivo application/config/database.php, en donde podremos modificar datos tales como el hostname, username, password y database.

```
$active_group = 'default';
$query_builder = TRUE;

$db['default'] = array(
    'dsn'   => '',
    'hostname' => 'localhost',
    'username' => 'root',
    'password' => '',
    'database' => 'codeinter',
    'dbdriver' => 'mysqli',
    'dbprefix' => '',
    'pconnect' => FALSE,
    'db_debug' => TRUE,
    'cache_on' => FALSE,
    'cachedir' => '',
    'char_set' => 'utf8',
    'dbcollat' => 'utf8_general_ci',
    'swap_pre' => '',
    'encrypt' => FALSE,
    'compress' => FALSE,
    'stricton' => FALSE,
    'failover' => array(),
    'save_queries' => TRUE
);
```

Interacción con la base de datos

CodeIgniter usa una versión modificada del Patrón de Base de Datos Active Record, lo cual permite interactuar con la base de datos de manera más “simple”.
Un ejemplo de esto es el siguiente:

```
function eliminarCurso($idCurso){
    $this->db->where('idCurso', $idCurso);
    $this->db->delete('Cursos');
  }
```

También permite consultas SQL como en el siguiente ejemplo:

```php
function eliminarCurso($idCurso){
    $query = "DELETE FROM Cursos WHERE idCurso = $idCurso";
    $this->db->query($query);
}
```

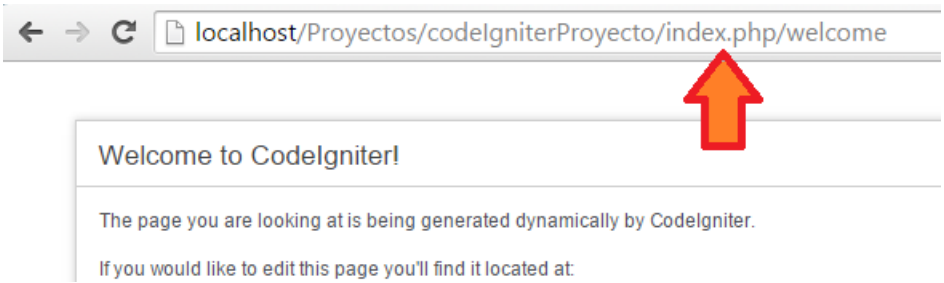
Rutas.

Dentro de application/config/routes.php podemos configurar las rutas de la URL para acceder al controlador que deseemos. Podemos indicar cuál va a ser el controlador que se ejecutara por defecto. También podemos crear rutas y re direccionarlas hacia algún método dentro de algún controlador.

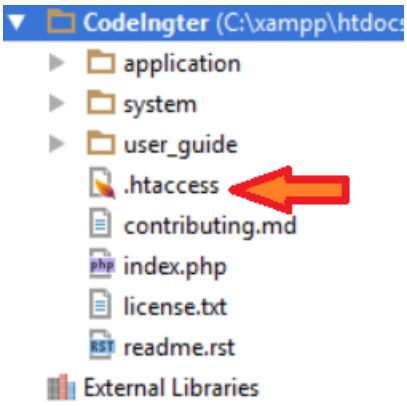
```php
$route['default_controller'] = 'welcome';
$route['nuevo'] = 'cursos/nuevo';
$route['404_override'] = '';
$route['translate_uri_dashes'] = FALSE;
```

Quitar el index.php de la URL.

Para lograr quitar el "index.php" de la URL como se ve en la siguiente imagen:



Debemos crear un archivo con el nombre .htaccess como este:

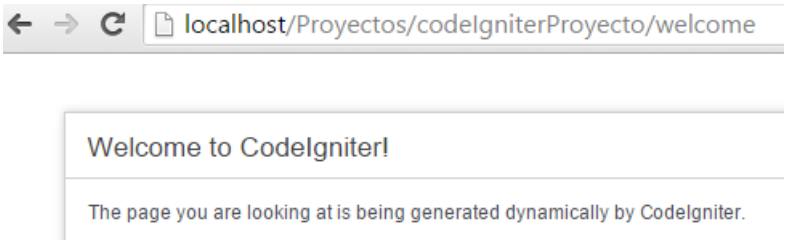


Una vez creado debemos ingresar el siguiente código dentro del archivo:

```
Options FollowSymLinks
<IfModule mod_rewrite.c>
   RewriteEngine on
   RewriteCond %{REQUEST_FILNAME} !-f
   RewriteCond %{REQUEST_FILNAME} !-d
   RewriteRule ^(.*)$ index.php?/$1 [L]
</IfModule>

<IfModule !mod_rewrite.c>
   ErrorDocument 404 /index.php
</IfModule>
```

Una vez hecho esto podremos accede al controlador sin la necesidad de colocar el "index.php" en la URL como se ve en la siguiente imagen



Conclusión

En general CodeIgniter es un framework fácil de instalar y fácil de usar. Existe bastante documentación lo cual lo vuelve un framework muy popular entre los desolladores web que utilizan php. Cuenta con una gran variedad de librerías y helpers que lo vuelve un framework muy completo y con mucha funcionalidad. Permite el desarrollo adecuado del patrón MVC lo cual hace a nuestras aplicaciones más robustas y bien estructuradas.