Frameworks: CodeIgniter

(UD 5)

Características de CodeIgniter

CodeIgniter es un framework MVC PHP caracterizado por ser:

- Pequeño y manejable.
- Fácil de instalar y configurar.
- Fácil de aprender.
- Opensource.
- Estable (se actualiza con menos frecuencia que otros)
- Profesional y potente (aunque menos que otros).

Instalación de CodeIgniter

1. Descargar (¡siempre del sitio del fabricante!)
2. Descomprimir en un directorio accesible por el servidor web.
3. Comprobar en el navegador (http://localhost/<directorio>)

Introducción a CodeIgniter

Directorios (algunos)

- **/application** → archivos de nuestra aplicación
  - **/application/config** → archivos de configuración globales
    - **/application/config/database** → datos de conexión a la BD
    - **/application/config/routes** → configuración del enrutador
    - **/application/config/config** → datos de configuración general
  - **/application/css** → hojas de estilo
  - **/application/helpers** → ayudantes
  - **/application/libraries** → bibliotecas
  - **/application/controllers** → controladores
  - **/application/models** → modelos
  - **/application/views** → vistas
- **/system** → el core de CodeIgniter. ¡No tocar!

Algunas convenciones de CodeIgniter

- Para cargar el método index() de un controlador:

  http://servidor/index.php/controlador

- Para cartar el método mi_metodo() de un controlador:

  http://servidor/index.php/controlador/método/parámetros

- Los identificadores no deben usar camelCase, sino guiones bajos (_)
- Los nombres de las clases deben empezar por mayúsculas. Los nombres de archivo deben ser iguales pero en minúscula.
- El index.php de la ruta sólo se puede quitar mediante .htaccess

Controladores

```
class Hola_mundo extends CI_Controller {
    public function index() {
        $this->load->view("hola_mundo_view");
    }
    public function otro_método() {
        ...
    }
}
```

Lo cargaríamos así:

- http://servidor/index.php/hola_mundo (ejecuta index())
- http://servidor (ejecuta index() si hemos configurado el enrutador de CI para que el controlador principal sea Hola_mundo)
- http://servidor/index.php/hola_mundo/otro_método (ejecuta otro_método())

Vistas

Son colecciones de documentos HTML cargados desde los controladores.

Pueden recibir datos empaquetados en arrays:

- Desde el controlador:

```
$datos[“titulo”] = “Titulo de la vista”;
$datos[“var1”] = “Valor de la variable 1”;
$this->load->view(“nombre_vista”, $datos);
```

- En la vista:

```
<title><?php echo $titulo; ?></title>
<body>
<?php echo “La variable var1 vale: $var1” ?>
etc.
```

Modelos

Los modelos pueden ser capas de acceso a la BD o cualquier otra implementación de la lógica del problema.

Algo muy habitual es implementar una **capa de acceso** a la BD usando la biblioteca Database, que sería la **capa de abstracción**.

El driver Database se configura en **config/database.php.**

(CI considera a Database un “Driver”, no una “library”):

```
class Mi_modelo extends CI_Model {
    public function __construct() {
        parent::__construct();
        $this->load->database();
    }
```

Introducción a CodeIgniter

A través de **$this->db** se puede ejecutar cualquier sentencia SQL.

Por ejemplo, una consulta que devuelva el contenido completo de una tabla puede tratarse así:

```
public function get_all_data() {
   $query = $this->db->query("SELECT * FROM mi_tabla");
   if ($query->num_rows() > 0)
   {
      foreach ($query->result() as $row)
      {
         $data[] = $row;
      }
      return $data;
   }
```

Introducción a CodeIgniter

También se puede escribir casi cualquier modelo sin una línea de SQL usando el patrón **Active Record** implementado por CI.

Por ejemplo, un método como el anterior, que recupera todos los datos de una tabla, quedaría así:

```
public function get_all_data() {
    $query = $this->db->get('mi_tabla');
    if ($query->num_rows() > 0) {
       foreach ($query as $row) {
          $data[] = $row;
       }
       return $data;
    }
}
```

Introducción a CodeIgniter

Con Active Record pueden encadenarse llamadas a los métodos de la clase para conformar una sentencia SELECT compleja sin escribir nada de SQL.

Por ejemplo:

```
$this->db->select('*');

$this->db->from('blogs');

$this->db->join('comments', 'comments.id = blogs.id');

$query = $this->db->get();


// Esto es equivalente a ejecutar:

// SELECT * FROM blogs

// INNER JOIN comments ON comments.id = blogs.id
```

Introducción a CodeIgniter

También se puede hacer INSERT o UPDATE, claro.

- Ejemplo de inserción mediante SQL:

```
$sql = "INSERT INTO...";
$this->db->query($sql);
echo $this->db->affected_rows();
```

- Ejemplo de inserción mediante Active Record:

```
$data = array(
                'campo1' => $valor1,
                'campo2' => $valor2,
                'campo3' => $valor3
        );
$this->db->insert('tabla', $data);
```

Helpers

Son “asistentes” para facilitar y agilizar tareas comunes en la programación de aplicaciones web.

Por ejemplo: **helper Form**

```
$this->load->helper(“form”);
```

A partir de ahí están disponibles funciones (no métodos) para crear formularios fácilmente sin escribir HTML:

```
echo form_open(“action”);
echo form_input(“nombre-input-text”);
echo form_sumbit(“id”, “etiqueta”);
```

O mediante un array:

```
$data = array('name' => 'username','id' => 'username',
              'value' => 'johndoe','maxlengt' => '100',
               'size' => '50','style' => 'width:50%');
echo form_input($data);
```

Libraries

Son clases para implementar tareas habituales del lado del servidor.

Por ejemplo: **library Form_validation**

```
$this->load->library(“form_validation”);
```

Estableciendo reglas de validación puede validarse el formulario del lado del servidor sin apenas escribir código:

```
$this->form_validation->set_rules('user', 'Usuario', 'required|
usr_no_existe');
$this->form_validation->set_rules('pass', 'Contraseña', 'required|
min_length[5]|strtolower);
if ($this->form_validation->run() == FALSE) {
        $this->load->view('myform');
}
else {
        $this->load->view('formsuccess');
}
```

Introducción a CodeIgniter

Para mostrar errores de validación en la vista:

```
echo validation_errors();
```

(puede personalizarse el mensaje con: set_message(“rule”, “msj”);)

Para repoblar los campos del formulario en la vista:

```
echo set_value(id);
```

Más información

La documentación oficial de CodeIgniter es extraordinariamente buena:

https://www.codeigniter.com/docs

Pondremos una copia para su consulta offline en el Aula Virtual, por si las moscas.