

CodeIgniter 3.1.8

Parte I: *estático*

Versión 1.0

Gustavo A. Casañ

CodeIgniter

- Framework: una librería que proporciona un entorno de trabajo para un lenguaje de programación.
- Es decir, clases y funciones que puedes emplear directamente.
- Página web: <https://www.codeigniter.com> o <https://codeigniter.es/> (CASTELLANO)
- Wiki:
<https://github.com/bcit-ci/CodeIgniter/wiki>

Arquitectura MVC

- Utiliza arquitectura MVC: Modelo-Vista-Controlador.
- Esto separa las funciones del código en tres “tipos” de ficheros.

Controlador

- Un controlador es una clase que permite delegar trabajo.
- A un controlador puede accederse con una url:

`http://example.com/[controller-class]/
[controller-method]/[arguments]/`

Ejemplo de Controlador

```
<?php
```

```
class Welcome extends CI_Controller {
```

```
    public function index() {
```

```
        $this->load->view('mi_vista');
```

```
    }
```

```
}
```

```
?>
```

Clase CI_Controller

```
class CI_Controller {  
var $config;      var $db;  
var $email;       var $form_validation;  
var $input;       var $load;  
var $router;      var $session;  
var $table;       var $unit;  
var $uri;         var $pagination;  
var $facebook;  
private static $instance;
```

Clase CI_Controller métodos

```
public function __construct() {  
    self::$instance =& $this;  
    foreach (is_loaded() as $var => $class) {  
        $this->$var =& load_class($class);  
    }  
    $this->load =& load_class('Loader', 'core');  
    $this->load->initialize();  
    log_message('debug', "Controller Class Initialized");  
}  
public static function &get_instance() {  
    return self::$instance;  
}  
} //Fin de la clase
```

Vista

- Una vista es un template (esquema) de página web (o un fragmento de ella).

Ejemplo de Vista

- Casi cualquier php con html.

```
<?php
defined('BASEPATH') OR exit('No direct script access
allowed'); ?>
<!DOCTYPE html>
<html lang="en">
<head> ... </head>
<body>
<div id="container">
...
</div>
</body>
</html>
```

Modelo

- Representa las estructuras de datos.
- Las consultas a la base de datos se deben colocar en un modelo, de forma que puedan reutilizarse después.
- CodeIgniter incluye una capa de abstracción de la base de datos, Query Builder (https://www.codeigniter.com/userguide3/database/query_builder.html).
- Esto permite crear aplicaciones independientes de la tecnología de BD.

Ejemplo de Modelo

```
class News_model extends CI_Model {  
    public function __construct() {  
        $this->load->database();    }  
  
    public function get_news($slug = FALSE) {  
        if ($slug === FALSE) {  
            $query = $this->db->get('news');  
            return $query->result_array();  
        }  
        $query = $this->db->get_where('news', array('slug'  
=> $slug));  
        return $query->row_array();  
    } ... }
```

Clase CI_Model

```
<?php if(!defined('BASEPATH')) exit('No direct script access allowed');
class CI_Model {
    var $config;          var $db;
    var $email;           var $form_validation;
    var $input;           var $load;
    var $router;          var $session;
    var $table;           var $unit;
    var $uri;             var $pagination;          var $facebook;
    function __construct() {
        log_message('debug', "Model Class Initialized");    }
    function __get($key) {
        $CI =& get_instance();
        return $CI->$key;
    }
}
```

No sólo estos ficheros MVC

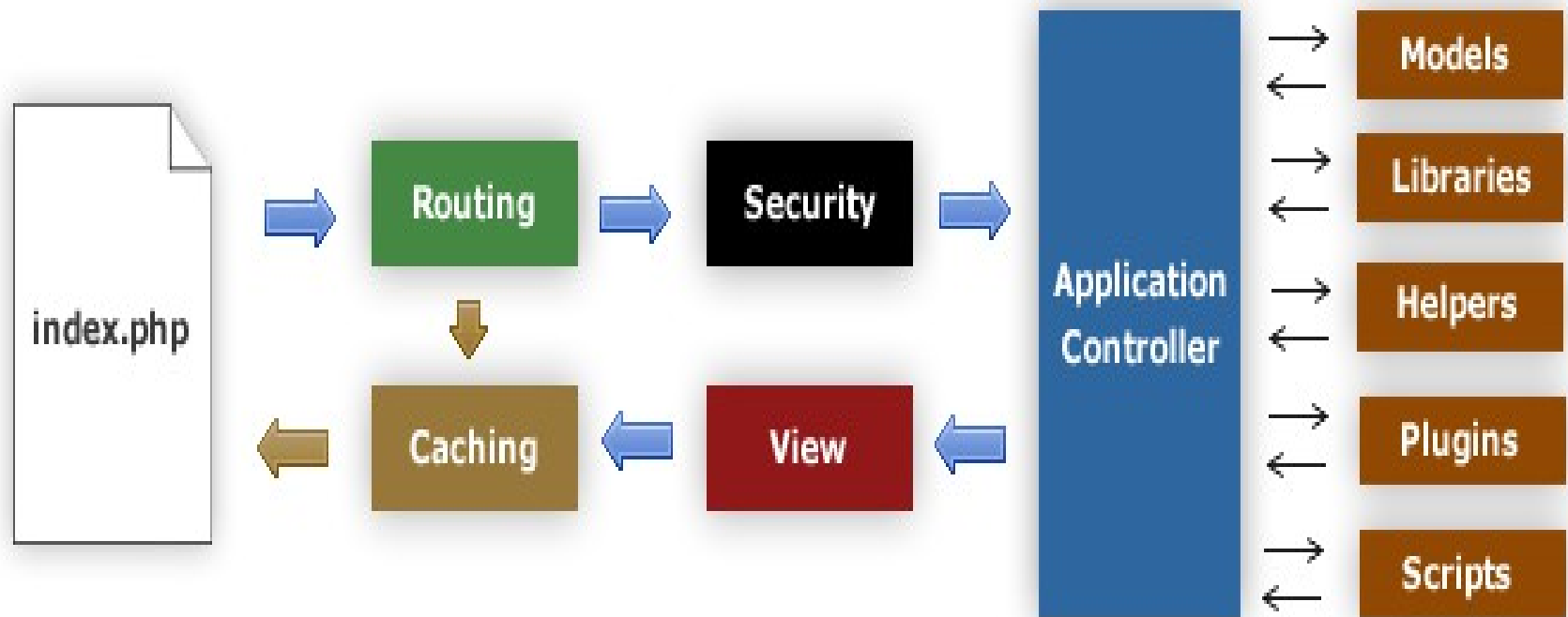
- **Helpers:** agrupan funciones de una categoría en especial: URL Helpers, que ayudan a crear enlaces, Form Helpers que ayudan a crear formularios, Text Helpers que realizan varias rutinas de texto, Cookie Helpers crean y leen, File Helpers...

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

```
$this->load->helper( array('helper1', 'helper2', 'helper3') );
```

- <https://www.codeigniter.com/userguide3/general/helpers.html>

Diagrama de flujo en una aplicación



Flujo en una aplicación

1. index.php es el controlador “front” inicializando los recursos base necesarios para CodeIgniter.
2. El Router examina la petición HTTP y determina que hacer con ella.
3. Si el fichero está en cache, se envía al navegador.
4. Antes de cargar el controlador de la aplicación, la petición HTTP y los datos son filtrados por seguridad.
5. El Controller carga modelo, librerías, helpers, y cualquier otro recurso necesario para la petición.
6. La vista finalizada se procesa y envía al navegador.

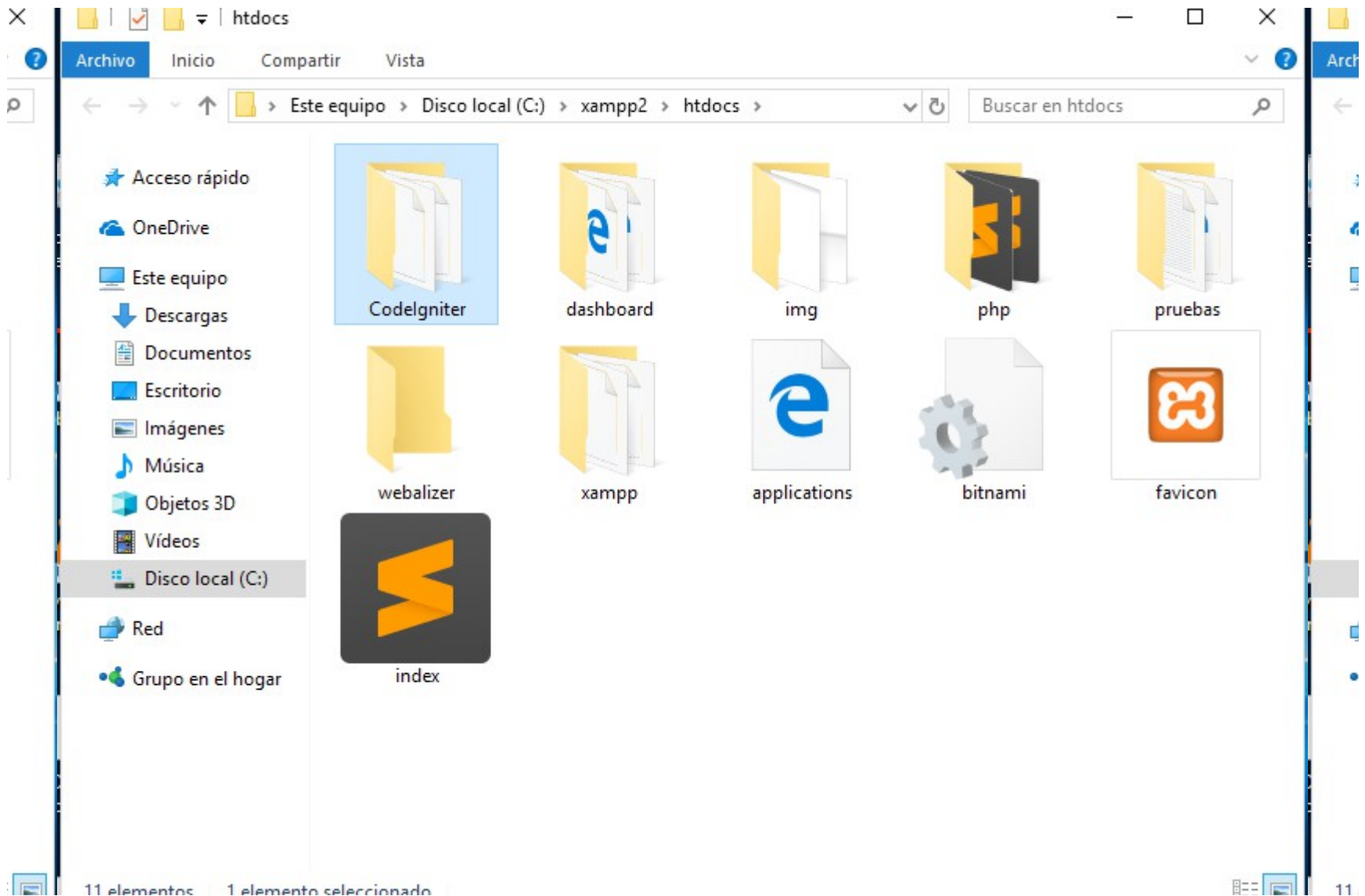
Previo a instalar CodeIgniter

- Todo este tutorial se ha hecho con Windows 10. En otros sistemas operativos será ligeramente distinto.
- Se ha instalado Xampp 5.6.36 en Windows 10.
<https://www.apachefriends.org/es/download.html>
- Por tanto se trabaja con PHP 5.

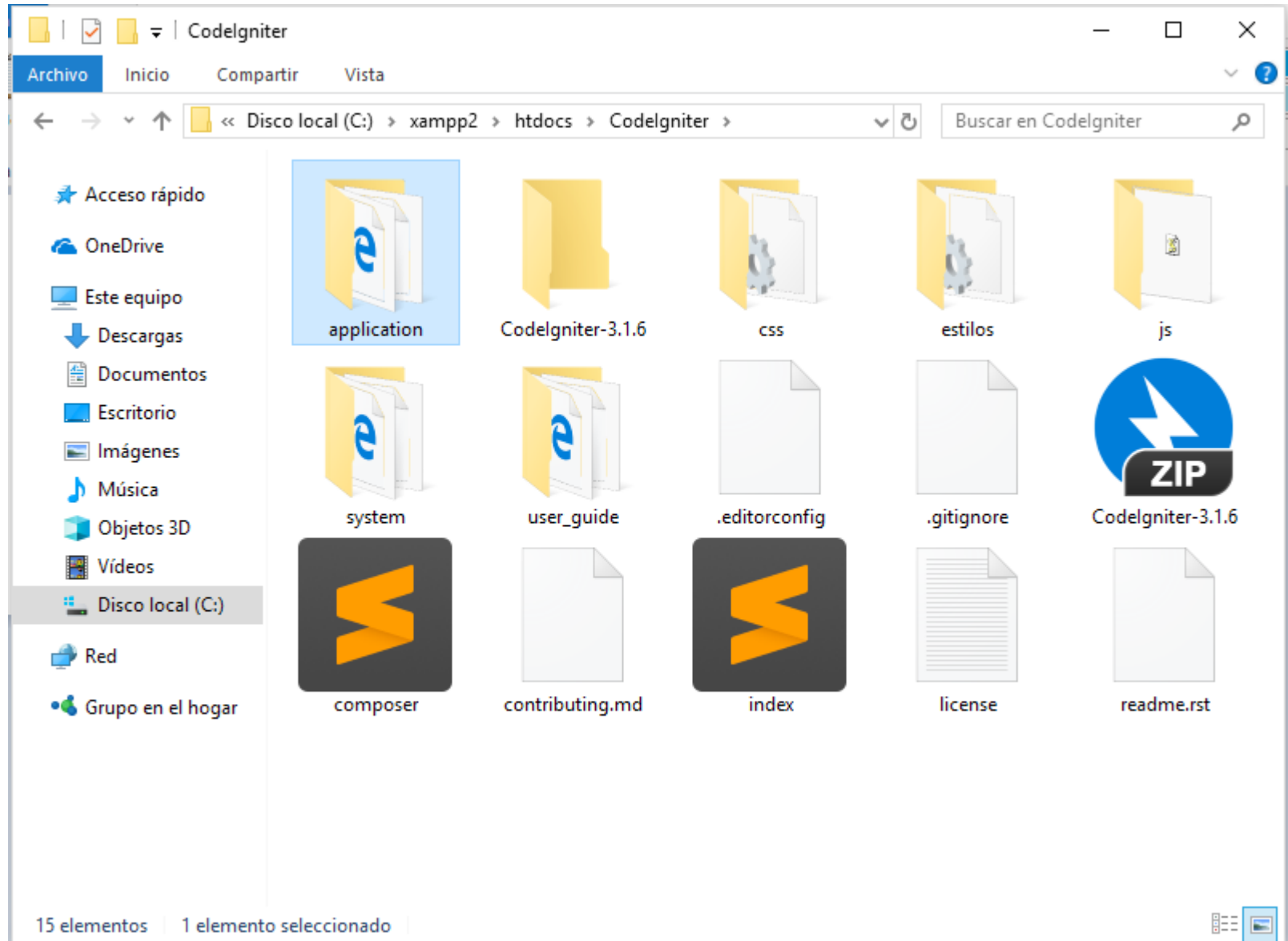
Instalar CodeIgniter

- Cuatro pasos: descargar, descomprimir y subir al servidor: index.php estará en root:
xampp/htdocs/codeigniter
- Abre el archivo application/config/config.php y coloca tu URL base
<http://localhost/codeigniter/>
- <https://www.codeigniter.com/userguide3/installation/index.html>

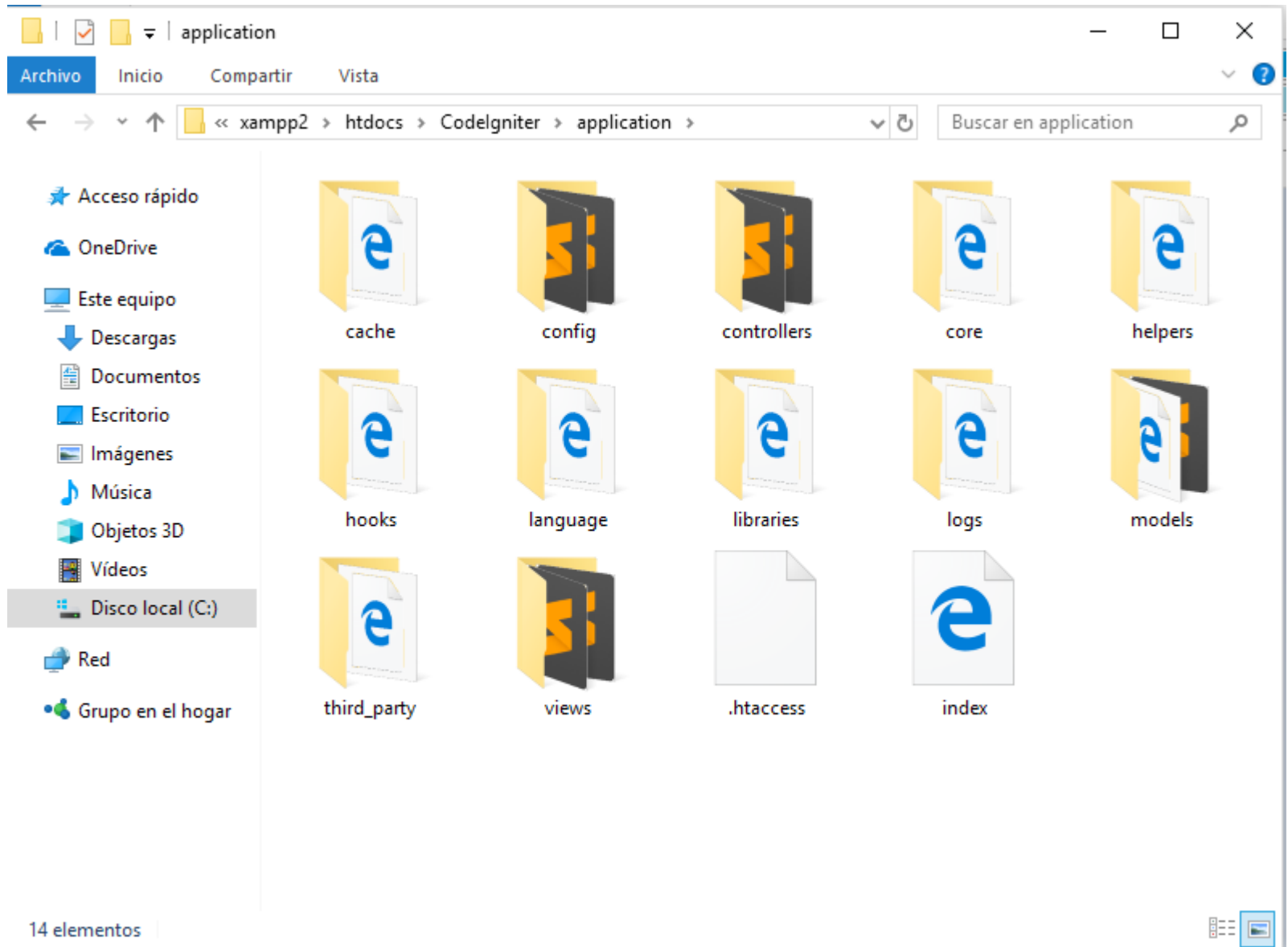
Mi carpeta root queda algo así:



Y CodeIgniter:



Trabajamos sobre todo en application:



Instalar Codeigniter II

- Configura tu base de datos en el fichero `application/config/database.php`.

```
$db['default'] = array(  
    'dsn' => "",  
    'hostname' => 'localhost',  
    'username' => "",  
    'password' => "",  
    'database' => "",  
    'dbdriver' => 'mysqli',  
    'dbprefix' => "", ...
```

- <https://www.codeigniter.com/userguide3/installation/index.html>

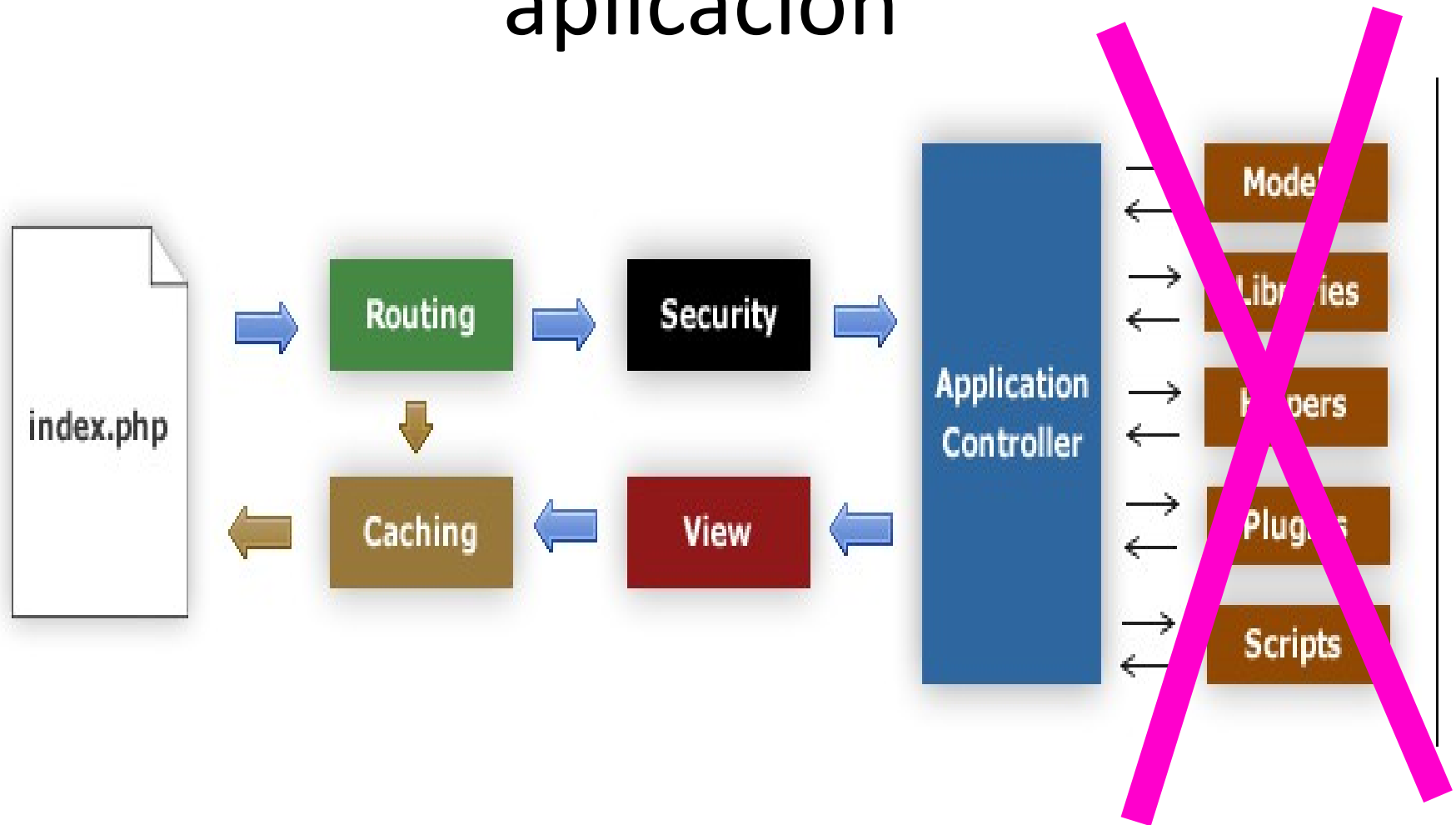
Instalar CodeIgniter III

- Podemos tener en cuenta la seguridad y modificar la instalación para que los ficheros no sean accesibles directamente.
- Podemos hacer que las direcciones web sean muy distintas de las posiciones reales de los ficheros.

Ejemplo: página estática

- La forma más sencilla de crear una página es mediante un único controlador en el directorio controllers:
C:\xampp\htdocs\CodeIgniter\application\controllers
- Fichero blog.php.
- Eso supone que no se utiliza gran parte de CodeIgniter, como se puede ver el siguiente diagrama de flujo.

Diagrama de flujo en una aplicación



Ejemplo: página estática blog.php

<http://localhost/codeigniter/index.php/blog/>

```
<?php
```

```
class Blog extends CI_Controller {
```

```
    function index() {
```

```
        echo 'Hello World!';
```

```
    }
```

```
?>
```

Tarea: Cambiar el mensaje a “Adiós, mundo cruel”.

Ejemplo: página estática II

- Las funciones dentro del fichero se interpretan como páginas.

<http://localhost/codeigniter/index.php/blog/comments/>

```
<?php
class Blog extends CI_Controller {
    function index() {
        echo 'Hello World!';
    }
    function comments() {
        echo 'Mira esto!';
    }
} ?>
```

Tarea: crea un función `adios()` que se despidan.

Ejemplo: página estática III

- Incluso los parámetros de las funciones se interpretan como páginas.

<http://localhost/codeigniter/index.php/blog/multiplica3/3/4/>

```
<?php
class Blog extends CI_Controller {
    function index() {
        echo 'Hello World!';
    }
    function multiplica3($p1, $p2) {
        echo $p1 * $p2 * 3;
    } ?>
```

Ejemplo: página estática IV

- Pero continúan siendo funciones, se pueden llamar.

<http://localhost/codeigniter/index.php/blog/multiplica/5/6/>

```
<?php
class Blog extends CI_Controller {
    function index() {
        echo 'Hello World!';
    }
    function multiplica($p1, $p2) {
        echo $p1 * $p2;
        $this->index();
    } ?>
```

Tarea: añade una función suma que sume dos parámetros.

Ejercicio 1: página estática

- Crea un fichero llamado prueba.php en el cual habrá las siguientes funciones (páginas):
 - Index: muestra un mensaje de bienvenida.
 - Dividir: te muestra el resultado de dividir dos números pasados como parámetros. Te advierte si intentas dividir por cero.
 - ParOImpar: dado un número pasado como parámetro, te dice si es par o impar.
 - Azar: nos devuelve uno de estos nombres al azar: Tomás, Gustavo, María, Juan, Belén.

Ejemplo: página estática V

- A veces no queremos que una función sea una página: hacemos que empiece por _ (barra horizontal baja).

```
<?php
```

```
class Blog extends CI_Controller {  
    function index() {  
        echo 'Hello World!';  
    }  
    function _meEscondido($p1, $p2) {  
        echo $p1 * $this->_invisible($p2);  
    }  
?>
```

Ejemplo: página estática VI

- Al ser una clase, una función invisible se llama con `$this->_nombreFuncion()`.

```
<?php
```

```
class Blog extends CI_Controller {  
    function _invisible($p1) {  
        return 3*$p1;  
    }  
    function multiplica3($p1, $p2) {  
        echo $p1 * $this->_invisible($p2);  
    }?  
}
```

Ejercicio 2: página estática

- Añade al fichero prueba.php la siguiente función “invisible”:
 - `_negativo($p)`: devuelve cero si el parámetro es un número negativo.
- Convierte la función `ParOImpar` en una invisible y utilízala en otra función llamada `comprobacion` que acepte un único parámetro.

Ejemplo: página estática VII

- ¿Dónde ponemos el html? Podríamos ponerlo aquí directamente.

```
<?php
```

```
class Blog extends CI_Controller {
```

```
function index() {
```

```
    echo '<h1>Hello World!</h1>';
```

```
} ?>
```

- Pero rompe MVC: **Modelo-Vista-Controlador**.

**¡NO LO ESTAMOS
HACIENDO BIEN!**

Ejemplo: página estática VIII

- El html se pone en la carpeta **view**.
- Y se carga desde el controlador.

```
<?php
defined('BASEPATH') OR exit('No direct script
access allowed');
class Welcome extends CI_Controller {
    public function index() {
        $this->load->view('welcome_message');
    }
} ?>
```

Ejemplo: página estática IX

- Se carga el fichero welcome_message.php

```
<?php
defined('BASEPATH') OR exit('No direct script access
allowed'); ?>
<!DOCTYPE html>
<html lang="en">
<head> ... </head>
<body>
<div id="container">
    <h1>Welcome ... </div>
</body>
</html>
```

Ejercicio 3: página estática

- Adapta la web formularioestatico.html para que nos funcione en CodeIgniter.

Ejemplo: página estática X

- Añadiendo ficheros css y JavaScript.

Añade código a application/config/routes.php

```
$route['profiler'] = "Profiler_controller";
```

```
$route['profiler/disable'] = "Profiler_controller/disable"
```

Crea el fichero controllers/test.php

https://www.tutorialspoint.com/codeigniter/codeigniter_adding_js_css.htm

```
<?php
```

```
class Test extends CI_Controller {
```

```
    public function index() {
```

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

```
        $this->load->view('test');    } }
```

```
?>
```

Ejemplo: página estática X (b)

```
<!DOCTYPE html>
<html lang = "en">
  <head>
    <meta charset = "utf-8">
    <title>CodeIgniter View Example</title>
    <link rel = "stylesheet" type = "text/css"
      href = "<?php echo base_url(); ?>css/style.css">
    <script type = 'text/javascript' src = "<?php echo
base_url(); ?>js/sample.js"></script>
  </head>
  <body>
    <a href = 'javascript:test()'>Click Here</a> to execute
the javascript function.
  </body>
</html>
```

Ejemplo: página estática X (c)

Crea los ficheros en los directorios correspondientes.

css/style.css

```
body {  
    background:#000;  
    color:#FFF;  
}
```

js/sample.js

```
function test() {  
    alert('test');  
}
```

Tarea: complica el ejemplo con más html y css.

Ejercicio 4: página estática

- Crea una página web de la ciudad de Castellón con css y JavaScript que muestre una barra de navegación, un carrusel con tres imágenes distintas de la ciudad y debajo una tabla con tres imágenes con textos explicativos (la Farola, la Madalena, el Ayuntamiento...). Cuando el ratón pase por encima de los textos, que aumente el tamaño de letra. Al hacer click sobre las imágenes de la tabla que se abra una nueva ventana con un enlace referente a esa página.

Comunicación vista-controlador

- Para pasar datos a la vista desde el controlador, se puede incluir como un simple parámetro en la llamada, en la forma de un array de datos. Las clases se transforman en arrays automáticamente.

```
$data = array(  
    'title' => 'My Title',  
    'heading' => 'My Heading',  
    'message' => 'My Message' );  
$this->load->view('blogview', $data);
```

Comunicación vista-controlador

- Para acceder a los datos en la vista, sólo tenemos que acceder a las variables que forman los índices del array.

...

```
echo "El título es ".$title;  
echo "<br>";  
echo "La cabecera: ".$heading;  
echo "Y el mensaje enviado...<br>";  
echo $message;
```

Ejemplo Comunicación

- Controlador Blog:

```
<?php
class Blog extends CI_Controller {
    public function index() {
        $data['title'] = "My Real Title";
        $data['heading'] = "My Real Heading";
        $this->load->view('blogview', $data);
    } } ?>
```

Vista blogview.php:

```
<html> <head> <title><?php echo $title;?></title> </head>
<body> <h1><?php echo $heading;?></h1>
</body> </html>
```

Ejemplo Comunicación: tarea

- Comprueba que esto funciona pasando un array de nombres y edades desde el controlador a una vista y en la vista mostrándolos en una tabla.

```
$lista = array(  
    array('nombre'=>'Juan', 'edad'=>'23'),  
    array('nombre'=>'Pablo', 'edad'=>'2'),  
    array('nombre'=>'Gustavo', 'edad'=>'43'),  
    array('nombre'=>'Esteban', 'edad'=>'27'),  
    array('nombre'=>'Maria', 'edad'=>'3') );
```

Ejercicio 5: página estática

- Si recordáis, hace ya unos meses hicimos un juego de Ping-Pong con JavaScript. Adáptalo para que funcione con CodeIgniter (en el drive están las instrucciones originales y mi código, por si lo habéis perdido). No vale poner imágenes en el directorio views, habrá que crearles un directorio propio, como al css y al JavaScript.

Ejercicio 6: página estática

- Si recordáis, hace ya unos meses hicimos un juego de localizar el perro escondido entre los gatos. Adáptalo para que funcione con CodeIgniter (en el drive están las instrucciones originales y mi código, por si lo habéis perdido). No vale poner imágenes en el directorio views, habrá que crearles un directorio propio, como al css y al JavaScript.

Helpers

- Hay multitud de helpers, cada uno con un propósito. https://codeigniter.com/user_guide/helpers/index.html

Array Helper

Cookie Helper

Directory Helper

Email Helper

Form Helper

Inflector Helper

Number Helper

Security Helper

String Helper

Typography Helper

XML Helper

CAPTCHA Helper

Date Helper

Download Helper

File Helper

HTML Helper

Language Helper

Path Helper

Smiley Helper

Text Helper

URL Helper

Helper: Form

- Crear un formulario:
- En el Controlador (empleado.php)

```
function nuevo_empleado(){  
    $this->load->helper('form');  
    $this->load->view("formulario2");  
}
```

Helper: Form

- En la Vista (formulario2.php):

```
<html> <body> <h1>Nuevo empleado</h1>
  <?php
    echo form_open('empleado/nuevo_empleado');
    echo form_label('Nombre', 'nombre');
    echo form_input('nombre');echo '<br>';
    echo form_label('Sueldo', 'sueldo');
    echo form_input('sueldo');echo '<br>';
    echo form_submit('botonSubmit', 'Enviar');
    echo form_close();
  ?>
</body>
</html>
```

Tarea: Form

- Añádele una función index que también nos lleve al formulario.
- Mejora el ejemplo anterior con algo de CSS: espacios, colores...
- Por ejemplo:

```
$attributes = array('class' => 'email', 'id' => 'myform');  
echo form_open('email/send', $attributes);  
echo form_open('email/send', 'class="email"  
id="myform"');
```

Ejercicio 7: crea un formulario

- Crear un formulario para realizar una reserva en un balneario. Te solicitará los datos personales y una fecha de llegada. Utiliza el helper Form, claro.

Funciones curiosas

- Text: `word_censor()`
- Array: `random_element()`
- HTML: `img()`, `ul()`
- String: `increment_string()`
- Url: `anchor_popup()`

Librerías

- Hay multitud de librerías, cada una con un propósito (aunque no vamos a explorarlas todas)

<https://www.codeigniter.com/userguide3/libraries/index.html>

Librerías

- Benchmarking Class
 - Calendaring Class
 - Config Class
 - Encrypt Class
 - File Uploading Class
 - FTP Class
 - **Input Class**
 - Language Class
 - Migrations Class
 - Pagination Class
 - Security Class
 - **HTML Table Class**
 - Typography Class
 - URI Class
 - XML-RPC and XML-RPC Server Classes
 - Zip Encoding Class
- Caching Driver
 - Shopping Cart Class
 - Email Class
 - Encryption Library
 - Form Validation**
 - Image Manipulation Class
 - Javascript Class
 - Loader Class
 - Output Class
 - Template Parser Class
 - Session Library
 - Trackback Class
 - Unit Testing Class
 - User Agent Class

Librerías

- Una librería muy usada es **Input**

<https://www.codeigniter.com/userguide3/libraries/input.html>

Input

- Sobre todo acceder a datos de un formulario:

```
$algo = isset($_POST['algo']) ? $_POST['algo'] :  
NULL;
```

```
$algo = $this->input->post('algo');
```

- Los métodos principales son:

```
$this->input->post()
```

```
$this->input->get()
```

```
$this->input->cookie()
```

```
$this->input->server()
```

Ejemplo: Input en el controller

```
// No hace falta cargar la librería input  
class Empleado extends CI_Controller { ...  
    public function nuevo_empleado()    {  
        $this->load->helper('form');  
        $nombre = $this->input->post('nombre');  
        $sueldo = $this->input->post('sueldo');  
        // $this->load->model('empleado_model');  
        // $this->empleado_model->  
>insertar_empleado($nombre, $sueldo);  
        $this->load->view("formulario2");  
    }  
}  
?>
```

Ejercicio 8: acceder a datos

- Modifica el programa del formulario creado en el ejercicio 7 para que acceda a los datos y los muestre con un echo.

Table: ejemplo en controller

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

```
$data = array(  
    array('Name', 'Color', 'Size'),  
    array('Fred', 'Blue', 'Small'),  
    array('Mary', 'Red', 'Large'),  
    array('John', 'Green', 'Medium')  
);
```

```
echo $this->table->generate($data);
```

Table: ejemplo II

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

```
$this->table->set_heading('Name', 'Color', 'Size');
```

```
$this->table->add_row('Fred', 'Blue', 'Small');
```

```
$this->table->add_row('Mary', 'Red', 'Large');
```

```
$this->table->add_row('John', 'Green', 'Medium');
```

```
echo $this->table->generate();
```

```
$this->table->clear();
```

Table: template

```
$template = array(  
    'table_open'          => '<table border="0"  
cellpadding="4" cellspacing="0">',  
    'thead_open'          => '<thead>',  
    'thead_close'         => '</thead>',  
    'heading_row_start'   => '<tr>',  
    'heading_row_end'     => '</tr>',  
    'heading_cell_start'  => '<th>',  
    'heading_cell_end'    => '</th>',  
    'tbody_open'          => '<tbody>',  
    'tbody_close'         => '</tbody>',  
    'row_start'           => '<tr>',  
    'row_end'             => '</tr>',
```

Table: template II

'row_end'	=> '</tr>'
'cell_start'	=> '<td>'
'cell_end'	=> '</td>'
'row_alt_start'	=> '<tr>'
'row_alt_end'	=> '</tr>'
'cell_alt_start'	=> '<td>'
'cell_alt_end'	=> '</td>'
'table_close'	=> '</table>'

);

\$this->table->set_template(\$template);

Tarea Table: template en controller

- Incorpora un template de tabla al ejemplo anterior para que quede más atractiva.
- Puedes empezar por algo como
`'table_open' => '<table border="1"
cellpadding="8" cellspacing="2">'`

Ejercicio 9: mostrar una tabla

- Modifica el programa del formulario creado en el ejercicio 7 (y 8) para que acceda a los datos y los muestre en una tabla.

Librerías

- Otra librería muy usada es **Form Validation**

https://www.codeigniter.com/userguide3/libraries/form_validation.html

- Vamos a explorarla a continuación.

Form Validation: escenario ideal en el cliente

- Se muestra un formulario.
- Lo rellenas y envías
- Si enviaste algo invalido o faltan datos se vuelve a mostrar el formulario junto con un mensaje describiendo un problema.
- Repetir hasta que se envía un formulario correcto.

Form Validation: escenario ideal en el servidor

- Se comprueban los datos requeridos.
- Se verifican que los datos son del tipo correcto.
- Se limpian los datos por seguridad.
- Se pre-formatean los datos si es necesario (¿Tiene espacios en blanco? ¿Códigos HTML?)
- Se preparan los datos para insertarlos en la base de datos.

Ejemplo: Form Validation I

- Trabajamos con tres ficheros distintos, dos views, myform.php y formsuccess.php y un controller, Form.php.

- formsuccess.php

```
<html> <head> <title>My Form</title> </head>
<body>
<h3>¡Tu formulario fue enviado con éxito!</h3>
<p><?php echo anchor('form', '¡Inténtalo de nuevo!'); ?
></p>
</body>
</html>
```

Ejemplo: Form Validation II

- myform.php

```
<?php echo validation_errors(); ?>
<?php echo form_open('form'); ?>
<h5>Nombre de usuario</h5>
<input type="text" name="username" value="" size="50" />
<h5>Password</h5>
<input type="text" name="password" value="" size="50" />
<h5>Confirmar Password</h5>
<input type="text" name="passconf" value="" size="50" />
<h5>Dirección de Email</h5>
<input type="text" name="email" value="" size="50" />
<div><input type="submit" value="Submit" /></div>
</form>
```

Ejemplo: Form Validation III

- `<?php echo validation_errors(); ?>` devuelve los mensajes de error del validador. O una cadena vacía si no hay errores.
- ¿Por qué usamos `<?php echo form_open('form'); ?>`

Nos genera directamente

```
<form  
action="http://localhost/codeigniter/index.php/form"  
method="post" accept-charset="utf-8">
```

Y esto va muy bien con la redirección dentro de nuestra página.

Tarea Form: usa helper

- En este ejemplo la creación de la página se hace con html. Utiliza el helper Form que ya hemos visto para crear el formulario.

Ejemplo: Form Validation IV

- Form.php

```
class Form extends CI_Controller {  
    public function index()    {  
        $this->load->helper(array('form', 'url'));  
        $this->load->library('form_validation');  
        if ($this->form_validation->run() == FALSE) {  
            $this->load->view('myform');  
        }  
        else { $this->load->view('formsuccess'); }  
    }  
}
```

Tarea: Comprueba que funciona. Añade un campo dirección de tamaño 100 al formulario.

Form Validation: REGLAS

- Pero aún no comprobamos nada.
- `$this->form_validation->set_rules(P1, P2, P3[, P4]);`

P1: Nombre del campo en el formulario.

P2: Un nombre más “humano” que se insertará en el mensaje de error.

P3: Las reglas de validación para este campo.

P4: (opcional) Mensajes de error personalizados.

```
$this->form_validation->set_rules('username', 'Username',  
'required');
```

set_rules(): Reglas de validación

- `$this->form_validation->set_rules(P1, P2, P3[, P4]);`

`required`

`matches`

`regex_match`

`differs`

`is_unique`

`min_length` `max_length` `exact_length`

`greater_than` `greater_than_equal_to` `less_than ...`

`in_list`

`alpha` `alpha_numeric` `alpha_numeric_spaces` `alpha_dash`

`numeric` `integer` `decimal` `is_natural` `is_natural_no_zero`

`valid_url` `valid_email` `valid_emails` `valid_ip` `valid_base64`

- También `$this->form_validation->required($string);`

set_rules(): Reglas de validación mensajes de error

- `$this->form_validation->set_rules(P1, P2, P3[, P4]);`
`$this->form_validation->set_rules('field_name', 'Field Label',
 'rule1|rule2|rule3',
 array('rule2' => 'Error Message on rule2 for this
field_name'));`

ó

`$this->form_validation->set_message('min_length', '{field}
must have at least {param} characters.');`

Ejemplo: Form Validation V

```
class Form extends CI_Controller {  
    public function index()    {  
        $this->load->helper(array('form', 'url'));  
        $this->load->library('form_validation');  
        $this->form_validation->set_rules('username', 'Username',  
'required');  
        $this->form_validation->set_rules('password', 'Password',  
'required', array('required' => 'You must provide a %s.') );  
        ...  
        if ($this->form_validation->run() == FALSE) {  
            $this->load->view('myform');  
        }  
        else { $this->load->view('formsuccess'); }  
    }  
}
```

Tarea: Añade validaciones para todos los campos.

Ejemplo: Form Validation VI

```
$this->form_validation->set_rules('username', 'Username',  
    'required | min_length[5] | max_length[12] |  
is_unique[users.username]',  
    array('required'    => 'You have not provided %s.',  
          'is_unique'   => 'This %s already exists.')  
);  
$this->form_validation->set_rules('password', 'Password',  
    'required');  
$this->form_validation->set_rules('passconf', 'Password  
Confirmation', 'required|matches[password]');  
$this->form_validation->set_rules('email', 'Email', 'required|  
valid_email|is_unique[users.email]');
```

Tarea: Valida más correctamente todos los campos del formulario ejemplo.

Ejercicio 10: Form Validation

- Con el ejercicio de crear un formulario para introducir los siguientes datos de una persona y validarlos:
 - Nombre
 - DNI
 - Dirección
 - Fecha de Nacimiento
 - Teléfono
- Que el controlador le devuelva a la vista estos datos validados y la vista los muestre en una tabla.

Ejemplo: Form Validation VII

- Por supuesto puedes crear tus propias funciones de validación.

```
$this->form_validation->set_rules('username', 'Username',  
'callback_username_check');
```

- Y en el controller creas la función correspondiente:

```
public function username_check($str) {  
    if ($str == 'test') {  
        $this->form_validation->set_message('username_check',  
'The {field} field can not be the word "test"');  
        return FALSE;  
    }  
    else { return TRUE; } }
```

Tarea: Incorpora este código al ejemplo.

Ejemplo: Form Validation VIII

- Introducir datos en un formulario: `set_value()`

...

```
<?php echo validation_errors(); ?>
```

```
<?php echo form_open('form'); ?>
```

```
<h5>Username</h5>
```

```
<input type="text" name="username" value="<?php echo  
set_value('username');?>" size="50" />
```

```
<h5>Password</h5>
```

```
<input type="text" name="password" value="<?php echo  
set_value('password');?>" size="50" />
```

...

Tarea: Reintroduce todos los valores en los campos del formulario ejemplo.

Ejercicio 11: Form

- Mejora el formulario anterior para que cuando se cometa un error de validación aparezcan los datos introducidos y no haya que volver a escribirlos todos.
- Crea funciones de validación propias para dos campos del formulario:
 - Dirección tiene que incluir Almazora
 - Telefono tiene que empezar por seis (6).

Ejercicio 12: Form²

- Añade Bootstrap al formulario anterior para mejorar su aspecto.
- Además, cuando se rellene el formulario correctamente que se muestre una página distinta con los datos introducidos en una tabla.

Ejercicio 13: tenis

- Crea un sitio web dedicado al tenis que tenga usuarios que deban acreditarse para entrar y acceder a las secciones privadas. La gama de colores de la página que sea verde y blanco. Y habrá una página de inicio con una cabecera con barra de navegación con los siguientes enlaces:

Ejercicio 13: tenis (continuación)

- El tenis (página de inicio): explicación del deporte y algunas fotos y enlaces (federación internacional, wiki, etc...), en dos columnas. Por ejemplo:

El tenis, también llamado tenis de campo, es un deporte de raqueta practicado sobre una pista rectangular (compuesta por distintas superficies las cuales pueden ser cemento, tierra, o césped), delimitada por líneas y dividida por una red. Se disputa entre dos jugadores (individuales) o entre dos parejas (dobles). El objetivo del juego es lanzar una pelota golpeándola con la raqueta de modo que rebote dentro de los límites permitidos del campo del rival, procurando que este no pueda devolverla para conseguir un segundo rebote en el suelo y por ende un punto.

Ejercicio 13: tenis (continuación)

- Galería: fotos de tenis organizadas en tres columnas. Cuando se reduzca de tamaño la pantalla primero que pasen a dos y finalmente a una. Cada foto tendrá una frase diciendo algo sobre ella. Al hacer click sobre el texto se abrirá una nueva ventana con la foto.
- Contacto: formulario típico (nombre, email, tema, texto del mensaje). Tras hacer click sobre enviar (si la validación es correcta) se vuelve a la página de inicio. Pero antes se deben eliminar las palabras “futbol” y “tonto” del cuerpo del mensaje (hay una función de CodeIgniter que puede hacer justo eso...) y escribirlo en la consola.

Ejercicio 13: tenis (continuación)

- Usuario: página formulario login/password con un enlace a la página Nuevo usuario en el texto “¿Aún no estás registrado?”.
- Nuevo usuario: página formulario con datos personales, un password y la confirmación del password. Tras hacer click en enviar y si la validación es correcta debe enviarnos a la página Usuario.

Ejercicio 13: tenis (continuación)

- Quienes somos: texto de presentación y una tabla con dos o tres nombres. Por ejemplo:
 - Somos unos enamorados del tenis que llevamos 10 años dedicando toda nuestra energía y pasión a este magnífico deporte. En esta página podrás mantenerte al día de las competiciones y las últimas novedades de material.
 - La tabla será esta (o lo más parecida posible).

Nombre	Cargo	Año incorporación
Gustavo Casañ	Presidente	2007
María de la O	Tesorero	2009
Rodrigo Perez	Secretario	2008

Ejercicio 13: tenis (continuación)

- Además habrá un pie de página con la información de copyright habitual (todos los derechos reservados, 2018, etc...) además de la fecha y hora actuales (CodeIgniter tiene funciones...).
- Se debe realizar la validación de todos los campos e informar de los errores al usuario en el propio formulario (que debe mostrar los datos introducidos).
- En todas las páginas se podrán ver tanto la cabecera como el pie de página.
- Si el visitante introduce como usuario la palabra administrador con password administrador, que se abra una nueva ventana con el mensaje “No te ⁸⁹pases de listo”.

CodeIgniter

Parte II: dinámico

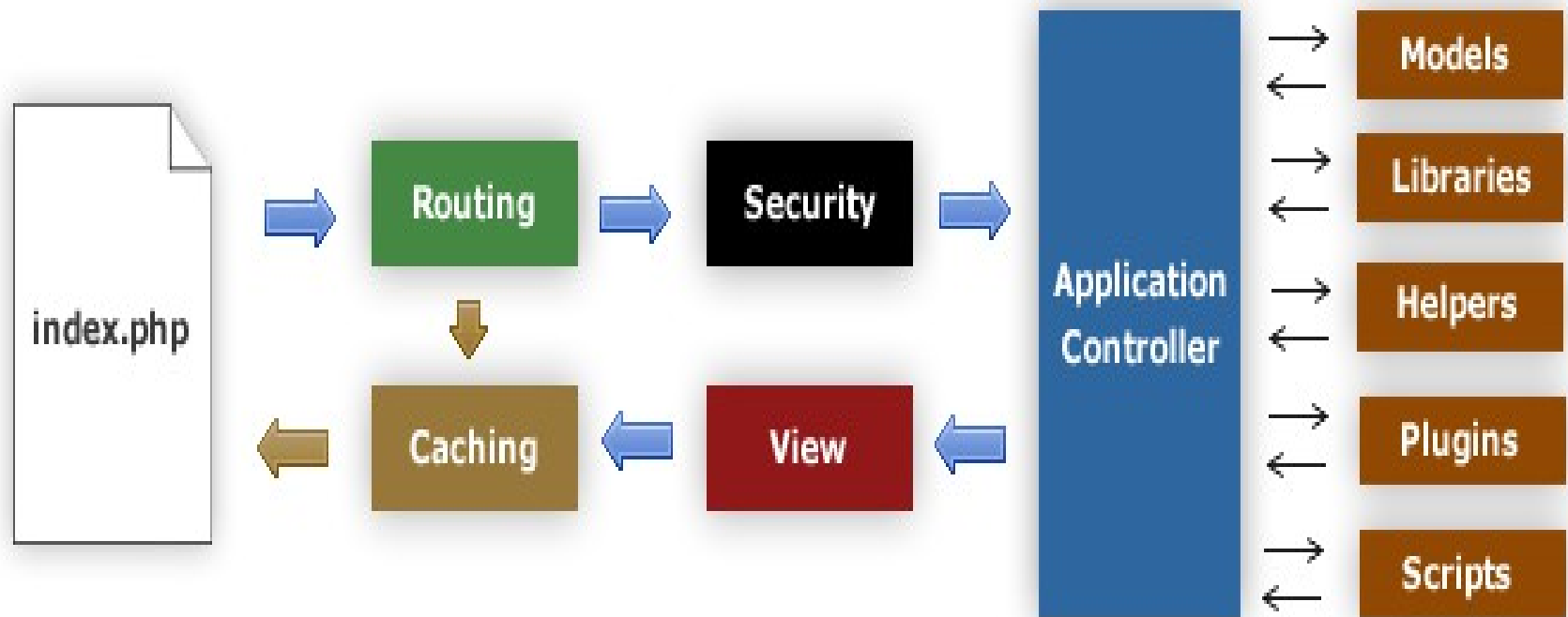
CodeIgniter

- Framework: una librería que proporciona un entorno de trabajo para un lenguaje de programación.
- Página web: <https://www.codeigniter.com>
- Utiliza arquitectura MVC: Modelo-Vista-Controlador.
- Permite crear páginas dinámicas utilizando la información extraída de la Base de Datos.

Modelo

- Representa las estructuras de datos.
- Las consultas a la base de datos se deben colocar en un modelo, de forma que puedan reutilizarse después.
- CodeIgniter incluye una capa de abstracción de la base de datos, Query Builder (
https://www.codeigniter.com/userguide3/database/query_builder.html).
- Esto permite crear aplicaciones independientes de la tecnología de BD.

Diagrama de flujo en una aplicación



Configurar la Base de Datos

- Configura tu base de datos en el fichero `application/config/database.php`.

```
$db['default'] = array(  
    'dsn' => '',  
    'hostname' => 'localhost',  
    'username' => 'alumno',  
    'password' => 'alumno',  
    'database' => 'test',  
    'dbdriver' => 'mysqli',  
    'dbprefix' => '', ...
```

- <https://www.codeigniter.com/userguide3/installation/index.html>

Configurar la Base de Datos II

- Aunque eso es suficiente la configuración puede ser muy compleja. Vale la pena mirar la documentación.

<https://www.codeigniter.com/userguide3/database/configuration.html>

- Cosas que se pueden determinar: casi todo. Por ejemplo, a que otras bases de datos/servidores conectarse si falla la actual.

Páginas dinámicas

- Las consultas **se deben colocar en el modelo**, no directamente en el controlador, para facilitar su reutilización.
- Los modelos son el lugar dónde se accede y modifica la información de la base de datos u otra fuente de datos. Representan tus datos.

Páginas dinámicas

- Se crea un nuevo modelo extendiendo CI_Model y cargando la librería de la base de datos.

- En el controlador:

```
$this->load->model('nombre_del_modelo');
```

- Esto hace a la BdD accesible a través del objeto \$this->db

```
$this->nombre_del_modelo
```

Páginas Dinámicas: ejemplo

```
class News_model extends CI_Model {  
    public function __construct() {  
        $this->load->database();    }  
  
    public function get_news($slug = FALSE) {  
        if ($slug === FALSE) {  
            $query = $this->db->get('news');  
            return $query->result_array();  
        }  
        $query = $this->db->get_where('news', array('slug'  
=> $slug));  
        return $query->row_array();  
    } ... }
```

Páginas dinámicas: ejemplo model

```
class Blog_model extends CI_Model {  
    public $title;    public $content;  
    public function get_last_ten_entries() {  
        $query = $this->db->get('entries', 10);  
        return $query->result();    }  
    public function insert_entry() {  
        $this->title  = $this->input->post('title');  
        $this->content = $this->input->post('content');  
        $this->db->insert('entries', $this);    }  
    public function update_entry() {  
        $this->title  = $this->input->post('title');  
        $this->content = $this->input->post('content');  
        $this->db->update('entries', $this, array('id' => $this->  
>input->post('content')));  
    } }  
}
```

PD: ejemplo controller

```
class Blog_controller extends CI_Controller {  
    public function index() {  
        ...  
    }  
    public function blog() {  
        $this->load->model('blog','',TRUE);  
        $data['query'] = $this->blog-  
>get_last_ten_entries();  
        $this->load->view('blog', $data);  
    }  
}
```

PD: ejemplo view

```
<html>
<body> <h1>      Nueva entrada de blog  </h1>
  <?php echo form_open('blog_controller/blog');
    echo form_label('Titulo', 'title');
    echo form_input('title'); echo '<br>';
    echo form_label('Contenido', 'content');
    echo form_input('content'); echo '<br>';
    echo form_submit('botonSubmit', 'Enviar');
    echo form_close();
    if(isset($query)) {
        echo "El título del primer blog es: ".$query[0]->title;
        echo "Y su contenido es: ".$query[0]->content;
    }  ?>
</body> </html>
```

PD: tarea ejemplo

- Adapta la vista del ejemplo para que muestre todos los datos devueltos por la consulta en una tabla.
- Crea un botón en la vista principal que te abra una vista que muestre la tabla (quita la tabla del formulario).
- Cambia el formulario para que al apretar submit se acabe llamando al método `insert_entry()` del modelo.

PD: tarea ejemplo

- Crea un método que te devuelva toda la tabla.
- Crea un método con un parámetro de entrada número que indique el número de filas a extraer de la tabla (y el controlador y la vista con formulario correspondiente).
- Vista y controlador para insertar una nueva fila.

Páginas dinámicas

- Se crea un nuevo modelo extendiendo `CI_Model` y cargando la librería de la base de datos. Esto hace a la BdD accesible a través del objeto `$this->db`

`$this->load->database();` ó se conecta automáticamente (por ejemplo con la opción `TRUE`).

Conexión a la BdD

- Se puede conectar a varias bases de datos a la vez: modificar el fichero de configuración database.php para crear distintos grupos.

```
$active_group = 'public';
$active_record = TRUE;
$db['public']['hostname'] = 'localhost';
$db['public']['username'] = 'root';
$db['public']['password'] = '';
$db['public']['database'] = 'my_public_db';
$db['public']['dbdriver'] = 'mysql';
...
$db['admin']['hostname'] = 'localhost';
$db['admin']['username'] = 'root';
$db['admin']['password'] = '';
$db['admin']['database'] = 'my_admin_db';
$db['admin']['dbdriver'] = 'mysql';
...
```

Conexión a la BdD

- En este caso conectarse como:

```
$DB1 = $this->load->database('public', TRUE);
```

```
$DB2 = $this->load->database('admin', TRUE);
```

O por ejemplo:

```
$this->db = $this->load->database('admin', TRUE);
```

```
$records = $this->my_admin_model->get_example_data();
```

```
$this->db = $this->load->database('public', TRUE);
```

Conexión a la BdD

- Se pueden cambiar los parámetros de conexión especificados en el fichero de configuración.
- Se puede realizar cualquier acción.
- Se utiliza la clase Query Builder (QB) para conectarse y manipular la Base de Datos (aunque no es imprescindible).

Query Builder: ejemplos de acceso

- Ejemplo “exótico”:

```
$data = array(  
    'title' => $title,  
    'name' => $name,  
    'date' => $date );  
$this->db->insert('mytable', $data);  
// Produce: INSERT INTO mytable (title, name, date)  
VALUES ('{$title}', '{$name}', '{$date}')
```

Query Builder

- Muy complejo, leeros la lista de funciones que os he pasado.
- Ya hemos visto, sin embargo...

```
$query = $this->db->query('SELECT name, title  
FROM my_table');
```

```
foreach ($query->result() as $row) {  
    echo $row->title;  
    echo $row->name;  
}
```

Query Builder: detalles

- En general los parámetros pueden pasarse como un array asociativo.
- Sentencias que ejecutan y otras que sólo preparan la consulta (`_compiled_`).
- Concatenado de métodos:

```
$this->db->select('title')->where('id',$id)->get_('mitabla');
```

Query Builder: ejemplo clases

```
function mostrar_persona() {  
    $this->load->database();  
    $query = $this->db->query('SELECT nombre, dni  
FROM persona');  
    foreach ($query->result() as $row) {  
        echo $row->nombre; echo " ";  
        echo $row->dni;  
        echo "<br>";  
    }  
    echo 'Total Results: ' . $query->num_rows();  
} }
```


Query Builder: ejemplo array

```
function mostrar_persona() {  
    $this->load->database();  
    $query = $this->db->query('SELECT nombre, dni  
FROM persona');  
    foreach ($query->result_array() as $row){  
        echo $row['nombre']; echo " ";  
        echo $row['dni'];  
        echo "<br>";  
    }  
    echo 'Total Results: ' . $query->num_rows();  
} }
```

Query builder query: detalles

Importante:

- `query()` devuelve un objeto, `TRUE` o `FALSE`, no el resultado de la consulta, eso se obtiene a partir del objeto con `result()` o `result_array()`.
- Esta pensada para usarse con `INSERT`, `DELETE` o `UPDATE`.
- Hay que controlar lo que le ponemos.

Ejemplo completo

- `blog_model.php`
- `Blog_controller.php`
- `Blog.php` (vista)
- Basados en la tabla **entries** (id, title, content) de la base de datos **test**.

Query builder query: error

Para controlar el error:

```
if ( ! $this->db->simple_query('
SELECT `example_field`
FROM `example_table`'))
{
    $error = $this->db->error();
    echo $error;
}
```

Query builder: resultados

Todo el resultado: `result_object()` (`result()`) y `result_array()`.

Línea a línea: `row_object()` (`row()`) y `row_array()`. Además...

- `$row = $query->row(numero_de_fila);`
- `$row = $query->first_row()`
- `$row = $query->last_row()`
- `$row = $query->next_row()`
- `$row = $query->previous_row()`

Query builder: Tarea

Utilizar el método **get()** de Query builder para hacer lo mismo.

```
$query = $this->db->get('table_name');
```

¿Esto quiere decir que no hace falta SQL?

Query builder: select más complejo

```
function consulta_compleja() {  
    $this->load->database();  
    $query = $this->db->query('SELECT nombre,  
dni FROM persona WHERE nombre="Gustavo  
Casañ");  
    foreach ($query->result() as $row) {  
        echo $row->nombre; echo " ";  
        echo $row->dni;  
        echo "<br>";  
    }  
}
```

Tarea: probar otras condiciones en el WHERE

Query builder: Tarea

Utiliza métodos de Query builder para realizar la consulta anterior.
Para empezar, ¿cuáles serían?

Ejercicio 14

- Crea un modelo para **mostrar** los datos de las personas en la tabla **persona** de la base de datos **test**. Crea el correspondiente controlador y vista y muéstralos en una tabla.
 - Nombre
 - DNI
 - Dirección
 - Fecha de Nacimiento
 - Teléfono
 - Email
- Tarea extra: utiliza css para darle buen aspecto.

Ejemplo: insert

```
$data = array(  
    'title' => 'My title',  
    'name' => 'My Name',  
    'date' => 'My date'  
);  
  
$this->db->insert('mytable', $data);
```

Ejercicio 15

- Crea un formulario para **introducir** los datos de una persona, validarlos e introducirlos en la tabla **persona** de la base de datos **test**:
 - Nombre
 - DNI
 - Dirección
 - Fecha de Nacimiento
 - Teléfono
 - Email

Ejemplo: actualizar

```
$this->db->set('name', $name);
```

```
$this->db->set('title', $title);
```

```
$this->db->set('status', $status);
```

```
$this->db->where('id', 2);
```

```
$this->db->update('mytable');
```

Ejercicio 16

- Crea un formulario para **modificar** los datos de una persona, validarlos e introducirlos en la tabla **persona** de la base de datos **test**:
 - Nombre
 - DNI
 - Dirección
 - Fecha de Nacimiento
 - Teléfono
 - Email

Ejemplo: varias tablas

```
function get_categories($id) {  
    $this->db->select('*');  
    $this->db->from('entries_categories');  
    $this->db->  
>where('entries_categories.entry_id = '.$id);  
    $this->db->join('entries', 'entries.id =  
entries_categories.entry_id');  
    $this->db->join('categories', 'categories.id =  
entries_categories.category_id');  
    $this->db->order_by("category", "asc");  
    return $this->db->get()->result();    }
```

Ejemplo: varias tablas

```
$this->db->select('u.*, c.company, r.description');
```

```
$this->db->from('users u, company c, roles r');
```

```
$this->db->where('c.id = u.id_company');
```

```
$this->db->where('r.permissions =  
u.permissions');
```

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

Ejercicio 17

- Crea un método que dada una persona, consulte las tablas **persona**, **ventas** y **productos** de la base de datos **test** (aseguraros que están relacionadas). Crea controlador y vista para, en una tabla, mostrar los siguientes campos:
 - Nombre de la persona
 - Fecha de la compra
 - Precio final cantidad*precio-descuento+IVA
 - Nombre del producto

Ejercicio 18: tenis

- En un ejercicio anterior creaste una página sobre el tenis. Ahora crea la base de datos adecuada (tabla usuarios con sus campos y tabla cargos con los campos id, nombre, cargo, fecha_inicio y fecha_final, que puede ser nulo) y conéctate a ella con CodeIgniter.
- Se deberá poder crear un nuevo usuario, comprobar si el usuario existe y dejarle acceder a otras páginas y mostrar la tabla de cargos a partir de la información en la BdD.

Ejercicio 18: tenis (continuación)

Primero:

- Añade la tabla **noticias** con los campos titulo (título de la noticia), fecha (fecha en la que se introdujo la noticia), texto (el texto de la noticia en sí), usuario (identificador del cargo que introdujo la noticia) y foto (un campo foto que puede ser nulo). Además de un identificador que será clave primaria, por supuesto.
- Un cargo tiene que tener asociado un usuario. Añade la clave externa adecuada.

Ejercicio 18: tenis (continuación)

Segundo:

- Una vez creada la base de datos, añada un enlace en la barra de navegación a una página web (vista) llamada noticias, en la que se mostrarán noticias relacionadas con el deporte. Estas noticias estarán extraídas de la base de datos, de la tabla noticias.
- Añade un botón para cambiar el orden en que aparecen las noticias, de más antigua a más nueva y viceversa.

Ejercicio 18: tenis (continuación)

Segundo (continuación):

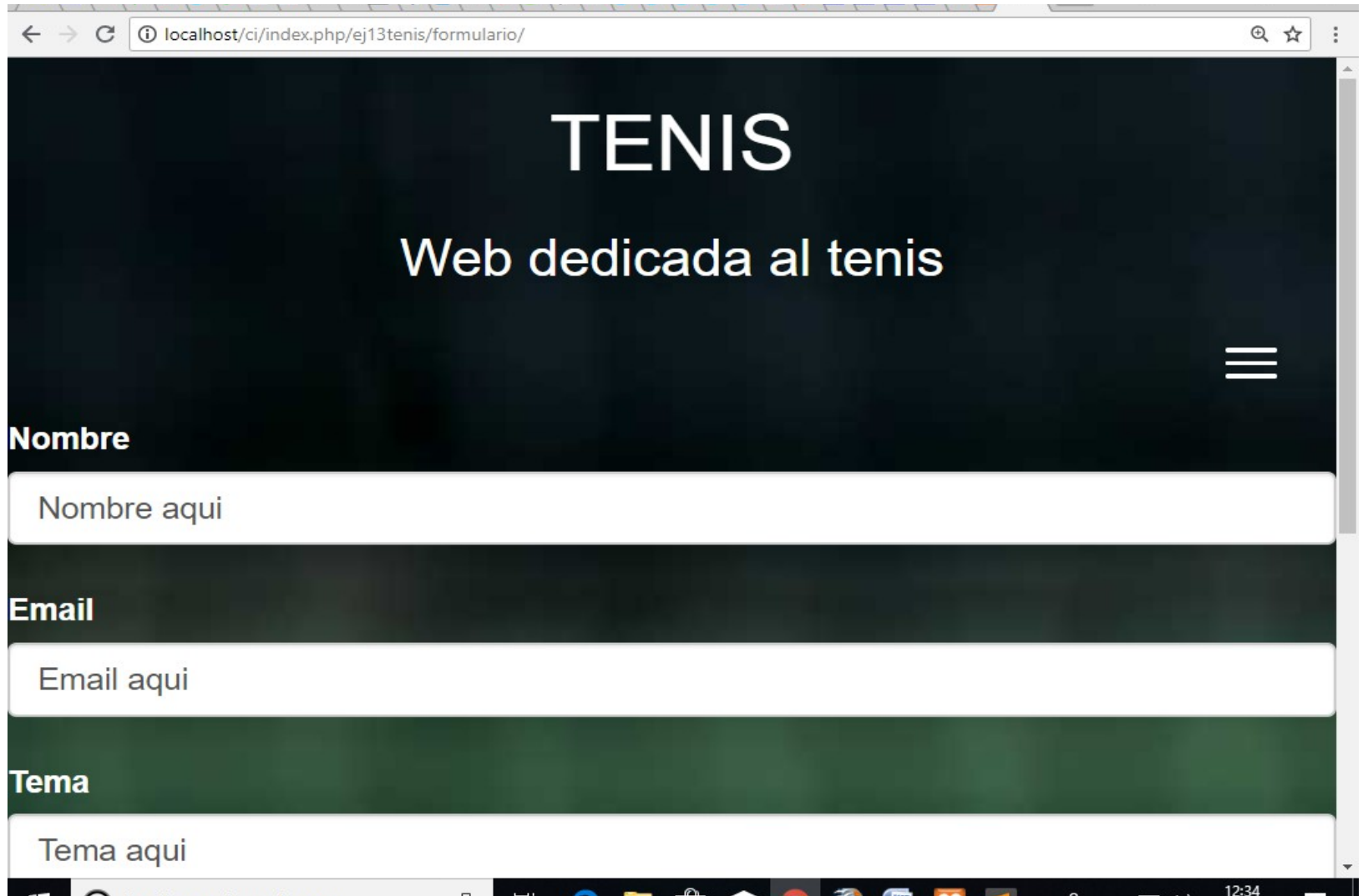
- Ya puestos, añade un filtro para localizar las noticias que te interesan.
- Crea una vista usuario que le permita al usuario identificado modificar sus datos, eliminarse como usuario del sistema y que tenga un enlace/botón para permitirle añadir noticias. También debe incluir un enlace botón que le muestre las noticias que él mismo ha introducido.

Ejercicio 18: tenis (continuación)

Segundo (continuación):

- Crea una vista formulario para añadir noticias a la base de datos. Debe permitir subir fotos y será sólo accesible a los usuarios identificados.
- El usuario que ha creado una noticia debe poder borrarla o modificarla, el resto no.

Ejercicio 18 (ejemplo)



A screenshot of a web browser displaying a registration form for a tennis website. The browser's address bar shows the URL `localhost/ci/index.php/ej13tenis/formulario/`. The page has a dark blue background with the word "TENIS" in large white letters and "Web dedicada al tenis" below it. A hamburger menu icon is on the right. The form consists of three sections: "Nombre" with a white input field containing "Nombre aqui", "Email" with a white input field containing "Email aqui", and "Tema" with a white input field containing "Tema aqui". The Windows taskbar is visible at the bottom with the time 12:34.

← → ↻ ⓘ localhost/ci/index.php/ej13tenis/formulario/ 🔍 ☆ ⋮

TENIS

Web dedicada al tenis

☰

Nombre

Email

Tema

12:34

Ejercicio 18 (ejemplo)

← → ↻ ⓘ localhost/ci/index.php/ej13tenis/quienesSomos/ 🔍 ☆ ⋮

TENIS

Web dedicada al tenis



Somos unos enamorados del tenis que llevamos 10 años dedicando toda nuestra energía y pasión a este magnífico deporte. En esta página podrás mantenerte al día de las competiciones y las últimas novedades de material.

Nombre	Cargo	Fecha incorporación
Gustavo Casañ	Presidente	2007
María de la O	Tesorerera	2014
Rodrigo Perez	Secretario	2017

Todos los derechos reservados - Posted by: Jordi - Contact information: mimail@example.com. 12:36:54 19-06-18

12:37

Ejercicio 18: tenis (ampliaciones)

1. Haz que las noticias aparezcan de tres en tres en la web, con botones de siguiente y anterior para moverse por la página.
2. Modifica el sistema para que un usuario normal no pueda añadir noticias, sólo un usuario que además tiene un cargo.
3. Añade lo necesario para que la administración de la base de datos (borrar usuarios, crearlos, cambiarles el password, etc...) pueda realizarse desde la web con un usuario con un cargo especial, administrador, que no debe aparecer en la tabla de cargos.

CI: subconsultas

- CodeIgniter NO contempla las subconsultas. Como hay versiones de SQL que no las permiten, ellos tampoco. Es decir, NO hay un método `$this->db->subquery()`
- Sin embargo, se pueden utilizar de una forma más o menos indirecta.

```
$query=$this->db->query('SELECT * FROM  
productos WHERE id NOT IN (SELECT productID  
FROM ventas)');
```

CI: subconsultas

```
public function subconsulta() {  
    $sub_query_from = '(SELECT id, product  
FROM product ) as product';  
    $this->db->select();  
    $this->db->from($sub_query_from);  
    $query = $this->db->get();  
    return $query->result();  
}
```

CI: subconsultas

```
public function subconsulta() {  
    $subquery = 'SELECT productID FROM  
ventas';  
    $this->db->select('*');  
    $this->db->where_not_in('id', $subquery);  
    $this->db->from('productos');  
    $query = $this->db->get();  
    return $query->result();  
}
```

Tarea: comprueba que esto funciona mostrando los resultados en una vista.

CI: subconsultas

```
public function subconsulta() {  
    $subquery = 'SELECT productID FROM  
ventas';  
    $this->db->select('*');  
    $this->db->where_not_in('id', $subquery);  
    $this->db->from('productos');  
    $query = $this->db->get();  
    return $query->result();  
}
```

Tarea: comprueba que esto funciona mostrando los resultados en una vista.

Convertir estas consultas a QB I

- `INSERT INTO coleccion(nombre, precio, IVA) VALUES ('Literatura', '12.45', '1.34');`
- `SELECT CONCAT(a.nombre, " ", a.apellidos) AS Autor, " es el autor de ", l.titulo AS Titulo FROM autor a, libro l, autor_libro au WHERE a.id=au.id_autor AND l.id=au.id_libro ORDER BY 1, l.titulo DESC`
- `SELECT l.titulo, CONCAT (a.nombre, " ", a.apellidos) AS Autor, FORMAT((c.precio + 10 + ROUND(RAND() * 10)),2) AS "Precio Final" FROM autor a, libro l, autor_libro au, coleccion c WHERE a.id=au.id_autor AND au.id_libro=l.id AND l.id_coleccion=c.id`

Convertir estas consultas a QB II

- `SELECT p.descripcion, "No activa" FROM promocion p
WHERE p.est_activo=0`

`UNION`

`SELECT p.descripcion, l.titulo FROM libro l, promocion p
WHERE l.id_promocion=p.id AND p.est_activo=1`

Realizar estas consultas en QB II

- Para la base de datos ENI, crea una función que nos devuelva los nombres de los autores y los temas que tienen sus libros.
- Para la base de datos ENI, crea una función que nos devuelva las promociones y los títulos de los libros que pertenecen a ellas. Y ya puestos, el precio y el IVA de cada libro (tendrás que cogerlo de la tabla coleccion).

Ejercicio 19

- Vamos a crear una web sobre la ciudad de Granada. La información podéis extraerla de <https://es.wikipedia.org/wiki/Granada>.
- Para facilitar la creación se aporta:
 1. un controlador (incompleto, que habrá que completar, fichero ejercicio_trozos.php)
 2. un fichero cabecera (que no se modificará excepto para añadir css, fichero cabecera_ejercicio.php)
 3. un fichero de pie (que no se modificará pie_ejercicio.php)

Ejercicio 19 (continuación)

- Habrá una página de entrada de usuarios sencilla (usuario/password) con posibilidad de crear un nuevo usuario.
- Añadir un mapa de google maps a la página, por ejemplo en región, que muestre una vista aérea de la provincia.

Fin...

- Aunque hay muchas cosas de CodeIgniter que no se han mencionado...