CodeIgniter
Parte I: estático

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

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

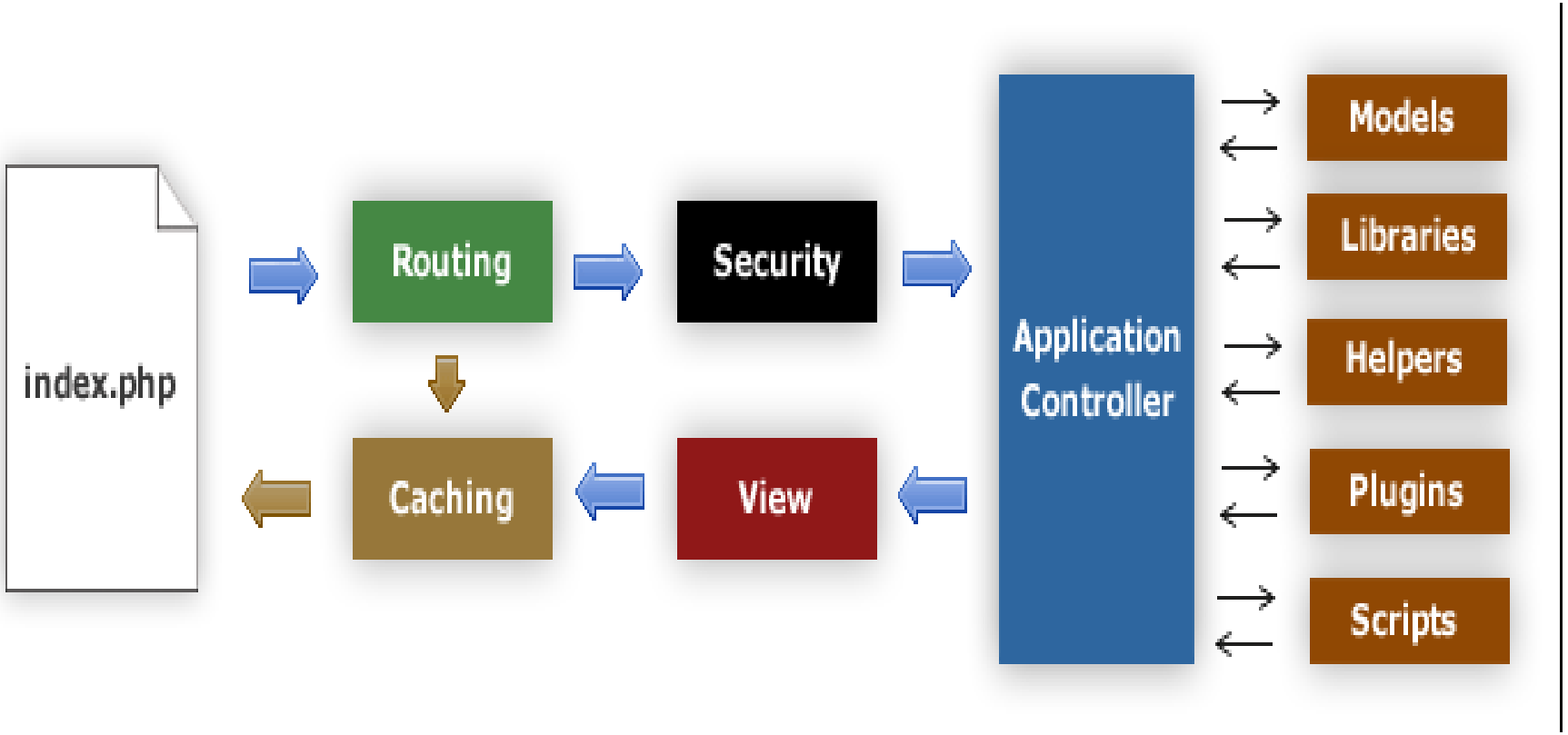
- **Helpers**: agrupan funciones de una categoría en especial: URL Helpers, que ayudan a crear enlaces, Form Helpers que ayudan a crear formurios, Text Helpers que realizan varias rutinas de texto, Cookie Helpers crean y leen, File Helpers...

```
$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



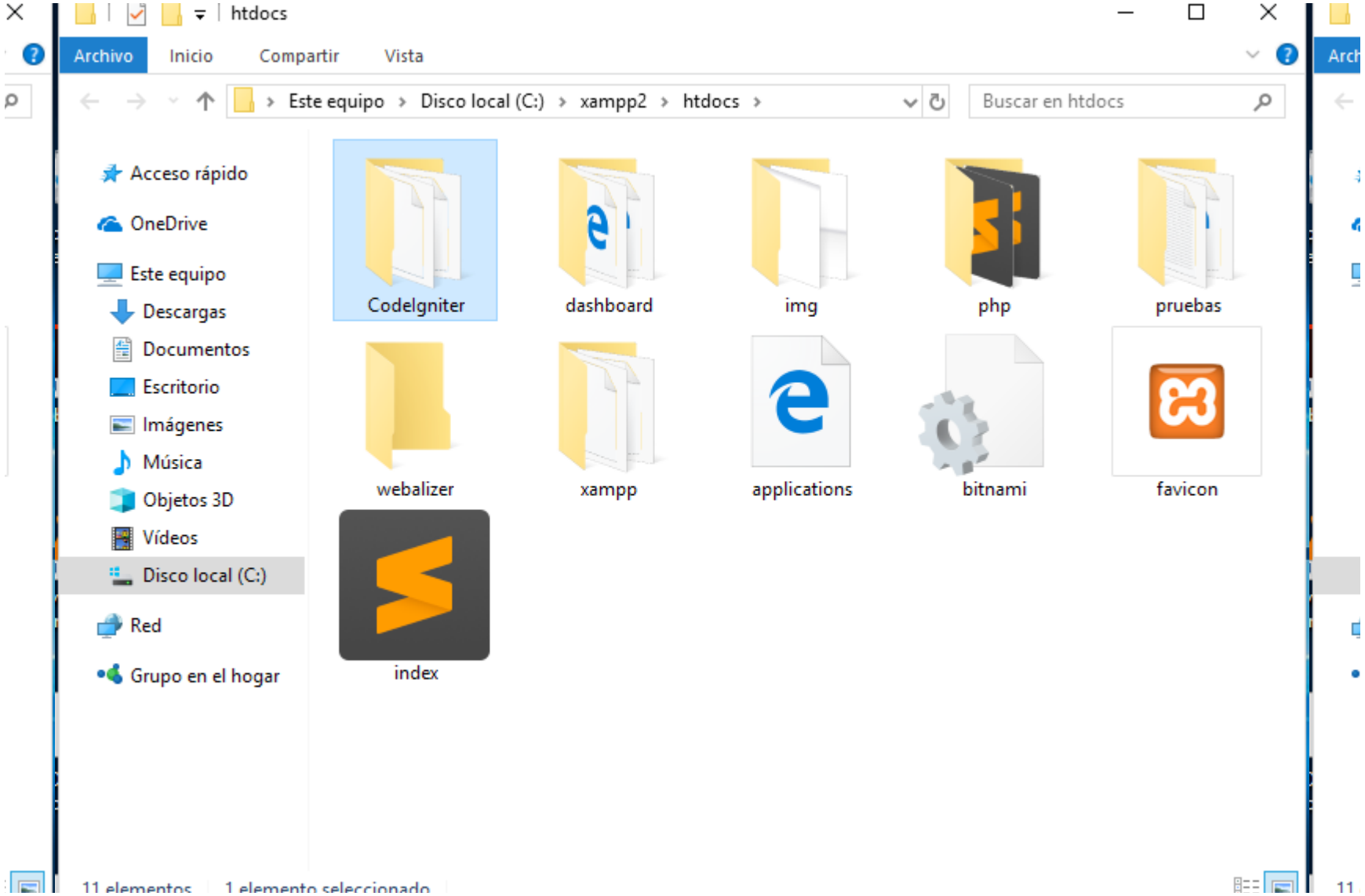
Flujo en una aplicación

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.

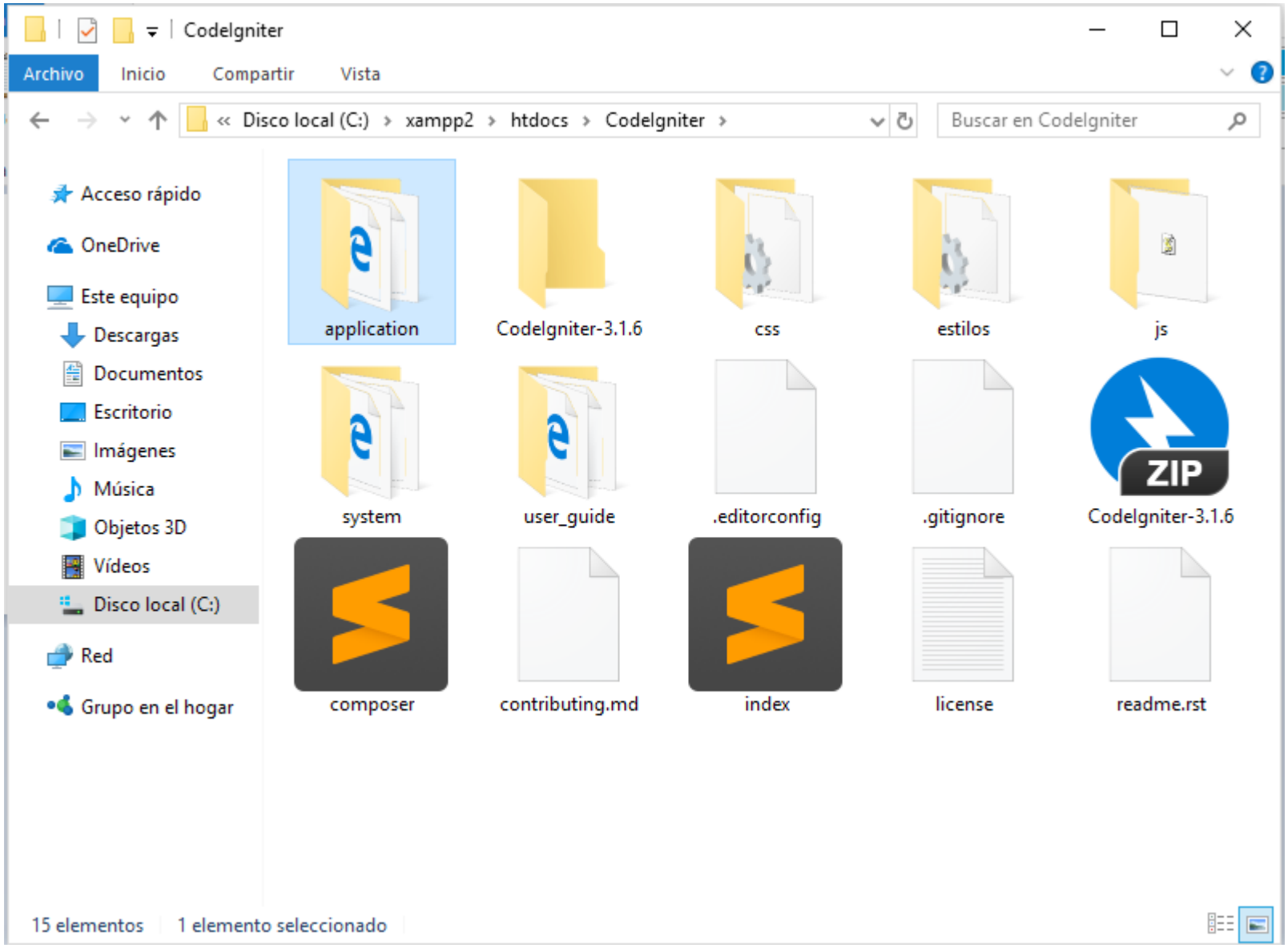
Instalar CodeIgniter

- 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

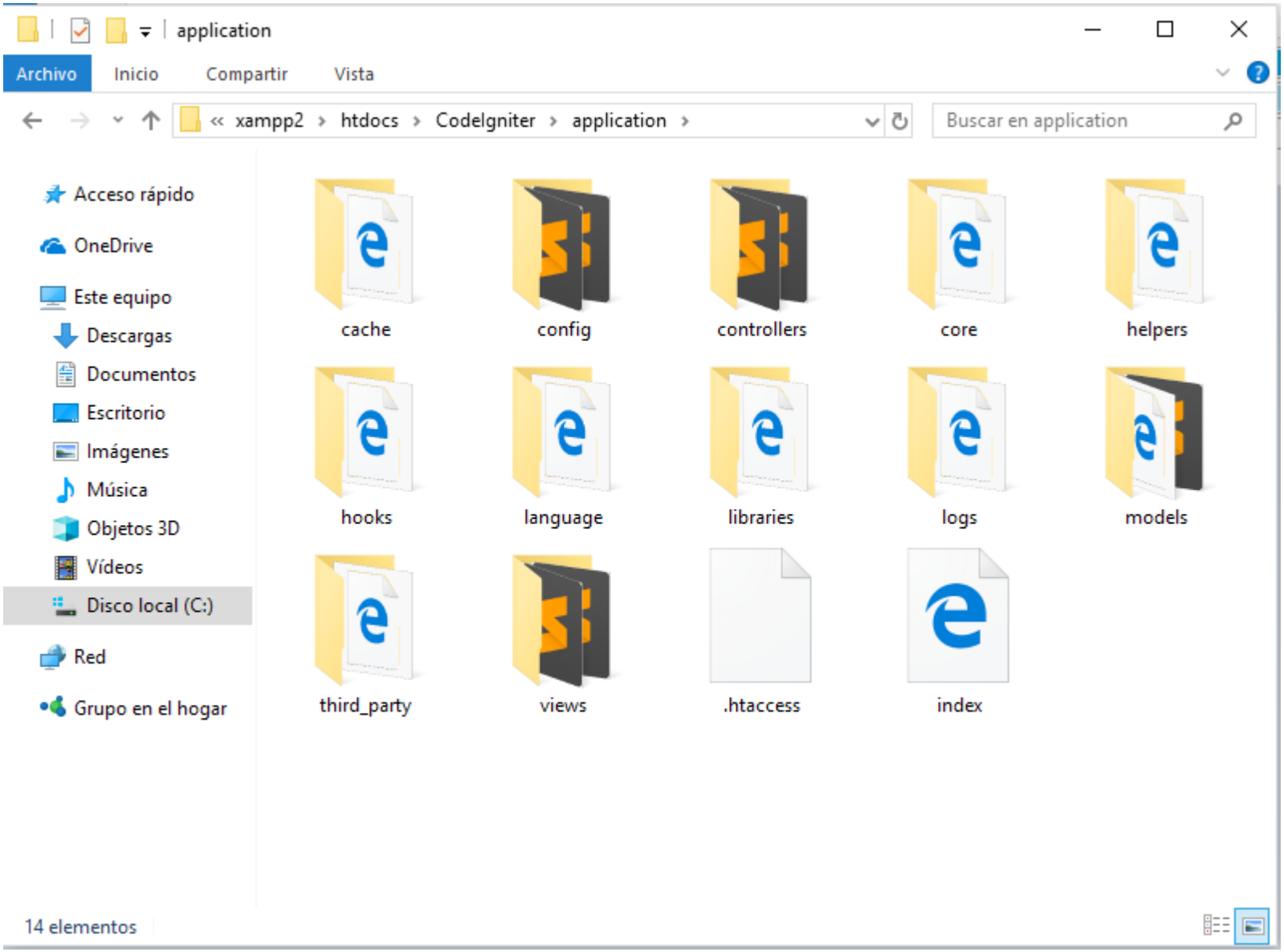
Mi carpeta root queda algo así:



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.

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.

Ejemplo: página estática

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

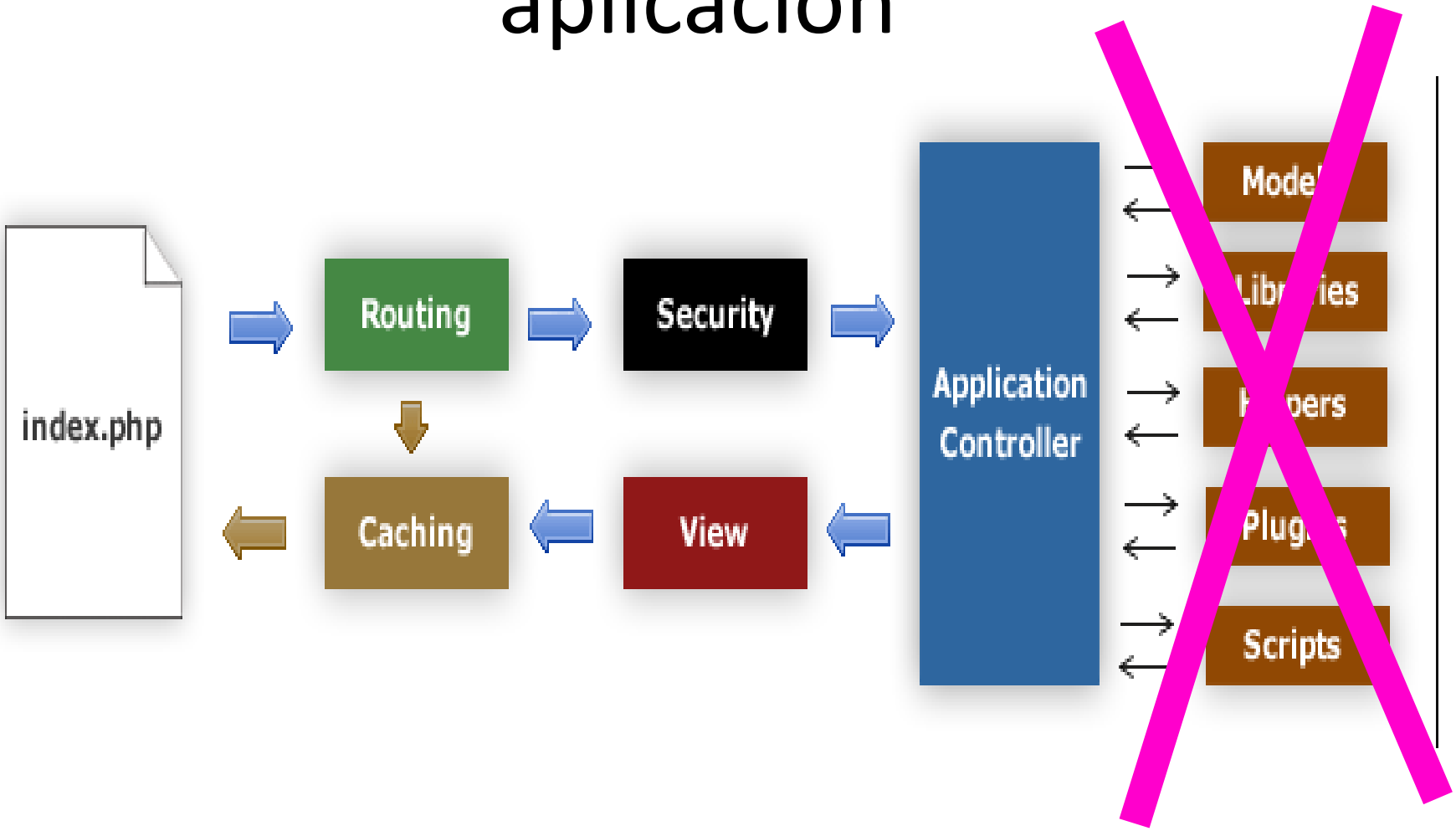
- Las funciones dentro del fichero se interpretan como páginas.

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 despida.

Diagrama de flujo en una aplicación



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 _meEscondo($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.

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

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

```
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

- Crea un formulario para introducir los siguientes datos de una persona y validarlos:
  - Nombre
  - DNI
  - Dirección
  - Fecha de Nacimiento
  - Teléfono

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.

Ejercicio 12: Form^2

- 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.