



Manual CodeIgniter

Antonio Roldán Andrade

Marzo 2022

Índice

1. Instalación CodeIgniter	2
1.1. Preparación del Entorno	2
1.2. Configuración CodeIgniter en XAMPP	2
2. Configuración de la Base de Datos	4
2.1. Creación de la Base De Datos	4
2.2. Acceso a la Base De Datos	4
3. Configuración Apache	5
3.1. Habilitando Mod Rewrite en XAMPP	5
4. Creacion del Primer controlador	5
5. Views en CodeIgniter	6
5.1. Creación de Una vista	6
5.2. Llamando a las vistas	6
6. Creación de Modelos	7
6.1. Configuración de La base de datos	7
6.2. Acceso al modelo	7
7. Creación de la vista overview	9
8. Creación de Artículos	9
8.1. Enrutamiento de creaarticulo.php	9
8.2. Creación de la vista	10
9. Autenticación de Usuarios	11
9.1. Vista Login view	11
9.2. Vista Register view	12
9.3. Vista intermedia	13
10. Notas y apuntes	13
11. Problemas Enfrentados	14
11.1. Problemas con AppServe	14
11.2. Problemas De Routing	14
11.3. Problema Con Overview	14
12. Conclusiones	14

1. Instalación CodeIgniter

1.1. Preparación del Entorno

Para iniciar con la instalación de CodeIgniter necesitaremos usar alguna aplicación que contenga servicios como Apache,MySQL-Server, etc.

Usaremos XAMPP, permite tener las mismas herramientas requeridas, herramienta previamente usada en esta asignatura, pero puede dar problemas con el uso de CodeIgniter (comentado en la clase del día 9 de marzo de 2022, por el profesor).

Para realizar la instalación simplemente iremos a la página de descarga ofrecida en su servicio web e instalaremos el software.

Descarga del software

1.2. Configuración CodeIgniter en XAMPP

Tras la descarga de CodeIgniter en su página web simplemente deberemos colocarnos en el directorio raíz, de forma predeterminada se encuentra en *C:/xampp* de AppServe, tras esto nos moveremos dentro de la carpeta *htdocs*. Una vez dentro simplemente crearemos una carpeta donde copiaremos el contenido del archivo descargado de Codeigniter para poder empezar con la creación de la URL .

Para corregir los errores previamente mencionados en la sección anterior deberemos modificar la configuración de PHP del XAMPP, esto lo haremos modificando el archivo *PHP.ini*, simplemente añadiremos al mismo la orden *extension=intl*, esto permite a CodeIgniter utilizar las herramientas necesarias para su correcto funcionamiento.

```

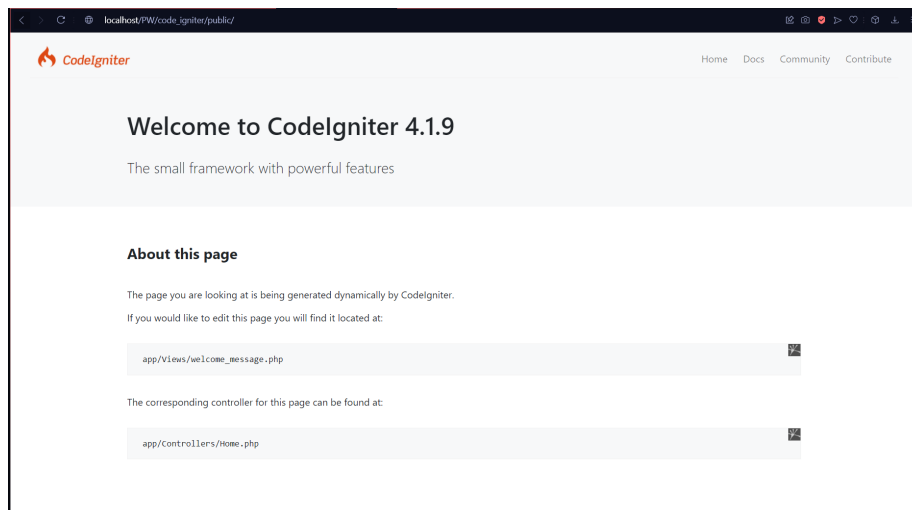
10 * -----
11 * Base Site URL
12 * -----
13 *
14 * URL to your CodeIgniter root. Typically this will be your
15 * WITH a trailing slash:
16 *
17 *   http://example.com/
18 *
19 * If this is not set then CodeIgniter will try guess the pro
20 * and path to your installation. However, you should always
21 * explicitly and never rely on auto-guessing, especially in
22 * environments.
23 *
24 * @var string
25 */
26 public $baseURL = 'http://localhost/code_igniter/';
27
28 /**
29 * -----
30 * Index File
31 * -----
32 *

```

Instalacion CodeIgniter

A continuación nos moveremos al archivo *App.php* donde editaremos la "BaseUrl", encargada de gestionar la dirección donde se aloja la app, y añadiremos la ruta donde "hostea" XAMPP.

Por último para que todo funcione de forma correcta necesitaremos, además de tener instalado PHP de forma nativa, Composer, en cualquier otro caso no funcionará de forma correcta.

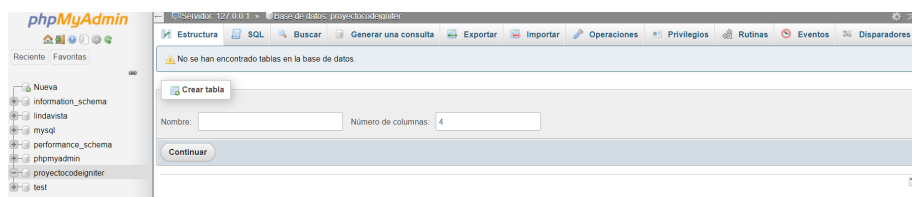


Portada Inicial CodeIgniter

2. Configuración de la Base de Datos

2.1. Creación de la Base De Datos

La tabla de base de datos la crearemos con PHPMyAdmin, usaremos el proporcionado por XAMPP, simplemente iremos al "DashBoardz crearemos de forma sencilla la tabla *ProyectoCodeIgniter*.



Base de Datos en PhpMyAdmin

2.2. Acceso a la Base De Datos

Para realizar la configuración deberemos irnos al archivo *database.php*, aquí le indicaremos que credenciales debe usar para realizar la conexión con la base de datos. En nuestro caso deberemos acceder a la dirección *127.0.0.1*, localhost, a la base de dato creada en el apartado anterior, con el nombre "ProyectoCodeIgniter". Para acceder usaremos el usuario previamente creado *codeigniterz* su contraseña, *codeigniter*.

```
33 public $default = [  
34     'DSN'      => '',  
35     'hostname' => 'localhost',  
36     'username' => 'codeigniter',  
37     'password' => 'codeigniter',  
38     'database' => 'proyectoCodeIgniter',  
39     'DBDriver' => 'MySQLi',  
40     'DBPrefix' => '',  
41     'pConnect' => false,  
42     'DBDebug'  => (ENVIRONMENT !== 'production'),  
43     'charset'  => 'utf8',  
44     'DBCollat' => 'utf8_general_ci',  
45     'swapPre'  => '',  
46     'encrypt'  => false,  
47     'compress' => false,  
48     'strictOn' => false,  
49     'failover' => [],  
50     'port'     => 3306,  
51 ];
```

3. Configuración Apache

Para una mayor simplificación con el sistema de enrutamiento en el manual proporcionado debemos modificar el sistema de configuración del servidor 'Apache', así evitaremos que las rutas de la página creada tengan de forma obligatoria *index.php* en su URL.

Esto se hace habilitando la extensión 'mod rewrite' de Apache, en mi caso, el servidor Apache es gestionado por XAMPP, cambiando ligeramente como deberemos habilitar la extensión.

3.1. Habilitando Mod Rewrite en XAMPP

Este paso lo realizaremos de forma sencilla dirigiéndonos al directorio donde se encuentra localizado el archivo *httpd.conf*, en mi caso se encuentra en la ruta *C:/xampp/apache/conf*.

Una vez localizado el archivo lo abriremos con el editor de texto que prefiramos y modificaremos/añadiremos la línea *LoadModule rewrite_module modules/mod_rewrite.so*, esta se encuentra comentada con *#*, simplemente eliminaremos ese carácter, tras esto buscaremos todas las ocurrencias de *AllowOverride none* y las cambiaremos por *AllowOverride All* permitiendo así la extensión.

4. Creacion del Primer controlador

Continuando con el tutorial debemos crear un controlador que muestre un mensaje de bienvenida, para así poder atender a la petición de la URL *localhost/proyecto/Articulo/view/3*.

Para esto simplemente usaremos el código proporcionado, quedando de la siguiente forma el archivo *Articulo.php* creado anteriormente.

```

<?php

use CodeIgniter\Controller;

class Artículo extends Controller{

    public function index(){
        return view('welcome_message');
    }

    public function view($page = 'home')
    {
        if (! is_file(APPPATH.'/Views/articulo/'.$page.'.php')){

            //Error Whoops we dont have a page
            throw new \CodeIgniter\Exceptions\PageNotFoundException($page);
        }

        $data['title'] = ucfirst($page);//Pone en mayuscula la primera letra

        echo view('templates/header', $data);
        echo view('articulo/'.$page, $data);
        echo view('templates/footer', $data);

    }

}

```

Archivo.php

5. Views en CodeIgniter

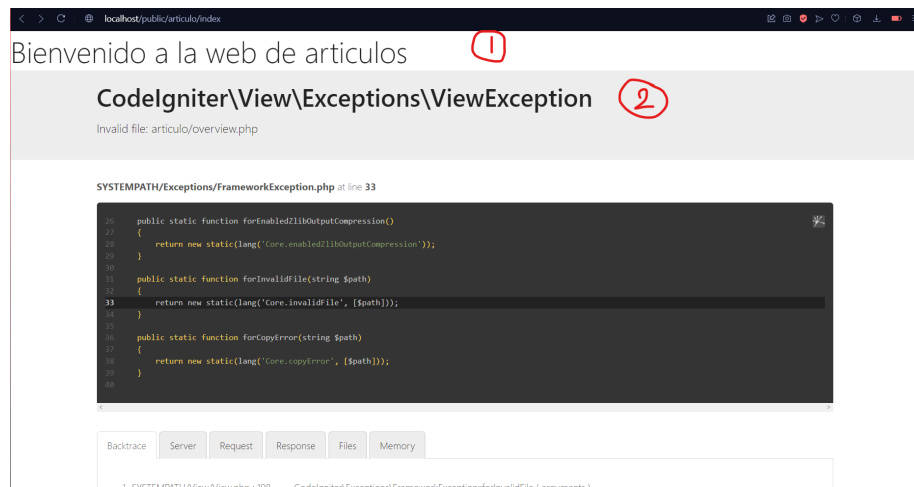
5.1. Creación de Una vista

La creación de una vista se realizará de la siguiente forma, cada vez que necesitemos hacer esto debemos movernos a la carpeta 'views', cuyo path es *App/Views*. Una vez dentro crearemos una carpeta por cada uno de los módulos que requiramos, en este caso necesitamos el módulo artículo, por lo que crearemos una carpeta con el mismo nombre, y dentro de la misma los archivos requeridos para el modulo, en este caso *header.php*, *home.php* y *footer.php*.

5.2. Llamando a las vistas

Realizar este apartado es más bien simple, debemos crear una función *index* donde llamaremos a la misma, usando la siguiente dirección:
localhost/public/articulo/[nombre-view].

Tenemos un pequeño problema con la view *overview*, esta aun no ha sido creada (se crea mas adelante), generando a consecuencia de esto una excepción en una de las llamadas que componen a la página web.



- 1) Corresponde al header de la página
- 2) Corresponde a overview.php (rompe la ejecución)

6. Creación de Modelos

6.1. Configuración de La base de datos

Simplemente nos moveremos al archivo `.env` e introducimos la configuración correspondiente a la base de datos que deseamos acceder, en la sección correspondiente.

```
#-----
# DATABASE
#-----

database.default.hostname = localhost
database.default.database = test
database.default.username = root
database.default.password = 
database.default.DBDriver = MySQLi
database.default.DBPrefix = 

# database.tests.hostname = localhost
# database.tests.database = ci4
# database.tests.username = root
# database.tests.password = root
# database.tests.DBDriver = MySQLi
# database.tests.DBPrefix =
```

Archivo de configuración

6.2. Acceso al modelo

Para realizar el acceso al modelo y poder usarlo en los controladores de CodeIgniter debemos crear cada uno de los modelos con las siguientes pautas:

1. Ficheros de modelo deben empezar en mayúscula y contener Model en su nombre.
2. Deben tener en su cabecera *namespace APP/Models*, además de *use CodeIgniter/Model*.
3. En el controlador deseado debemos incluir el modelo deseado mediante *App/Models/[nombre]*.

Quedando de la siguiente forma:

```
<?php

namespace App\Models;

use CodeIgniter\Model;

class ArtículoModel extends Model
{
    protected $table = 'articulo';

    protected $allowedFields=['title','descripcion','cuerpo'];

    public function getArticulos()
    {
        return $this->findAll();
    }
}
```

Modelo configurado

```
1 <?php
2
3 namespace App\Controllers;
4 use CodeIgniter\Controller;
5
6 use App\Models\ArticuloModel;
7
8 class Articulo extends Controller{
9
10     public function index(){
11
12         $model = new ArticuloModel();
13
14         $data = [
15             'articulos' => $model->getArticulos(),
16             'title'=>'Bienvenido a la web de articulos'
17         ];
18         echo view('templates/header',$data);
19         //echo view('articulo/overview',$data);
20         echo view('templates/footer',$data);
21     }
22 }
```

Controlador configurado

7. Creación de la vista overview

Crearemos la vista para que en la bienvenida de la página y se muestren los artículos de la base de datos, quedando de la siguiente forma:

```
<p><a href="/articulo/create">Create article</a></p>

<?php if(! empty($articulos) && is_array($articulos)) : ?>

    <?php foreach($articulos as $articulos): ?>
        <h3><?php esc($articulos_item['item']); ?></h3>

        <div class="main">
            <?php
                esc($articulos_item['cuerpo']);
            ?>
        </div>

        <p><a href="/articulo/<?php esc($articulos_item['id'],'url'); ?>">View article</a></p>
    <?php endforeach; ?>

<?php else: ?>

    <h3>No articulo</h3>
    <p>No hay articulos</p>
<?php endif ?>
```

overview.php

Quedando la vista en el navegador de la siguiente forma:



8. Creación de Artículos

8.1. Enrutamiento de creaarticulo.php

Tras la creación de la vista encargada de crear objetos articulo en la base de datos, debemos enrutarla para que CodeIgniter pueda encontrarla y mostrarla por pantalla, para esto volvemos al fichero *routes.php* y añadimos la nueva ruta.

```
// We get a performance increase by specifying the default
// route since we don't have to scan directories.
$routes->get('/', 'Home::index');
$routes->match(['get', 'post'], 'articulo/create', 'Articulo::create');
$routes->get('articulo/{:segment}', 'Articulo::view/$1');
$routes->get('articulos', 'Articulo::index');
```

Estado final routes.php

8.2. Creación de la vista

Usaremos esta vista para añadir elementos *articulo* a la base de datos de nuestra página web.

Para un correcto funcionamiento es recomendable crear además una vista auxiliar que muestre un mensaje de error e inserte el nuevo artículo en la base de datos.

Quedando la vista de la siguiente forma:



Creación de un artículo

Titulo

Descripción

Cuerpo

Create articulo item

© 2021

Vista Creación de artículo

9. Autenticación de Usuarios

Para la correcta autenticación de los usuarios e implementación de un sistema de registro y login.

Para el correcto funcionamiento del acceso a través del login debemos forzar al usuario a pasar por el, esto se realiza mediante la modificación de la ruta base (explicado en la pagina 38 del manual).

Obviamente para que esta colección de vistas y métodos funcionen debemos crear un nuevo modelo, el modelo *UserModel*, tras esto, crearemos el controlador encargado de manejar todos las vistas relacionadas con los usuarios.

9.1. Vista Login view

La utilidad de esta vista será la de permitir a cuyo usuario tenga ya una cuenta registrada en el sistema de base de datos.

Código:

```
app > Views > usuario > login_view.php
1  <?= \Config\Services::validation()>listErrors(); ?>
2  <form action="/articulo/create" method="post">
3      <?= csrf_field() ?>
4      <label for="user">Usuario</label>
5      <input type="input" name="user" /><br/>
6      <br></br>
7      <label for="pass">Contraseña</label>
8      <input type="input" name="pass" /><br />
9      <br></br>
10     <input type="submit" name="submit" value="Entrar" />
11     <p><a href="/usuario/create">Registrarse</a></p>
12 </form>
```

loginview.php

Vista:

Login usuario

Usuario

Contraseña

[Registrarse](#)

© 2022

Vista Login Usuario

9.2. Vista Register view

La vista es usada para permitir el registro a los usuarios de esta web registrarse, para acceder a posteriori a la página web.

Código:

```
app > Views > usuario > register_view.php
1  <?php \Config\Services::validation()->listErrors(); ?>
2  <form action="/usuario/create" method="post">
3      <?php csrf_field() ?>
4      <label for="name">Nombre</label>
5      <input type="input" name="name" /><br />
6      <br></br>
7      <label for="correo">Correo</label>
8      <input type="input" name="correo" /><br />
9      <br></br>
10     <label for="user">Usuario</label>
11     <input type="input" name="user" /><br />
12     <br></br>
13     <label for="pass">Contraseña</label>
14     <input type="input" name="pass" /><br />
15     <br></br>
16     <label for="pass2"> Repetir Contraseña</label>
17     <input type="input" name="pass2" /><br />
18     <br></br>
19     <input type="submit" name="submit" value="Create usuario item" />
20     <p><a href="/">Inicio Sesión</a></p>
21 </form>
```

registerview.php

Vista:



Vista Registro de usuarios

9.3. Vista intermedia

Sirve para comprobar que se ha iniciado sesión de forma correcta en el sitio
Código:

```
app > Views > usuario > entrar_view.php
1
2 <p><a href="/articulo/create">Crea Un articulo</a></p>
3
4 <p><a href="/articulos">Mostrar articulos</a></p>
```

entrarview.php

10. Notas y apuntes

Con respecto a las capturas de pantalla, en el footer se muestra el símbolo de copyright del año 2021, a pesar de que se ha realizado en 2022, esto se debe a que cuando se hubieron hecho las capturas no se cambio correctamente el año actual, por esto las capturas iniciales se ven con 2021 y las ultimas con 2022, además como mencioné anteriormente he realizado múltiples páginas con CodeIgniter para entender y mejorar el tutorial, dando pie a este tipo de *fallos*, siento la molestia.

Algunos de los fallos no se encuentran en el apartado de Problemas, ya que parte de ellos se han ido subsanando conforme se realizaba el ejercicio, careciendo de sentido repetir los mismos en la sección anteriormente mencionada.

11. Problemas Enfrentados

11.1. Problemas con AppServe

El problema principal con AppServe es que no podía abrirlo de forma correcta, ya que la configuración de puertos coincidía con mi instalación de XAMPP, para solucionar esto decidí simplemente no utilizar esta aplicación y seleccionar XAMPP, ya que con esta estoy mas familiarizado (tanto por la universidad como trabajos externos, cursos, ...), cambiando ligeramente el tutorial para poder realizar correctamente lo solicitado.

Sobre todo tuve problemas con la selección del directorio raíz de XAMPP, para cambiarlo tuve que modificar el archivo *httpd.conf* y colocar la raíz del mismo en /public de CodeIgniter y no en la predeterminada de httdocs.

11.2. Problemas De Routing

Otro de los problemas mas difíciles de subsanar fue el problema derivado de las rutas, ya que no era capaz de conseguir que el entorno encontrara los modelos creados por mi, la solución una vez la ves es muy simple, pero al nunca haber trabajado con este framework sobreentendía que los modelos creados no necesitaban ser incluidos manualmente. La solución fue como se espera incluirlos en el controlador mediante la linea *use App/Models/[modelo a incluir];* , tras esto se elimina el error de inclusión del controlador.

11.3. Problema Con Overview

Este problema mas bien fue un error de lectura, aunque a mi parecer no se indica lo suficientemente claro.

En la función *index* del controlador usuario existe una función denominada overview, esta es la encargada de mostrar por pantalla los elementos de tipo articulo existentes en la base de datos, el problema reside en que se llama a la vista en una pagina mucho anterior a aquella donde se muestra la creación de la misma, creando así un poco de confusión, cosa que a mi al menos me costo ver, ya que la captura de pantalla da a entender que en la primera llamada que hacemos al índice debe funcionar y esto no es así.

12. Conclusiones

Este tutorial ha conseguido que tras múltiples intentos infructuosos con la configuración del entorno (en mi caso XAMPP) y bastantes cambios en la estructura de las vistas, por falta de atención propia, he conseguido entender de forma sencilla la configuración e implementación de webs usando el entorno dado por el FrameWork, el cual nos presenta muchas ventajas con respecto a PHP puro.

Espero que haya sido amena la lectura de mi tutorial, gracias por la atención =)