UNIVERSIDAD DE SONSONATE

Implantación de Sistemas

Guía 4 –Introducción a CodeIgniter

Instructor: Ing. David Gildardo Rajo

Facultad de Ingeniería y Ciencias Naturales

Contenido

CODEIGNITER.

CodeIgniter es un framework para desarrollo de aplicaciones - un conjunto de herramientas - para gente que construye sitios web usando PHP. Su objetivo es permitirle desarrollar proyectos mucho más rápido que lo que podría hacer si escribiera el código desde cero, proveyéndole un rico conjunto de bibliotecas para tareas comunes, así como y una interfaz sencilla y una estructura lógica para acceder a esas bibliotecas. CodeIgniter le permite enfocarse creativamente en su proyecto al minimizar la cantidad de código necesaria para una tarea dada.

CUANDO UTILIZAR UN FRAMEWORK COMO CODEIGNITER.

CodeIgniter es un framework que se debe de utilizar en las siguientes circunstancias, si usted como desarrollador:

• Necesita un desempeño excepcional.

• Necesita amplia compatibilidad con cuentas estándar de alojamiento que corren una variedad de versiones de PHP y configuraciones.

• Necesita un framework que casi no necesite configuración.

• Necesita un framework que no le obligue a usar la línea de comandos.

• Necesita un framework que no le obligue a adquirir reglas de codificación restrictivas.

• No está interesado en bibliotecas monolíticas de gran tamaño como PEAR.

• No quiere verse forzado a aprender un lenguaje de plantillas (aunque hay un motor de plantillas disponible si desea uno).

• Evita la complejidad, favoreciendo las soluciones simples.

• Necesita una documentación clara y completa.

INSTALACION DE CODEIGNITER.

Para la instalación de CodeIgniter solo se necesitan realizar los siguientes pasos:

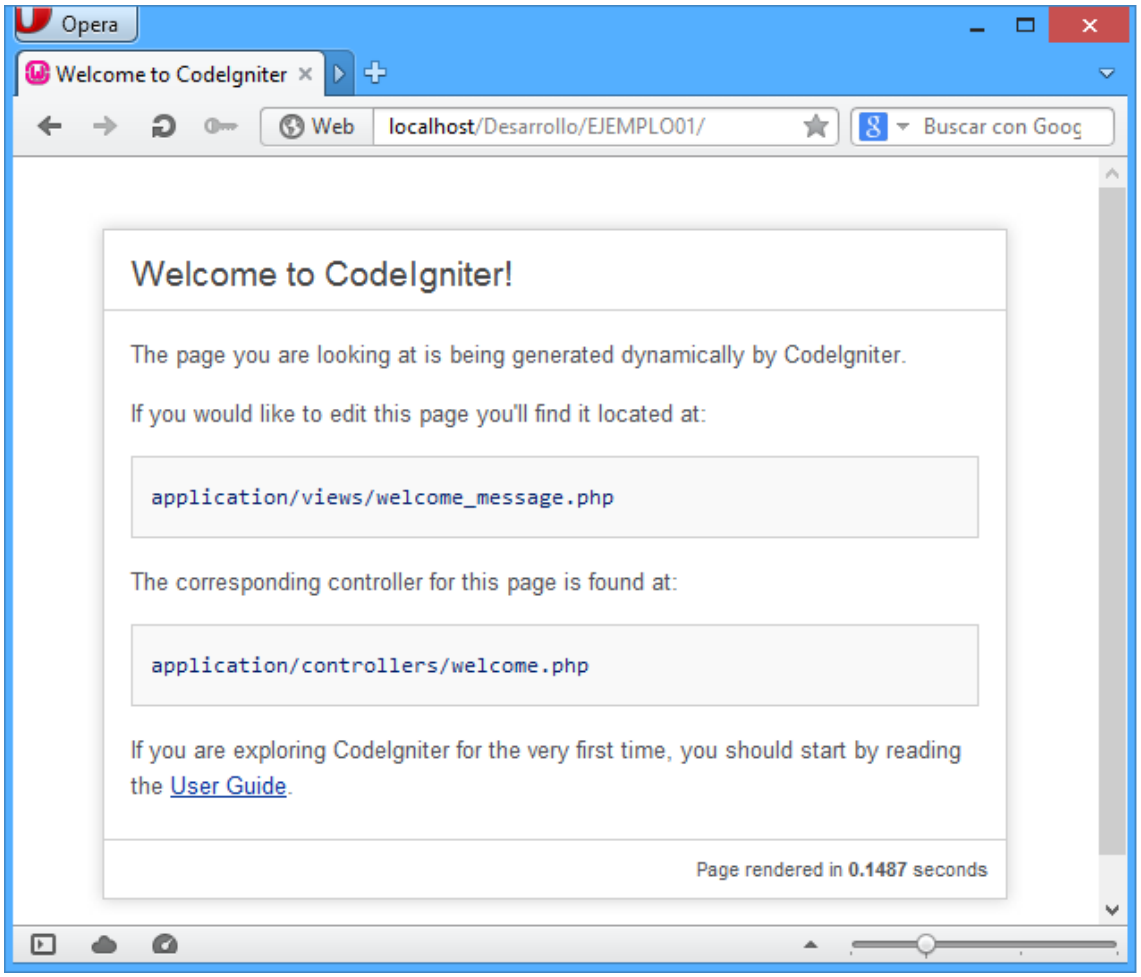
• Descomprima el paquete que contiene el Framework CodeIgniter.

• Suba las carpetas y archivos de CodeIgniter al servidor. Normalmente el archivo ***g s*** será la raíz.

• Si tiene intención de usar una base de datos, abra el archivo ***ffi ffi g ffiffi ffi s*** con un editor de texto y establezca los parámetros de la base de datos.

Cada uno de estos pasos se debe realizar para cada proyecto que se va a realizar, ya que provee un conjunto de bibliotecas para tareas comúnmente necesarias, tanto como una interfaz sencilla y una estructura lógica para acceder a esas bibliotecas.

En nuestro caso que utilizamos una aplicación como *J* solo necesitamos colocar los archivos de CodeIgniter en la carpeta en la que vayamos a desarrollar un proyecto, por ejemplo, si tuvieras un proyecto *W* y WAMP se encuentra instalado en nuestro disco raíz(C:\), la ruta donde copiaríamos dichos archivos seria *fiti W* Si todo se ha realizado de forma correcta e ingresamos a la dirección de nuestro servidor web se debería de desplegar la siguiente imagen:



COMO FUNCIONA CODEIGNITER.

CodeIgniter usa el enfoque Modelo-Vista-Controlador, que permite una gran separación entre la lógica y la presentación. Es particularmente bueno para proyectos en los que los diseñadores trabajan en sus archivos de plantillas, ya que el código en estos archivos será mínimo.

Cuando ejecutamos una aplicación o proyecto, como *W* , se realizan los siguientes procesos dentro de la aplicación:

1. El index.php sirve como el controlador frontal, inicializando los recursos básicos que necesita CodeIgniter para ejecutar.

2. El Ruteador examina la solicitud HTTP para determinar que debería hacer con ella.

3. Si existe el archivo de caché, se lo envía directamente al navegador, sin pasar por la ejecución normal del sistema.

4. Seguridad. Antes que se cargue el controlador de la aplicación, por razones de seguridad se filtran la solicitud HTTP y cualquier otro dato enviado por el usuarios.

5. El controlador carga el modelo, las bibliotecas del núcleo, helpers, y cualquier otro recurso requerido para procesar una solicitud específica.

6. La Vista terminada se procesa y se envía al navegador para que se pueda ver. Si el caché está habilitado, la vista se cachea primero para que las siguientes solicitudes que la necesiten puedan ser servidas.

Después de conocer todo el funcionamiento que realiza CodeIgneter tras ejecutar una aplicación, tendremos un acercamiento al uso de los controladores, para lo cual nos apoyaremos con el siguiente ejemplo.

EJEMPLO 1

Para nuestro primer ejemplo nos basaremos en la creación de una carpeta **EJEMPLO01,** y como se mencionó a un inicio copiaremos el CodeIgniter en dicha carpeta, un proyecto de CodeIgniter consta de las siguientes carpetas Importantes:

Carpeta	Descripción
Config	Almacena los archivos de configuración para el framework, donde se pueden habilitar ciertas librerías.
controller	Almacena todo los controladores del proyecto.
models	Almacena los archivos donde se define la lógica del negocio.
views	Almacena las vistas del proyecto.

En el ejemplo que desarrollaremos, será un proyecto que nos permita realizar la sumatoria de dos números, para ello iniciamos creando nuestro controlador, creamos un controlador con el nombre de ***ffi tti ffi s*** el código será el siguiente:

```php
<?php

class Aritmetica extends CI_Controller{

    #Funcion para cargar la vista determinada
    public function view($page='home'){

        if(!file_exists( 'application/views/pagina/'. $page. '.php' )){
            show_404();
        }
        $data['titulo'] = "Guia 06";
        $this->load->view('plantilla/header', $data);
        $this->load->view('pagina/'.$page ,$data);
        $this->load->view('plantilla/footer', $data);
    }

    #Funcion para realizar una suma
    public function sumar(){
        $data = array(
                    'titulo'=>"Guia 06",
                    'resp'=>"Error"
                        );
        if($_POST){
            $num1 = $_POST['num1'];
            $num2 = $_POST['num2'];
            $valor = $num1 + $num2;
            $data = array(
                    'titulo'=>"Guia 06",
                    'resp'=>$valor
                        );
        }
        $this->load->view('plantilla/header', $data);
        $this->load->view('plantilla/respuesta',$data);
        $this->load->view('plantilla/footer', $data);
    }
}
```

Como podemos observar el controlador hereda de una clase llamada esta es un clase del framework CodeIgniter, la cual nos permite cargar las vistas y los modelos que creamos convenientes; en el ejemplo solo cargamos las vistas, si deseamos cargar ya sea un modelo o una vista debemos utilizar siempre la siguiente línea de codigoa ***s x ffig*** y complementamos con la función de acuerdo al elemento que desesamos cargar.

La función que utilizamos en este momento posee 2 parametros, los cuales son:

1) El código de la vista que será renderizado en el navegador.
2) Los datos que se le pasaran a la vista a renderizar, este valor es un Array en el cual se deben definir los elementos que se pasaran tanto en el controlador como en la vista que se va a renderizar.

Luego de crear nuestro controlador crearemos nuestras vistas, para nuestro ejemplo dentro de la carpeta *x* crearemos dos carpetas, una con el nombre de *ffi ffi* (Se almacenaran las vistas que se usan con frecuencia) y la otra carpeta con el nombre de *ffi ffi* (Se almacenaran todas las vistas para cada proceso especifico).

ffi ffi ffi ffi En la carpeta Plantilla crearemos tres archivos con los siguientes nombres y extensiones *s ffij s s x x ffi s* Los códigos para cada una de ellos serán los siguientes:

s ffij s

```
<html>
<head>
    <title><?php echo $titulo ?> - Ejemplo 01</title>
</head>
<body>
    <h1>Guia de Implantacion 06</h1>
```

s

```
<strong>Universidad de Sonsonate &copy </strong>
</body>
</html>
```

x x ffi s

```
<body>
    <h3>La Respuesta es <?php echo $resp ?></h3>
</body>
```

Como podemos observar en los archivos header.php y respuesta.php, existe código php en el cual se utiliza para imprimir el valor que poseen las variables $titulo y $resp, dichos valores son almacenados en el controlador del proyecto.

ffi ffi ffi ffi En esta carpeta por el momento solo agregaremos un archivo php, llamado *x ttiffi s* El código es el siguiente:

```
<body>
    <form action="../sumar" method="post" >
        <p>Numero 1: <input name="num1" type="text"/></p>
        <p>Numero 2: <input name="num2" type="text"/></p>
        <p><input type="submit" value="Sumar"/></p>
    </form>
</body>
```

Para probar nuestra aplicación debemos acceder a la siguiente URL:

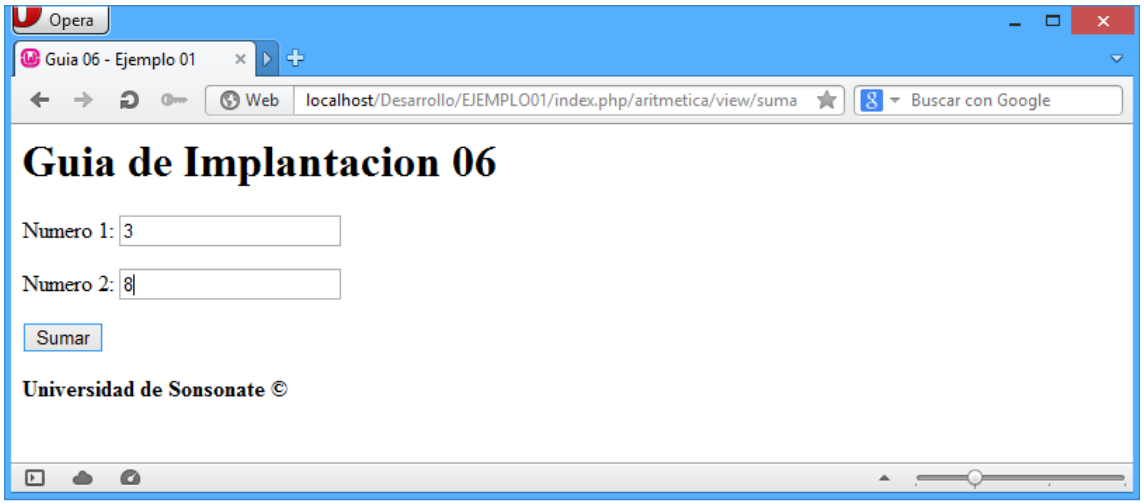
http://localhost/ EJEMPLO01/index.php/aritmetica/view/suma

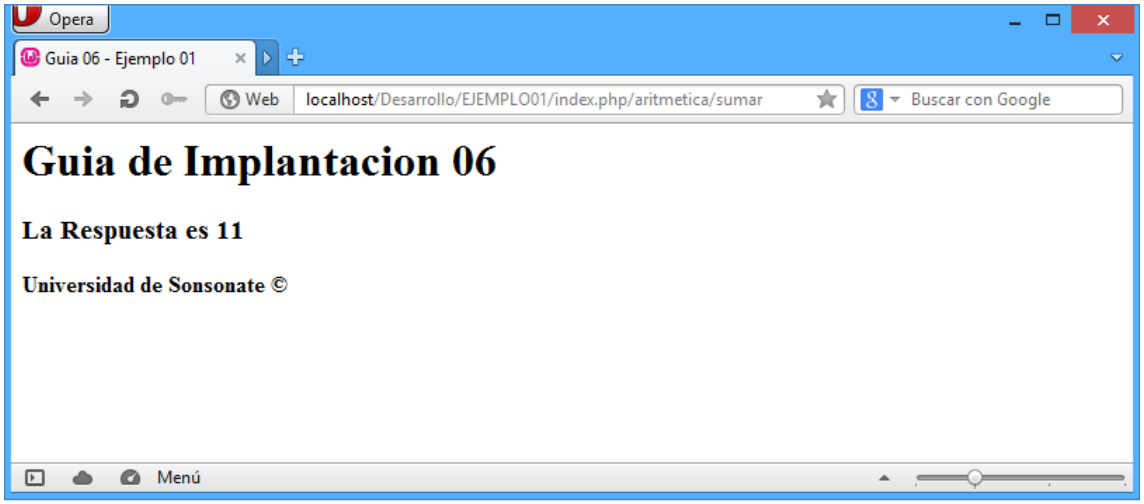
Recordemos que a un inicio se mencionaba que el archivo index.php, funcionaba como un controlador frontal, más adelante estudiaremos la manera de configurar las URL, para establecer una ruta determinada para las aplicaciones.

Un punto importante es recordar que CodeIgniter está basado en patrón de diseño Modelo-Vista-Controlador por lo cual la mayoría de frameworks que utilizan este patrón presentan un uso de urls de la siguiente manera:

http://ejemplo.com/[clase-controlador]/[método-controlador]/[argumentos]

Si hemos codificado de forma correcta el resultado debería de ser el siguiente:





EJERCICIOS.

1. Codifique las siguientes vistas en la carpeta ***ffi***
   - Multiplicar.php: Debe poder realizar la inserción de 5 números.
   - Division.php: Debe Permitir la inserción de 3 números.
   - Resta.php: Debe permitir la inserción de 2 números.
2. Codifique las funciones necesarias en el controlador para poder realizar las operaciones que llevan como nombre las vistas.
3. Modifique el archivo ***s ffg s*** y colóquele los siguientes datos:
   - Alumno: “Nombre del Alumno”
   - Codigo: “Código del Alumno”
   - Materia: Implantación de Sistemas”