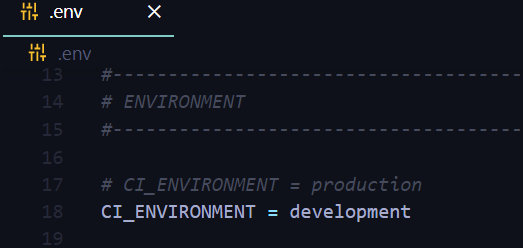
**PASOS PARA TRABAJAR CON CODEIGNITER**

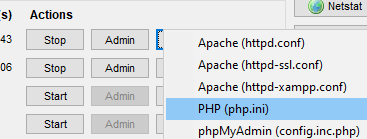
**PASOS INICIALES**

1. Descargar e instalar de la página oficial <https://codeigniter.com/download>
2. Tener instalado Xampp
3. En VSC instalar plugins como Bootstrap snippest
4. Instalar plugin CodeIgniter Spark en VSC
5. Se descomprime el proyecto y se ubica en la carpeta raíz de Htdocs

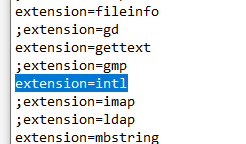
Para iniciar es posible que haya que hacer una modificación en el archivo .env. Se copia la línea comentada como # CI\_ENVIROMENT y se le quita el numeral a la copia, y donde dice production cambiamos por development,



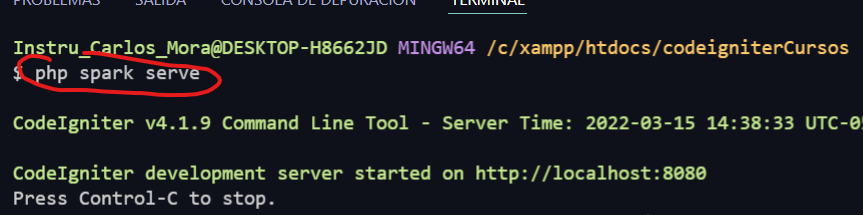
Si sigue arrojando algún error, se debe hacer una configuración en el archivo php.ini. Así que en el Panel de Control de Xampp, para el servicio de Apache, clickeamos en config:



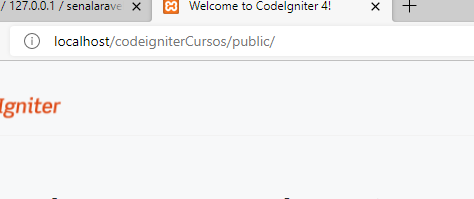
En ese archivo se hace el siguiente cambio, se quita el punto y coma (;) que está antes de la línea extensión=intl



No olvidar que se debe iniciar el servidor en la ventana de comandos, con php spark serve:



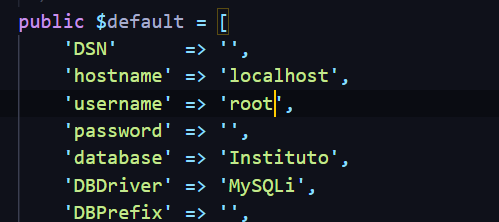
Se visualiza en el navegador indicando la ruta donde está el proyecto y la ruta public. Ejemplo:



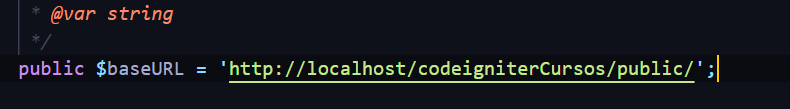
|  |
| --- |
| ***Las siguientes configuraciones son opcionales para resolver el problema de que el Framework no inicia en el navegador***  y también en app/config/Boot/development.php, en la siguiente captura, en display\_errors, si hay un cero poner un 1.    Y en la misma ruta pero en production.php también en display\_error, si hay un cero poner un 1    Y en Models/userModel.php, cambiamos en el use, el Codeigniter por CodeIgniter |

**CONFIGURACION DE BASE DE DATOS**

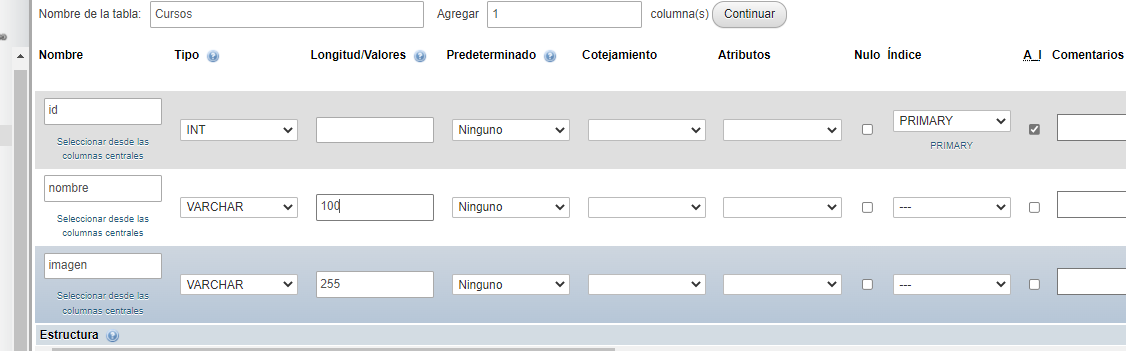
1. En App/Config/Database.php se hace la siguiente configuración (en este caso nuestra bd se llamará Instituto).



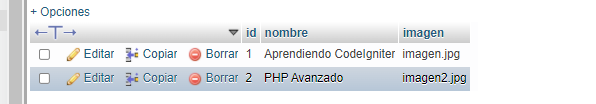
1. En App/Config/App.php cambiamos la dirección de la baseURL:



1. Usando la interfaz de phpMyAdmin, creamos la bd Cursos manualmente:



1. Ingresamos datos manualmente en la BD. La ruta de la imagen la inventamos, solo con fines de modificación:

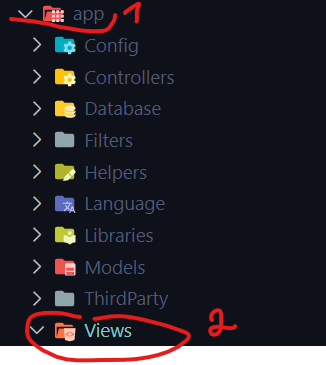


El query ejecutado es algo como:

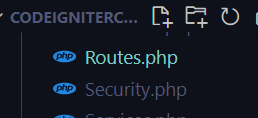
INSERT INTO `cursos` (`id`, `nombre`, `imagen`) VALUES (NULL, 'PHP Avanzado', 'imagen2.jpg');

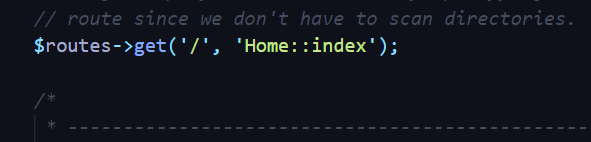
**UBICACIÓN DE LAS VISTAS Y LAS RUTAS**

**En CodeIgniter** las vistas están dentro de la misma carpeta app, y luego en Views:

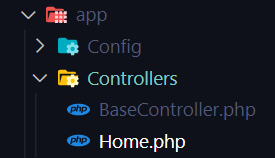


Por otra parte, las rutas las encontramos también en app, dentro de la subcarpeta Config. Allí está el archivo Routes.php



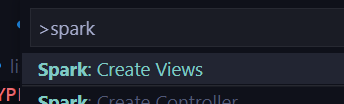


**LOS CONTROLADORES**

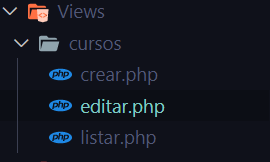
Están directamente en la carpeta app, en Controllers. Por default, allí está Home.php 

**Cómo crear una vista usando la extensión Spark de VSC**

Para acceder a las funcionalidades del plugin spark, presionamos F1 (o Fn F1), y en la barra de búsque escribimos spark. Luego damos el nombre de la vista y finalmente decimos que deseamos incluir la estructura HTML (clic en yes):

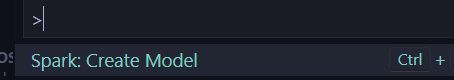


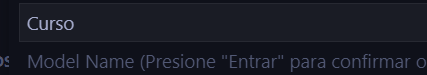
1. Estas son las vistas creadas, las cuales se incluyen en una subcarpeta Cursos:



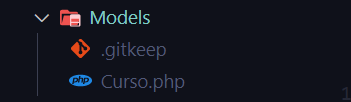
**Creando un Modelo y un Controlador**

Es necesario crear un modelo (Curso) el cual permitirá acceder a la tabla que debe especificarse con su nombre (cursos)





Hay que decirle cómo se llama la tabla con la que conectará y si se creará una migración. Como no estamos desde la ventana de comandos, no es necesario crear una migración. Además, cuando pregunta si deseamos crear el controlador, le decimos que sí y se llamará Cursos.



**Creación de la ruta con Spark**

Creamos la respectiva ruta ingresando los datos que nos pida. Simplemente oprimimos F1, luego spark create rout, y seguimos los pasos.

**Para Personalizar La vista Listas**

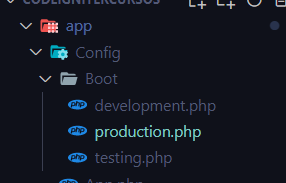
Incluimos Bootstrap.

Creamos una tabla (b-table header, usando la extensión de Bootstrap para VSC):



**CONFIGURANDO PARA DEPURAR ERRORES (O VER LOS ERRORES)**

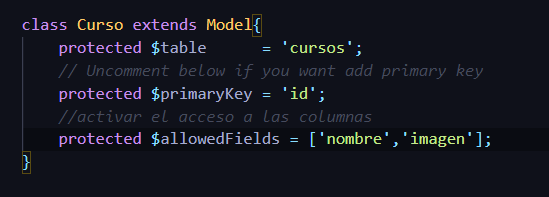
1. Ir a app/config/Boot/production.php



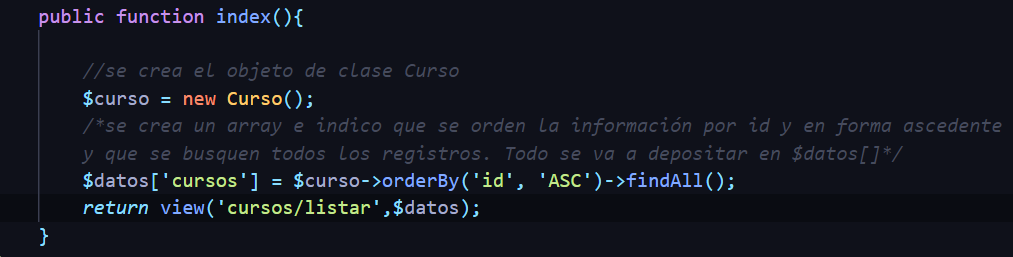
1. Cambiar 0 por 1.

**CONSULTAR LA BASE DE DATOS**

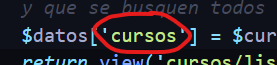
1. Lo primero en el modelo Curso (el único creado hasta ahora) hacemos lo siguiente:

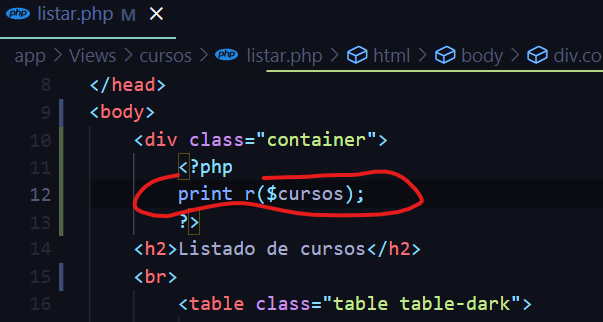
****

1. Luego se debe llamar la información desde el controlador (en el ejemplo, Cursos), en el método index():



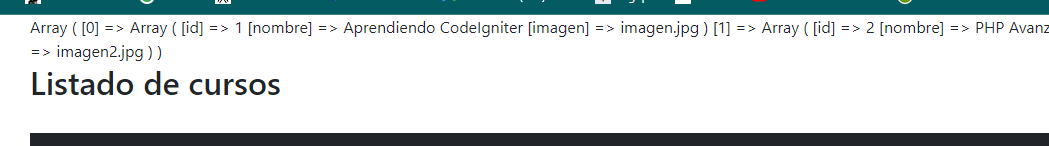
1. El índice del array $datos, que en este ejemplo lo llamamos ‘cursos’, y que está en los corchetes, lo debemos tratar como una variable dentro de la vista listar.php. Por tanto lo llamamos en la vista como $cursos, y así podremos mostrar la información de la tabla, en la vista:



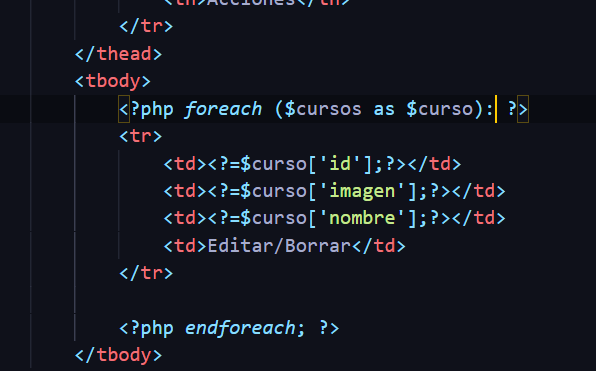


Como puede apreciarse, se utiliza el método print\_r para hacer una comprobación y ver que funciona en el navegador. Los usuarios finales no deberían ver ese tipo de información.

1. Ya en la vista se puede apreciar la información de la tabla respectiva:

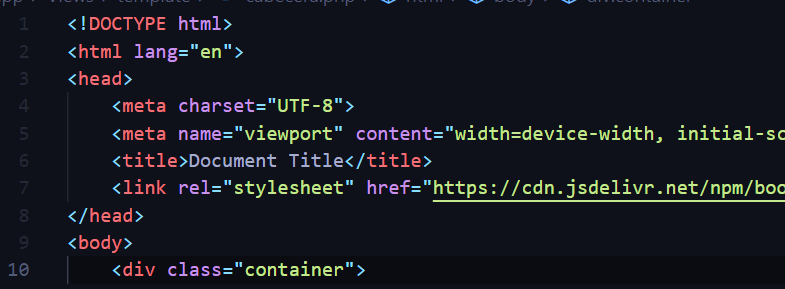


1. Para verlo ya dentro de la tabla de estilo Bootstrap como tal, comentamos la línea anterior de print\_r y digitamos lo siguiente dentro de tr:

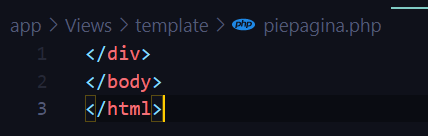


**CREACIÓN DE TEMPLATES**

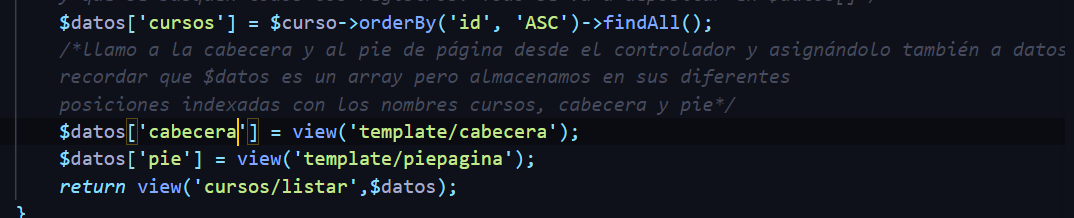
1. Se crea la carpeta templates en Views
2. Dento se crea cabecera.php y piepagina.php
3. El siguiente código se corta (Ctrl X) de listar.php y se pega en cabecera.php



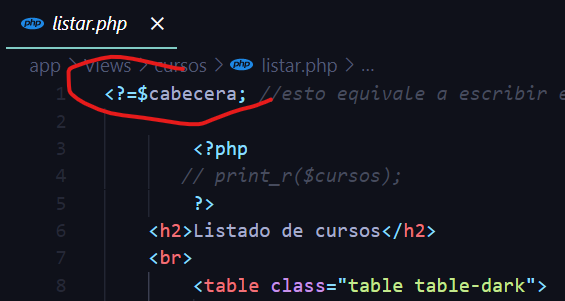
1. El siguiente código se corta (Ctrl X) de listar.php y se pega en piepagina.php



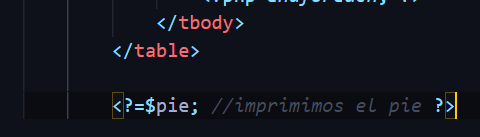
1. Luego en el controlador, dentro del método index() se llama a la cabecera y al pie de página creados así:



1. Luego usamos lo anterior directamente en la vista listar.php de la siguiente manera

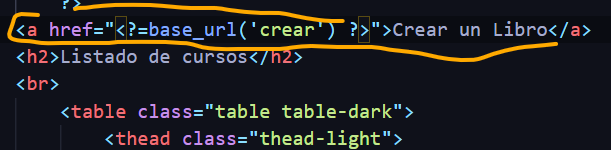
****

**…**

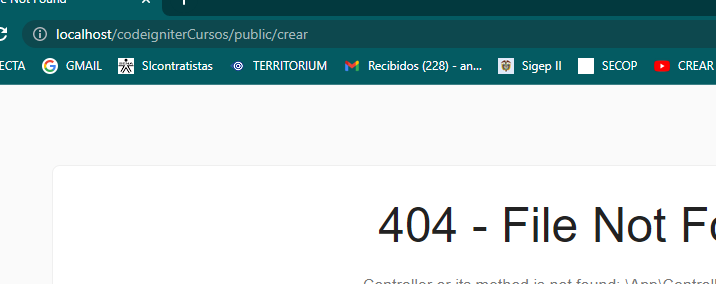
****

**CREANDO ENLACES (LINKS) A OTRAS VISTAS**

1. Creamos un link en la página Listar.php, que lleva a la vista crear.php

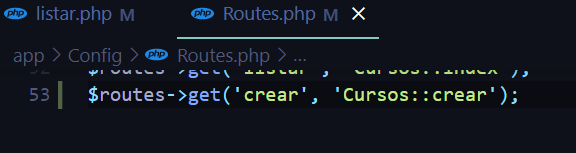
****

1. Luego probamos



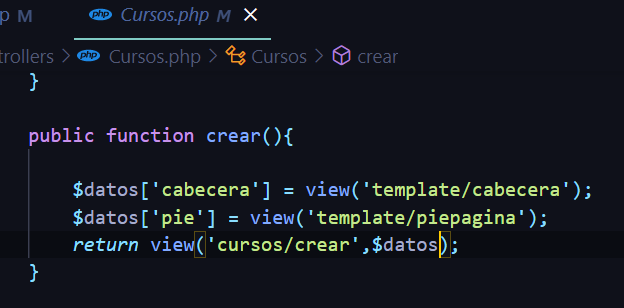
1. Abrimos el archivo Routes.php que está en Boot.
2. Oprimimos F1 para poder usar los diferentes comandos
3. Luego escribimos spark
4. Elegimos Create Route

El nombre sería crear, el controlador sería Cursos. El método sería crear

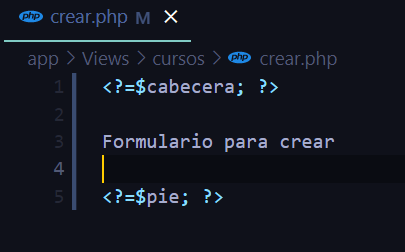


**MODIFICANDO EL FORMULARIO DE CREAR Y AGREGANDO EL FORMULARIO**

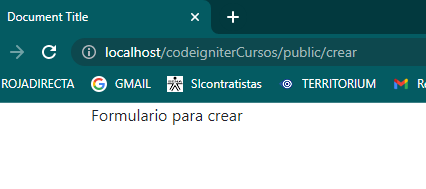
1. Se incluye primero cabecera y pie en el controlador, en el método crear()



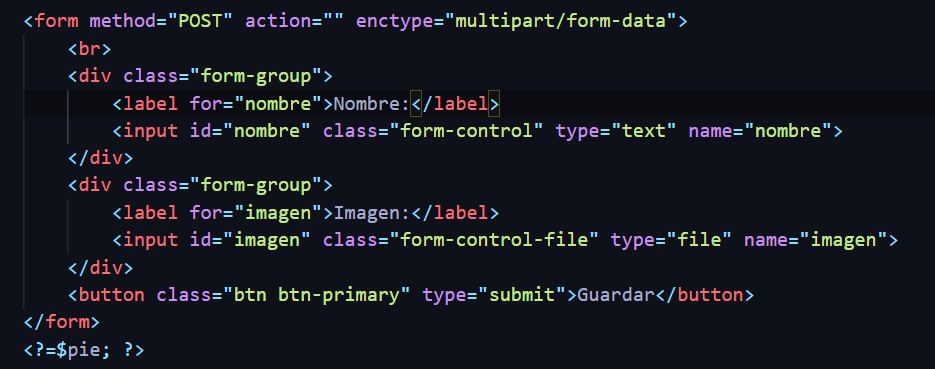
1. Se borra lo que había en la vista crear y se agregan cabecera y pie:



1. El resultado es:



1. Se agrega el código para el formulario en crear.php:



**ENVÍO LA INFORMACIÓN POR POST**

¿Cómo envío y almaceno la información del formulario?

1. Se debe indicar en el action a través de site\_url y la ruta guardar. Se debe crear dicha ruta.



1. Se abre el archivo de Routes.php. La ruta se crea con el nombre guardar, asociada al controlador Cursos y al método guardar que luego debemos crearlo también.
2. En el método guardar del controlador Cursos, se digita lo siguiente para probar si está llegando la información:

Texto

Descripción generada automáticamente

1. Al hacer la prueba, veremos que no funciona. Y es porque la ruta /guardar está usando el método get cuando se debe usar post. Por eso, es necesario hacer el cambio en la ruta:

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente