

MANUAL DE CAKE PHP

Diego Estrella

email: [diegoe4@hotmail.com](mailto:diegoe4@hotmail.com)

Mi Canal:

<https://youtu.be/cjpVuBVX2tY> (Resolución720x480)

https://youtu.be/6y-GWRJemlI (Resolución 1920x1080)

GIT: [diegoe4/proyectoweb: Ingweb (github.com)](https://github.com/diegoe4/proyectoweb)

Contenido

[**Introducción al Framework**](#_heading=h.t2talusckl71)2

[¿Qué es Cake PHP?](#_heading=h.30j0zll) 5

[Características de Cake PHP](#_heading=h.8jevh0qoubs9) 2

[Modelo MVC en CakePHP](#_heading=h.1gwpkvp2laar) 3

[Ciclo de una petición CakePHP](#_heading=h.7exi3blazipk) 4

[Instalación de Cake PHP](#_heading=h.30j0zll) 5

[Estructura del proyecto CakePHP](#_heading=h.6wawkqveuoat) 9

[Creación y configuración de la base de datos](#_heading=h.ohsie3umr07y) 11

[Creación de CRUD de manera automática](#_heading=h.s79w0ak5qd6c) 13

[Respaldo de la base de datos en phpMyAdmin](#_heading=h.c161fidl52jg) 15

[Login del Proyecto](#_heading=h.qs3gyhpu4q02) 16

# 

# Introducción al Framework

## ¿Qué es Cake PHP?

CakePHP es un marco de desarrollo [framework] rápido para PHP, libre, de código abierto. Se trata de una estructura que sirve de base a los programadores para que éstos puedan crear aplicaciones Web

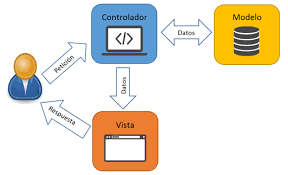
## 

## Características de Cake PHP

Esta es una lista breve con las características de las que disfrutarás al utilizar CakePHP:

* Comunidad activa y amistosa
* Licencia flexible
* Compatible con PHP4 y PHP5
* CRUD integrado para la interacción con la base de datos
* Soporte de aplicación [scaffolding]
* Generación de código
* Arquitectura Modelo Vista Controlador (MVC)
* Despachador de peticiones [dispatcher], con URLs y rutas personalizadas y limpias
* Validación integrada
* Plantillas rápidas y flexibles (sintaxis de PHP, con ayudantes[helpers])
* Ayudantes para AJAX, Javascript, formularios HTML y más
* Componentes de Email, Cookie, Seguridad, Sesión y Manejo de solicitudes
* Listas de control de acceso flexibles.

## Modelo MVC en CakePHP



**La capa Modelo**

La capa Modelo representa la parte de tu aplicación que implementa la lógica de negocio. Es la responsable de obtener datos y convertirlos en los conceptos que utiliza tu aplicación. Esto incluye procesar, validar, asociar u otras tareas relacionadas con el manejo de datos. En el caso de una red social la capa modelo se encargaría de tareas como guardar los datos del usuario, las asociaciones de amigos, almacenar y obtener fotos, buscar sugerencias de amistad, etc. Los objetos modelo serían «Amigo», «Usuario», «Comentario» o «Foto».

**La capa Vista**

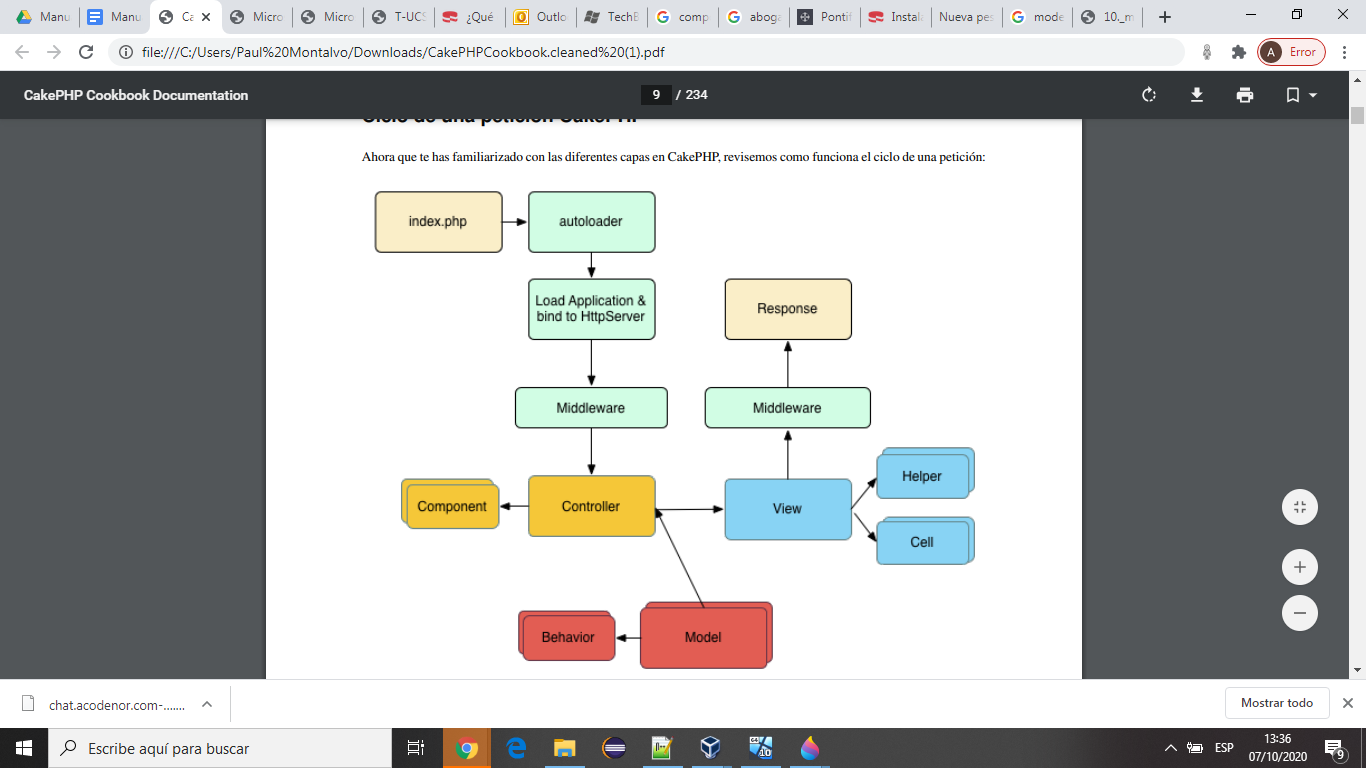
La capa Vista renderiza una presentación de datos modelados. Separada de los objetos Modelo, es la responsable de usar la información disponible para producir cualquier interfaz de presentación que pueda necesitar tu aplicación.

**La capa Controlador**

La capa Controlador maneja peticiones de usuarios. Es la responsable de elaborar una respuesta con la ayuda de las capas Modelo y Vista. Un controlador puede verse como un gestor que asegura que todos los recursos necesarios para completar una tarea son delegados a los trabajadores oportunos. Espera por las peticiones de los clientes, comprobar la validez de acuerdo con las reglas de autenticación y autorización, delega la búsqueda o procesado de datos al modelo, selecciona el tipo de presentación que el cliente acepta y finalmente delega el proceso de renderizado a la capa Vista.

## Ciclo de una petición CakePHP

Ahora que te has familiarizado con las diferentes capas en CakePHP, revisemos cómo funciona el ciclo de una petición:



El ciclo de petición típico de CakePHP comienza con un usuario solicitando una página o recurso en tu aplicación. A un alto nivel cada petición sigue los siguientes pasos:

1. Las reglas de rescritura del servidor web envían la petición a webroot/index.php.
2. Tu aplicación es cargada y ligada a un **HttpServer**
3. Se inicializa el **midleware** de tu aplicación.
4. Una petición y respuesta son procesadas a través del **Middleware PSR-7** que tu aplicación utiliza. Normalmente esto incluye la captura de errores y enrutamiento.
5. Si no recibe ninguna respuesta del **middleware** y la petición contiene información de enrutamiento, se selecciona un controlador y una acción.
6. La acción del controlador es ejecutada y el controlador interactúa con los Modelos y Componentes necesarios
7. El controlador delega la creación de la respuesta a la Vista para generar la salida a partir de los datos del modelo.
8. La vista utiliza **Helpers y Cells** para generar el cuerpo y las cabeceras de la respuesta.
9. La respuesta es devuelta a través del /controllers/middleware.
10. El **HttpServer** envía la respuesta al servidor web.

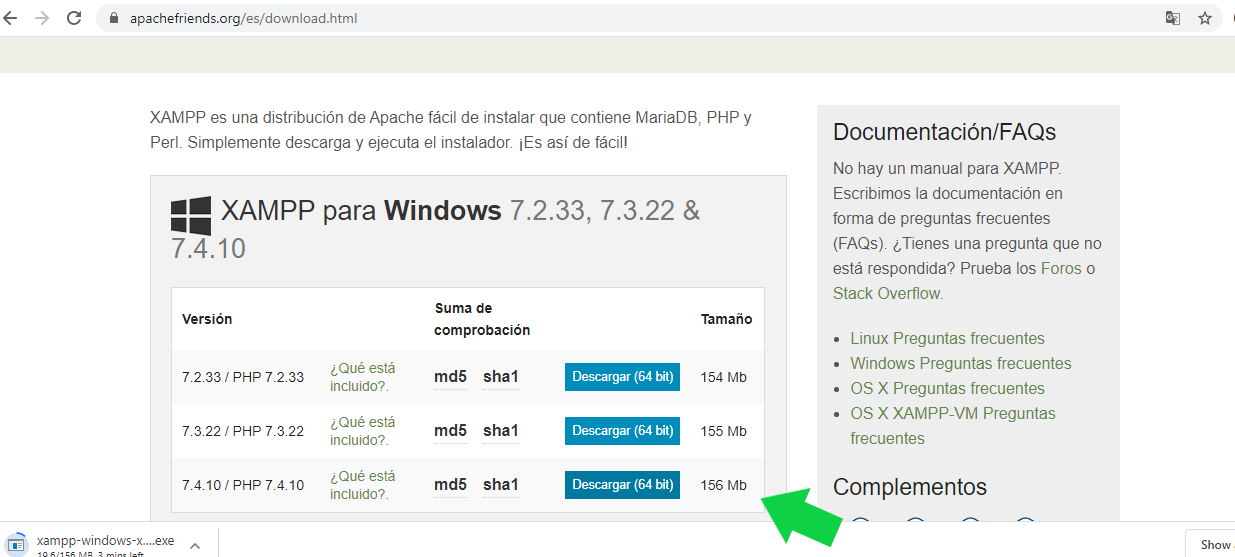
## Instalación de Cake PHP

La aplicación que vamos a desarrollar la trabajaremos en un sistema Windows en el que realizaremos el siguiente proceso de instalación.

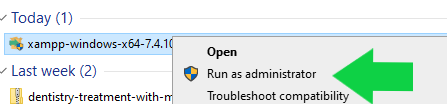
1. Descargar Oracle VirtualBox.
   1. Dependiendo del sistema operativo que tengamos instalado seleccionamos el instalador y procedemos a realizar la instalación. Nos podemos descargar de la siguiente página web: <https://www.virtualbox.org/>
2. Instalamos Windows 10 en cualquiera de sus versiones dentro de Oracle VirtualBox
3. Dentro de la máquina virtual de Windows **instalamos** XAMPP (XAMPP es una distribución de Apache completamente gratuita y fácil de instalar que contiene MariaDB, PHP y Perl. El paquete de instalación de XAMPP ha sido diseñado para ser increíblemente fácil de instalar y usar.)

Nos descargamos de la siguiente página el instalador para nuestro Windows: <https://www.apachefriends.org/es/index.html>

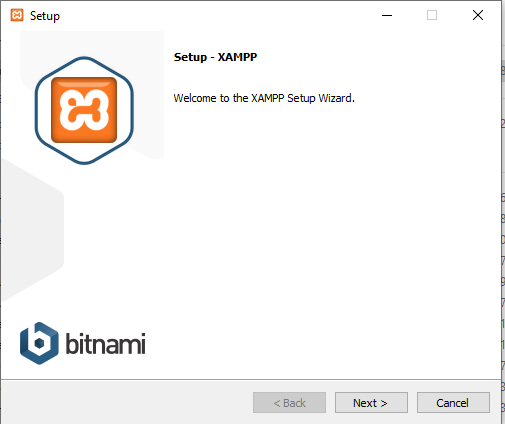
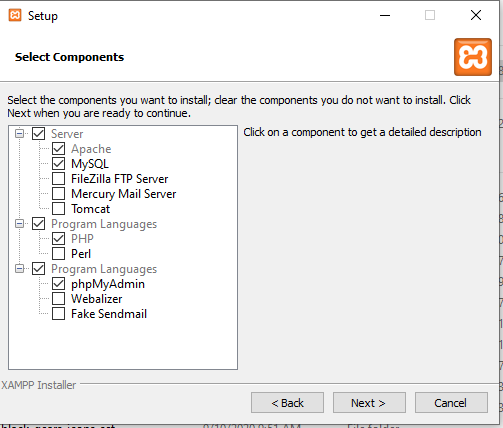
Siempre descargar la última versión



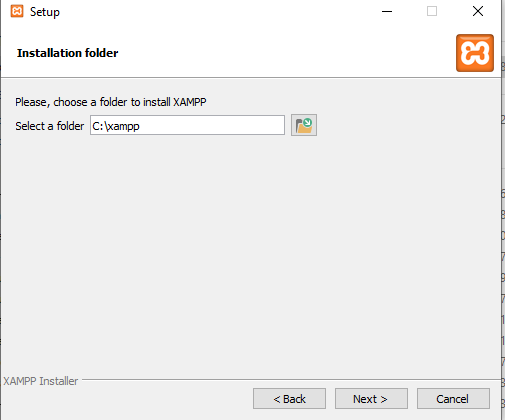
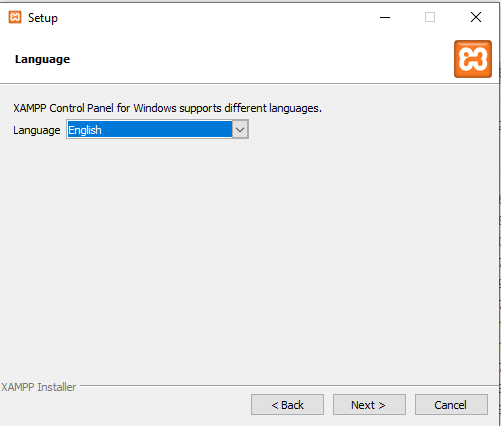
Luego procedemos damos clic derecho en el instalador descargado y lo ejecutamos como Administrador



Luego procedemos a iniciar la instalación de XAMPP. Damos clic en Next. A continuación seleccionamos los componentes que se instalarán para la realización del proyecto.

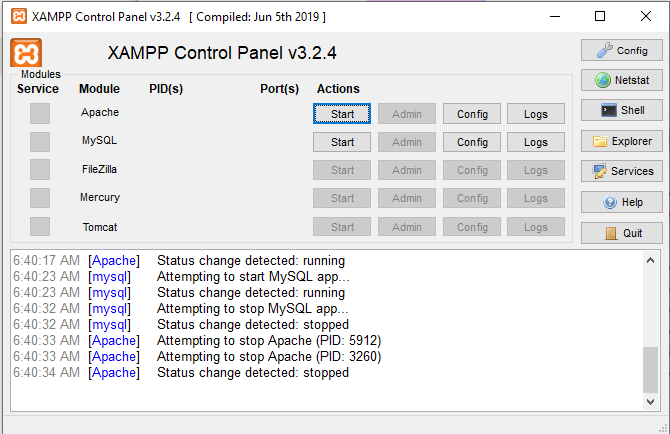
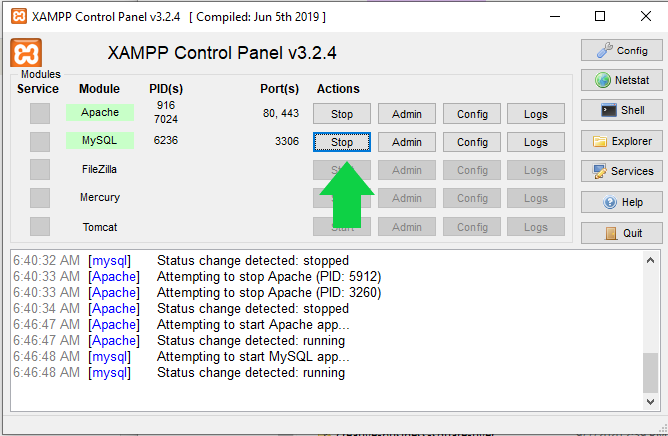
 

Procedemos a poner la ruta de instalación de XAMPP por lo que dejaremos por defecto que sea dentro de la unidad C:. Luego seleccionamos el idioma de nuestro panel de control de XAMPP. Luego damos clic en las siguientes pantallas hasta que se proceda con la instalación.

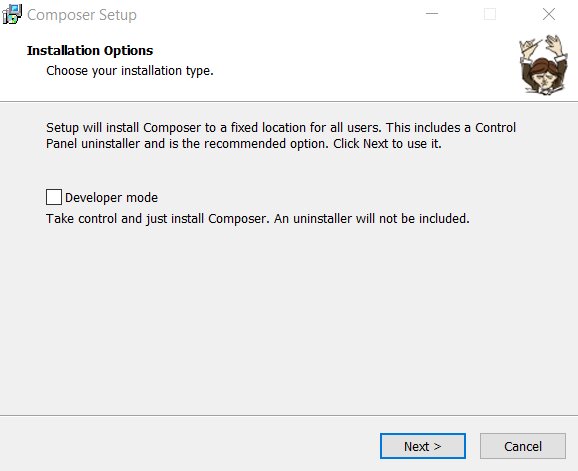
1. Levantamos los servicios de XAMPP

Damos clic en el botón Start los servicios que queremos subir. En nuestro caso subimos los servicios Apache y MySQL.

1. Creación del proyecto mediante Composer (Composer es un sistema de gestión de paquetes para programar en PHP el cual provee los formatos estándar necesarios para manejar dependencias y librerías de PHP).

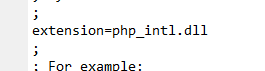
Una vez que XAMPP esté instalado, descargamos la [última versión de Composer](https://getcomposer.org/Composer-Setup.exe) del siguiente link <https://getcomposer.org/Composer-Setup.exe>.

Ejecutamos el asistente de instalación de Composer como administrador. Cuando te solicite que actives el modo desarrollador, ignoralo y continúa con el proceso de instalación.  


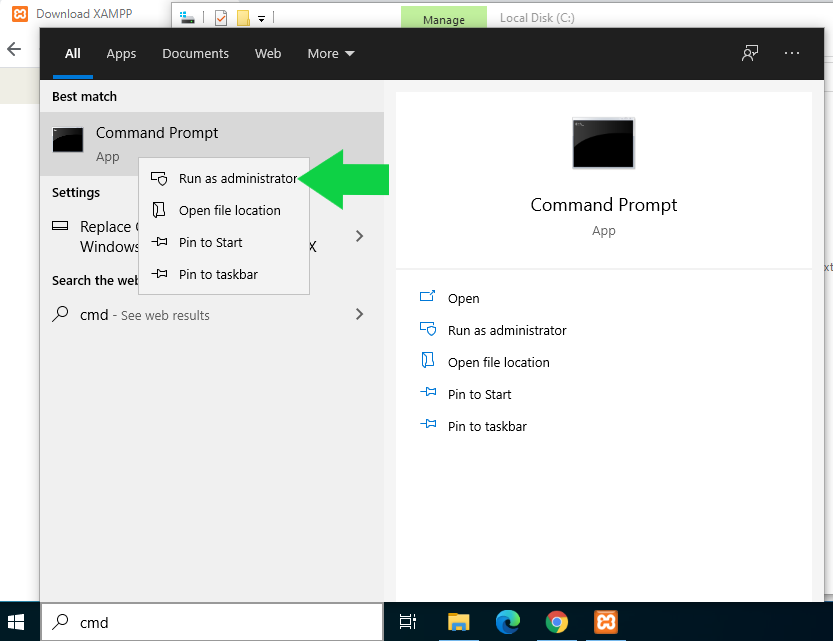
Aparecerá otra ventana y te pedirá que ubiques la línea de comando PHP. Por defecto, está en **C:/xampp/php/php.exe.** Después de especificar la ubicación, haz clic en Siguiente.

Se te solicitará la configuración de proxy. Deja la casilla sin marcar y omite esta parte presionando **Siguiente**. Luego, en la última ventana, haz clic en **Instalar**.

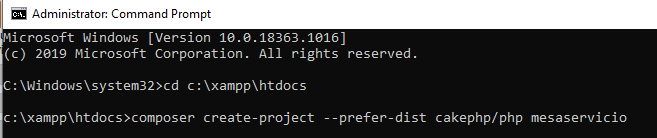
NOTA: Dentro de XAMPP modificamos el archivo c:\xampp\php\php.ini. Añadimos la línea (extension=php\_intl.dll) o si existe descomentamos.

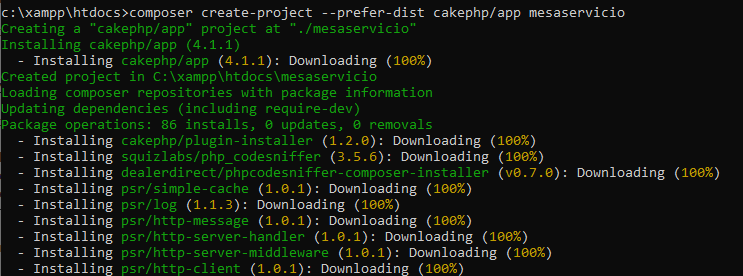


Abrimos la consola de comandos de windows y damos clic derecho y ejecutamos como Administrador.



Luego nos ubicamos dentro de la ruta c:\xampp\htdocs

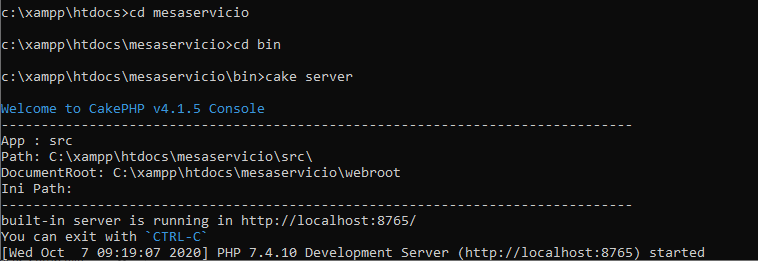
Creamos el proyecto mesaservicio mediante composer y que utilice cakephp ejecutando el siguiente comando: **composer create-project --prefer-dist cakephp/app mesaservicio**



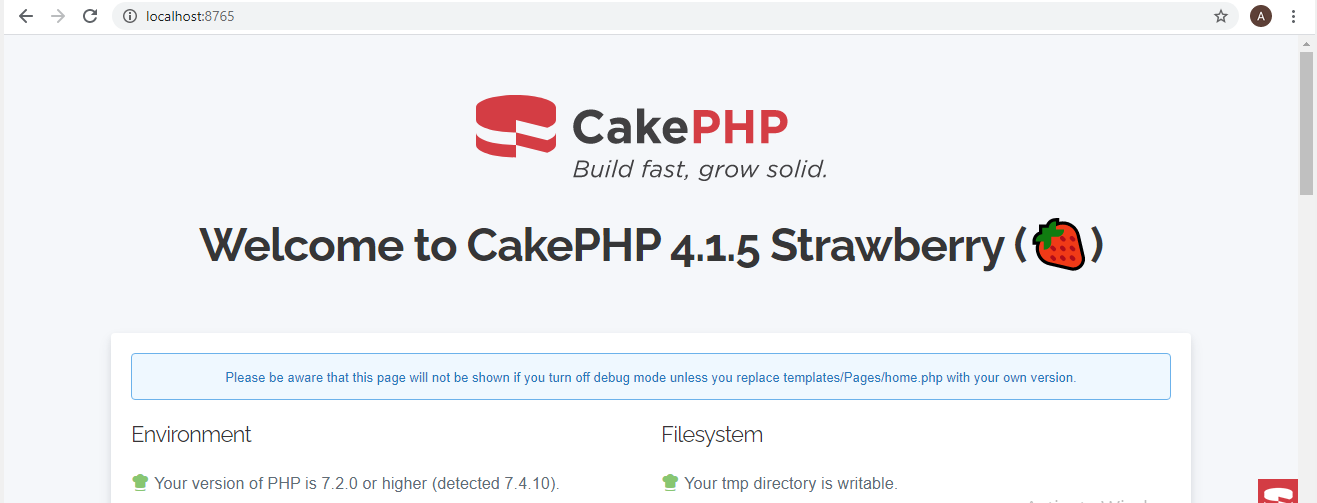
Luego dentro de la ruta c:\xampp\htdocs se encontrará nuestro proyecto mesaservicio creado.

**Comprobar la instalación**

Podemos comprobar rápidamente que nuestra instalación ha sido correcta accediendo a la página principal que se crea por defecto. Pero antes necesitarás inicializar el servidor de desarrollo ejecutando el siguiente comando **cake server** en la ruta donde se encuentran el programa principal de cake: c:\xampp\htdocs\mesaservicio\bin

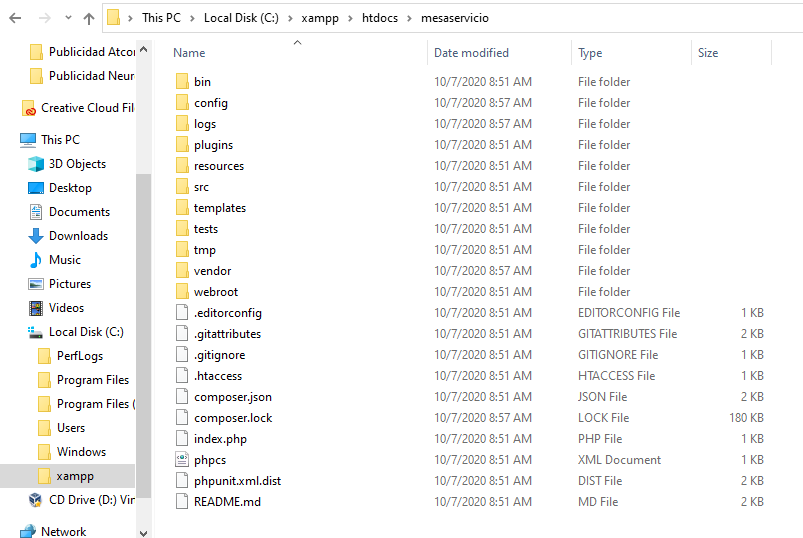


Esto arrancará el servidor integrado en el puerto 8765. Luego accederemos a la página http://localhost:8765 a través de tu navegador para ver la página de bienvenida.



## Estructura del proyecto CakePHP

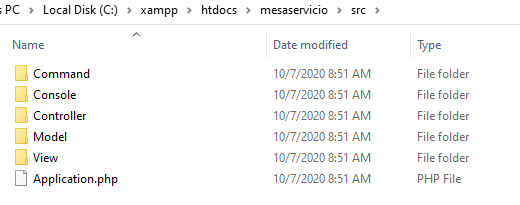
Luego de instalado el proyecto se genera la estructura completa. A continuación se detalla cada una carpeta generada.



|  |  |
| --- | --- |
| **Carpeta** | **Descripción** |
| bin | Contiene los ejecutables por consola de Cake. |
| config | Contiene los documentos de Configuración que utiliza CakePHP. Detalles de la conexión a la Base de Datos, bootstrapping, archivos de configuración del core y otros, serán almacenados aquí. |
| plugins | Se almacenan los Plugins que utiliza tu aplicación |
| logs | Contiene los archivos de log |
| src | Se almacenan los datos de la aplicación |
| tests | Donde se pondrán los test para la aplicación |
| tmp | CakePHP almacenará temporalmente la información. La información actual que almacenará dependerá de cómo se configure CakePHP, pero esta carpeta es normalmente utilizada para almacenar descripciones de modelos y a veces información de sesión |
| vendor | CakePHP y otras dependencias de la aplicación serán instaladas |
| webroot | Es la raíz de los documentos públicos de tu aplicación. Contiene todos los archivos que quieres que sean accesibles públicamente |

***La carpeta src***

La carpeta src de CakePHP es donde tú harás la mayor parte del desarrollo de tu aplicación. Observemos más detenidamente dentro de la carpeta src.

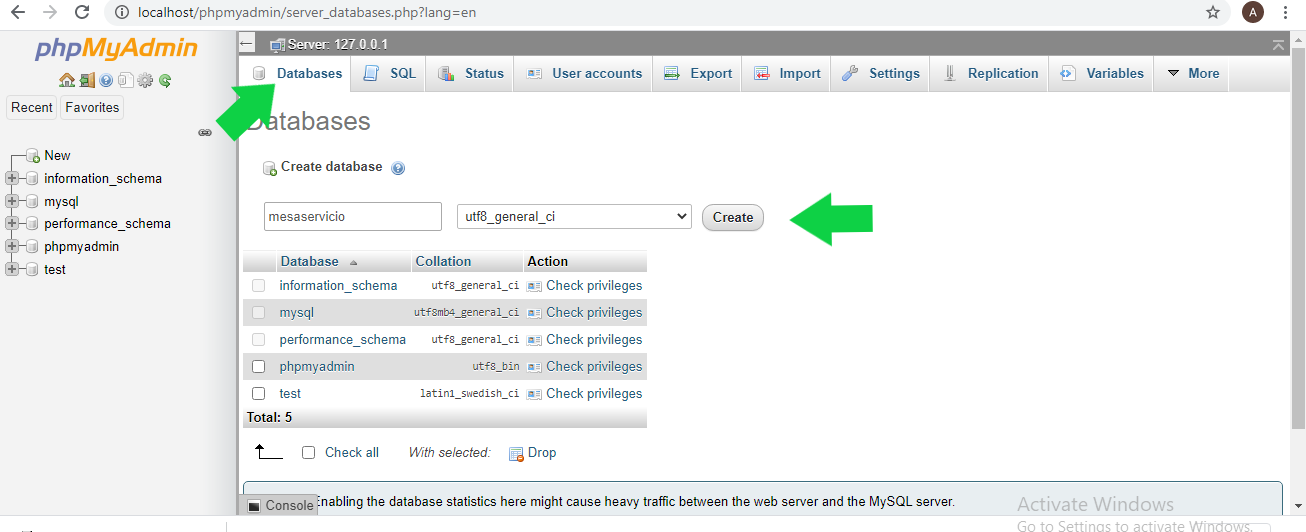


|  |  |
| --- | --- |
| **Carpeta** | **Descripción** |
| Console | Contiene los comandos de consola y las tareas de consola de tu aplicación |
| Controller | Contiene los controladores de tu aplicación y sus componentes. |
| Model | Contiene las tablas, entidades y funcionamiento de tu aplicación |
| View | Las clases de presentación se ubican aquí: cells, helpers y templates. |

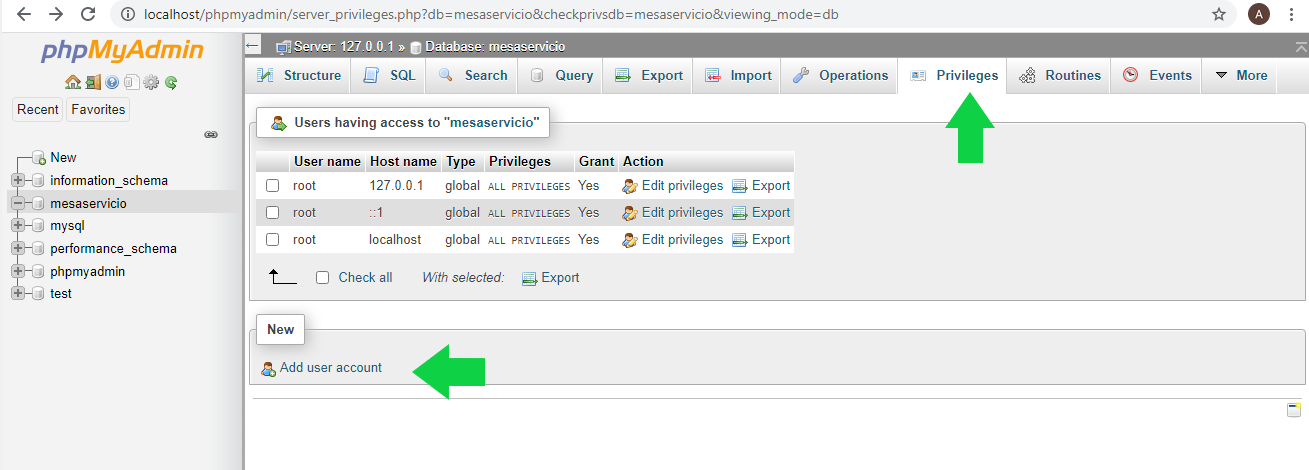
## Creación y configuración de la base de datos

Para la creación de la base de datos accedemos a phpMyAdmin(herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas web, utilizando un navegador web) el cual se podrá acceder mediante la siguiente URL: <http://localhost/phpmyadmin>. Por defecto el phpMyAdmin no posee clave por defecto.

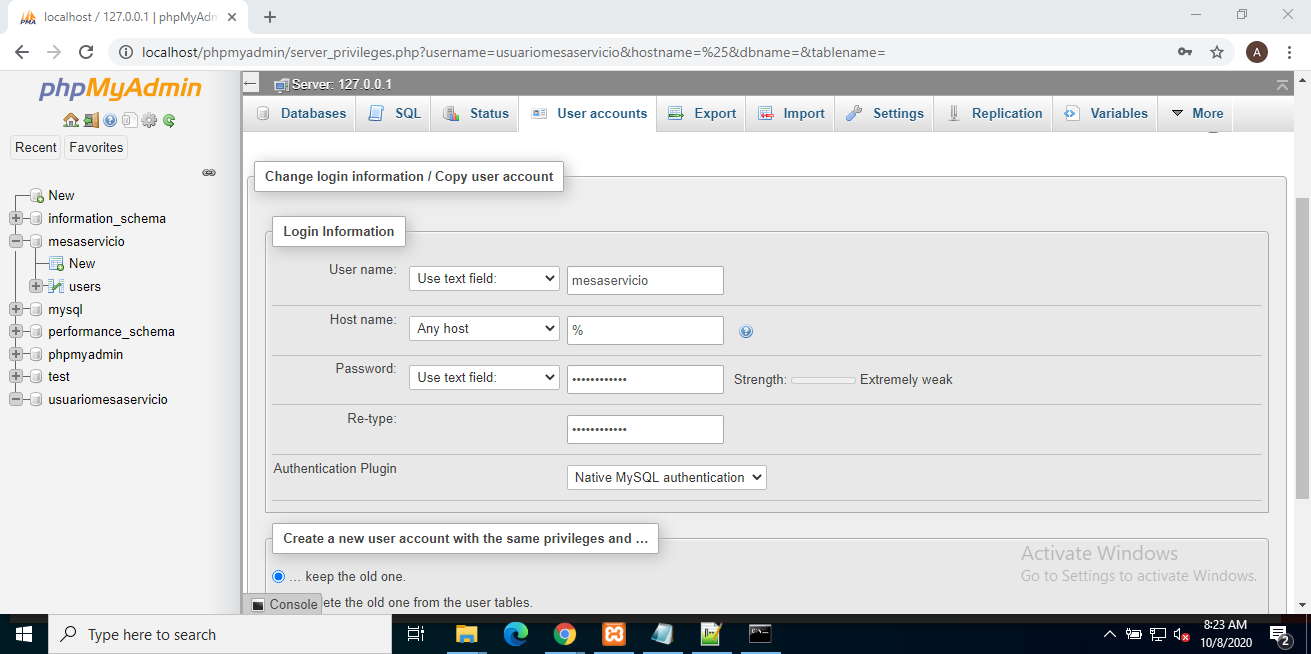
Seleccionamos la pestaña **Databases** y dentro **Create database** ingresamos el nombre de nuestra base de datos en este caso **mesaservicio** y la codificación escogemos **utf8\_general\_ci** porque vamos a registrar tildes y caracteres especiales. Luego damos clic en el botón **Create**.

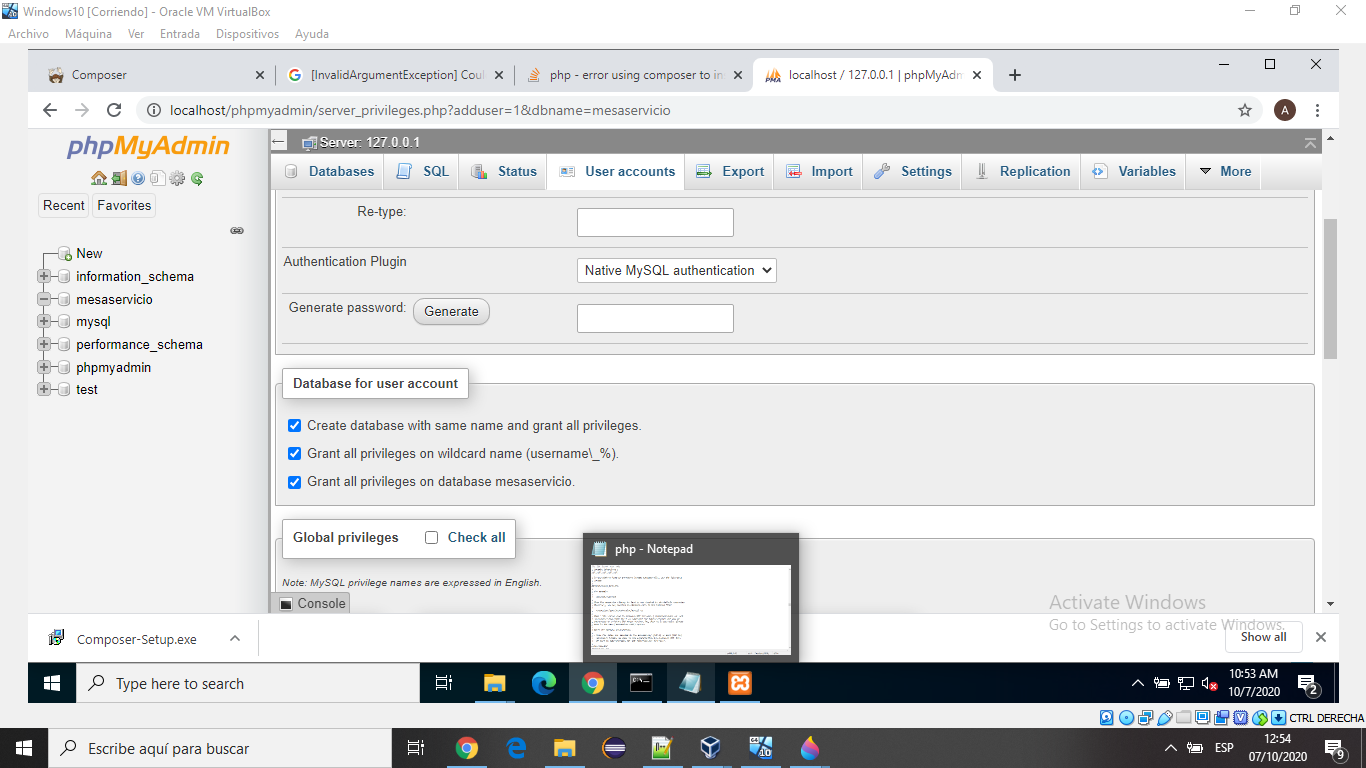


Luego creamos un usuario (mesaservicio) que acceda a nuestra base creada y que tenga todos los permisos necesarios para que pueda interactuar con la base de datos.Accedemos dentro de nuestra base de datos y seleccionamos **Privileges**. Dentro de esta pestaña se mostrarán todos los usuarios de la base de datos creados y que pueden ver esta base de datos creada. Damos clic en **Add user account**.



Creamos el usuario mesaservicio dentro de esta nueva pantalla y damos clic en el botón **Go** que se encuentra al final de la página.





La configuración de la base de datos dentro de cakePHP es bastante sencilla: sólo cambia los valores del array **Datasources.default** en el archivo **config/app\_local.php** por los datos de la base de datos que previamente hayas creado.

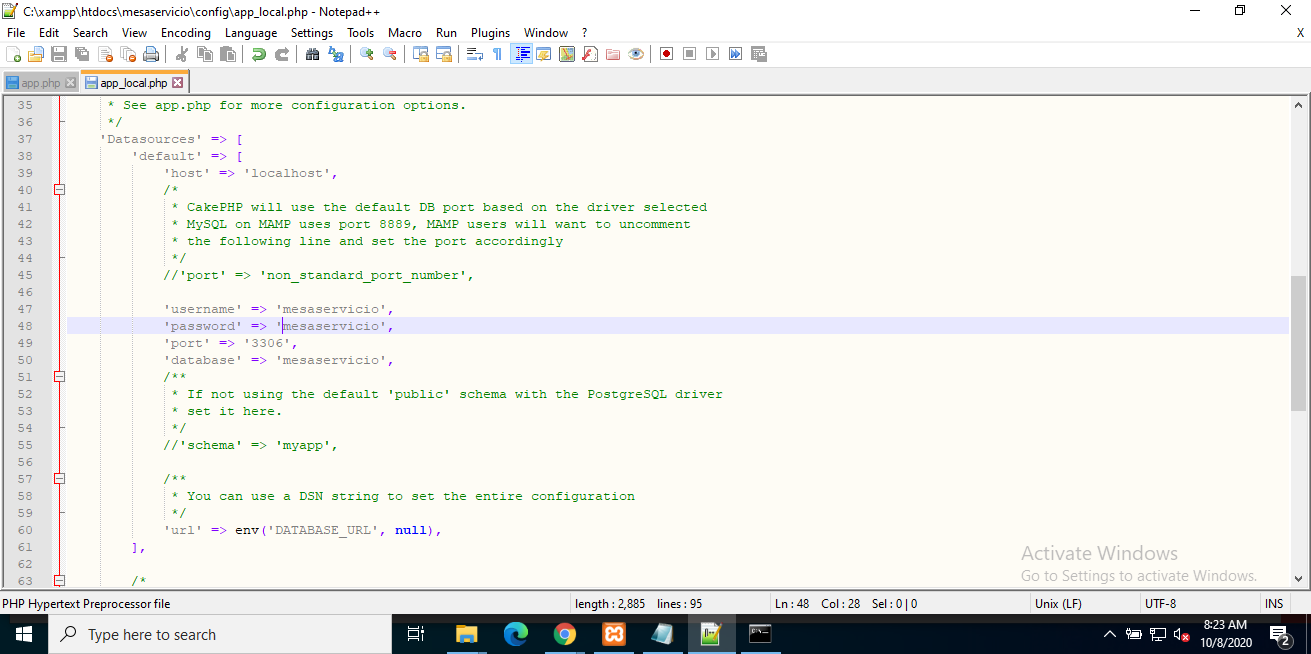
'host' => 'localhost',

'username' => 'mesaservicio',

'password' => 'mesaservicio',

'port' => '3306',

'database' => 'mesaservicio',



## Creación de CRUD de manera automática

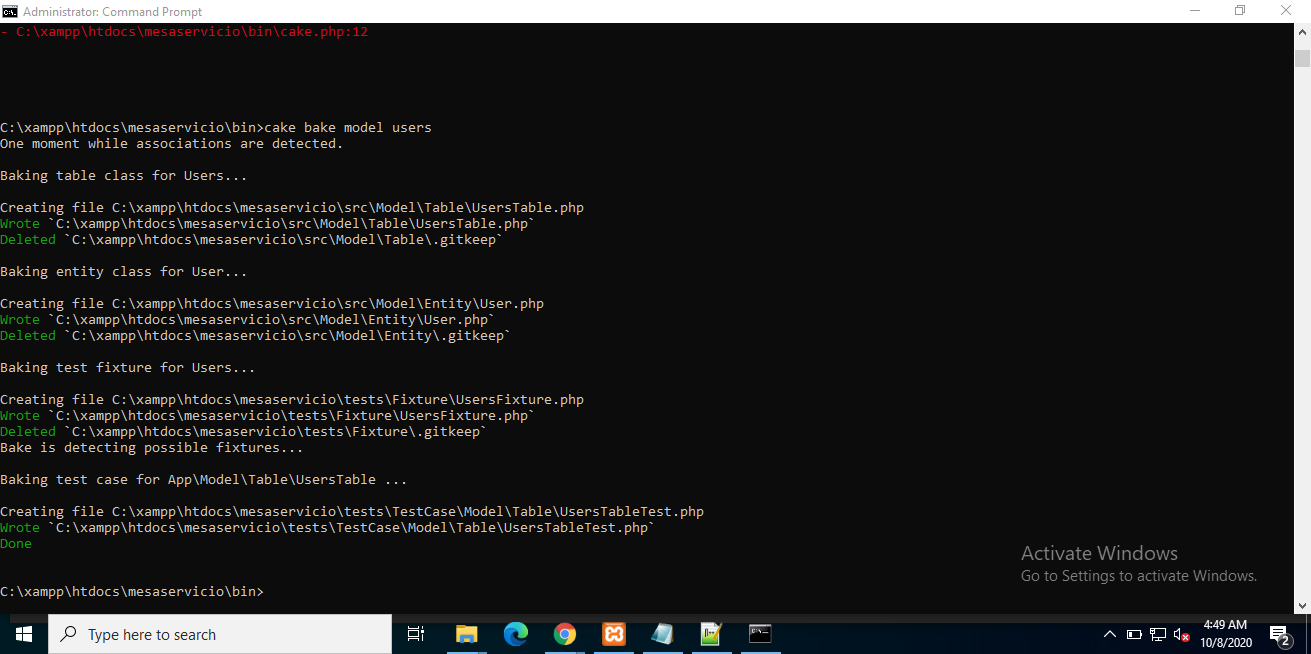
Vamos a utilizar las funcionalidades de CakePHP en la generación de código. Creamos la tabla users dentro de phpMyAdmin.

Puedes importar el respaldo de la base de datos en PHPMyAdmin

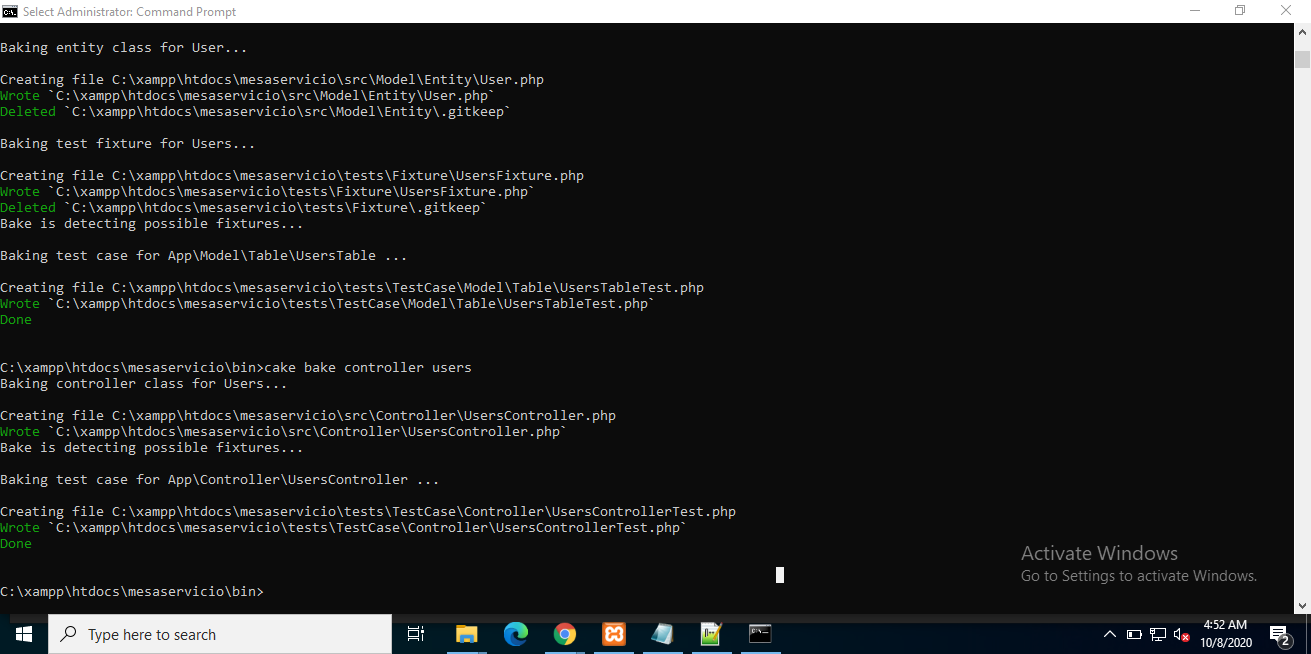
|  |
| --- |
| CREATE TABLE IF NOT EXISTS `users` (  `id` int(11) NOT NULL AUTO\_INCREMENT PRIMARY KEY,  `nombre` varchar(32) NOT NULL,  `apellido` varchar(32) NOT NULL,  `email` varchar(32) NOT NULL,  `usuario` varchar(32) NOT NULL,  `clave` varchar(255) NOT NULL,  `creacion` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP,  `modificacion` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,  `activo` tinyint(1) NOT NULL DEFAULT 1,  PRIMARY KEY (`id`)  ) ENGINE=InnoDB ; |



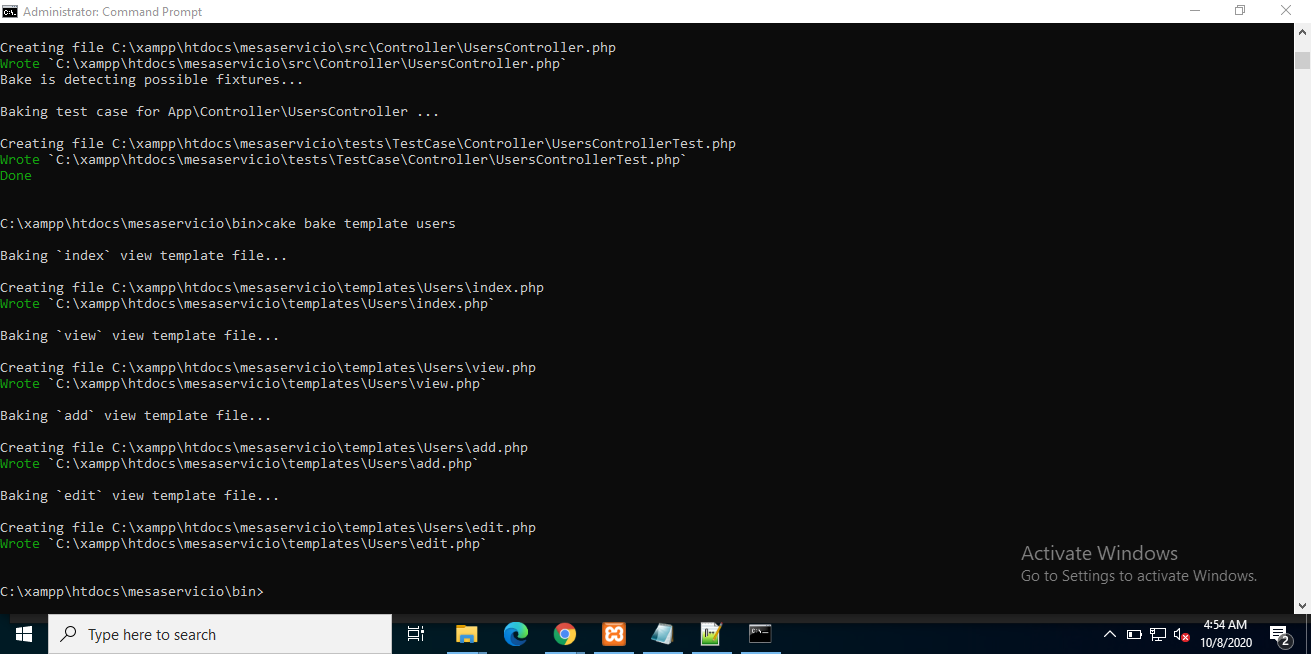
Luego ejecutamos los siguientes códigos dentro de la carpeta bin de cakephp que se encuentra en la siguiente ruta: C:\xampp\htdocs\mesaservicio\bin>**cake bake model users**

****

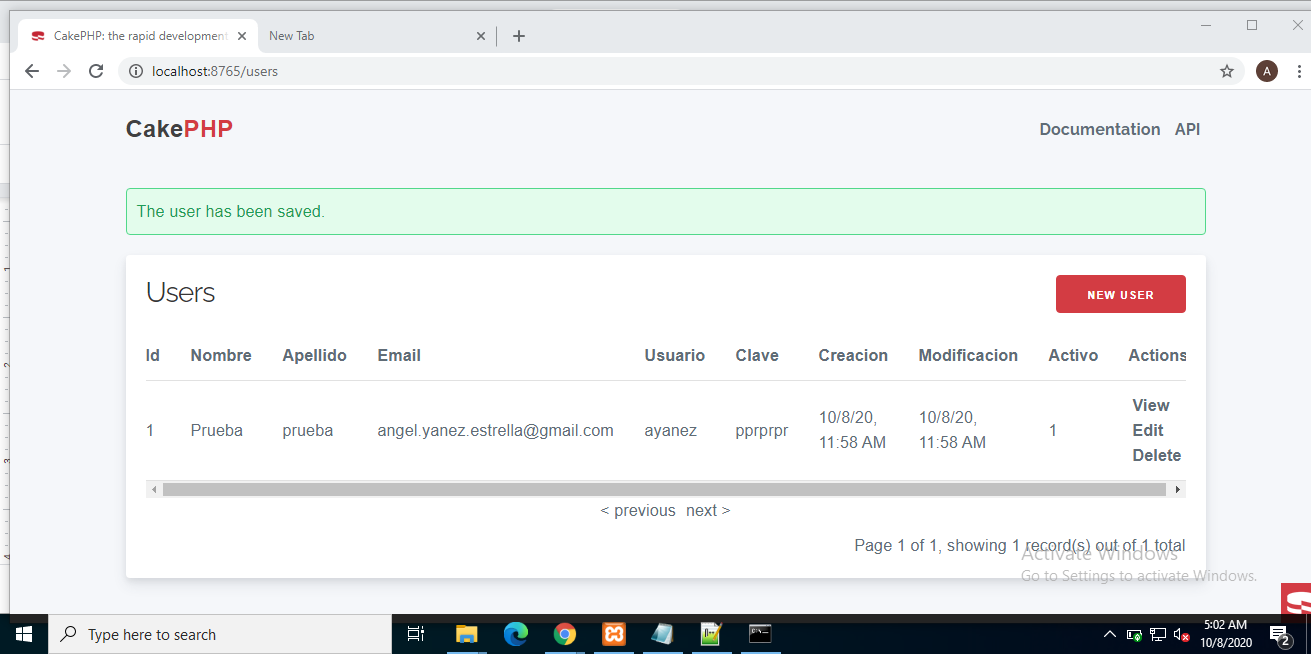
C:\xampp\htdocs\mesaservicio\bin>**cake bake controller users**

****

C:\xampp\htdocs\mesaservicio\bin>**cake bake template users**

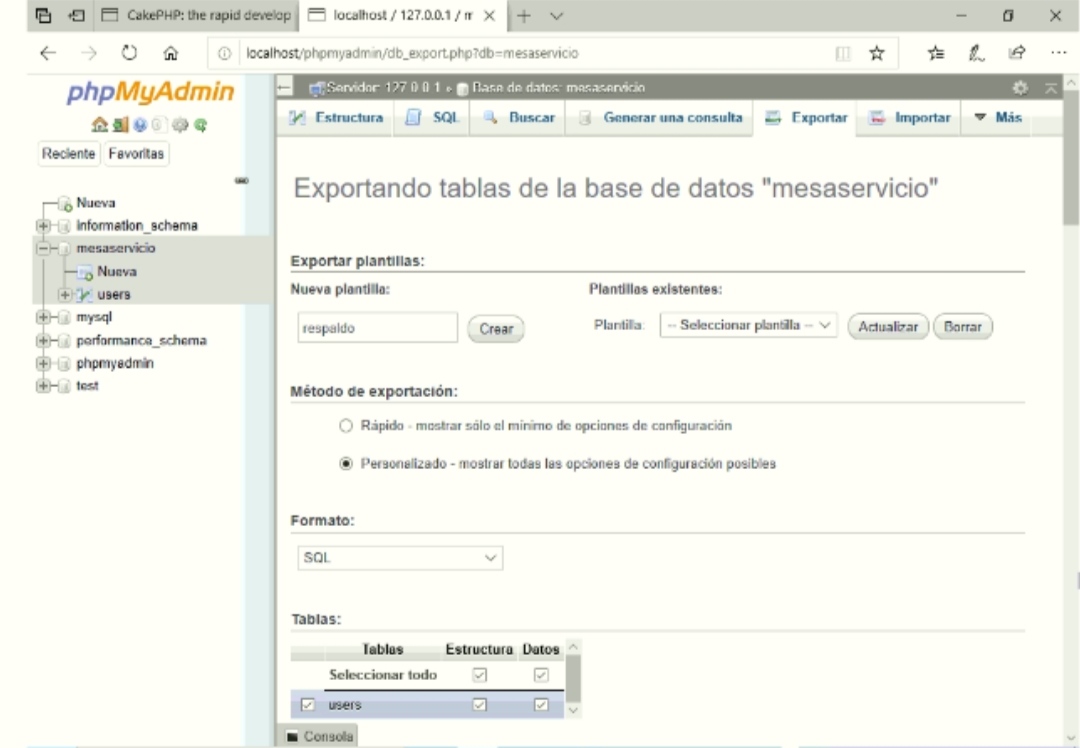
****

Luego accedemos a la página del CRUD creada <http://localhost:8765/users>

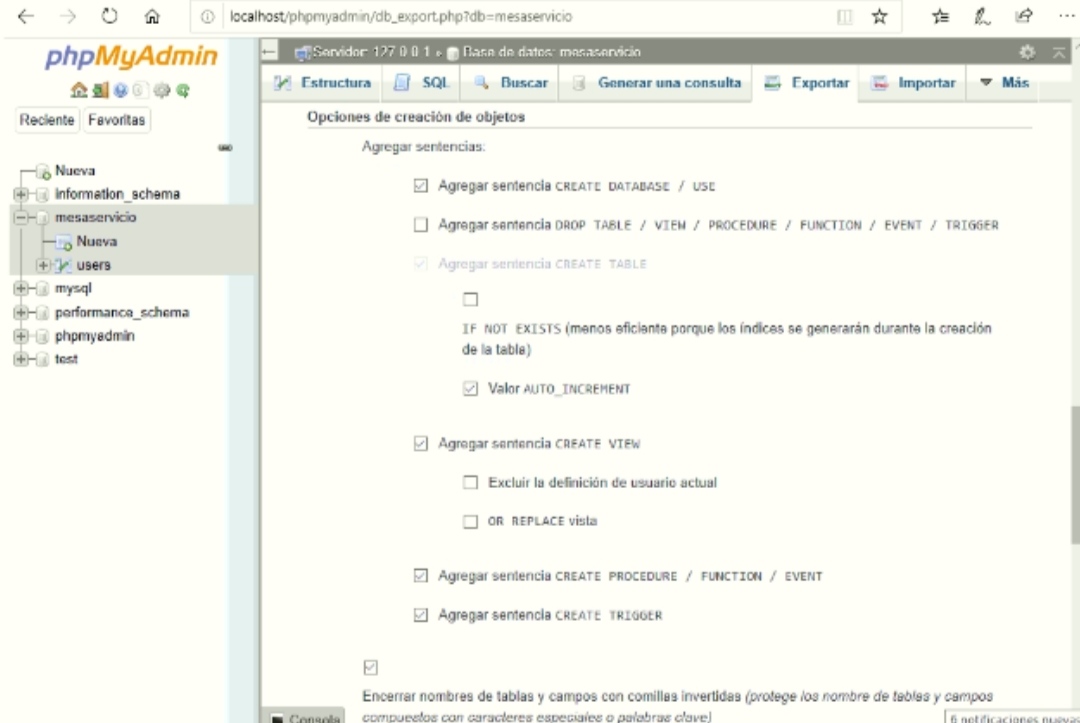


## Respaldo de la base de datos en phpMyAdmin

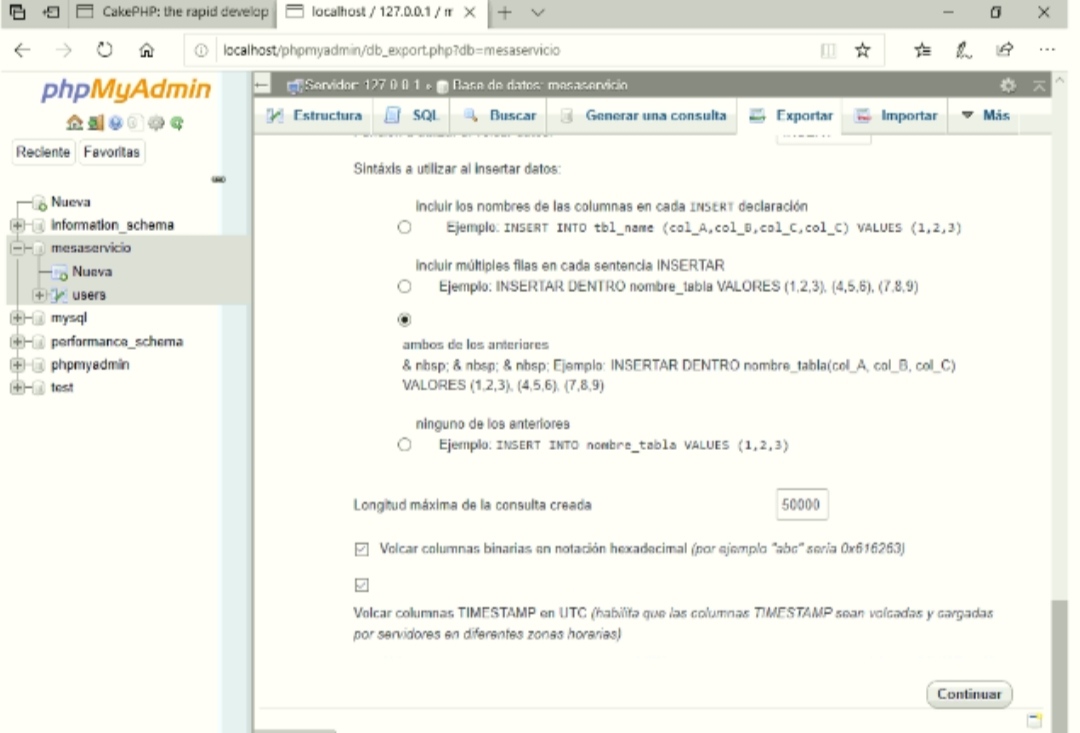
Dentro de phpMyAdmin seleccionamos la base de datos que creamos. Luego seleccionamos la pestaña **Exportar**. Aquí ingresamos el nombre de nuestro respaldo y seleccionamos el método de exportación **Personalizado**.



Luego seleccionamos dentro de Opciones de creación de objetos la opción **Agregar sentencia CREATE DATABASE / USE.**



Luego dejamos por defecto las demás opciones y damos clic en el botón **Continuar**.

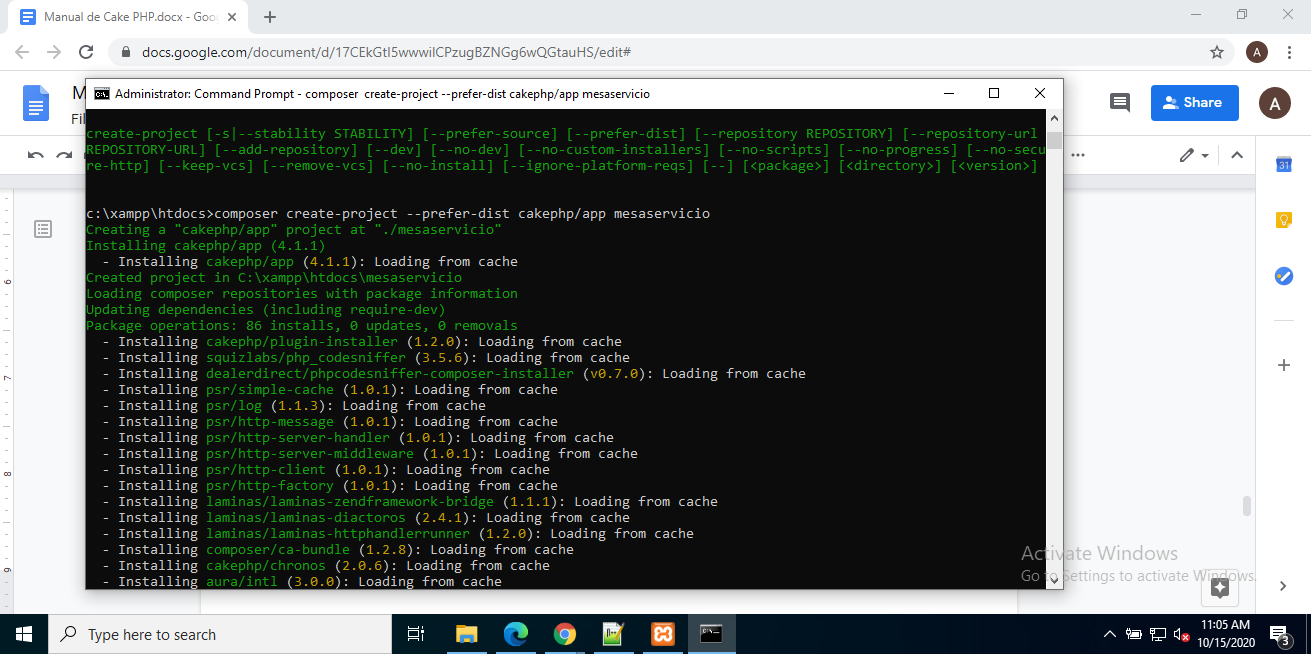


## Login del Proyecto

Vamos a utilizar el plugin cakedc que se encuentra en el siguiente link:

Creamos un nuevo proyecto mesaservicio mediante composer

**composer create-project --prefer-dist cakephp/app mesaservicio**



Se dan los permisos a la carpeta aplastando y cuando está finalizando la descarga de los componentes de composer.

Configuramos el Datasource en el archivo **app\_local.php** que se encuentra en la siguiente ruta: **C:\xampp\htdocs\mesaservicio\config**

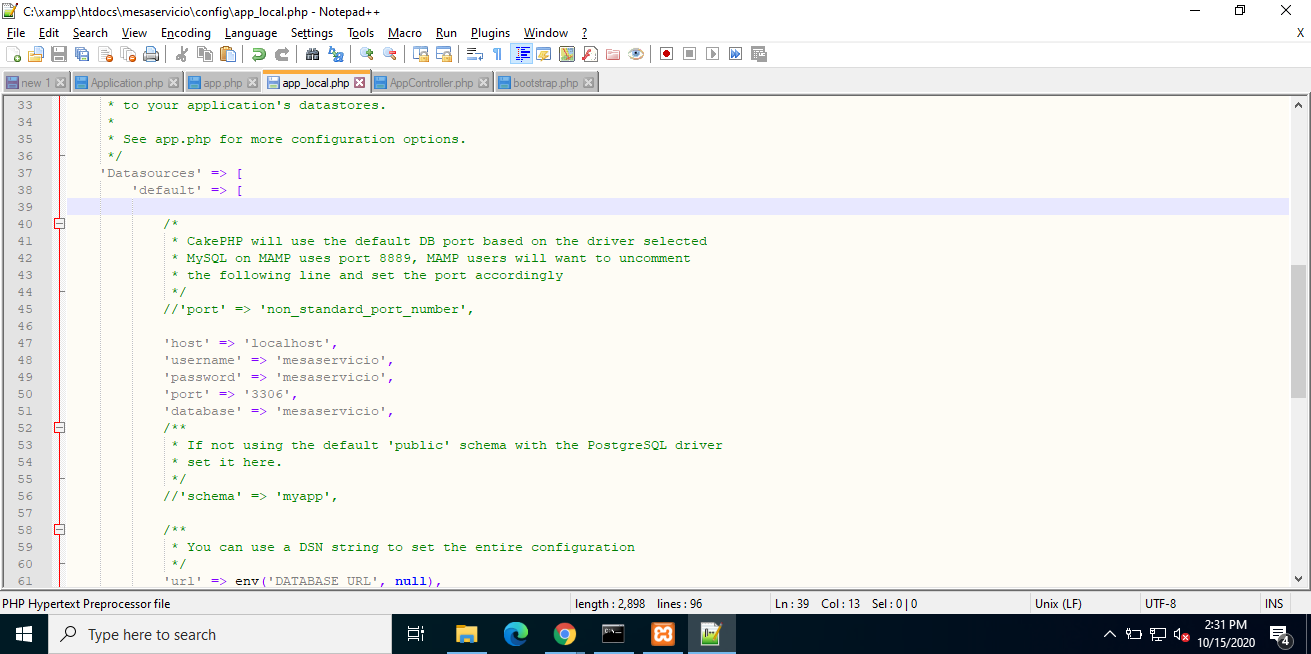
'host' => 'localhost',

'username' => 'mesaservicio',

'password' => 'mesaservicio',

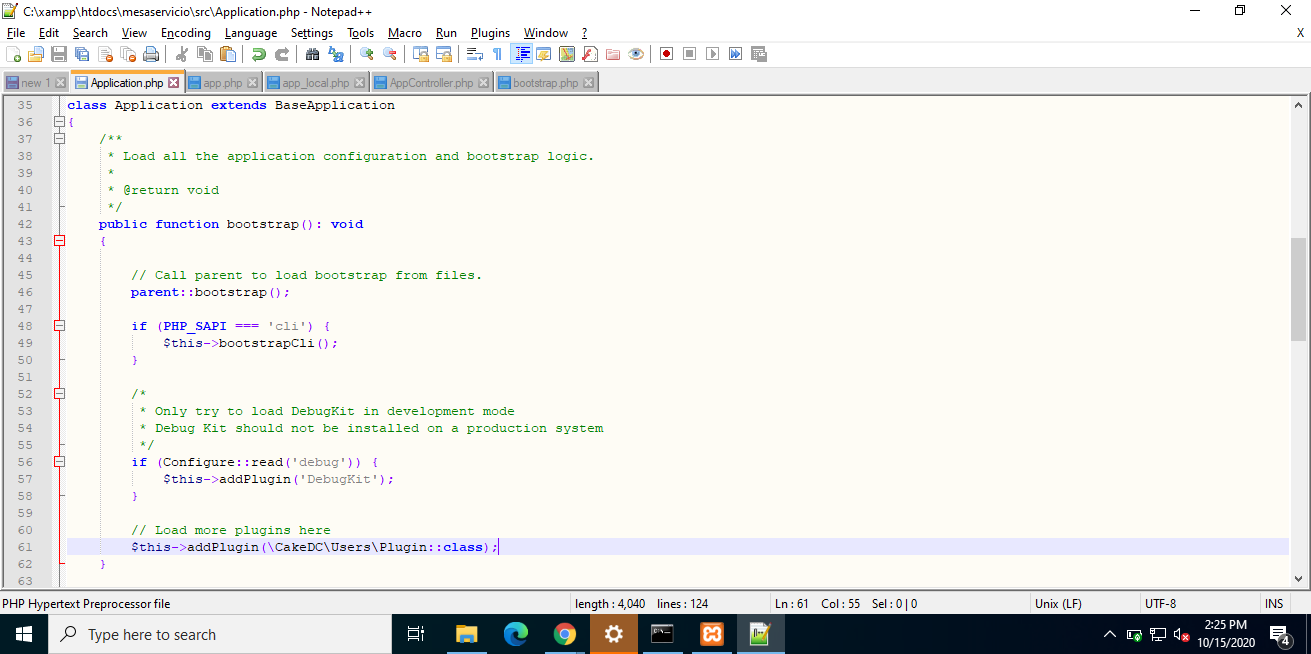
'port' => '3306',

'database' => 'mesaservicio',



Agregamos el plugin dentro de la aplicación en el archivo **Application.php** que se encuentra en la siguiente ruta: **C:\xampp\htdocs\mesaservicio\src**

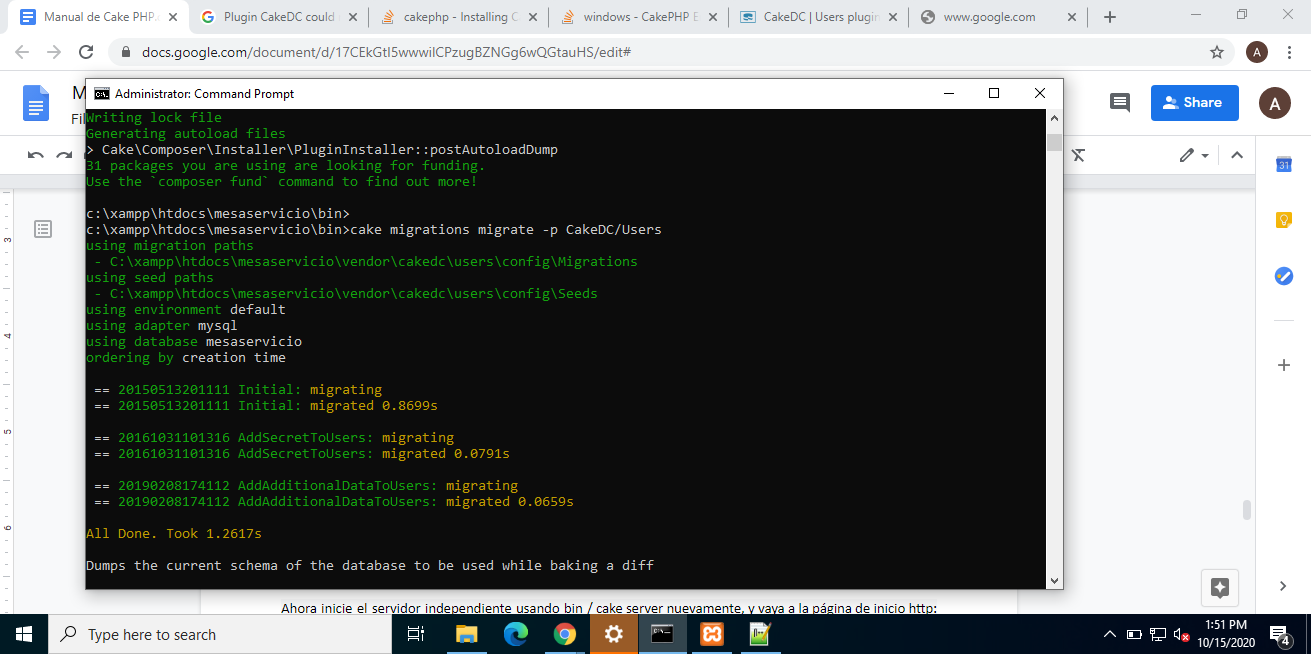
Agregamos la siguiente linea: **$this->addPlugin(\CakeDC\Users\Plugin::class);**



Instalamos el plugin mediante el siguiente comando: **composer require cakedc/users:^9**

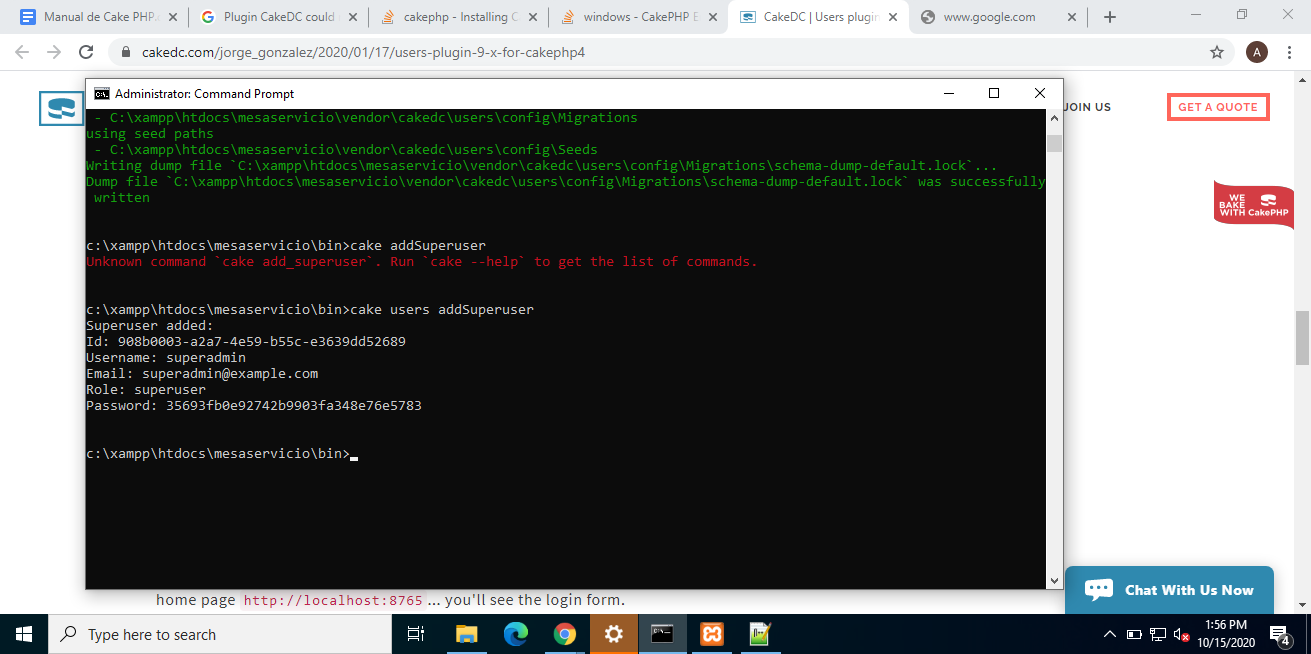
Creamos las tablas requeridas en su fuente de datos configurada usando el comando migrate:

**cake migrations migrate -p CakeDC/Users** en la siguiente ruta **c:\xampp\htdocs\mesaservicio\bin>**

****

Esto creará una tabla **ʻusers**` vacía y `**social\_accounts**` para guardar sus datos de autenticación.

Ahora creamos un usuario superadministrador mediante el siguiente comando dentro de la carpeta bin: **cake user addSuperuser**



Superuser added:

Id: 9ed5a9ac-f87d-4499-8967-8816b0db646c

**Username:** superadmin

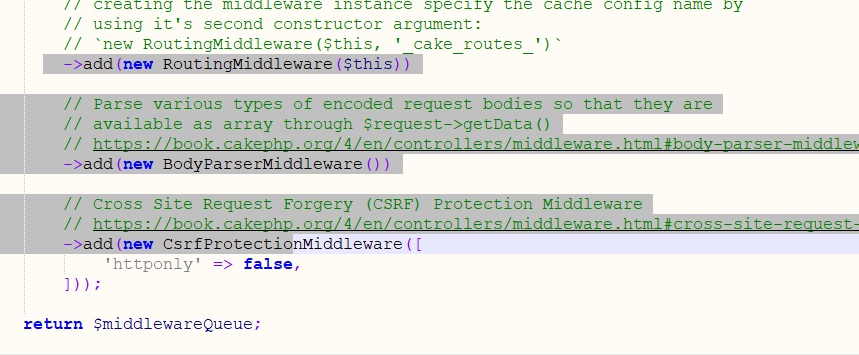
**Email:** superadmin@example.com

**Role:** superuser

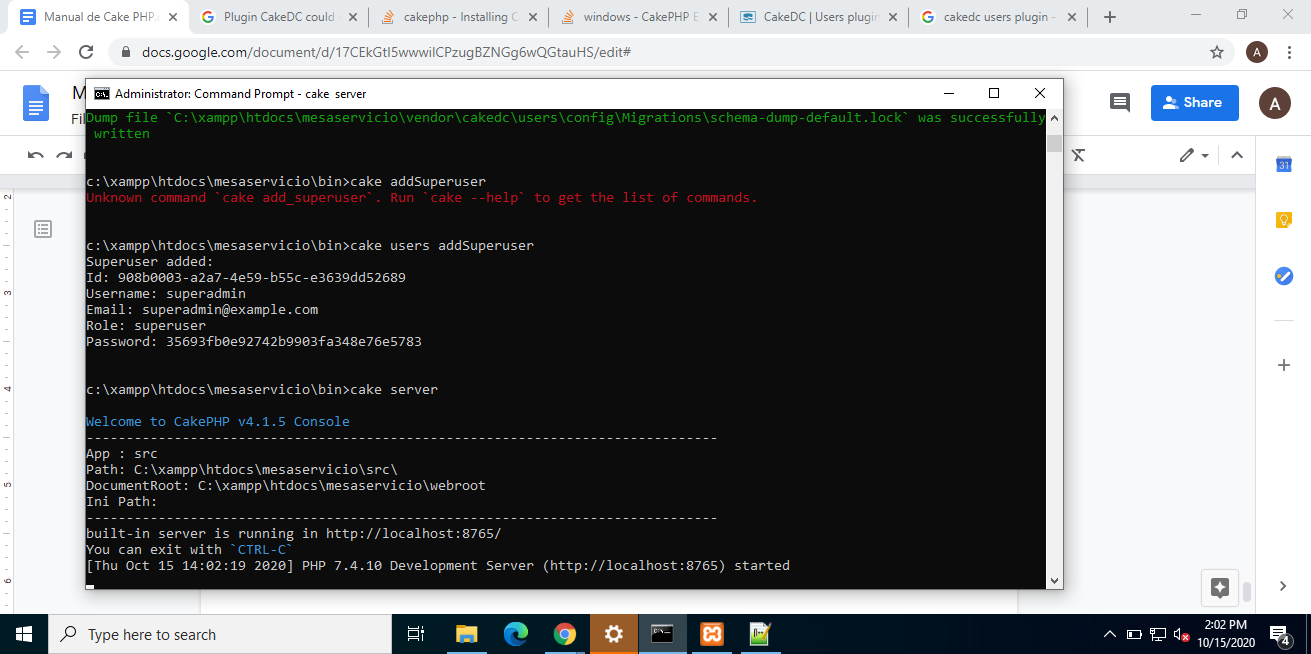
**Password:** dbd4ce512cbc4838a318a25993d6ea38

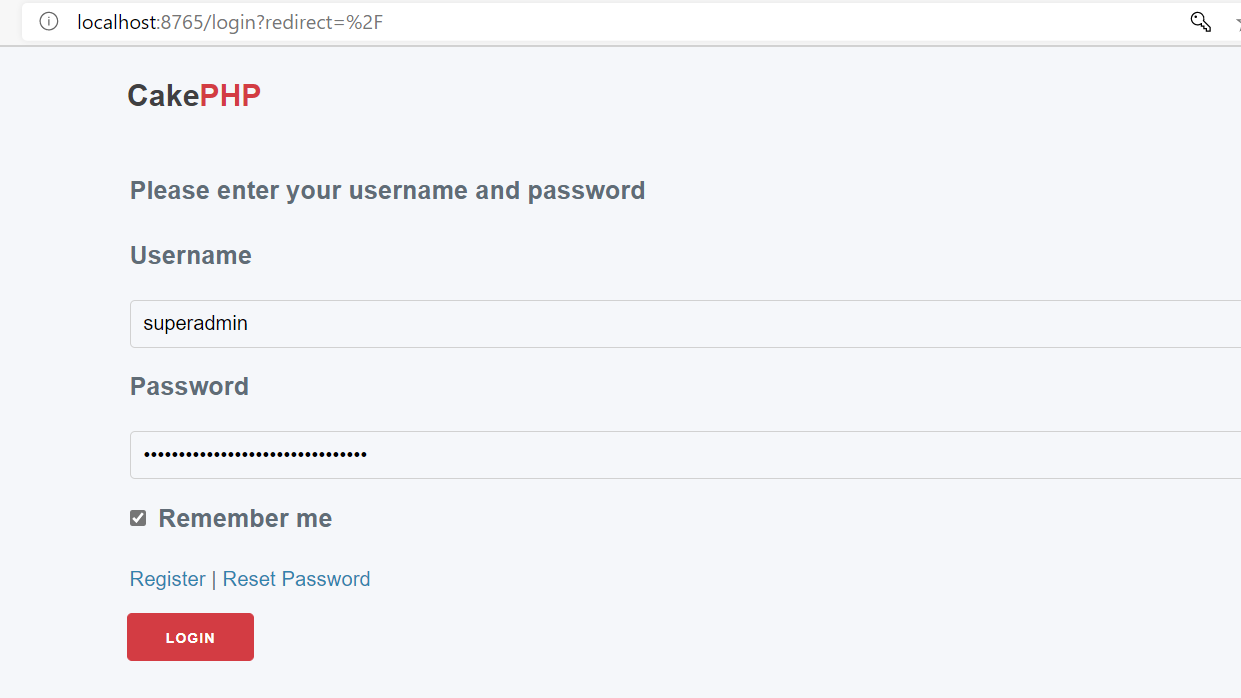
Se creará un usuario superadministrador y una contraseña aleatoria, copiará el usuario y la contraseña a un lugar seguro.

Luego modificamos en la sección httponly a false para deshabilitar la protección **CRSF** en el archivo **Application.php** que se encuentra en la ruta **C:\xampp\htdocs\mesaservicio\src**



Ahora inicie el servidor independiente usando **cake server** en la ruta c:\xampp\htdocs\mesaservicio\bin nuevamente, y vaya a la página de inicio **http: // localhost: 8765** ... verá el formulario de inicio de sesión.





Luego modificamos la página principal del sistema. Modificamos el archivo **home.php** que se encuentra en la ruta **C:\xampp\htdocs\mesaservicio\templates\Pages** con un menú y un botón de cerrar sesión.

**NOTA**: CakeDC pone por defecto el archivo home.php como página principal cuando se accede al sistema.

***Modificar página de cambio de clave de CakeDC con textos personalizados***

Modificamos el archivo **change\_password.php** que se encuentra en la siguiente ruta:

C:\xampp\htdocs\mesaservicio\vendor\cakedc\users\templates\Users

***Modificar página de login del sitio de CakeDC con textos personalizados***

Modificamos el archivo **login.php** que se encuentra en la siguiente ruta:

C:\xampp\htdocs\mesaservicio\vendor\cakedc\users\templates\Users

**BootStrap dentro del Proyecto**

La última versión de bootstrap lo podemos descargar del siguiente link: <https://getbootstrap.com/>

Agregamos bootstrap dentro de las diferentes partes del proyecto. Copiamos en las siguientes carpetas:

|  |  |
| --- | --- |
| **Archivo** | **Carpeta** |
| bootstrap.min.css | C:\xampp\htdocs\mesaservicio\webroot\css |
| bootstrap.min.js | C:\xampp\htdocs\mesaservicio\webroot\js |

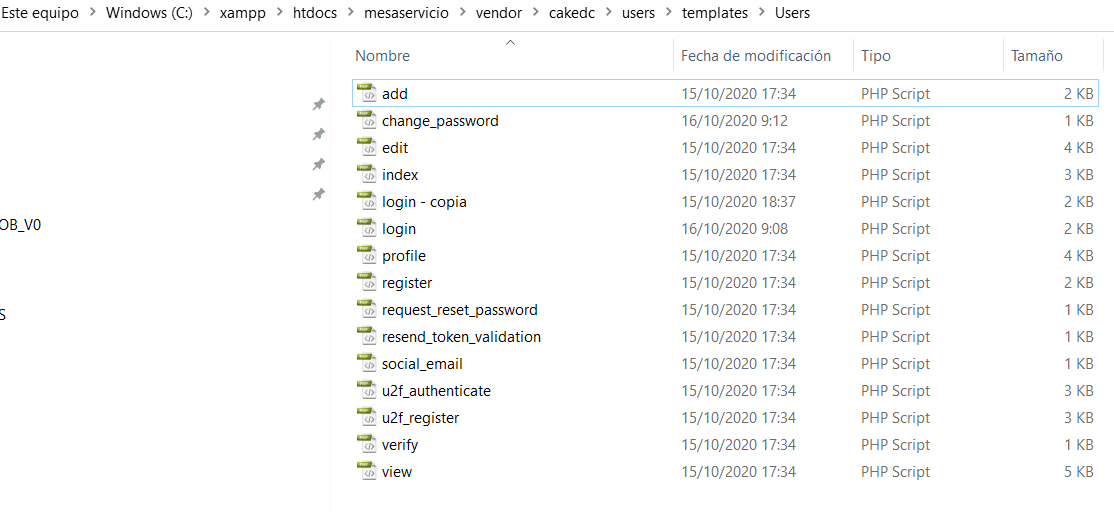
Luego agregamos dentro del archivo **home.php** la referencia a bootstrap.

Luego en el archivo home.php de la ruta agregamos lo siguiente para agregar bootstrap en el head de archivo



Luego modificamos el CRUD de Usuarios generados por la herramienta dentro de la siguiente ruta: **C:\xampp\htdocs\mesaservicio\vendor\cakedc\users\templates\Users** y modificamos cada uno de los archivos generados.

Hay que copiar el archivo **permissions.php** del cakedc que se encuentra en la ruta **C:\xampp\htdocs\mesaservicio\vendor\cakedc\users\config** a la ruta de la estructura principal del proyecto **C:\xampp\htdocs\mesaservicio\config**



Copiamos el archivo permission.php que se encuentra en la ruta **C:\xampp\htdocs\mesaservicio\vendor\cakedc\users\config** a la ruta **C:\xampp\htdocs\mesaservicio\config.** Luego iniciamos nuevamente el aplicativo

