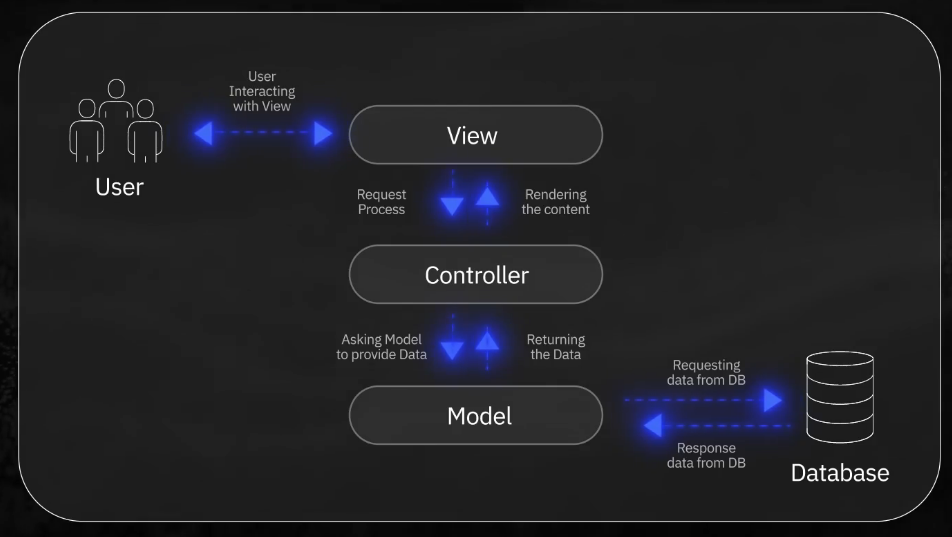
**Curso de PHP y Yii2**

Texto

Descripción generada automáticamente <https://www.yiiframework.com/doc/guide/2.0/es>

**Arquitectura modelo–vista–controlador**



Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Usar MySQL que viene con XAMPP es una excelente opción para simplificar tu entorno de desarrollo. Vamos a proceder con la instalación de Yii2 usando MySQL en tu configuración existente de XAMPP. Aquí tienes los pasos a seguir:

1. Asegúrate de que XAMPP está funcionando correctamente:
   * Inicia Apache y MySQL desde el panel de control de XAMPP.
   * Verifica que puedes acceder a <http://localhost> y a <http://localhost/phpmyadmin>
2. Instalar Composer (si aún no lo tienes):
   * Descarga e instala Composer desde <https://getcomposer.org/download/>
   * Asegúrate de seleccionar la opción para añadir Composer al PATH del sistema durante la instalación.
3. Instalar Yii2:
   * Abre una terminal (CMD o PowerShell)
   * Navega a la carpeta htdocs de XAMPP (usualmente C:\xampp\htdocs)
   * Ejecuta el siguiente comando para instalar Yii2 Basic Template:

**composer create-project --prefer-dist yiisoft/yii2-app-basic nombre-proyecto**

**composer create-project --prefer-dist yiisoft/yii2-app-basic thebookclub**

* + Reemplaza "nombre-proyecto" con el nombre que desees para tu proyecto.

1. Configurar la base de datos:

* Crea una nueva base de datos para tu proyecto usando phpMyAdmin (<http://localhost/phpmyadmin>)
* Edita el archivo config/db.php en tu proyecto Yii2:

**return [**

**'class' => 'yii\db\Connection',**

**'dsn' => 'mysql:host=localhost;dbname=nombre\_de\_tu\_base\_de\_datos',**

**'username' => 'root',**

**'password' => '',**

**'charset' => 'utf8',**

**];**

* Reemplaza 'nombre\_de\_tu\_base\_de\_datos' con el nombre de la base de datos que creaste.

1. Verificar la instalación:

* Abre un navegador y ve a <http://localhost/nombre-proyecto/web>
* Deberías ver la página de bienvenida de Yii2

1. Configurar el entorno de desarrollo (opcional pero recomendado):

* Copia el archivo web/.htaccess.example a web/.htaccess
* Edita el archivo web/.htaccess y descomenta las líneas para habilitar URLs amigables

1. Instalar extensiones adicionales de Yii2 (según necesidades):

* Puedes instalar extensiones adicionales usando Composer, por ejemplo:

**composer require --prefer-dist yiisoft/yii2-bootstrap4**

1. Configurar un IDE (recomendado):

* PhpStorm es excelente para desarrollo con Yii2, pero es de pago.
* Visual Studio Code con extensiones PHP es una buena opción gratuita.

En **php.ini** copie el contenido de php.ini-development y luego habilite

extension=fileinfo

extension=mbstring

extension=exif ; Must be after mbstring as it depends on it

extension=mysqli

extension=openssl

extension=pdo\_mysql

extension=gd ; Para manipulación de imágenes

extension=intl ; Para internacionalización

extension=curl ; Para hacer peticiones HTTP

extension=pdo\_sqlite ; Si planeas usar SQLite

**cambiar los puertos de MySQL y Apache**:

Para Apache:

* Abre el archivo d:\xampp\apache\conf\httpd.conf
* Busca la línea Listen 80 y cámbiala por el puerto que desees (**port=8085**), por ejemplo Listen 8080

Para MySQL:

* Abre el archivo d:\xampp\mysql\bin\my.ini
* Busca la línea port=3306 y cámbiala por el puerto que prefieras, por ejemplo **port=3307**

**Después de hacer estos cambios**:

* Guarda los archivos modificados
* Reinicia los servicios de Apache y MySQL desde el panel de control de XAMPP

Interfaz de usuario gráfica

Descripción generada automáticamente

**Definir usuario y clave de la BD**

D:\xampp\phpMyAdmin

Texto

Descripción generada automáticamente

**Crear usuario platzi en BD**

d:\xampp\mysql\bin

mysql -u root

CREATE USER ‘platzi'@'localhost' IDENTIFIED BY 'Yii2Rulez';

GRANT ALL PRIVILEGES ON \*.\* TO 'platzi'@'localhost' WITH GRANT OPTION;

FLUSH PRIVILEGES;

Cambiar clave de root

**ALTER USER 'root'@'localhost' IDENTIFIED BY 'platziRoot1029';**

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

MariaDB [(none)]> SELECT user, host FROM mysql.user;

+--------+-----------+

| User | Host |

+--------+-----------+

| root | 127.0.0.1 |

| root | ::1 |

| platzi | localhost |

| pma | localhost |

| root | localhost |

+--------+-----------+

5 rows in set (0.094 sec)

MariaDB [(none)]> EXIT;

Bye

principal@DESKTOP-CAVELI d:\xampp\mysql\bin

#

<http://localhost:8085/phpmyadmin/>

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente con confianza media

Seleccionamos la pestaña SQL, se copia el contenido de schema.sql

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Se da clic en el botón de Continuar

Texto, Aplicación

Descripción generada automáticamente

Se crea la BD y las tablas según el script

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Tabla

Descripción generada automáticamente

**Configuración adicional recomendada en php.ini:**

display\_errors = On ; Muestra errores (solo para desarrollo)

error\_reporting = E\_ALL ; Reporta todos los errores

post\_max\_size = 20M ; Aumenta el tamaño máximo de POST si es necesario

upload\_max\_filesize = 20M ; Aumenta el tamaño máximo de subida de archivos

max\_execution\_time = 300 ; Aumenta el tiempo máximo de ejecución si es necesario

memory\_limit = 256M ; Aumenta el límite de memoria si es necesario

**Para Yii2 específicamente:**

* Asegúrate de que mod\_rewrite esté habilitado en Apache para URLs amigables
* En httpd.conf, busca la sección <Directory "d:/xampp/htdocs"> y asegúrate de que AllowOverride esté configurado como All

**Verifica la configuración:**

* Crea un archivo phpinfo.php en C:\xampp\htdocs con el contenido:

<?php phpinfo(); ?>

* Abre http://localhost:8085/phpinfo.php en tu navegador para ver qué extensiones están habilitadas y otras configuraciones

Para **asegurarte de que mod\_rewrite esté habilitado en Apache** para URLs amigables en XAMPP, sigue estos pasos:

1. Verificar si mod\_rewrite está habilitado:
   * Abre el archivo C:\xampp\apache\conf\httpd.conf con un editor de texto.
   * Busca la línea LoadModule rewrite\_module modules/mod\_rewrite.so
   * Si la línea está comentada (comienza con #), descoméntala quitando el #.
2. Permitir el uso de .htaccess:
   * En el mismo archivo httpd.conf, busca la sección <Directory "C:/xampp/htdocs">.
   * Asegúrate de que AllowOverride esté configurado como All:

apache

Copy

<Directory "C:/xampp/htdocs">

Options Indexes FollowSymLinks Includes ExecCGI

AllowOverride All

Require all granted

</Directory>

1. Configurar el archivo .htaccess de Yii2:
   * Ve a la carpeta web de tu proyecto Yii2 (ej: C:\xampp\htdocs\tu-proyecto-yii2\web).
   * Si no existe un archivo .htaccess, crea uno nuevo.
   * Añade el siguiente contenido:

RewriteEngine on

# If a directory or a file exists, use it directly

RewriteCond %{REQUEST\_FILENAME} !-f

RewriteCond %{REQUEST\_FILENAME} !-d

# Otherwise forward it to index.php

RewriteRule . index.php

1. Configurar Yii2 para usar URLs amigables:
   * Abre el archivo config/web.php de tu proyecto Yii2.
   * Busca la configuración de urlManager y modifícala así:

'urlManager' => [

'enablePrettyUrl' => true,

'showScriptName' => false,

'rules' => [

*// Tus reglas de URL aquí*

],

],

1. Reiniciar Apache:
   * Abre el panel de control de XAMPP.
   * Detén y vuelve a iniciar el servicio de Apache.
2. Probar las URLs amigables:
   * Abre tu navegador y accede a tu aplicación Yii2.
   * Intenta navegar a diferentes partes de tu aplicación.
   * Las URLs deberían verse "amigables" sin "index.php" en ellas.
3. Solución de problemas:
   * Si encuentras un error 500, verifica los logs de Apache en C:\xampp\apache\logs\error.log.
   * Asegúrate de que todos los pasos anteriores se hayan seguido correctamente.
4. Nota sobre permisos:
   * En Windows, generalmente no necesitas preocuparte por los permisos de archivos.
   * Sin embargo, si tienes problemas, asegúrate de que el usuario de Apache tenga permisos de lectura en la carpeta de tu proyecto.

Después de seguir estos pasos, deberías tener URLs amigables funcionando en tu proyecto Yii2. Si encuentras algún problema o tienes preguntas adicionales, no dudes en preguntar.

D:\xampp\htdocs>php --version

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

D:\xampp\htdocs>composer -V

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**composer create-project --prefer-dist yiisoft/yii2-app-basic thebookclub**

Se crea D:\xampp\htdocs\thebookclub

Interfaz de usuario gráfica, Aplicación, Correo electrónico

Descripción generada automáticamente

Token Git: ghp\_5eAKv9lTztQFxxc4zwuIkbBgxr5RWp2EsdTu

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Texto

Descripción generada automáticamente

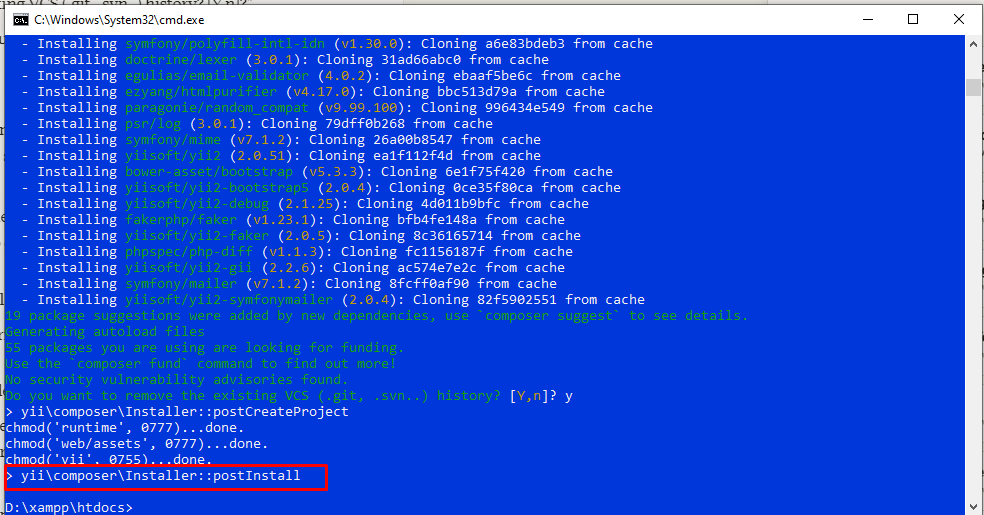
**Composer está intentando acceder a repositorios de GitHub y necesita autenticación**. Dado que estás trabajando con Yii2, que es un proyecto público, solo necesitas un token con permisos de lectura para repositorios públicos. Aquí te guío paso a paso:

1. Generar un token de GitHub:
   * Haz clic en el primer enlace proporcionado en el mensaje: <https://github.com/settings/tokens/new?scopes=&description=Composer+on+DESKTOP-CAVELI+2024-08-23+0208>
   * Esto te llevará a la página de GitHub para crear un nuevo token de acceso personal.
2. En la página de GitHub:
   * Asegúrate de estar conectado a tu cuenta de GitHub.
   * Verás un formulario para crear un nuevo token.
   * El nombre del token ya debería estar rellenado.
   * No necesitas seleccionar ningún scope adicional para repositorios públicos.
3. Generar el token:
   * Desplázate hasta el final de la página y haz clic en "Generate token".
4. Copiar el token:
   * GitHub te mostrará el token generado. Cópialo inmediatamente, ya que no podrás verlo de nuevo.
5. Volver a la línea de comandos:
   * Pega el token que copiaste cuando Composer te pida "Token (hidden):".
   * Presiona Enter.
6. Composer guardará este token:
   * El token se guardará en "C:/Users/principal/AppData/Roaming/Composer/auth.json" para uso futuro.
7. Continuar con la instalación:
   * Composer debería ahora poder continuar con la instalación de los paquetes necesarios.

Notas importantes:

* Este token tiene permisos de solo lectura para información pública, lo cual es suficiente para instalar Yii2 y otros paquetes públicos.
* Mantén este token seguro. Aunque solo tiene acceso a información pública, es una buena práctica de seguridad no compartirlo.
* Si en el futuro necesitas acceder a repositorios privados, tendrías que generar un nuevo token con permisos más amplios.

Una vez que hayas proporcionado el token, Composer debería poder completar la instalación de Yii2 y sus dependencias.



Recomendación: Para un nuevo proyecto, responde "Y" o simplemente presiona Enter. Esto te dará un proyecto limpio sin el historial de git del template.

Después de responder:

1. La instalación se completará.
2. Navega a la carpeta de tu proyecto: cd thebookclub
3. Si deseas iniciar un nuevo repositorio git para tu proyecto:

git init

git add .

git commit -m "Inicio del proyecto The Book Club con Yii2"

**Ejecutar Servidor Yii**

D:\xampp\htdocs\thebookclub>**yii serve**

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

<http://localhost:8080/thebookclub/web> | http://localhost:8080/index.php

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Una vez que se ha creado un nuevo proyecto usando composer y yii2-basic-project, es importante seguir la estructura de directorios y convenciones que Yii2 propone. Esto implica:

1. Colocar los modelos en el directorio models/
2. Poner los controladores en controllers/
3. Ubicar las vistas en views/
4. Situar los componentes en components/
5. Colocar los módulos (si los hay) en modules/
6. Poner los assets (recursos como CSS, JS) en assets/

Seguir esta estructura predeterminada es crucial porque:

* Facilita la organización del código
* Mejora la legibilidad y mantenibilidad del proyecto
* Permite que Yii2 funcione correctamente, ya que busca ciertas clases en ubicaciones específicas
* Hace que el proyecto sea más fácil de entender para otros desarrolladores familiarizados con Yii2

<https://www.random.org/strings/?num=10&len=10&digits=on&upperalpha=on&loweralpha=on&unique=on&format=html&rnd=new>

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

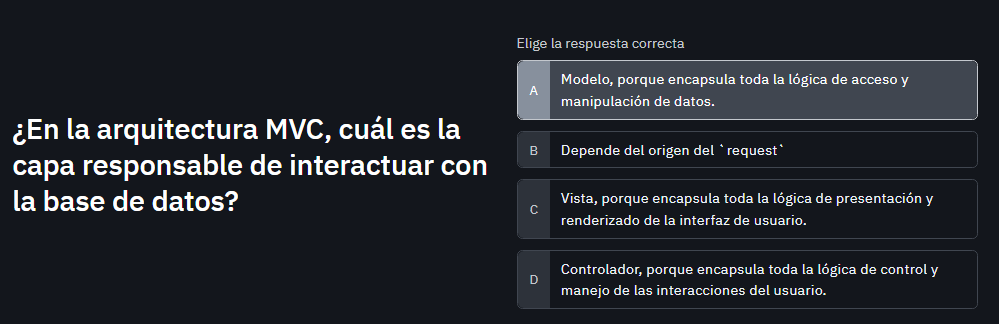
Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**db.php**





Captura de pantalla de un celular

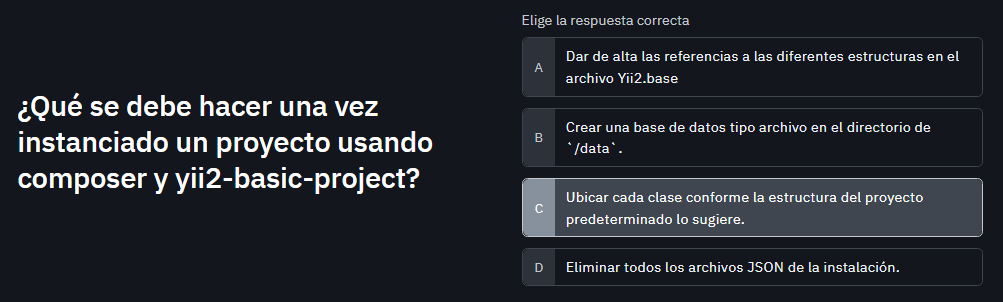
Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

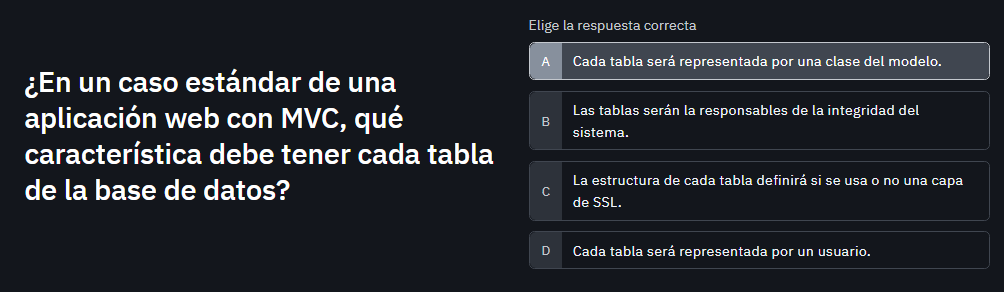
Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente



Esta es una característica fundamental del patrón Modelo-Vista-Controlador (MVC) y de los frameworks que lo implementan, como Yii2:

1. Representación en el modelo: Cada tabla en la base de datos se mapea a una clase de modelo en la aplicación. Esta clase encapsula la lógica de negocio y las operaciones de datos relacionadas con esa entidad específica.
2. Abstracción de la base de datos: El modelo proporciona una capa de abstracción entre la aplicación y la base de datos, permitiendo trabajar con objetos en lugar de consultas SQL directas.
3. Encapsulación de la lógica: La clase del modelo contiene métodos para realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) y cualquier otra lógica específica de esa entidad.
4. Validación de datos: Los modelos en Yii2 también manejan la validación de datos antes de la interacción con la base de datos.
5. Relaciones entre tablas: Las relaciones entre tablas (como uno a muchos, muchos a muchos) se definen en estas clases de modelo.



Tener trazabilidad al registrar exactamente cuándo se creó o modificó cada registro. Ayuda en la depuración y resolución de problemas al identificar el momento preciso de los cambios; permite ordenar y filtrar registros por fecha. Es útil para cuando se modifican usuarios, por ejemplo.

Esta es la utilidad principal de las columnas de timestamp como created\_at y modified\_at:

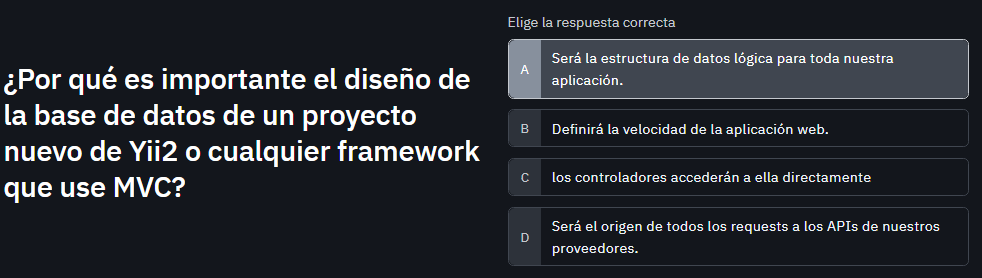
1. Trazabilidad: Permiten rastrear cuándo se creó o modificó por última vez cada registro.
2. Depuración: Facilitan la identificación de cuándo ocurrieron cambios específicos, lo que es crucial para resolver problemas.
3. Ordenamiento y filtrado: Permiten ordenar y filtrar registros basándose en su fecha de creación o modificación.
4. Auditoría: Son útiles para mantener un registro de auditoría de los cambios en la base de datos.
5. Lógica de negocio: Pueden utilizarse en la lógica de la aplicación, por ejemplo, para mostrar contenido reciente o identificar registros inactivos.
6. Sincronización de datos: Son útiles en sistemas que requieren sincronización de datos entre diferentes bases de datos o aplicaciones.

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

Esta es la razón fundamental por la que el diseño de la base de datos es crucial en un proyecto MVC como Yii2:

1. Fundamento estructural: La base de datos define la estructura fundamental de los datos que manejará la aplicación.
2. Reflejo del modelo de dominio: Representa el modelo de dominio de la aplicación, influyendo directamente en cómo se organizan y relacionan los datos.
3. Impacto en los modelos: En MVC, los modelos suelen corresponder directamente con las tablas de la base de datos, por lo que su diseño afecta la lógica de negocio.
4. Eficiencia y rendimiento: Un buen diseño de base de datos es crucial para la eficiencia en el acceso y manipulación de datos.
5. Escalabilidad: Un diseño sólido facilita la expansión futura de la aplicación.
6. Integridad de datos: Ayuda a mantener la consistencia y la integridad de los datos en toda la aplicación.



Platzicontroller.php

<?php

namespace app\commands;

use yii\console\Controller;

use yii\console\ExitCode;

/\*\* Comando para la clase de prueba \*/

class PlatziController extends Controller {

    /\*\* Suma los valores de los dos parámetros \*/

    public function actionSuma($a, $b) {

        $result = $a + $b;

        printf("%f\n", $result);

        return ExitCode::OK; // 0 (EXIT\_SUCCESS)

    }

}

If you trust this command, run the following command instead:

PS D:\xampp\htdocs\thebookclub> .\yii platzi/suma 10 5

15.000000

PS D:\xampp\htdocs\thebookclub> php yii platzi/suma 10 5

15.000000

PS D:\xampp\htdocs\thebookclub>

Imagen que contiene Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Book.php

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Author.php

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

**BookController.php**

    public function actionAll()

    {

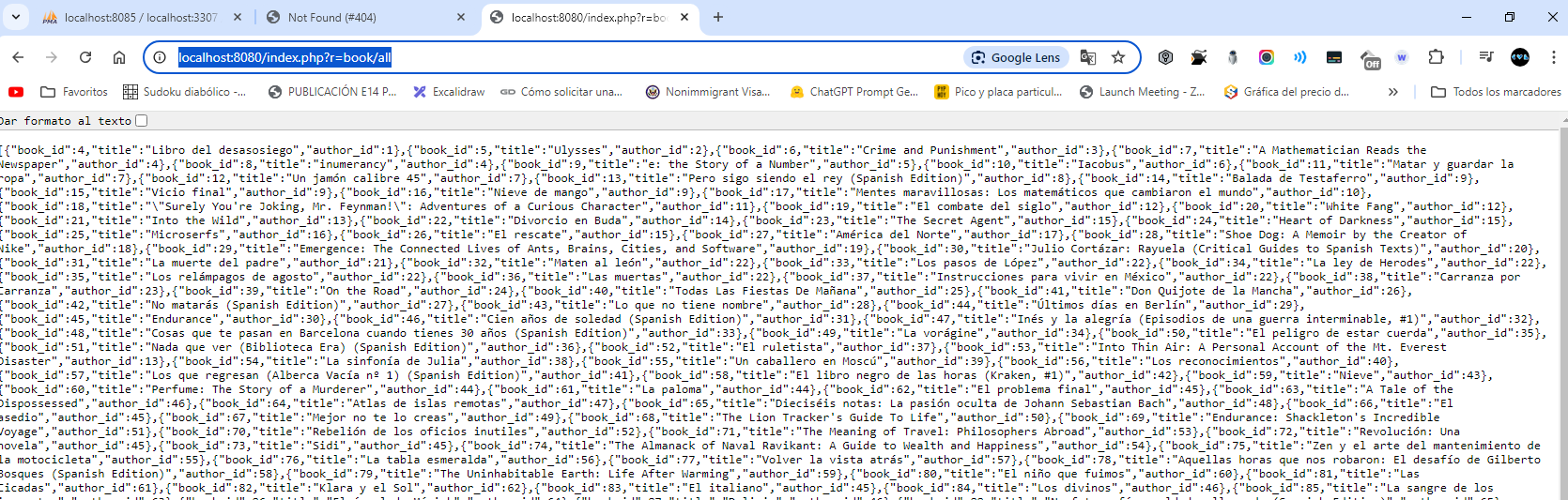
        \Yii::$app->response->format = \yii\web\Response::FORMAT\_JSON;

        $books = Book::find()->all();

        return $books;

    }

<http://localhost:8080/index.php?r=book/all>



<http://localhost:8080/index.php?r=author/detail&id=43>

    public function actionDetail($id) {

        $book = Book::findOne($id);

        if(empty($book)) {

            return $this->redirect(['site/index']);

        }

        return $book->title;

    }

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

<http://localhost:8080/index.php?r=book/detail&id=8>

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

<http://localhost:8080/index.php?r=book/detail&id=555555555555>

    public function actionDetail($id) {

        $book = Book::findOne($id);

        if(empty($book)) {

            Yii::$app->session->setFlash('error', 'ese libro no existe');

            return $this->goHome();

        }

        return $book->toString();

    }

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

<http://localhost:8080/index.php?r=author/all>

  public function actionAll()

  {

      \Yii::$app->response->format = \yii\web\Response::FORMAT\_JSON;

      $authors = Author::find()->all();

      return $authors;

  }

Texto

Descripción generada automáticamente

<http://localhost:8080/index.php?r=author/all&search=Jack>

  public function actionAll($search = null) {

    \Yii::$app->response->format = \yii\web\Response::FORMAT\_JSON;

    if($search !=  null) {

      $authors = Author::find()

        ->where(['like', 'name', $search])

        ->all();

    } else {

      $authors = Author::find()->all();

    }

    return $authors;

  }

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Enrutamiento y creación de URLS con urlManager**

<http://localhost:8080/books>

Texto, Pizarra

Descripción generada automáticamente

Gráfico de dispersión

Descripción generada automáticamente

<http://localhost:8080/book/12>

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

<http://localhost:8080/author/search/Gamboa>

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

<http://localhost:8080/author/10>

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

<http://localhost:8080/author/4>

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

<http://localhost:8080/author/4>

<?php

use yii\helpers\Html;

?>

<h1><?= $authorvw->toString() ?></h1>

<h2>Libros:</h2>

<ol>

    <?php foreach($authorvw->books as $book) {?>

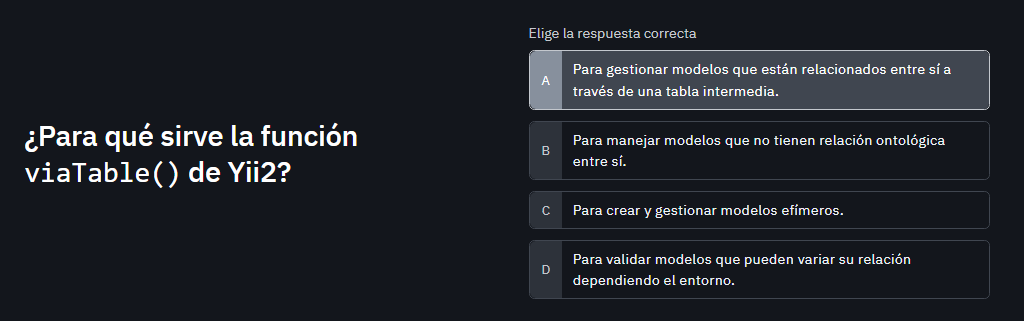
        <li><?= Html::a($book->title, ['book/detail', 'id' => $book->id]) ?></li>

    <?php } ?>

</ol>

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente



Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

**Despliegue de aplicación de Yii2 a Heroku**