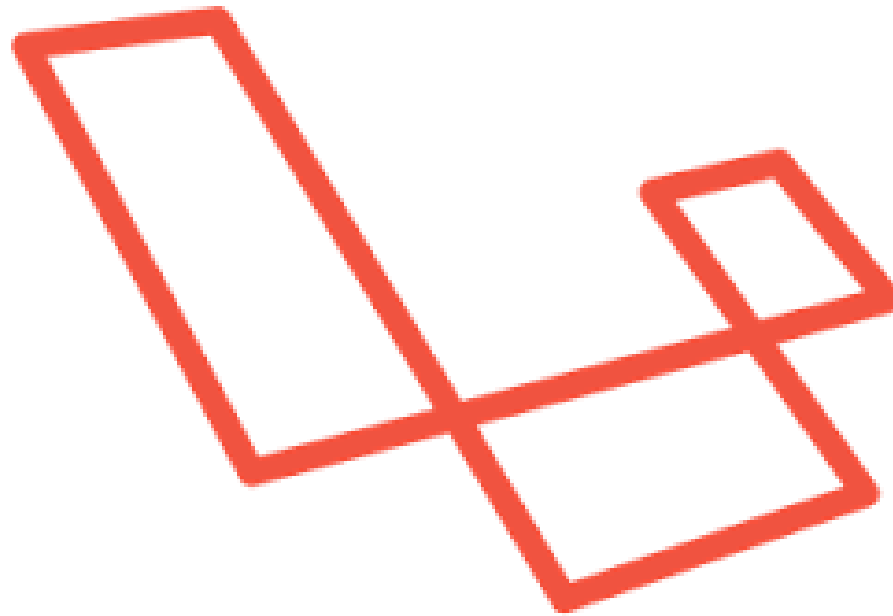


Curso de Laravel



laravel

Capítulo 2: Contexto MVC de Laravel



Laravel: Arquitectura

Estructura de directorios

- app directorio principal de los fuentes de la aplicación
- config directorio de configuraciones
- database directorio con los esquemas de la DB
- public directorio público de la aplicación
- resources tiene 3 subdirectorios
 - views directorio de las vistas de la app
 - lang directorio para soporte multi idioma
 - assets directorio para almacenar fuentes css less o sass



Laravel: Arquitectura

Estructura de directorios

- bootstrap incluye el código que procesa cada petición del proyecto
- routes incluye los archivos que manejan las rutas del proyecto
- storage almacena elementos para la ejecución del proyecto tales como la caché, sesiones, vistas compiladas, logs, etc
- test directorio para la realización de pruebas unitarias con PHPUnit
- vendor directorio que almacena las librerías y dependencias del framework y terceros gestionados con composer



Laravel: Arquitectura

Archivos del framework

- `.env` Archivo que almacena las configuraciones de variables de entorno y otro (nombre de la app, credenciales de la conexión a la base de datos y otros)
- `composer.json` Archivo de configuración de composer. Acá se listan las dependencias del framework y se pueden añadir otras necesarias para el proyecto y que serán gestionadas por composer.



Laravel: Arquitectura

El directorio App

- El directorio App almacena los códigos fuentes del proyecto, las clases del modelo, controladores, filtros y rutas, el directorio principal es el Http, que contiene:
- `app/Http/Controllers` Contiene los archivos de los controladores, que a su vez contienen la lógica del negocio e interactúan con el modelo y la vista
- `app/Http/Middleware` Son los filtros o clases intermedias, que permite ejecutar rutinas antes o después que corra la lógica de los controladores o vistas.
- `app/Http/Kernel.php` Define los filtros a correr en cada petición



Laravel: Arquitectura

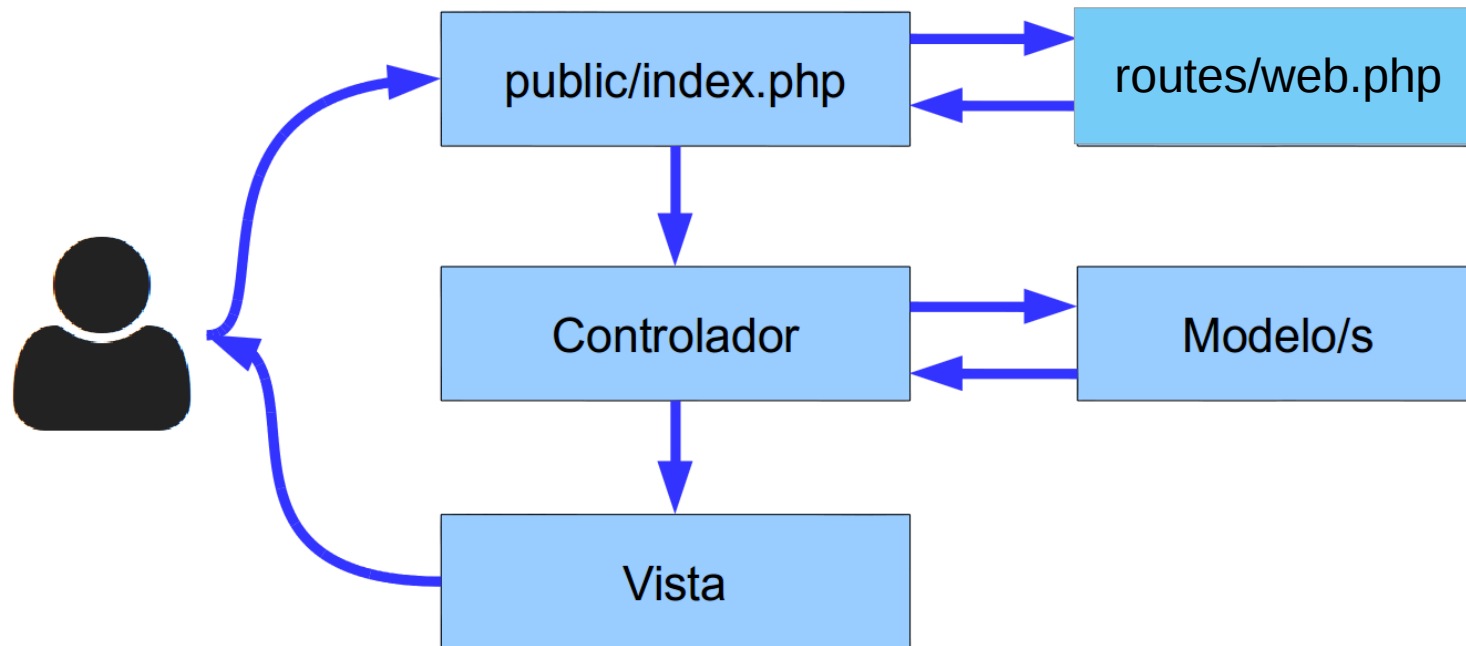
Funcionamiento Básico

- Todas las peticiones son atendidas en primera instancia por el archivo
`public/index.php`
- Este archivo, comprobará la existencia de la URL solicitada en el archivo de rutas
`route/web.php`
- De existir la ruta, se realizará lo definido en ese archivo (llamada a vista, llamada a controlador, llamada a función u otro)



Laravel: Arquitectura

Esquema de Funcionamiento Básico





Laravel: Arquitectura

Rutas

- Las rutas deben definirse en el archivo `route/web.php`
- Si se hace una petición a una ruta que no esté definida en ese archivo, Laravel arrojará un error 404.
- Las rutas definidas pueden devolver un valor propio, hacer una llamada a una vista o a un controlador.



Laravel: Arquitectura

Ruta Básica

- Se define en el archivo `route/web.php` la url, junto con el tipo de petición (GET o POST)

```
Route::get('/', function () {  
    return 'Hola Mundo con GET';  
});
```

```
Route::post('/', function () {  
    return 'Hola Mundo con POST';  
});
```



Laravel: Arquitectura

Ruta Básica

- También es posible definir llamadas con los métodos PUT o DELETE

```
Route::put('foo/bar', function () {  
    //  
});
```

```
Route::delete('foo/bar', function () {  
    //  
});
```



Laravel: Arquitectura

Ruta Básica

- Definir varios métodos al mismo tiempo

```
Route::match(Array('GET', 'POST'), '/'),  
function () {  
    //  
});
```

- O definir una ruta para cualquier tipo de petición

```
Route::any('/', function () {  
    return 'Hola Mundo';  
});
```



Laravel: Arquitectura

Añadir parámetros a las rutas

- Se pueden añadir parámetros a la URL de la siguiente forma:

```
Route::get('libro/{i}', function ($i)
{
    return 'Mostrar libro'. $i;
});
```



Laravel: Arquitectura

Artisan

- Laravel incluye un CLI (Command Line Interface), que funciona como asistente y facilita tareas del desarrollo (como crear un controlador, listar rutas, actualizar el modelo u otros)

```
C:\php artisan route:list
```

```
C:\php artisan make:model Modelo
```

```
C:\php artisan make:controller Controlador
```



Laravel: Arquitectura

Vistas

- Las vistas se guardan en el directorio `resources/views` como archivos con extensión `.php`
- Ejemplo, se creará una vista `home` en el archivo `resources/views/home.php`

```
<html>
  <head>
    <title>Ejemplo 1</title>
  </head>
  <body>
    <h1>¡Hola <?php echo $nombre; ?>!</h1>
  </body>
</html>
```



Laravel: Arquitectura

Referenciar y devolver la vista en la ruta

- Se abre el archivo routes/web.php
- Se genera dentro de la ruta la llamada a la vista como se indica a continuación

```
Route::get('/', function()  
{  
    return view('home', array('nombre' => 'Juan Perez'));  
});
```



Ejercicio

Realizar lo siguiente

- Capturar una variable de la URL (nombre) y mostrarlo en el archivo de vista.



Laravel: vistas

Traspasar datos a una vista

- Otra alternativa es traspasar los datos con el método with

```
Route::get('/{nombre}', function ($nombre) {  
    return view('home')->with("nombre", $nombre)  
        ->with("editable", true);  
});
```