

Estudio Teorico Laravel

Sumario

Objetivo.....	3
Introducción.....	4
Competidores.....	5
Características.....	6
Características de Laravel.....	6
Estructura de directorios.....	7
Componentes, relación y utilidad.....	9
Webgrafía, fuentes, tutoriales, cursos, recomendados.....	10
Conclusión (teoría).....	11

Objetivo

El objetivo de este estudio es aprender como funciona Laravel, su estructura y posibilidades que ofrece

Introducción

Laravel es un framework de código abierto para el desarrollo de aplicaciones web en PHP, basado en el patrón MVC (Modelo-Vista-Controlador). Fue creado por Taylor Otwell y lanzado el 9 de junio de 2011.

Laravel se destaca por su sintaxis fácil de comprender, que facilita tareas comunes como la autenticación, el enrutamiento, las sesiones y el almacenamiento en caché.

Competidores

A pesar de que Laravel sea el framework de PHP más usado, compite con varios frameworks, entre ellos:

CodeIgniter: Es fácil de aprender y muy eficiente, por ello está enfocado en el rendimiento

Symfony: Es usado en proyectos a gran escala, es más complejo que Laravel pero proporciona una mayor flexibilidad y control. Laravel incorpora componentes de Symfony

Zend: Tiene la característica de que permite cargar solo los componentes que vayan a usarse en la aplicación, gracias a esto la reutilización de código se facilita mucho

Phalcon: Escrito en C, es un framework muy ligero y con un muy buen rendimiento, sin embargo, su documentación no es tan completa como la de Laravel

CakePHP: Uno de los primeros frameworks MVC de PHP, cuenta con plantillas y validación integrada, tiene una serie de configuraciones ya preestablecidas que agilizan la creación de proyectos

Características

Características de Laravel

- **Eloquent ORM:** Un ORM (Object-Relational Mapping) que facilita la interacción con bases de datos.
- **Blade:** Un motor de plantillas simple y potente, permite crear vistas reutilizables y mantener la lógica de presentación separada de la lógica de negocio .
- **Artisan:** Una interfaz de línea de comandos que proporciona herramientas útiles para el desarrollo, proporciona comandos útiles para tareas comunes como la generación de código, migraciones y pruebas .
- **Migrations:** Sistema de control de versiones para la base de datos, permite definir y modificar la estructura de la base de datos de manera programática .
- **Middleware:** Permite filtrar solicitudes HTTP, permite aplicar lógica antes de que las solicitudes lleguen a los controladores, como la autenticación y la autorización .
- **Seguridad:** Incluye características como protección CSRF, hashing de contraseñas y autenticación.
- **Routing:** Gestiona las rutas de la aplicación, mapeando URLs a controladores específicos
- **Validation:** Proporciona una forma sencilla de validar datos de entrada antes de procesarlos
- **Queues:** Permite ejecutar tareas en segundo plano para mejorar el rendimiento y la experiencia del usuario
- **Events:** Permite suscribirse y escuchar eventos en la aplicación, facilitando la implementación de patrones de diseño
- **Broadcasting:** Facilita la implementación de aplicaciones en tiempo real
- **Testing:** Proporciona herramientas para asegurar el funcionamiento del código mediante pruebas automatizadas

Estructura de directorios

Hablar de su estructura AMP, directorios que crea, etc

La última versión estable es Laravel 11, lanzada el 12 de Marzo de 2024

Nada mas crear un proyecto de laravel, se nos presentara esta estructura:

app	14/03/2024 08:51	Carpeta de archivos	
bootstrap	14/03/2024 08:51	Carpeta de archivos	
config	14/03/2024 08:51	Carpeta de archivos	
database	14/03/2024 10:32	Carpeta de archivos	
public	14/03/2024 08:51	Carpeta de archivos	
resources	14/03/2024 08:51	Carpeta de archivos	
routes	14/03/2024 08:51	Carpeta de archivos	
storage	14/03/2024 08:51	Carpeta de archivos	
tests	14/03/2024 08:51	Carpeta de archivos	
vendor	14/03/2024 10:32	Carpeta de archivos	
.editorconfig	14/03/2024 08:51	Archivo de origen ...	1 KB
.env	14/03/2024 10:32	Archivo ENV	2 KB
.env.example	14/03/2024 08:51	Archivo EXAMPLE	2 KB
.gitattributes	14/03/2024 08:51	Archivo de origen ...	1 KB
.gitignore	14/03/2024 08:51	Archivo de origen ...	1 KB
artisan	14/03/2024 08:51	Archivo	1 KB
composer.json	14/03/2024 08:51	Archivo de origen ...	2 KB
composer.lock	14/03/2024 10:31	Archivo LOCK	292 KB
package.json	14/03/2024 08:51	Archivo de origen ...	1 KB
phpunit.xml	14/03/2024 08:51	xmlfile	2 KB
README.md	14/03/2024 08:51	Archivo de origen ...	5 KB
vite.config.js	14/03/2024 08:51	JSFile	1 KB

Estos directorios y archivos corresponden a lo siguiente:

- **app:** Contiene el código principal de la aplicación
 - Console/: Comandos de Artisan.
 - Exceptions/: Manejo de excepciones.
 - Http/: Controladores, middleware y requests.
 - Models/: Modelos Eloquent.
 - Providers/: Proveedores de servicios.
- **bootstrap:** Contiene el archivo `app.php` que inicializa el framework y la carpeta `cache` para archivos de caché generados por el framework
- **config:** Contiene la configuración de la aplicación
- **database:** Contiene las migraciones de la base de datos
 - factories/: Definiciones de fábricas para modelos.
 - migrations/: Archivos de migración de la base de datos.
 - seeders/: Clases para poblar la base de datos con datos de prueba.
- **public:** Contiene el archivo `index.php` (índice), así como cualquier archivo público
- **resources:** Vistas, archivos de idioma
 - views/: Plantillas Blade.
 - lang/: Archivos de localización
 - sass/ y js/: Archivos de Sass y JavaScript sin compilar
- **routes:** Contiene las rutas de la aplicación
- **storage:** Contiene los archivos de almacenamiento de la aplicación, como imágenes o audios
- **tests:** Contiene las pruebas de la aplicación
- **vendor:** Contiene las dependencias de la aplicación instaladas por Composer.
- **.env:** Archivo de entorno de la aplicación, que contiene la configuración específica de la máquina, como la información de la base de datos y la configuración de correo electrónico.
- **.env.example:** Ejemplo del archivo de entorno de la aplicación.
- **composer.json:** Archivo de configuración de Composer,
- **composer.lock:** Archivo generado por Composer que especifica las versiones exactas de las dependencias instaladas.
- **package.json:** Archivo de configuración de npm, que especifica las dependencias de JavaScript de la aplicación.
- **README.md:** Archivo de documentación de la aplicación.

Componentes, relación y utilidad

Webgrafía, fuentes, tutoriales, cursos, recomendados

Competidores de Laravel:

<https://www.hostinger.es/tutoriales/mejores-frameworks-php>

<https://colorvivo.com/mejores-frameworks-php-desarrollo-web/>

Características:

https://www.tutorialspoint.com/laravel/laravel_application_structure.htm

<https://codersfree.com/courses-status/aprende-laravel-avanzado/estructura-de-carpetas>

Conclusión (teoría)

Laravel se ha consolidado como uno de los frameworks PHP más populares gracias a su facilidad de uso, su amplia gama de características y su comunidad activa. Ofrece herramientas poderosas que simplifican el desarrollo de aplicaciones web. Con una buena documentación y una gran cantidad de recursos disponibles, Laravel es una excelente opción tanto para desarrolladores principiantes como experimentados.