

Conceptos de Laravel

por Guillermo Romero Cuevas

¿Que es Laravel?

Laravel es un Marco de Trabajo(Framework) De Fuente Abierta Basado en PHP, que Provee Herramientas de Desarrollo y Paquetes. Funciona bajo el Modelo MVC (Model View Controller). Versiones de Laravel y requisitos Laravel Tiene Varias Versiones Principales, Se usa Versiones Semánticas Para diferenciarlas. La Version Mayor mas actual es Laravel 12, La cual Necesita de PHP Minimo 8.2 y maximo 8.4.

Segun el Sitio, La Siguiente Version de Laravel Sera la 13, y tendra como Minimo de PHP 8.3 en vez de 8.2. Segun este mismo, Laravel 13 Se Publicara en el primer Trimestre de 2026.

Composer

Es una herramienta de Manejo de Paquetes, la Cual ayuda con la instalación y Actualizaciones de los paquetes de PHP. Esta Herramienta se puede utilizar para Instalar Laravel de manera directa.

PHP Artisan

Es un comando que Permite construir nuevos Archivos rapidamente, utilizando patrones de construccion, Como por ejemplo el patron de Fabrica, Seeder, Test.

Viene incluido con Laravel, y se puede extender su funcionalidad al instalar Otras herramientas, como Tinker(REPL) y Laravel Sail

¿Cómo se crea un proyecto de Laravel?

Se Crea Mediante la Invocación del Comando 'Composer'. En Software como Herd Este Proceso esta Automatizado, Sin Embargo si se desea Hacer manualmente, se debe de tener instalado PHP, Composer y Laravel. Si ya se Tiene Composer y PHP instalado, se puede Ejecutar el Siguiente Comando para Instalar el instalador de Laravel

```
composer global require laravel/installer
```

A Seguido se Puede empezar el Proyecto en un Ruta que Se Desea

```
laravel new NombreDeCarpetaParaElProyecto
```

una Vez se termine el Proceso que Laravel Pide, Podemos entrar a la Carpeta

```
cd NombreDeCarpetaParaElProyecto  
npm install && npm run build
```

```
composer run dev
```

Una Vez se Terminen de ejecutar Estos Procesos, Se Ha Creado un Proyecto Con Laravel

¿Cuál es la estructura del proyecto Laravel y para qué sirve cada carpeta en lo general?

La Estructura General del Archivo Raiz de un Proyecto de Laravel se ve como sigue:

```
app/  
bootstrap/  
config/  
database/  
node_modules/  
public/  
resources/  
routes/  
storage/  
tests/  
vendor/
```

A Continuación una Explicación corta de Que Hace cada Archivo.

app/

Este Archivo tiene los Archivos Fuente de Controladores, Fabricas, Modelos, Etc. Esencialmente es el Alma de la Aplicación, donde la mayoria de la Logica y codigo debe de ejecutarse.

bootstrap/

Dicta el Inicio de la Aplicación, Configurando Paquetes y Autoarranques para el Marco de Trabajo.

config/

Aqui Viven todas las Configuraciones Por Defecto de la Aplicación. Si se necesita Expandir Configuraciones y Estandares Base, aquí es un Buen lugar para buscar.

database/

Aqui se crean las Fabricas de Objetos, Migraciones y Seeders. Esencialmente todos los elementos Relacionados con la Base de Datos. Tambien se puede incluir aquí Mismo la base de Datos si asi se desea.

node_modules/

Esta es una Carpeta generada por NPM, el Manejador de Paquetes de Javascript. Generalmente se debe de Ignorar por GIT Mediante el Archivo .gitignore.

public/

Es el punto de Acceso para todas las rutas del proyecto, Mediante el Archivo index.php. Tambien se pueden Agregar los Recursos usados por Javascript y CSS para acceder a estos de Manera mas ordenada.

resources/

Aqui Viven todos los archivos de Frontend. Tiene 3 Sub-carpetas, CSS, JS y views. De Manera Importante, Views es la carpeta que contiene todos los Archivos de vistas, Escritas en PHP con ayuda de el Motor de Plantillas Blade, el cual esta incluido con Laravel.

routes/

Aqui viven los archivos que determinan las rutas, Funciones y Controladores que se deben de invocar cuando un Cliente pida acceso a un Documento dentro del sitio web. Tambien se Encuentra el Archivo console.php, El cual puede usarse para Hacer Tareas Agendadas si estas son Necesarias.

storage/

Contiene Los Archivos Cache, Plantillas Compiladas y Logs. Sirve como la carpeta donde Iso Archivos de Acceso Rapido deben de Encontrarse.

tests/

Aqui se Escriben los Códigos de Prueba, Utilizando ya sea Pest o PHP unit para hacer comprobaciones de Código automáticas. Si estas Fallan, Significa que Algun Elemento del Código esta mal Hecho o tiene algún Fallo detectable. Se deben de considerar los Casos Esquina que se pueden presentar y predecir.

vendor/

Aqui es donde todas las dependencias que Composer necesita se encuentran.