

Laravel

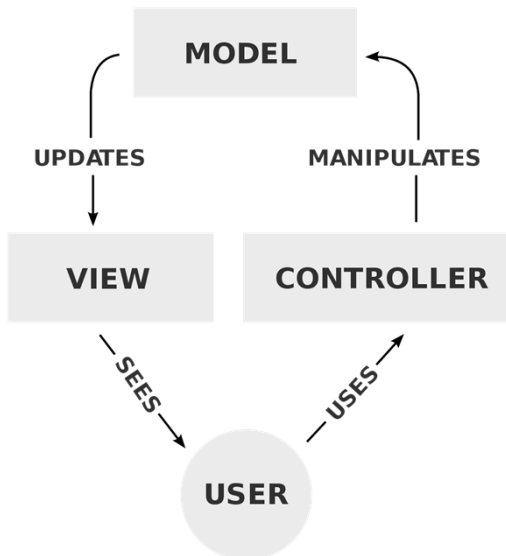
Framework PHP

Laravel

- Es un marco de desarrollo utilizado para crear sitios y aplicaciones web. El creador de Laravel, Taylor Otwell, lanzó su versión Beta el 9 de junio de 2011.
- Laravel 10 = publicado en febrero de 2023.
- El lanzamiento de Laravel 11 está previsto para febrero de 2024.

Laravel

- En la actualidad, Laravel está alojado en GitHub, donde cuenta con 65.900 usuarios y ha sido descargado 144 millones de veces. Tiene licencia MIT.
- Laravel utiliza PHP con la arquitectura MVC.

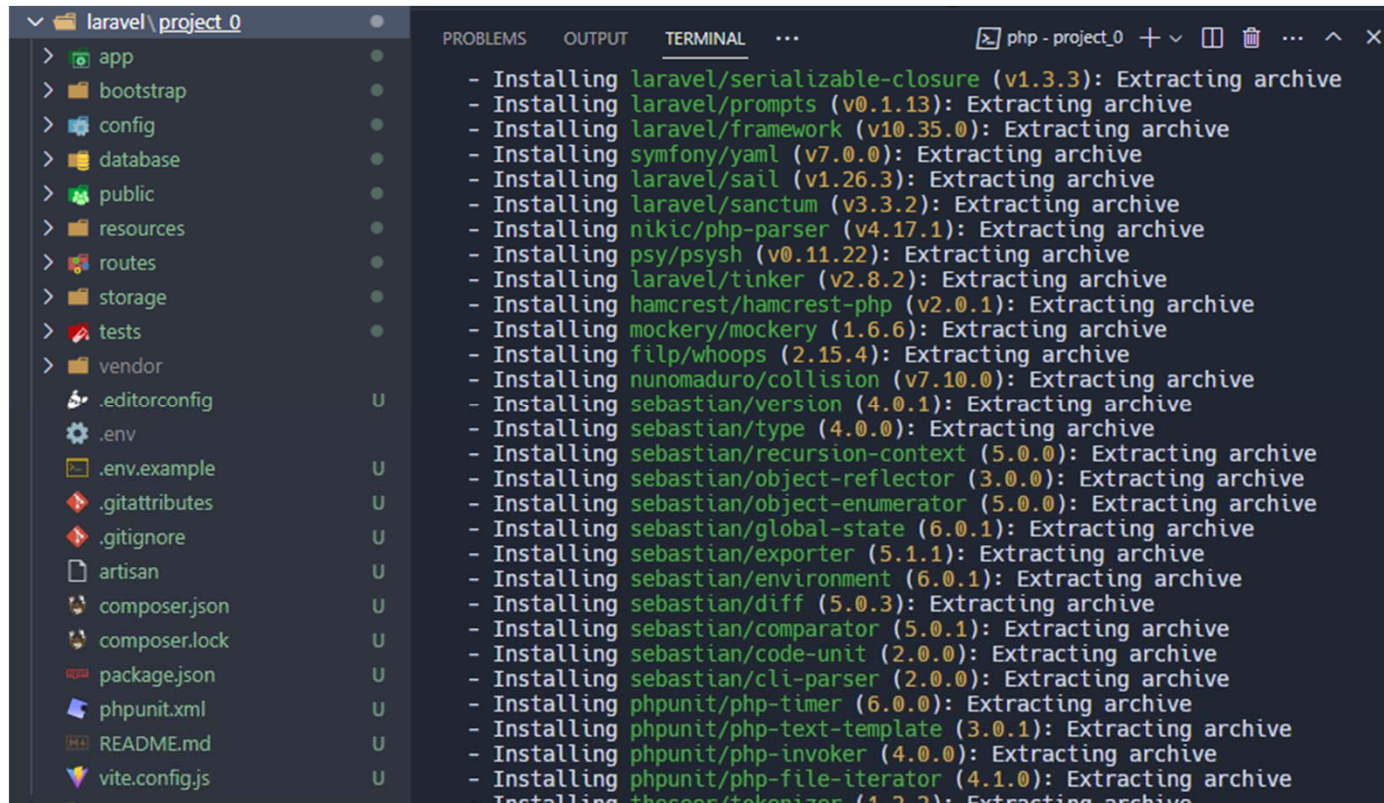


Características de Laravel

- Sistema completo de autenticación
- Mapeo objeto-relacional (ORM)
- Biblioteca de módulos
- Interfaz de línea de comandos (CLI) con comandos preconstruidos
- Pruebas automáticas
- Entorno de desarrollo virtual

Tu primer proyecto Laravel

composer create-project laravel/laravel .



Tu primer proyecto Laravel

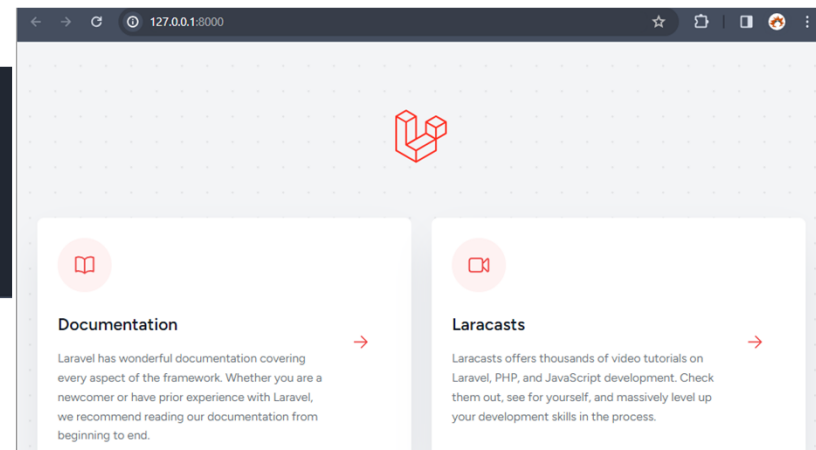
- Una vez creado el proyecto, inicia el servidor de desarrollo local de Laravel mediante el comando serve de la CLI Artisan de Laravel.

```
[kralos]--[main = ● ]
[C:\xampp\htdocs\examples-PHP\code\laravel\project_0]
○ ▶ php .\artisan serve

INFO Server running on [http://127.0.0.1:8000].

Press Ctrl+C to stop the server

2023-12-07 12:33:10 ..... ~ 2s
```



Carpetas más importantes

- app: Contiene la lógica principal de la aplicación, como los controladores, las excepciones, las solicitudes, etc.
- bootstrap: Contiene los archivos de inicialización de la aplicación, incluyendo la carga de archivos de entorno y la configuración de la configuración.
- config: Contiene la configuración de la aplicación, incluyendo la configuración de la base de datos, el correo electrónico y la aplicación.
- database: Contiene las migraciones de la base de datos, las semillas y las fábricas.
- public: Contiene el archivo index.php que es el punto de entrada de la aplicación, así como cualquier archivo público, como CSS, JavaScript e imágenes.

Carpetas más importantes

- resources: Contiene los recursos de la aplicación, incluyendo las vistas, los archivos CSS, los archivos JavaScript y los archivos de idioma.
- routes: Contiene las rutas de la aplicación, incluyendo las rutas web y API.
- storage: Contiene los archivos de almacenamiento de la aplicación, como las imágenes cargadas y los registros.
- 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.

Carpetas más importantes

- `.env.example`: Ejemplo del archivo de entorno de la aplicación.
- `composer.json`: Archivo de configuración de Composer, que especifica las dependencias de la aplicación.
- `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.

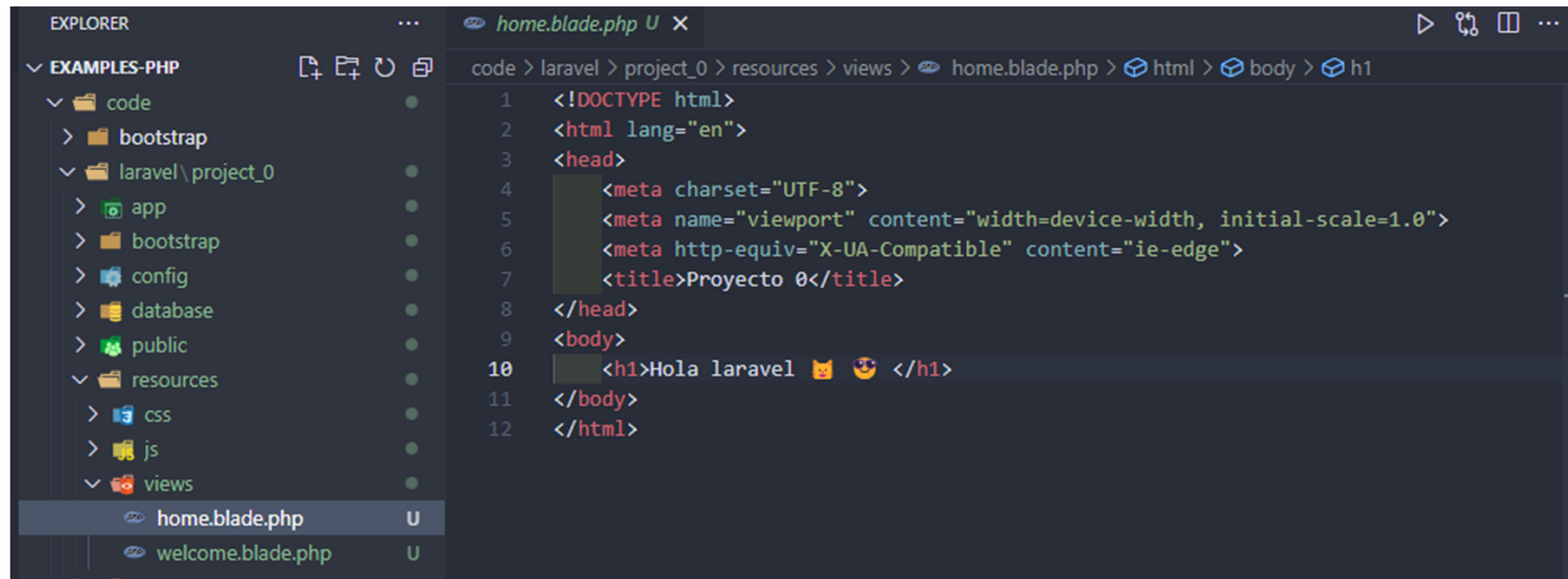
Primeros pasos {Código inicial}

examples-PHP\code\laravel\project_0\resources\views\welcome.blade.php

```
welcome.blade.php 3, U x
code > laravel > project_0 > resources > views > welcome.blade.php > ...
33
34     <div class="max-w-7xl mx-auto p-6 lg:p-8">
35         <div class="flex justify-center">
36             <svg viewBox="0 0 62 65" fill="none" xmlns="http://www.w3.org/2000/svg" class="h-16 w-aut
37                 <path d="M61.8548 14.6253C61.8778 14.7102 61.8895 14.7978 61.8897 14.8858V28.5615C61.
38             </svg>
39         </div>
40
41     <div class="mt-16">
42         <div class="grid grid-cols-1 md:grid-cols-2 gap-6 lg:gap-8">
43             <a href="https://laravel.com/docs" class="scale-100 p-6 bg-white dark:bg-gray-800/50
44                 <div>
45                     <div class="h-16 w-16 bg-red-50 dark:bg-red-800/20 flex items-center justify-
46                         <svg xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24" s
47                             <path stroke-linecap="round" stroke-linejoin="round" d="M12 6.042A8.9
48                         </svg>
49                     </div>
50                 </a>
51             <h2 class="mt-6 text-xl font-semibold text-gray-900 dark:text-white">Document
52
53             <p class="mt-4 text-gray-500 dark:text-gray-400 text-sm leading-relaxed">
54                 Laravel has wonderful documentation covering every aspect of the framewor
55             </p>
56         </div>
57
58         <svg xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24" stroke-wi
59             <path stroke-linecap="round" stroke-linejoin="round" d="M4.5 12h15m0 0l-6.75-
60         </svg>
61     </a>
62
63     <a href="https://laracasts.com" class="scale-100 p-6 bg-white dark:bg-gray-800/50 dar
64         <div>
65             <div class="h-16 w-16 bg-red-50 dark:bg-red-800/20 flex items-center justify-
66                 <svg xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24" s
67                     <path stroke-linecap="round" d="M15.75 10.5147A4.72 4.72a.75 .75 0 011.28
68                 </svg>
69             </div>
70         </a>
71     <h2 class="mt-6 text-xl font-semibold text-gray-900 dark:text-white">Laracast
72
```

Primeros pasos {Código inicial}

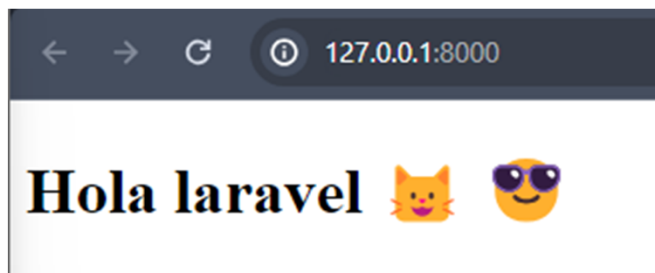
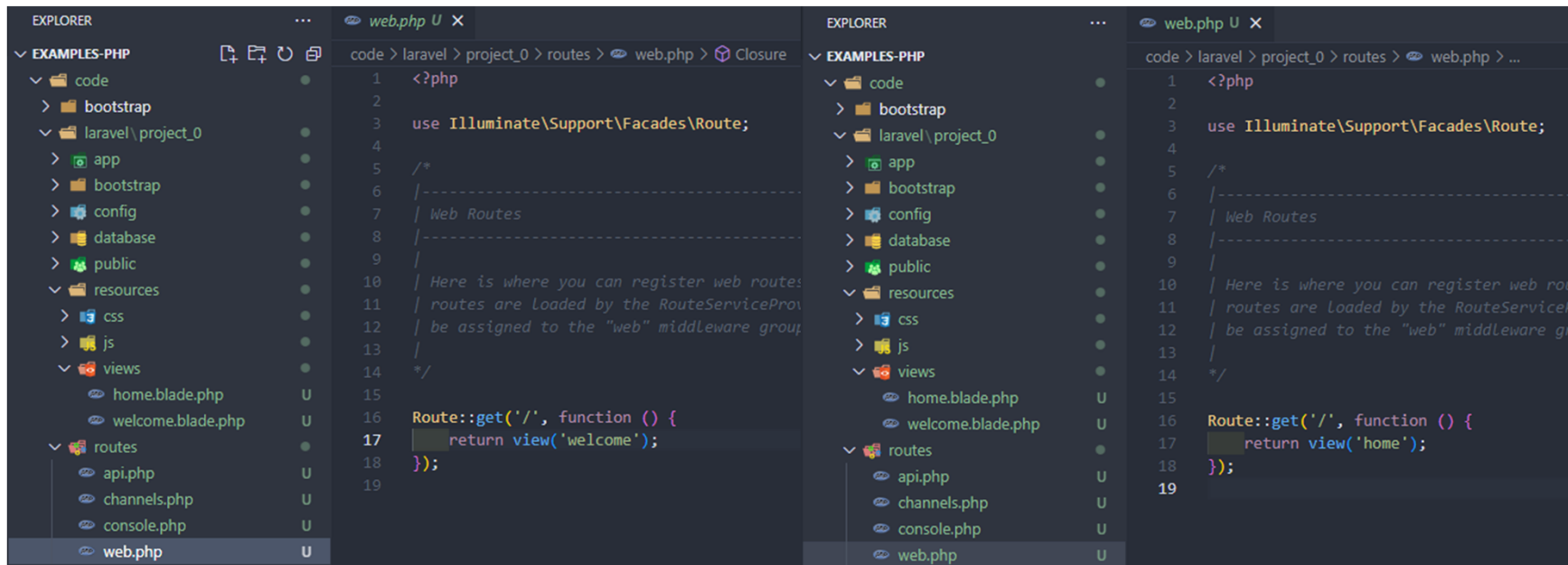
examples-PHP\code\laravel\project_0\resources\views\home.blade.php



The screenshot shows a code editor with a dark theme. On the left, the 'EXPLORER' sidebar displays a file tree for a project named 'EXAMPLES-PHP'. The tree includes folders for 'code', 'bootstrap', and 'laravel\project_0'. Under 'laravel\project_0', there are folders for 'app', 'bootstrap', 'config', 'database', 'public', and 'resources'. The 'resources' folder is expanded, showing subfolders for 'css', 'js', and 'views'. The 'views' folder is selected, and the file 'home.blade.php' is highlighted. On the right, the code editor displays the content of 'home.blade.php'. The breadcrumb navigation at the top of the editor reads 'code > laravel > project_0 > resources > views > home.blade.php > html > body > h1'. The code itself is as follows:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <meta http-equiv="X-UA-Compatible" content="ie-edge">
7     <title>Proyecto 0</title>
8 </head>
9 <body>
10    <h1>Hola laravel 🐱 😊 </h1>
11 </body>
12 </html>
```

Primeros pasos {Código inicial}



{¿Qué es artisan?}

- La interfaz de línea de comandos de Laravel.
- Artisan está basado en el componente Console de Symfony y nos ofrece un conjunto de comandos que nos pueden ayudar a realizar diferentes tareas durante el desarrollo e incluso cuando la aplicación se encuentra en producción.
- Para conocer el listado completo de los comandos disponibles ejecutamos en consola, en el directorio raíz de un proyecto de Laravel: ***php artisan list***

{¿Qué es artisan?}

```
[kralos]--[?main ≡ ● ]
[C:\xampp\htdocs\examples-PHP\code\laravel\project_0]
▶ php artisan list
Laravel Framework 10.35.0

Usage:
  command [options] [arguments]

Options:
  -h, --help                Display help for the given command. When no command is given display help for the list comm
and
  -q, --quiet               Do not output any message
  -V, --version              Display this application version
      --ansi|--no-ansi      Force (or disable --no-ansi) ANSI output
  -n, --no-interaction      Do not ask any interactive question
      --env[=ENV]           The environment the command should run under
  -v|vv|vvv, --verbose      Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and 3 fo
r debug
```

```
[kralos]--[?main ≡ ● ]
[C:\xampp\htdocs\examples-PHP\code\laravel\project_0]
▶ php artisan route:list

GET|HEAD / .....
POST _ignition/execute-solution ignition.executeSolution > Spatie\LaravelIgnition > ExecuteSolutionContr...
GET|HEAD _ignition/health-check ..... ignition.healthCheck > Spatie\LaravelIgnition > HealthCheckController
POST _ignition/update-config .... ignition.updateConfig > Spatie\LaravelIgnition > UpdateConfigController
GET|HEAD api/user .....
GET|HEAD sanctum/csrf-cookie ..... sanctum.csrf-cookie > Laravel\Sanctum > CsrfCookieController@show

Showing [6] routes
```

{¿Qué es artisan?}

```
[kralos]--[?main ≡ • ]
[C:\xampp\htdocs\examples-PHP\code\laravel\project_0]
▶ php artisan make:view main

INFO View [C:\xampp\htdocs\examples-PHP\code\laravel\project_0\resources\views\main.blade.php] created successfully.
```

