

# LARAVEL



**José Luis Padilla López  
Fernanda Monserrat Ramírez Ochoa  
Francisco Manuel Guzmán Hernández**

**Martes y Jueves 7:00 a.m. – 8:40 a.m.**

**Viernes 7:50 a.m. – 8:40 a.m.**





# ¿Que es Laravel?

Laravel es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP 5 y PHP 7. Su filosofía es desarrollar código PHP de forma elegante y simple, evitando el "código espagueti". Fue creado en 2011 y tiene una gran influencia de frameworks como Ruby on Rails, Sinatra y ASP.NET MVC.2

Framework: estructura conceptual y tecnológica de asistencia definida, normalmente, con artefactos o módulos concretos de software, que puede servir de base para la organización y desarrollo de software.



# ¿Que es Laravel?

Laravel propone en el desarrollo usar 'Routes with Closures'(ruta con clausura), en lugar de un MVC tradicional con el objetivo de hacer el código más claro. Aun así permite el uso de MVC tradicional.

Es una técnica para implementar ámbitos léxicos en un lenguaje de programación con funciones de primera clase.

Una cerradura es un registro que contiene una función junto con el ámbito donde fue declarada. La clausura permite que la función acceda a los valores de las variables declaradas en el mismo ámbito, aun cuando la invocación ocurra fuera de este.

## Características:

- Sistema de ruteo, también RESTful
- Blade, Motor de plantillas
- Peticiones Fluent
- Eloquent ORM
- Basado en Composer
- Soporte para el caché
- Soporte para MVC
- Usa componentes de Symfony
- Adopta las especificaciones PSR-2 y PSR-4



# Instalación:

The screenshot shows the Laravel website at <https://laravel.com>. The page features the red Laravel logo at the top left. The main navigation bar includes links for Documentación, Vapor, Ecosistema, Noticias, and Socios, along with a search bar labeled "Search Docs". A blue banner on the left side announces "¡Laravel Nova ya está disponible! Obtenga su copia hoy!" with a right-pointing arrow. The central content area has a large title "El marco PHP para artesanos web" and a descriptive paragraph about Laravel's expressive syntax and base-setting nature. To the right of the text is a 3D illustration of a white cube with several red pipes of varying sizes protruding from its sides, symbolizing connectivity or infrastructure.

Aplicaciones Google Traductor de Google Administrador de d... Social Player's Blogs DEV UNI Google Play Convertidor de You... AirDroid Web | Man...

Laravel Documentación Vapor Ecosistema Noticias Socios Search Docs

¡Laravel Nova ya está disponible! Obtenga su copia hoy! →

# El marco PHP para artesanos web

Laravel es un marco de aplicación web con sintaxis expresiva y elegante. Ya hemos sentado las bases, liberándote para crear sin sudar las cosas pequeñas.

<https://laravel.com/>



# Instalación:

## Requerimientos:

- PHP => 7.2.0
- Extensión PHP BCMath
- Extensión PHP Ctype
- Extensión PHP JSON
- Extensión PHP Mbstring
- Extensión PHP OpenSSL
- Extencion PDO PHP
- Tokenizer PHP Extension
- Extensión XML PHP



# Instalación:

## Verificación de la versión de PHP usando XAMPP

### Abrimos el “Shell”

XAMPP Control Panel v3.2.4 [ Compiled: Jun 5th 2019 ]

### XAMPP Control Panel v3.2.4

Service	Module	PID(s)	Port(s)	Actions			
	Apache	14408 7620	4438, 8080	Stop	Admin	Config	Logs
	MySQL	1372	3306	Stop	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

**Shell** (highlighted with a red box)

Config  
Netstat  
Explorer  
Services  
Help  
Quit

07:11:47 a. m. [main] Initializing Control Panel



# Instalación:

## Introducimos el comando ( php -v )

XAMPP for Windows

```
Setting environment for using XAMPP for Windows.  
newto@MYTBILL-LAP c:\xampp  
# php -v  
PHP 7.3.9 (cli) (built: Aug 28 2019 09:28:48) ( ZTS MSVC15 (Visual C+ 2017) x64 )  
Copyright (c) 1997-2018 The PHP Group  
Zend Engine v3.3.9, Copyright (c) 1998-2018 Zend Technologies  
  
newto@MYTBILL-LAP c:\xampp  
#
```

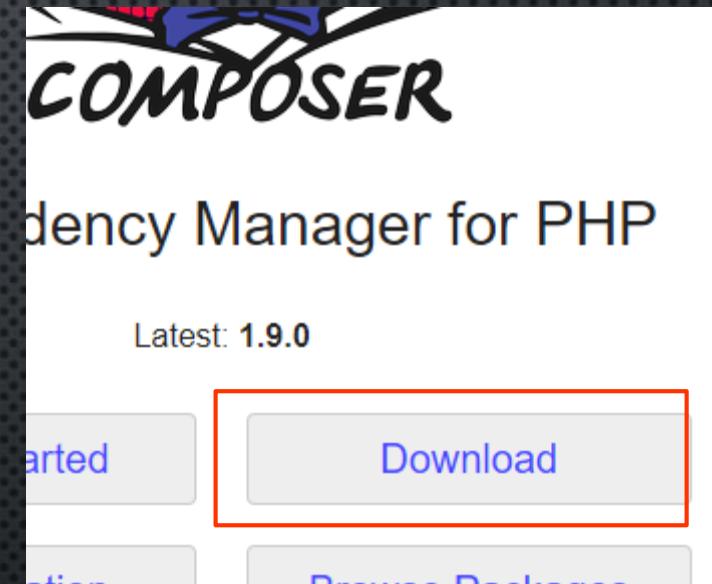


Descargamos el instalador y lo ejecutamos y sin cambiar nada solo presionamos siguiente hasta que termine.

## Instalación:



<https://getcomposer.org/>





# Instalación:

## Instalar de forma global Laravel

Introducimos la siguiente instrucción en cmd para que comience a descargar laravel.

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



# Instalación:

Antes de comenzar con la instalación debemos arrancar nuestro servicios en XAMPP

XAMPP Control Panel v3.2.4 [ Compiled: Jun 5th 2019 ]

### XAMPP Control Panel v3.2.4

Service	Module	PID(s)	Port(s)	Actions			
	Apache	14408 7620	4438, 8080	Stop	Admin	Config	Logs
	MySQL	1372	3306	Stop	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

Config  
Netstat  
Shell  
Explorer  
Services  
Help  
Quit

07-11-17 2:30 p.m. [main] Initializing Control Panel

Microsoft Windows [Versión 10.0.18362.356]

(c) 2019 Microsoft Corporation. Todos los derechos reservados.

```
C:\Users\newto>composer global require laravel/installer
```

Using version ^2.1 for laravel/installer

./composer.json has been updated

- ```
- Installing symfony/process (v4.3.5): Downloading (100%)
- Installing symfony/polyfill-ctype (v1.12.0): Downloading (100%)
- Installing symfony/filesystem (v4.3.5): Downloading (100%)
- Installing psr/container (1.0.0): Downloading (100%)
- Installing symfony/service-contracts (v1.1.7): Downloading (100%)
- Installing symfony/polyfill-php73 (v1.12.0): Downloading (100%)
- Installing symfony/polyfill-mbstring (v1.12.0): Downloading (100%)
- Installing symfony/console (v4.3.5): Downloading (100%)
- Installing guzzlehttp/promises (v1.3.1): Downloading (100%)
- Installing ralouphie/getallheaders (3.0.3): Downloading (100%)
- Installing psr/http-message (1.0.1): Downloading (100%)
- Installing guzzlehttp/psr7 (1.6.1): Downloading (100%)
- Installing guzzlehttp/guzzle (6.3.3): Downloading (100%)
- Installing laravel/installer (v2.1.0): Downloading (100%)
```

[symfony/console](#) suggests installing [symfony/lock](#)

Writing lock file

Generating autoload file



# Creamos una carpeta para nuestros proyectos

A screenshot of a Windows File Explorer window. The title bar says "proyectos". The menu bar includes "Archivo", "Inicio", "Compartir", "Vista", and a help icon. The address bar shows the path: "Este equipo > Acer (C:) > xampp > htdocs > proyectos". The search bar says "Buscar en ...". The main area has columns for "Nombre", "Fecha de modificación", "Tipo", and "Tamaño". A single item, "laravel", is listed: it is a "Carpeta de archivos" modified on "10/10/2019 05:05 p. m.".

| Nombre  | Fecha de modificación  | Tipo                | Tamaño |
|---------|------------------------|---------------------|--------|
| laravel | 10/10/2019 05:05 p. m. | Carpeta de archivos |        |



# Instalación:

## Instalar de forma global Laravel

Usaremos la siguiente instrucción mas el nombre de nuestro proyecto desde cmd en la dirección que usaremos para crear nuestros proyecto:

```
laravel new blog
```

```
C:\ Símbolo del sistema

Microsoft Windows [Versión 10.0.18362.356]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\newto>cd..

C:\Users>cd..

C:\>cd C:\xampp\htdocs\proyectos\laravel

C:\xampp\htdocs\proyectos\laravel>laravel new blogPracticas
```

Símbolo del sistema

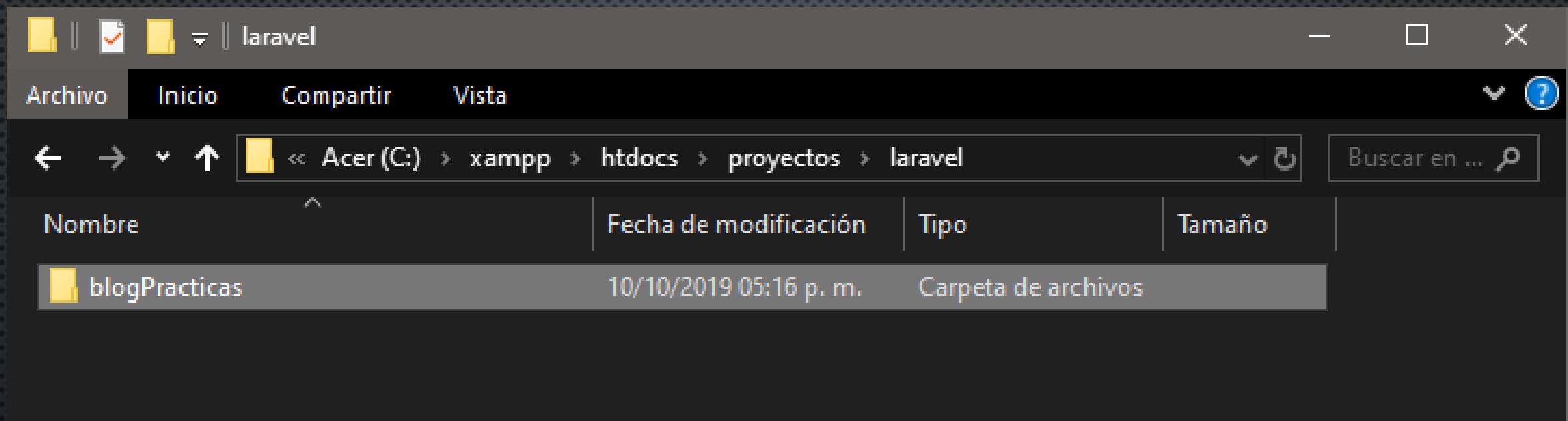
```
laravel/framework suggests installing pusher/pusher-php-server (Required to use the Pusher broadcast driver (^4.0).)
laravel/framework suggests installing symfony/cache (Required to PSR-6 cache bridge (^4.3.4).)
laravel/framework suggests installing symfony/psr-http-message-bridge (Required to use PSR-7 bridging features (^1.2).)
laravel/framework suggests installing wildbit/swiftmailer-postmark (Required to use Postmark mail driver (^3.0).)
psy/psysh suggests installing ext-pcntl (Enabling the PCNTL extension makes PsySH a lot happier :))
psy/psysh suggests installing ext-pdo-sqlite (The doc command requires SQLite to work.)
psy/psysh suggests installing ext-posix (If you have PCNTL, you'll want the POSIX extension as well.)
psy/psysh suggests installing hoa/console (A pure PHP readline implementation. You'll want this if your PHP install does
n't already support readline or libedit.)
filp/whoops suggests installing whoops/soap (Formats errors as SOAP responses)
facade/ignition suggests installing laravel/telescope (^2.0)
sebastian/global-state suggests installing ext-uopz (*)
sebastian/environment suggests installing ext-posix (*)
phpunit/php-code-coverage suggests installing ext-xdebug (^2.7.2)
phpunit/phpunit suggests installing ext-soap (*)
phpunit/phpunit suggests installing ext-xdebug (*)
phpunit/phpunit suggests installing phpunit/php-invoker (^2.0.0)
Generating optimized autoload files
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
> @php artisan key:generate --ansi
Application key set successfully.
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi
Discovered Package: facade/ignition
Discovered Package: fideloper/proxy
Discovered Package: laravel/tinker
Discovered Package: nesbot/carbon
Discovered Package: nunomaduro/collision
Package manifest generated successfully.
Application ready! Build something amazing.

C:\xampp\htdocs\proyectos\laravel>
```

Listo!



Una vez terminada la instalación revisamos que se creo la nueva carpeta.





# Revisando la instalación

localhost:8080/proyectos/laravel/blogPracticas/public/

Aplicaciones Google Traductor de Google Administrador de d... Social Player's Blogs DEV UNI Google Play Convertidor de You... AirDroid Web | Ma

# Laravel

DOCS LARACASTS NOTICIAS BLOG NOVA FORGE VAPOR GITHUB



# Revisando la instalación (alternativa)

Desde la línea de comando entramos a la carpeta de nuestro proyecto e introducimos el comando “php artisan serve”

```
C:\ Símbolo del sistema  
Microsoft Windows [Versión 10.0.18362.356]  
(c) 2019 Microsoft Corporation. Todos los derechos reservados.  
C:\Users\newto>cd C:\xampp\htdocs\proyectos\laravel\blogPracticas  
C:\xampp\htdocs\proyectos\laravel\blogPracticas>
```

Este comando nos genera una dirección ip en la cual se ejecuta nuestro proyecto.

Copiamos la dirección y la pegamos en nuestro navegador.

```
C:\ Símbolo del sistema - php artisan serve  
Microsoft Windows [Versión 10.0.18362.356]  
(c) 2019 Microsoft Corporation. Todos los derechos reservados.  
C:\Users\newto>cd C:\xampp\htdocs\proyectos\laravel\blogPracticas  
C:\xampp\htdocs\proyectos\laravel\blogPracticas>php artisan serve  
Laravel development server started: http://127.0.0.1:8000
```



# Revisando la instalación

- → C ⓘ 127.0.0.1:8000

Aplicaciones Google Traductor de Google Administrador de d... Social Player's Blogs DEV UNI Google Play Convertidor de You... 🔍

# Laravel

DOCS LARACASTS NOTICIAS BLOG NOVA FORGE VAPOR GITHUB



# Ejemplos

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File Edit Selection View Go Debug Terminal Help
- Title Bar:** web.php - blogPractices - Visual Studio Code
- Sidebar (Left):**
  - EXPLORER:** Shows the project structure under "BLOGPRACTICAS".
    - OPEN EDITORS: welcome.blade.php, routes > web.php (highlighted)
    - BLOGPRACTICAS:
      - > app
      - > bootstrap
      - > config
      - > database
      - > public
      - > resources
        - > js
        - > lang
        - > sass
      - > views
        - welcome.blade.php
    - routes
      - api.php
      - channels.php
      - console.php
      - web.php (highlighted)
    - > storage
    - > tests
    - > vendor
    - .editorconfig
    - .env
    - .env.example
    - .gitattributes
  - OUTLINE:** Shows the outline of the current file.
  - NPM SCRIPTS:** Shows the available npm scripts.- Editor Area (Right):** Displays the content of the file "web.php".

```
routes > web.php
1 <?php
2
3 /*
4 |-----
5 | Web Routes
6 |-----
7 |
8 | Here is where you can register web routes for your application. These
9 | routes are loaded by the RouteServiceProvider within a group which
10 | contains the "web" middleware group. Now create something great!
11 |
12 */
13
14 Route::get('/', function () {
15     return view('welcome');
16 });
17
```
- Bottom Status Bar:** Shows file statistics (0 changes), line 1, column 1, spaces: 4, encoding: UTF-8, line separator: LF, file type: PHP, and icons for save, close, and notifications.

File Edit Selection View Go Debug Terminal Help welcome.blade.php - blogPracticas - Visual Studio Code

EXPLORER

OPEN EDITORS

- welcome.blade.php resources\views
- web.php routes

BLOGPRACTICAS

- app
- bootstrap
- config
- database
- public
- resources
  - js
  - lang
  - sass
  - views
    - welcome.blade.php
- routes
  - api.php
  - channels.php
  - console.php
  - web.php
- storage
- tests
- vendor
- .editorconfig
- .env
- .env.example
- .gitattributes

OUTLINE

NPM SCRIPTS

welcome.blade.php x web.php

resources > views > welcome.blade.php

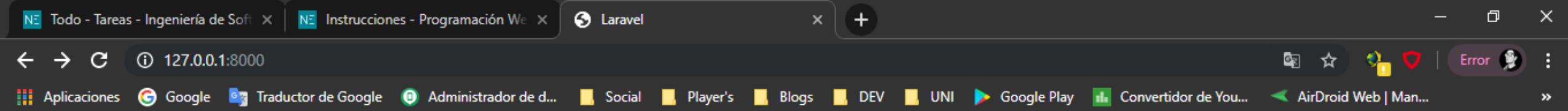
```
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

```
<div class="top-right links">
    @auth
        <a href="{{ url('/home') }}">Home</a>
    @else
        <a href="{{ route('login') }}">Login</a>
        @if (Route::has('register'))
            <a href="{{ route('register') }}">Register</a>
        @endif
    @endauth
</div>
@endif

<div class="content">
    <div class="title m-b-md">
        Programacion web usando Laravel
    </div>

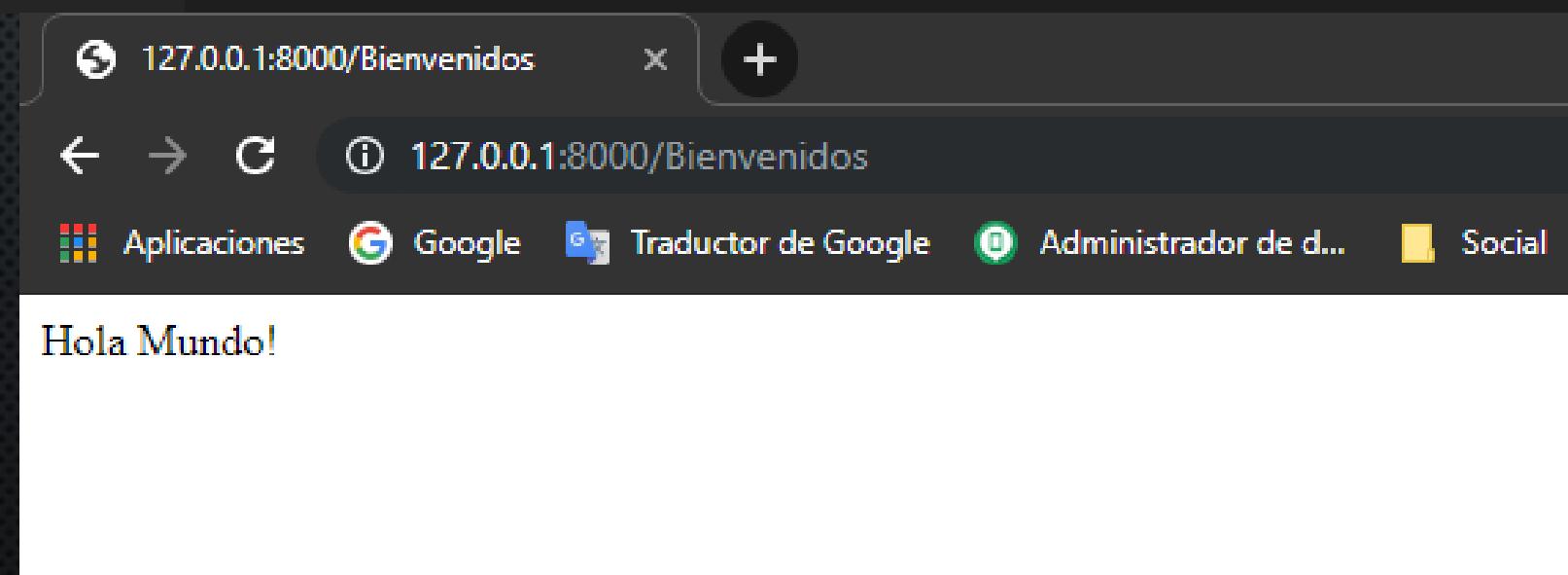
    <div class="links">
        <a href="https://laravel.com/docs">Docs</a>
        <a href="https://laracasts.com">Laracasts</a>
        <a href="https://laravel-news.com">News</a>
        <a href="https://blog.laravel.com">Blog</a>
        <a href="https://nova.laravel.com">Nova</a>
        <a href="https://forge.laravel.com">Forge</a>
        <a href="https://vapor.laravel.com">Vapor</a>
        <a href="https://github.com/laravel/laravel">GitHub</a>
    </div>
</div>
</body>
</html>
```

Ln 100, Col 8 Spaces: 4 UTF-8 LF PHP ⚡



# Programacion web usando Laravel

[DOCS](#)[LARACASTS](#)[NEWS](#)[BLOG](#)[NOVA](#)[FORGE](#)[VAPOR](#)[GITHUB](#)



A screenshot of a Laravel application running in a browser. The browser window shows a single tab with the URL `127.0.0.1:8000/Bienvenidos`. The page content displays the text "Hola Mundo!". The browser interface includes a back/forward button, a refresh button, and a search/address bar. Below the address bar are links for "Aplicaciones", "Google", "Traductor de Google", "Administrador de d...", and "Social".

EXPLORER

OPEN EDITORS

- welcome.blade.php resources\views
- web.php routes

BLOGPRACTICAS

- public
- resources
  - js
  - lang
  - sass
  - views
    - welcome.blade.php
- routes
  - api.php
  - channels.php
  - console.php
  - web.php
- storage
- tests
- vendor

welcome.blade.php web.php

```
routes > web.php
1 <?php
2
3 /*
4 |
5 | Web Routes
6 |
7 |
8 | Here is where you can register web routes for your application. These
9 | routes are loaded by the RouteServiceProvider within a group which
10 | contains the "web" middleware group. Now create something great!
11 |
12 */
13
14 Route::get('/', function () {
15     return view('welcome');
16 });
17 Route::get('Bienvenidos', function () {
18     return "Hola Mundo!";
19 });
20
```

The screenshot shows a development environment with a code editor and a browser window.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure under "BLOGPRACTICAS".
- Open Editors:** Shows "welcome.blade.php" and "web.php".
- Content:** The "web.php" file contains the following PHP code:

```
1 <?php
2
3 /*
4 |-----
5 | Web Routes
6 |-----
7 |
8 | Here is where you can register web routes for your application. These
9 | routes are loaded by the RouteServiceProvider within a group which
10 | contains the "web" middleware group. Now create something great!
11 |
12 */
13
14 Route::get('/', function () {
15     return view('welcome');
16 });
17 Route::get('Bienvenidos', function () {
18     return "Hola Mundo!";
19 });
20
```

**Browser:**

- Address Bar:** Shows "127.0.0.1:8000/Bienvenidos".
- Content:** Displays the text "Hola Mundo!".
- Toolbar:** Includes links for "Aplicaciones", "Google", "Traductor de Google", "Administrador de d...", and "Social".

# Rutas

Help

• web.php - blogPrácticas - Visual Studio Code

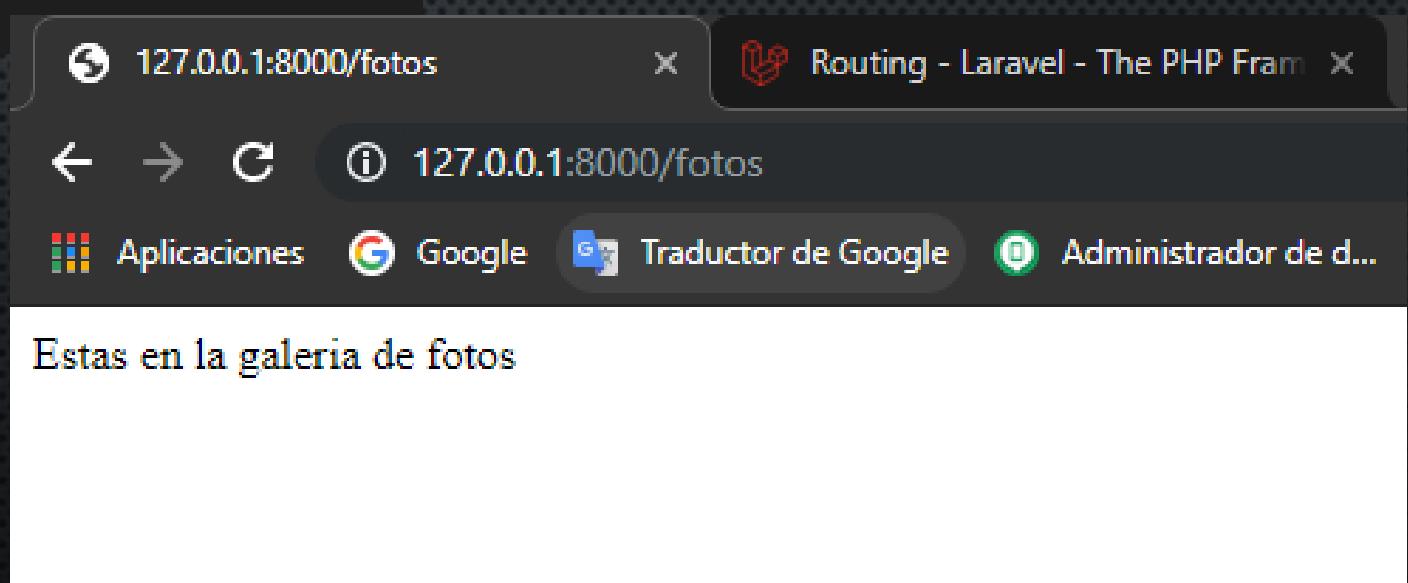
welcome.blade.php

web.php

routes > web.php

```
21
22 Route::get('fotos', function(){
23     return 'Estas en la galeria de fotos';
24 });
```

```
Route::get('foo', function () {
    return 'Hello World';
});
```



# Parámetros

nal Help

web.php - blogPracticas - Visual Studio Code

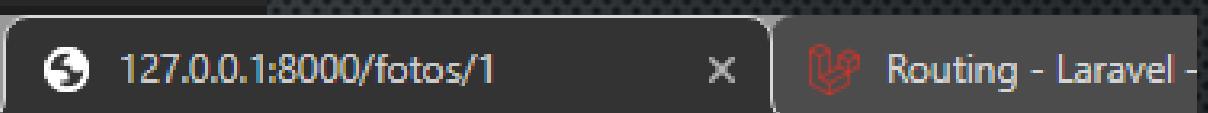
welcome.blade.php

web.php X

routes > welcome.blade.php

```
21  
22 <Route::get('fotos/{numero}', function($numero){  
23 |     return 'Estas en la galeria de fotos: '.$numero;  
24 });|
```

```
Route::get('user/{id}', function ($id) {  
    return 'User '.$id;  
});
```



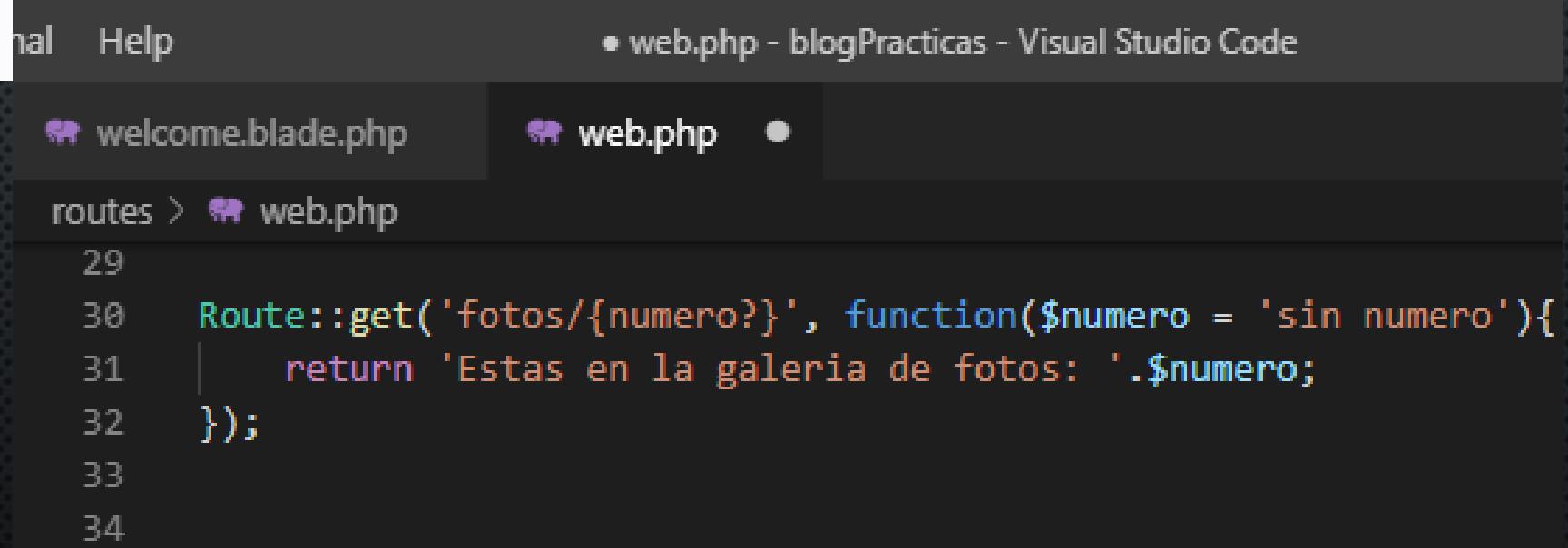
← → C ⌂ 127.0.0.1:8000/fotos/1

Aplicaciones Google Traductor de Google A

Estas en la galeria de fotos: 1

# Parámetros opcionales

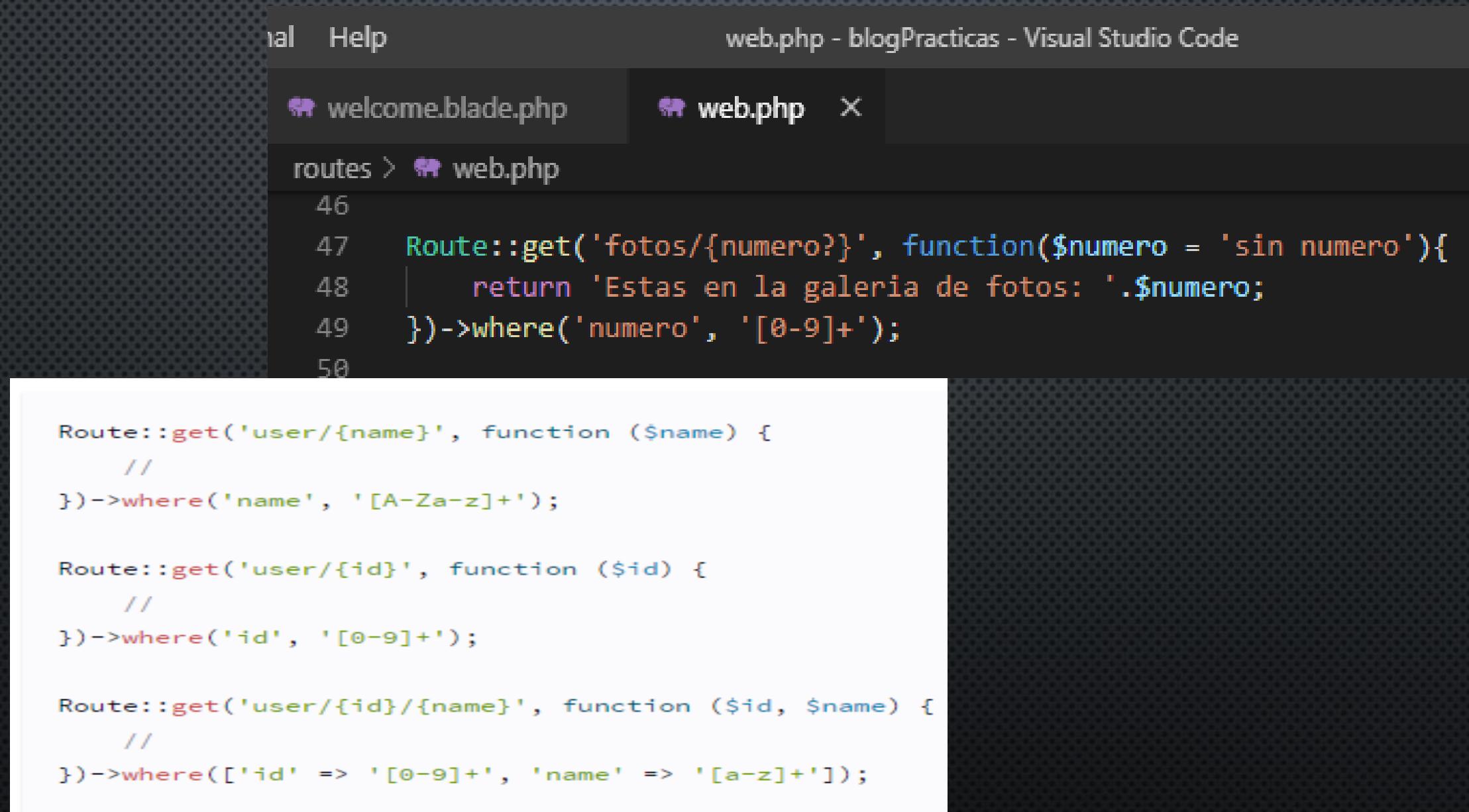
```
Route::get('user/{name?}', function ($name = null) {  
    return $name;  
});  
  
Route::get('user/{name?}', function ($name = 'John') {  
    return $name;  
});
```



A screenshot of the Visual Studio Code interface. The main area shows PHP code for defining routes. Below the code, the status bar displays the file path: routes > web.php. The bottom left corner shows line numbers from 29 to 34. The bottom right corner shows the current file name: web.php.

```
29  
30     Route::get('fotos/{numero?}', function($numero = 'sin numero'){  
31         return 'Estas en la galeria de fotos: '.$numero;  
32     });  
33  
34
```

# Parámetros condicionales(validaciones)



The screenshot shows a Visual Studio Code interface with two tabs: "welcome.blade.php" and "web.php". The "web.php" tab is active, displaying PHP code for route validation. The code uses the Route::get() method to define routes and the where() method to add validation conditions to the parameters.

```
46
47 Route::get('fotos/{numero?}', function($numero = 'sin numero'){
48     return 'Estas en la galeria de fotos: '.$numero;
49 })->where('numero', '[0-9]+');
50
Route::get('user/{name}', function ($name) {
    //
})->where('name', '[A-Za-z]+');

Route::get('user/{id}', function ($id) {
    //
})->where('id', '[0-9]+');

Route::get('user/{id}/{name}', function ($id, $name) {
    //
})->where(['id' => '[0-9]+', 'name' => '[a-z]+']);
```

# Vistas

The image shows a code editor interface with two panes. Both panes display the same file, `welcome.blade.php`, located at `resources\views\welcome.blade.php`. The code consists of the following lines:

```
routes > welcome.blade.php
20
21
22
23
24
25
26
27
28
29
```

The left pane's sidebar shows the project structure under the `BLOGPRACTICAS` folder:

- `> public`
- `< resources`
- `> js`
- `> lang`
- `> sass`
- `< views`

A context menu is open over the `views` folder, with the `New File` option highlighted.

The right pane also shows the same project structure, with the `fotos.php` file selected in the `views` folder.

# Vistas

nal Help • fotos.php - blogPracticas - Visual Studio Code

welcome.blade.php web.php fotos.php

resources > views > fotos.php

```
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>Document</title>
8  </head>
9  <body>
10 <h1>Estas en la galeria de fotos</h1>
11 </body>
12 </html>
```

Document Routing - Laravel - The PHP Framework

← → C ⚒ 127.0.0.1:8000/galeria

Aplicaciones Google Traductor de Google Administración

Estas en la galeria de fotos

nal Help • web.php - blogPracticas

welcome.blade.php web.php

routes > web.php

```
51
52     Route::view('galeria', 'fotos');
53
```

# Plantillas(Blade)

✓ views

- ≡ blog.blade.php
- ≡ fotos.blade.php
- ≡ plantilla.blade.php

# Plantillas(Blade)

```
resources > views > platilla.blade.php
1  doctype html>
2  tml lang="en">
3  <head>
4      <!-- Required meta tags -->
5      <meta charset="utf-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
7
8      <!-- Bootstrap CSS -->
9      <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1143x" crossorigin="anonymous">
10
11     <title>Hello, world!</title>
12 </head>
13 <body>
14     <h1>Hello, world!</h1>
15
16     <!-- Optional JavaScript -->
17     <!-- jQuery first, then Popper.js, then Bootstrap JS -->
18     <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965Dz00rT7abK41JStQ" crossorigin="anonymous">
19     <script src="https://cdn.jsdelivr.net/npm/popper.js@1.14.7/dist/umd/popper.min.js" integrity="sha384-UO2eT0CpHqdSJQyDKK3tafofInQFjTLtsg4KgIQukcxZK3eXKqZ9v9+wN" crossorigin="anonymous">
20     <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha384-J3tSClP5L/2J2q5h6BdWQo+JZG9hQZnV+DmZKpZQnXJZ" crossorigin="anonymous">
21 </body>
22 html>
```

# Plantillas(Blade)

Help      plantilla.blade.php - blogPracticas - Visual Studio Code

web.php    fotos.blade.php    blog.blade.php    plantilla.blade.php ×

resources > views > plantilla.blade.php > html > body > div.container

```
4   <!-- Required meta tags -->
5   <meta charset="utf-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
7
8   <!-- Bootstrap CSS -->
9   <link rel="stylesheet"
10  href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
11  integrity="sha384-gg0yR0iXCBMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T
12  hktb" crossorigin="anonymous">
13
14  <title>Hello, world!</title>
15  </head>
16  <body>
17  <div class="container">
18  | <a href="{{route('foto')}}" class="btn btn-primary">Photos</a>
19  | <a href="{{route('noticias')}}" class="btn btn-primary">Blog</a>
20  </div>
21  <div class="container">
22  | @yield('seccion')
23  </div>
24
```

# Estructuras de control

## views

- blog.blade.php
- fotos.blade.php
- nosotros.blade.php
- plantilla.blade.php
- welcome.blade.php

# ESTRUCTURAS DE CONTROL

Help • nosotros.blade.php - blogPracticas - Visual Studio Code

web.php    nosotros.blade.php ●    fotos.blade.php    blog.blade.php    plantilla.blade.php

resources > views > nosotros.blade.php > ...

```
1 @extends('plantilla')
2
3 @section('seccion')
4 <br><h1>Este es mi equipo de trabajo</h1><br>
5
6 @foreach ($equipo as $item)
7 <a href="{{route('nosotros', $item)}}" class="h4 text-danger">{{$item}}</a><br>
8 @endforeach
9
10@if(!empty($nombre))
11
12@switch($nombre)
13    @case($nombre == 'Luis')
14        <h2 class="mt-5">El nombre es {{$nombre}}:</h2>
15        <p>mide 1.70 metros</p>
16        @break
17
18    @case($nombre == 'Fernanda')
19        <h2 class="mt-5">El nombre es {{$nombre}}:</h2>
20        <p>mide 1.69 metros</p>
21        @break
22
23    @case($nombre == 'Francisco')
24        <h2 class="mt-5">El nombre es {{$nombre}}:</h2>
25        <p>mide 1.73 metros</p>
26        @break
27
28@endswitch
29
30@endif
31
32@endsection
```

Help routes > web.php

web.php    nosotros.blade.php ●    fotos.blade.php    blog.blade.php    plantilla.blade.php

```
62
63 Route::get('nosotros/{nombre?}', function($nombre = null){
64     $equipo = ['Luis', 'Fernanda', 'Francisco'];
65
66     return view('nosotros', compact('equipo', 'nombre'));
67 })->name('nosotros');
```

Help plantilla.blade.php - blogPracticas - visual Studio Code

web.php    nosotros.blade.php ●    fotos.blade.php    blog.blade.php    plantilla.blade.php

sources > views > plantilla.blade.php > html > body

```
L7 <div class="container">
L8     <a href="{{route('foto')}}" class="btn btn-primary">Fotos</a>
L9     <a href="{{route('noticias')}}" class="btn btn-primary">Blog</a>
L0     <a href="{{route('nosotros')}}" class="btn btn-primary">Nosotros</a>
L1 </div>
```

# Estructuras de control

A screenshot of a web browser window. The title bar shows three tabs: '¡Hola Mundo!', 'Routing - Laravel - The PHP Fre...', and 'Introducción · Bootstrap'. The address bar displays '127.0.0.1:8000/nosotros'. Below the address bar is a navigation bar with links: 'Aplicaciones', 'Google', 'Traductor de Google', 'Administrador de d...', 'Social', 'Player's', and a search bar. The main content area contains the heading 'Este es mi equipo de trabajo' and a list of names: Luis, Fernanda, and Francisco.

Este es mi equipo de trabajo

Luis  
Fernanda  
Francisco

A screenshot of a web browser window, similar to the one above. The title bar shows the same three tabs: '¡Hola Mundo!', 'Routing - Laravel - The PHP Fre...', and 'Introducción · Bootstrap'. The address bar now displays '127.0.0.1:8000/nosotros/Luis'. The main content area contains the heading 'Este es mi equipo de trabajo' and a list of names: Luis, Fernanda, and Francisco. Below this, there is additional text: 'El nombre es Luis:' followed by 'mide 1.70 metros'.

Este es mi equipo de trabajo

Luis  
Fernanda  
Francisco

El nombre es Luis:  
mide 1.70 metros

# Controlador

En lugar de definir toda su lógica de manejo de solicitudes como Cierres en archivos de ruta, es posible que desee organizar este comportamiento utilizando las clases de Controlador. Los controladores pueden agrupar la lógica de manejo de solicitudes relacionadas en una sola clase. Los controladores se almacenan en el directorio.

.app/Http/Controllers

```
<?php

namespace App\Http\Controllers;

use App\Http\Controllers\Controller;
use App\User;

class UserController extends Controller
{
    /**
     * Show the profile for the given user.
     *
     * @param int $id
     * @return View
     */
    public function show($id)
    {
        return view('user.profile', ['user' => User::findOrFail($id)]);
    }
}
```

# Controlador

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

Copyright (C) Microsoft Corporation. Todos los derechos reservados.

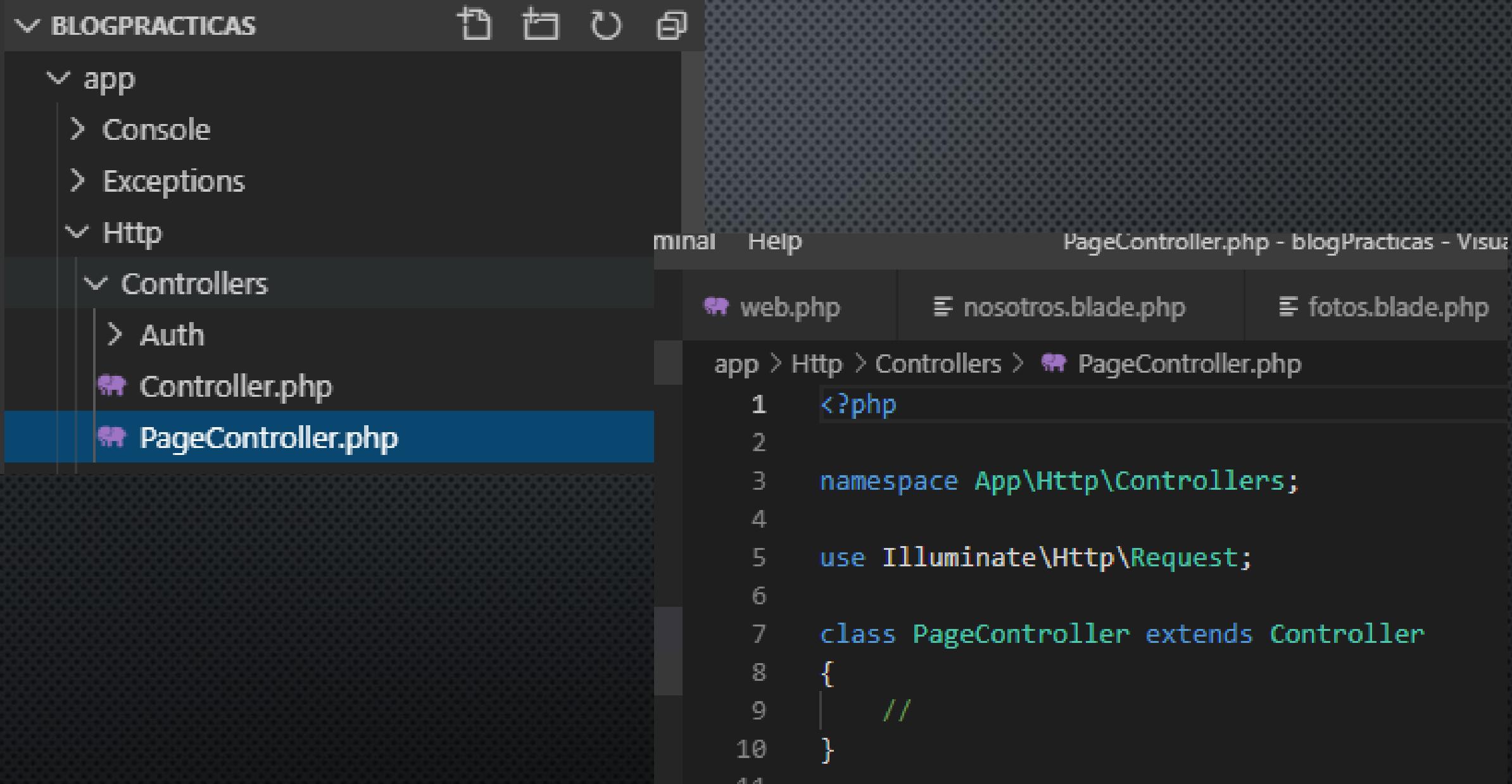
Prueba la nueva tecnología PowerShell multiplataforma <https://aka.ms/pscore6>

```
PS C:\xampp\htdocs\proyectos\laravel\blogPracticas>
```

```
PS C:\xampp\htdocs\proyectos\laravel\blogPracticas> php artisan make:controller PageController  
Controller created successfully.
```

```
PS C:\xampp\htdocs\proyectos\laravel\blogPracticas> 
```

# Controlador



The screenshot shows a code editor interface with a dark theme. On the left, there's a sidebar titled "BLOGPRACTICAS" containing a tree view of project structure. The "app" folder is expanded, showing "Console", "Exceptions", "Http", and "Controllers". "Controllers" is also expanded, showing "Auth", "Controller.php", and "PageController.php". "PageController.php" is highlighted with a blue background. The main editor area has tabs for "web.php", "nosotros.blade.php", and "fotos.blade.php". The current file is "PageController.php". The status bar at the bottom shows line numbers from 1 to 10.

```
mininal Help PageController.php - blogPracticas - Visual Studio Code
web.php nosotros.blade.php fotos.blade.php
app > Http > Controllers > PageController.php
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 class PageController extends Controller
8 {
9 |     //
10 }
```

# Controlador

File Help PageController.php - blogPracticas - Visual Studio Code

app > Http > Controllers > PageController.php

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 class PageController extends Controller
8 {
9     public function inicio(){
10         return view('welcome');
11     }
12     public function fotos(){
13         return view('fotos');
14     }
15     public function noticias(){
16         return view('blog');
17     }
18     public function nosotros[$nombre = null]{
19         $equipo = ['Luis', 'Fernanda', 'Francisco'];
20
21         return view('nosotros',compact('equipo', 'nombre'));
22     }
23 }
```

File Help web.php - blogPracticas - Visual Studio Code

routes > web.php

```
71
72     Route::get('/', 'PageController@inicio');
73
74     Route::get('fotos', 'PageController@fotos')->name('foto');
75
76     Route::get('blog', 'PageController@noticias')->name('noticias');
77
78     Route::get('nosotros/{nombre?}', 'PageController@nosotros')->name('nosotros');
```



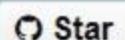
# Alternativas a Laravel



**CodeIgniter es un framework PHP que usa una arquitectura de Model View Controller (MVC).** En términos sencillos, eso significa que CodeIgniter utiliza diferentes componentes para manejar tareas de desarrollo específicas. Este enfoque es muy popular entre los desarrolladores porque permite crear aplicaciones web altamente escalables con un tamaño más reducido.

## CodeIgniter Rocks

CodeIgniter is a powerful PHP framework with a very small footprint, built for developers who need a simple and elegant toolkit to create full-featured web applications.



Star

16,504



Fork

7,505



Follow

22.5K followers

### Características principales:

- Utiliza un framework ligero, hecho pensando en el rendimiento.
- Comienza rápidamente, gracias a la simplicidad del framework y la excelente documentación.
- Crea aplicaciones escalables utilizando la arquitectura basada en MVC.



Cuenta con un paquete y un sistema de componentes que te permite elegir las funciones de PHP que necesitas, o simplemente usar todo el framework.

Para garantizar que tus aplicaciones se ejecuten correctamente, el framework también incluye la funcionalidad de prueba incorporada.

Symfony

Sponsored by SensioLabs

What is Symfony? Documentation Community Business Solutions Events News Account DOWNLOAD

Symfony is a set of reusable **PHP components**...

The standard foundation on which the best PHP applications are built. Choose any of the 50 stand-alone components available for your own applications.

[Browse components →](#)

... and a **PHP framework** for web projects

Speed up the creation and maintenance of your PHP web applications. End repetitive coding tasks and enjoy the power of controlling your code.

[What Is Symfony →](#)

## Características principales:

- Utiliza un framework altamente flexible que te permite configurar componentes individuales.
- Aprovecha la funcionalidad de prueba incorporada en el framework.
- Aprende a usar la plataforma a través de su excelente documentación.



# Zend

Mucha gente se refiere a Zend como un framework de “pegamento”, que es una forma de referirse a su naturaleza basada en componentes. Zend es un framework basado en MVC, orientado a objetos, que permite cargar solo los componentes que quieras como bibliotecas individuales.

## Zend Framework

Professional PHP packages ready for [PHP 7](#)

Focused on Simplicity, Reusability, and Performance

289,807,249 installs from



### Características principales:

- Utiliza un framework PHP orientado a objetos con una arquitectura MVC.
- Reutiliza tu código gracias al diseño de la plataforma.
- Integra Zend con bibliotecas externas fácilmente.
- Usa solo los componentes que deseas e ignora todo lo demás.



# Phalcon

Phalcon es un poco extraño en el mundo de los frameworks PHP. Su código fuente está escrito en C, por lo que es básicamente una extensión C de PHP. Esto suena raro, pero en la práctica, resulta ser uno de los frameworks más rápidos que existen hasta la fecha.

## A full-stack PHP framework delivered as a C-extension

Its innovative architecture makes Phalcon the fastest PHP framework ever built! See for yourself...

 Get Phalcon 3.3

### Características principales:

- Usa un framework PHP basado en C.
- Aprovecha el fantástico rendimiento de Phalcon y la reduce sobrecarga de recursos.
- Utiliza solo los módulos y bibliotecas que necesitas.



# CakePHP

A principios de la década del 2000, CakePHP fue el primer framework MVC de PHP en salir al mercado. En ese entonces fue una revelación, y sigue siendo uno de los mejores frameworks PHP que puedes usar (y uno de los más populares).

Las nuevas versiones de CakePHP han mejorado su rendimiento a lo largo del tiempo y han agregado muchos componentes nuevos. Sin embargo, donde CakePHP realmente se destaca es en la forma en que aborda las convenciones de programación. Esto significa que con CakePHP, una vez que domines tu conjunto de convenciones, puedes centrarte en el desarrollo y hacer más trabajo más rápido.

New CakePHP 3.6 Red Velvet.  
Faster. Stronger. Tastier.

## Características principales:

- Aprovecha un amplio conjunto de componentes.
- Usa las convenciones de CakePHP para programar proyectos más rápido.



# Bibliografía y recursos.

Laravel.com: <https://laravel.com/>

getcomposer.org: <https://getcomposer.org/>

Wikipedia: <https://es.wikipedia.org/wiki/Laravel>

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

Bluuweb!: <https://www.youtube.com/user/Bluuweb>