

Creación de un Layout MVC en Laravel

proyectolayout/

```
|
|
├── app/
|   ├── Http/
|       ├── Controllers/
|           ├── Controller.php
|           ├── RegistroController.php
|           └── UsuarioController.php
|
|
├── public/
|   ├── index.php
|   ├── css/
|       └── estilos.css
|
|
├── resources/
|   ├── views/
|       ├── layout.blade.php
|       ├── inicio.blade.php
|       ├── registro.blade.php
|       ├── login.blade.php
|       └── logadoOk.blade.php
|
|
├── routes/
|   └── web.php
|
|
├── bootstrap/
|   └── app.php
|
|
├── artisan
└── composer.json
```

└── .env

Las vistas en Laravel:

Blade es el **motor de plantillas** de Laravel.

Permite crear vistas en HTML combinadas con código PHP de una forma más limpia y ordenada, usando **directivas** (palabras que comienzan con @) en lugar de escribir directamente bloques de PHP.

Ventajas de Blade:

- Separa la lógica del diseño.
- Permite **heredar estructuras comunes**.
- Facilita **reutilizar código** (por ejemplo, cabeceras o pies de página).
- Es fácil de leer, ya que sustituye las etiquetas de PHP por una sintaxis sencilla.

Directivas blade iniciales

{{ \$nombre }} muestra el valor de una variable enviada desde el controlador.

@extends indica que una vista **hereda** de un layout o de otra plantilla. Es decir, que su contenido se mostrará dentro del layout principal.

@yield se coloca **en el layout** y **marca un espacio reservado** donde las vistas hijas insertarán su contenido.

```
<main>
```

```
    @yield('contenido')
```

```
</main>
```

@section define el contenido que se colocará dentro de una sección concreta del layout. Por ejemplo, si en el layout hay un **@yield('contenido')**, en la vista hija se escribirá:

```
@section('contenido')
```

```
    <h2>Página de inicio</h2>
```

```
    <p>Bienvenido al portal del Instituto Infanta Elena.</p>
```

```
@endsection
```

@include permite **insertar una vista dentro de otra**, ideal para fragmentos que se repiten, como un menú, una cabecera o un pie.

asset() es un **helper de Laravel** que crea automáticamente la **ruta pública correcta** hacia un recurso (imagen, CSS, JavaScript, etc.), sin que tengas que escribir la ruta completa.

```
public/
```

```
|── css/
```

```
|   └── app.css
```

```
|── images/
```

```
|   └── logo.png
```

```
└── js/
    └── main.js
```

```

```

<http://localhost:8000/images/logo.png>

Paso 1: Crear el proyecto en mi carpeta htdocs

Me sitúo en la carpeta htdocs y ejecuto el comando:

```
composer create-project laravel/laravel layouts1
```

Paso 2: Abrir el proyecto en VisualCodec

Paso 3: Subir style.css al proyecto

Paso 3: Lenvatar servidor web de artisan

```
php artisan serve --port=8085
```

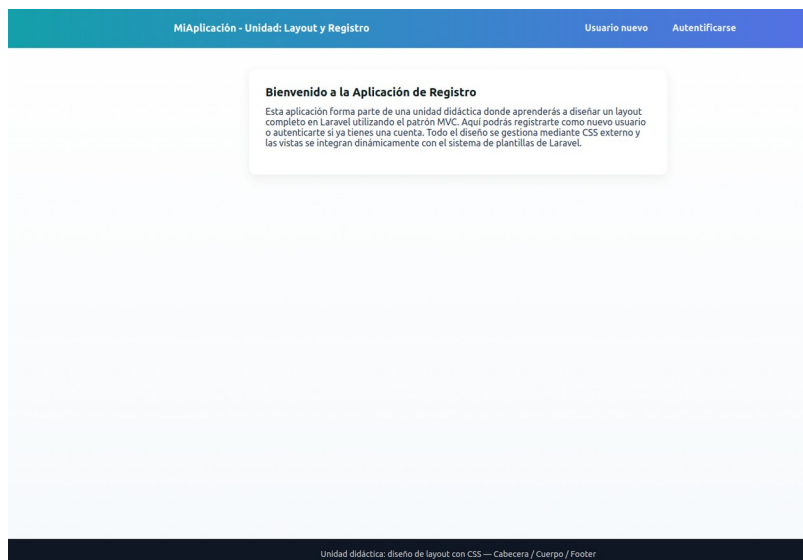
Paso 4: Crear mi layout

Tendrá una zona de cabecera, que variará. Podrá tener dos enlaces (nuevo usuario, autenticación), un texto que indique que formulario estamos usando(registrarse, autenticarse), o el nombre del usuario registrado.

Tendrá una zona de cuerpo, que podrá tener un texto general, un formulario de registro, un formulario de autenticación, un mensaje de bienvenida, un mensaje de error de registro.....

Tendrá una zona de footer siempre igual en todas las páginas.

El primer layout que vamos a preparar es para la usarlo en la página de inicio:



(Fichero Layout.blade.php)

Paso 5: Crear pagina inicial usando el layout.

(Fichero inicio1.blade.php)

Paso 6: Crear el controlador inicial RegistroController.php

```
php artisan make:controller RegistroController
```

(Fichero RegistroController1.php)

Paso1: Preparar las rutas de inicio por get y las de los dos enlaces que tenemos en la cabecera.

(Fichero web1.html)

Como nuestra página de inicio lleva dos url, debemos preparar la entrada para cada uno de los enlaces.

Paso 7: Crear las vistas **registro.blade.php** y **login.blade.php**

(Ficheros registro1.blade.php y login1.blade.php)

MiAplicación - Unidad: Layout y Registro

Nuevo Usuario

Registro de Nuevo Usuario

Solo se permite el registro de personas mayores de edad. Por favor, completa los siguientes datos:

Nombre:

rocio

Edad:

14

Registrarse

Volver al inicio

Unidad didáctica: diseño de layout con CSS — Cabecera / Cuerpo / Footer

MiAplicación - Unidad: Layout y Registro

Bienvenida

Autenticación de Usuario

Introduce tus credenciales para acceder a la aplicación.

Nombre de usuario:

Contraseña:

Entrar

Volver al inicio

Unidad didáctica: diseño de layout con CSS — Cabecera / Cuerpo / Footer