

Laravel: Plantillas BLADE

Blade es un motor de plantillas:

- Facilita la incrustación de PHP en HTML
- Ofrece varios atajos, que permiten, con poco texto, hacer muchas cosas.

Diseño Principal:

Si tenemos muchas vistas, tendremos mucho HTML, con partes comunes para una visualización homogénea de mi aplicación. Es tedioso andar copiando y pegando de una vista a otra, además es poco limpio, y difícil de mantener. ¿ No sería mejor hacer un include de las partes comunes ?

EJEMPLO:

Para ilustrar este concepto con un ejemplo práctico, vamos a organizar nuestras vistas. Crearemos un directorio dentro de views llamado layouts (diseño), y crearemos un fichero, cuyo nombre es típico entre usuarios de Laravel:

`views/layouts/app.blade.php`

El `app.blade.php` será la *plantilla principal* de nuestra aplicación. Veamos un ejemplo:

`app.blade.php`

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Plantilla</title>
</head>
<body>
    <div class="container">
        @yield('content')
    </div>

    @yield('footer')
</body>
</html>
```

Nota: La clase “container” del div se añade en previsión del uso de bootstrap.

Una vez creada una “plantilla”, vamos a modificar una de nuestras vistas, para que la utilice:

`contacto.blade.php`

```
@extends('layouts.app')

@section('content')
    <h1>Página de Contactos</h1>

@stop
```

@extends se utiliza de forma equivalente a un include. En este caso, le dice a blade que utilizará el código HTML incluido en `app.blade.php` que está dentro de la subcarpeta de views llamada **layouts**

@section ... @stop Se utiliza para escribir código (HTML, JavaScript, etc), que será colocado en el lugar de la plantilla donde haya un `@yield` con un parámetro de igual nombre. En nuestro ejemplo, tenemos una sección para el contenido llamada “content”, que irá dentro del `<div>` de la plantilla, justo donde hemos puesto el `@yield('content')`.

La forma de añadir JavaScript sería:

```
@section('footer')
    <script>alert('Hola amigo')</script>
@endsection
```

En este caso, el script se añadiría en la parte de la plantilla donde hemos escrito `@yield('footer')`.

EJERCICIO

Modifica la vista “ciudad” que utilizaste en la ruta viaje del último ejercicio, para que utilice la plantilla app que hemos creado.

La información debe visualizarse igual que antes.

Más opciones con Blade:

EJEMPLO:

Vamos a modificar el controlador.

Vamos al método contacto, y añadimos un array y algo más de código:

PostController.php

```
public function contacto() {
    $gente = ['Pedro', 'Carlos', 'Juan', 'Nacho', 'Silvia'];
    return view('contacto', compact('gente'));
}
```

En esta ocasión, la variable gente no es un parámetro de la función, sino una variable local definida dentro. En el ejemplo utilizamos un array con 5 elementos.

Ahora modificaré la vista de contacto para que muestre esta información.

Una de las ventajas de blade, es que tiene un lenguaje propio que me permite hacer bucles, bifurcaciones, sus propios includes, y más cosas.

contacto.blade.php

```
@extends('layouts.app')

@section('content')
    <h1>Página de Contactos</h1>

    @if (count($gente))
        <ul>
            @foreach($gente as $persona)
                <li>{{$persona}}</li>
            @endforeach
        </ul>
    @endif

@endsection

@section('footer')
    {{--<script>alert('Hello visitor')</script>--}}
@endsection
```

Hemos utilizado los **comentarios** de blade {{-- comentarios --}}

Hemos utilizado un **foreach** de blade

Hemos utilizado un **if** de blade

El resultado es el siguiente:



Las plantillas hechas con blade se utilizan mucho en las aplicaciones Laravel.

Cómo incluir CSS o JavaScript en nuestras vistas

Podemos utilizar ficheros javascript y css en nuestras vistas. Sin embargo, para hacer referencia al fichero tenemos que tener en cuenta lo siguiente:

1. Lo que no esté en la carpeta public de Laravel, no va a ser accesible
2. No vamos a crear una ruta para enlazar a un fichero CSS

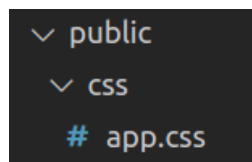
Para poder utilizar estos recursos:

1. Crearemos subcarpetas en el directorio public de mi aplicación, accesible desde el servidor web
2. Haremos referencia desde HTML, utilizando las funciones que nos retornan la ruta a los recursos, entre llaves dobles:

`{{asset('css/miCss.css')}} o bien {{url('css/miCss.css')}}`

Ejemplo: Utilizar un fichero CSS

Lo primero que tenemos que hacer, es crear una carpeta css colgando de public, y poner dentro nuestros ficheros css. Esto hará accesible el fichero.



Lo siguiente, tendremos que linkarlo en nuestro HTML. Para ello:

miVista.blade.php

```
...
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>MiWeb</title>
  <link rel='stylesheet' href="{{asset('css/app.css')}}">
</head>
...
```

La función **asset** (recurso) recupera la URL correspondiente a este fichero CSS, en este caso sería:

<http://localhost:8000/css/miCss.css>

asset lo buscará, siempre, en la carpeta **public**, que será la única accesible desde el servidor web.

Otra función que podemos utilizar es **url**, que también recuperará la ruta del css.

EJERCICIO:

Para aprovechar la flexibilidad de blade, vas a crear una miniweb con una página principal, y otras dos páginas: visitas e inscripción.

Elabora el ejercicio con las siguientes pautas:

Plantilla:

Elaborarás tres vistas, una para cada página, que estarán basadas en la siguiente plantilla:

1. Un menú en la parte superior con dos opciones:
 - a. Visitas: El href llevará a la ruta “/visitas”
 - b. Inscripción: El href llevará a la ruta “/inscripcion”
2. Una parte de contenido, diferente según la página que haya cargada

Vistas:

La página principal mostrará un saludo de bienvenida.

La página de visitas mostrará una lista de varios lugares a visitar. De cada lugar se mostrará nombre y descripción. No escribirás estos datos en la vista, los pasarás dentro de un array desde el controlador.

La página de inscripción mostrará un formulario de inscripción.

NOTA: Para escribir en tus vistas una URL definida como ruta en web.php, puedes utilizar el siguiente código:

```
<a href={{ url('visitas') }}>Visitas</a>
```

SOLUCIÓN PROPUESTA DEL EJERCICIO

A continuación se muestra el código de una solución propuesta.

La plantilla estará en “view/layouts”. El resto de vistas estarán en “view/miniapli”

web.php

```
// Miniaplicación: Visitas y Contacto
Route::get('/', '\App\Http\Controllers\PostController@principal');
Route::get('/visitas', '\App\Http\Controllers\PostController@visitas');
Route::get('/inscripcion', '\App\Http\Controllers\PostController@inscripcion');
```

PostController.php

```
public function principal() {
    return view('miniapli/principal');
}

public function visitas() {
    $datos = [
        'Prado'=>'Museo de Arte de Madrid',
        'Colon'=>'Plaza ajardinada en el centro de Madrid',
        'Cibeles'=>'Emblemática fuente de Madrid'
    ];
    return view('miniapli/visitas',compact('datos'));
}

public function inscripcion() {
    return view('miniapli/inscripcion');
}
```

plantilla.blade.php

```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>MiniApli</title>
</head>
<body>
    <!-- Esto será igual en todas las páginas de mi aplicación -->
    <nav>
        <a href={{ url('/') }}>Página Principal</a>
        <a href={{ url('visitas') }}>Visitas</a>
        <a href={{ url('inscripcion') }}>Contacto</a>
    </nav>
    <!-- El contenido cambiará según la página de mi aplicación -->
    <div class="container">
        @yield('contenido')
    </div>
</body>
</html>
```

principal.blade.php

```
@extends('layouts.plantilla')

@section('contenido')
    <h1> Bienvenido a mi MiniAplicación </h1>
    <hr>
@endsection
```

visitas.blade.php

```
@extends('layouts.plantilla')

@section('contenido')

<table border="1">
    @foreach($datos as $nombre=>$descripcion)
        <tr>
            <td>{{ $nombre }}</td>
            <td>{{ $descripcion }}</td>
        </tr>
    @endforeach
</table>

@endsection
```

inscripcion.blade.php

```
@extends('layouts.plantilla')

@section('contenido')
    <form>
        Nombre: <input name="nombre"/><br>
        Apellidos: <input name="apellidos"/><br>
        <br>
        <input type="submit" value="Inscribir"/>
    </form>

@endsection
```