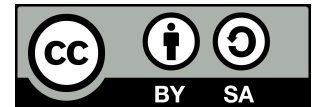


CURSO .PHP

MOTORES DE PLANTILLAS

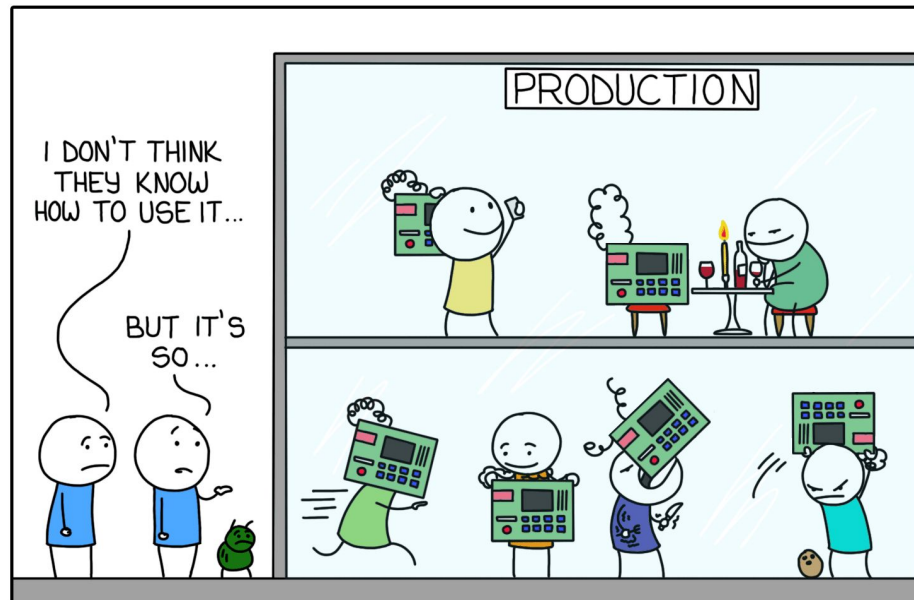


Autor: Jon Vadillo
www.jonvadillo.com

Contenidos

- Fundamentos básicos
- Instalar Blade
- Hello Blade!
- Directivas básicas
- Sub-vistas
- Layouts

OBVIOUS



MONKEYUSER.COM

Plantillas (templates)

hola.blade.php

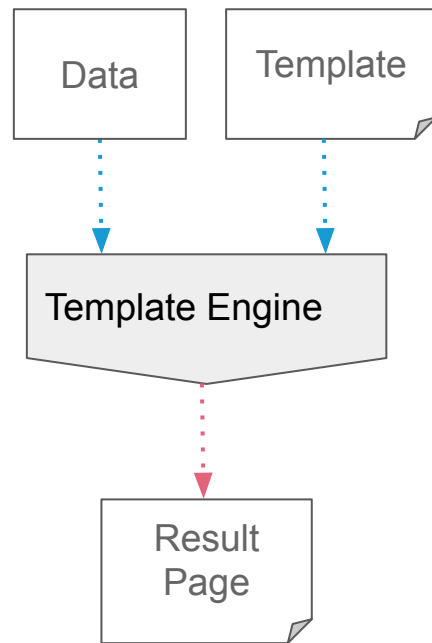
```
<p>Hola {{ $nombre }} </p>
```

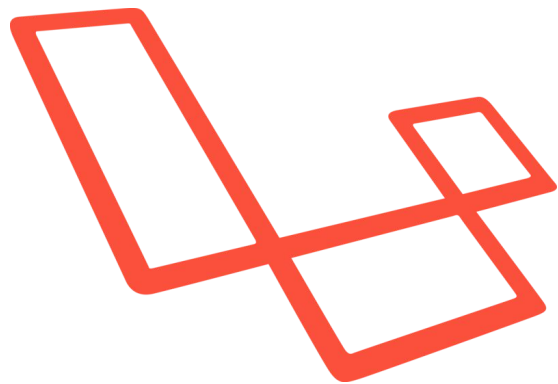


```
<p>Hola Amaia</p>
```

Plantillas (templates)

- Los **motores de plantillas** (templates engines) leen un fichero de texto (**plantilla**) que **define la estructura o diseño** e insertan en él los **datos de forma dinámica**.
- Las plantillas incluyen **variables** que serán reemplazadas por la información y **etiquetas** para controlar la lógica





Ejemplo de plantilla

listado.blade.php

```
<h1>Lista de pacientes:</h1>
<ul>
@foreach($pacientes as $paciente)
    <li>{{ $paciente->nombre }}</li>
@endforeach
</ul>
```

index.php

```
echo Blade::render("listado", [
    "pacientes" => $listado_pacientes
]);
```

Instalar Blade (sin usar Laravel)

- Blade es el motor de plantillas incluido en Laravel.
- Hay versiones standalone que permiten usarlo sin utilizar Laravel, como por ejemplo **eftec\bladeone**.
- Instalación:

```
$ composer require eftec\bladeone
```

- Las plantillas deberán estar en el directorio `/views`
- Es necesario crear un directorio `/compiles` para que deje el resultado de la compilación de las vistas.

Hello Blade

hello.blade.php

```
<h1>Hello {{ $nombre }}</h1>
```

index.php

```
require 'vendor/autoload.php';  
use eftec\bladeone\BladeOne;  
  
$blade = new BladeOne();  
echo $blade->run("hello", [  
    "nombre" => "Blade"  
]);
```

Nota para Vagrant



Tip

- En caso de tener errores de permisos, sigue los siguientes pasos:
 - a. Modifica el usuario y el grupo de usuario con el que se ejecuta Apache, estableciéndolo como vagrant:

```
sudo nano /etc/apache2/envvars
```

```
export APACHE_RUN_USER=vagrant  
export APACHE_RUN_GROUP=vagrant
```

- b. Reinicia el servicio de Apache: `sudo service apache2 restart`

Hands on!



Ejercicio 1

- Crea un nuevo proyecto e instala la librería `eftec\bladeone`.
- Crea una plantilla que muestre un `<h1>` con el siguiente mensaje: “¡Bienvenido *NOMBRE!*”, siendo *NOMBRE* el nombre de una persona enviada como variable.

Hands on!



Ejercicio 2

- Crea un nuevo proyecto e instala la librería `eftec\bladeone`.
- Crea la clase `Zapatilla` con los atributos: `marca`, `modelo`, `tipo` y `precio` (p. ej.: “Adidas”, “MARQUEE BOOST”, “Zapatilla de baloncesto”, “140€”).
- Crea una plantilla que muestre que reciba la información de una instancia de la clase `Zapatilla` y muestre el siguiente resultado:

Marca	Adidas
Modelo	MARQUEE BOOST
Tipo	Zapatilla de baloncesto
Precio	140€

Directivas básicas

<code>{{-- text --}}</code>
<code>{{ \$variable1 }}</code>
<code>{{ \$name or 'Default' }}</code>
<code>@if (boolean) / @else</code>
<code>@elseif (boolean)</code>
<code>@for(\$variable;\$condition;\$inc) / @endfor</code>
<code>@foreach(\$array as \$item) / @endforeach</code>
<code>@forelse(\$arr as \$item) / @empty / @endforelse</code>
<code>@while(\$condition) / @endwhile</code>

Ejemplos

```
@foreach($users as $user)
    @if($user->type == 1)
        @continue
    @endif
    <li>{{ $user->name }}</li>

    @if($user->number == 5) // ends loop
        @break
    @endif
@endforeach
```

```
@switch($countrySelected)
    @case(1)
        first country selected <br>
        @break
    @case(2)
        second country selected <br>
        @break
    @default
        other country selected <br>
@endswitch
```

Hands on!



Ejercicio 3

- Crea un nuevo proyecto e instala la librería `eftec\bladeone`.
- Crea una colección de 10 zapatillas utilizando la clase creada en el ejercicio anterior. Almacénalas en un array.
- Crea una vista que muestre una tabla con todas las zapatillas del array.

Marca	Modelo	Tipo	Precio
Adidas	MARQUEE BOOST	Zapatilla de baloncesto	140€
Nike	Messi X7 Plus	Zapatilla de fútbol	99€
Nike	Air Jordan V1	Zapatilla de baloncesto	135€

Incluir sub-vistas

```
<div>

  @include('shared.errors')

  <form>
    <!-- Form -->
  </form>
</div>
```

```
@include('view.name', ['some' => 'data'])

@includeIf('view.name', ['some' => 'data'])

@includeWhen($boolean, 'view.name', [
    'some' => 'data'
])
```


Hands on!



Ejercicio 4

- Incluye a la vista del ejercicio anterior una subvista como la de la siguiente imagen (la sub-vista recibirá el nombre de la tienda en una variable):

Listado de zapatillas de la tienda ZAPAPLUS:

Marca	Modelo	Tipo	Precio
Adidas	MARQUEE BOOST	Zapatilla de baloncesto	140€
Nike	Messi X7 Plus	Zapatilla de fútbol	99€
Nike	Air Jordan V1	Zapatilla de baloncesto	135€

Incluir sub-vistas en colecciones

- Es posible incluir una sub-vista para cada elemento de una colección.
- También es posible incluir un parámetro extra donde se especifique la vista a incluir si la colección está vacía.

```
@each('view.name', $jobs, 'job')
```

```
@each('view.name', $jobs, 'job', 'view.empty')
```

Definir layouts

- La mayoría de las aplicaciones mantienen una **estructura común** en la mayoría de sus páginas. Los layouts permiten definir la estructura y reutilizarla.
- `@yield` sirve para mostrar el contenido de una sección.

```
<html>
<head>
<title>Mi app!</title>
</head>
<body>
<div class="container">
  @yield('content')
</div>
</body>
</html>
```

Extender un layout

- Se definen vistas que hereden del layout.
- `@extends` indica el layout del que está heredando.
- Se indica el contenido a incluir en cada sección del layout (definidas con `@yield`) mediante la directiva `@section` / `@endsection`.

```
@extends('layouts.app')  
  
@section('content')  
    <p>This is my body content.</p>  
@endsection
```

Hands on!



Ejercicio 5

- Partiendo del ejercicio anterior, crea un layout común que tendrá toda la estructura de la página HTML y un `<h1>`.
- La vista del ejercicio anterior extenderá la plantilla creada.

Hands on!



Ejercicio 6

- Partiendo de los ejercicios anteriores, crearás una aplicación con 2 vistas:
 - Listado de zapatillas: idéntica a la del ejercicio 5 pero con una columna adicional que incluirá un enlace a la vista de detalle.
 - Detalle de zapatilla: mostrará únicamente los datos de la zapatilla indicada (como en el ejercicio 2). Tendrá un enlace para volver al listado de zapatillas.
 - Nota: tendrás que añadir la propiedad ID a la clase Zapatilla para crear el enlace
- Ambas vistas tendrán un layout común que incluirá un `<h1>` con el título de la tienda y un `<h2>` que muestre al usuario la página actual en la que está.

Sources

- [eftec\bladeone.](http://duncan3dc.github.io): <http://duncan3dc.github.io>
- [Laravel Blade official documentation](https://github.com/EFTEC/BladeOne): <https://github.com/EFTEC/BladeOne>