



# Desarrollo de aplicaciones con Symfony 4

Creando y usando templates



# Templates

Aunque lo veremos más adelante, vamos a ver algunos conceptos básicos de Twig.

# Templates: Sintaxis

Twig define tres tipos de sintaxis especial:

`{{. ..}}`

"Dice algo": imprime una variable o el resultado de una expresión en la plantilla.

`{% ...%}`

"Hace algo": una etiqueta que controla la lógica de la plantilla; se utiliza para ejecutar instrucciones como for-loops por ejemplo.

`{# ... #}`

"Comentar algo": es el equivalente de la sintaxis PHP `/* comment */`. Se utiliza para agregar comentarios de una o varias líneas.

# Templates: Sintaxis

Como hemos visto en el ejemplo de nuestra primera página:

```
{# templates/page/index.html.twig #}  
<h1>Hola mundo, soy {{ name }}</h1>
```

{{ }} se ha usado para imprimir una variable que le hemos pasado  
y  
{# #} para poner un comentario.

# Templates: Sintaxis

```
{% if name == 'David' %}  
    <li>Hola, soy {{ name }}</li>  
{% else %}  
    <li>No se quien es {{ name }}</li>  
{% endif %}
```

# Templates: Sintaxis

Si en lugar de un string le hubiésemos pasado un array y queremos recorrerlo:

```
<ul>
```

```
    {% for user in users %}
```

```
        <li>{{ user.username }}</li>
```

```
    {% endfor %}
```

```
</ul>
```

# Templates: Filters

Twig también contiene filtros, que modifican el contenido antes de ser procesados. Lo siguiente hace que la variable nombre esté en mayúscula antes de representarla:

```
{{ name|upper }}
```

Twig tiene una serie de filtros definidos, pero nosotros podremos crear los que deseemos.

# Templates: Herencia

La mayoría de las veces, las plantillas en un proyecto comparten elementos comunes, como el encabezado, el pie de página, la barra lateral o más.

En Symfony, este problema se piensa de manera diferente: una plantilla puede ser decorada por otra.

Esto funciona exactamente igual que las clases de PHP: la herencia de plantillas le permite crear una plantilla de "diseño" de base que contiene todos los elementos comunes de su sitio definidos como bloques.

Una plantilla secundaria puede extender el diseño base y anular cualquiera de sus bloques



# Templates: Herencia

```
{# templates/base.html.twig #}
```

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>{% block title %}Test Application{% endblock %}</title>
```

```
  </head>
```

```
  <body>
```

```
    {% block sidebar %}Sidebar por defecto{% endblock %}
```

```
    <div id="content">
```

```
      {% block body %}Mi body{% endblock %}
```

```
    </div>
```

```
  </body>
```

```
</html>
```

# Templates: Herencia

```
{# templates/blog/index.html.twig #}
```

```
{% extends 'base.html.twig' %}
```

```
{% block title %}Mi página {% endblock %}
```

```
{% block body %}
```

El contenido de mi página

```
{% endblock %}
```

¿Preguntas?

