

## **Curso Symfony 4 - Clase 2**

**Motor de plantillas:** Las plantillas, o más exactamente los motores de plantillas (templates engines) leen un fichero de texto, que contiene la presentación ya preparada en HTML (o meta información), e inserta en él la información dinámica que le ordena el controlador.

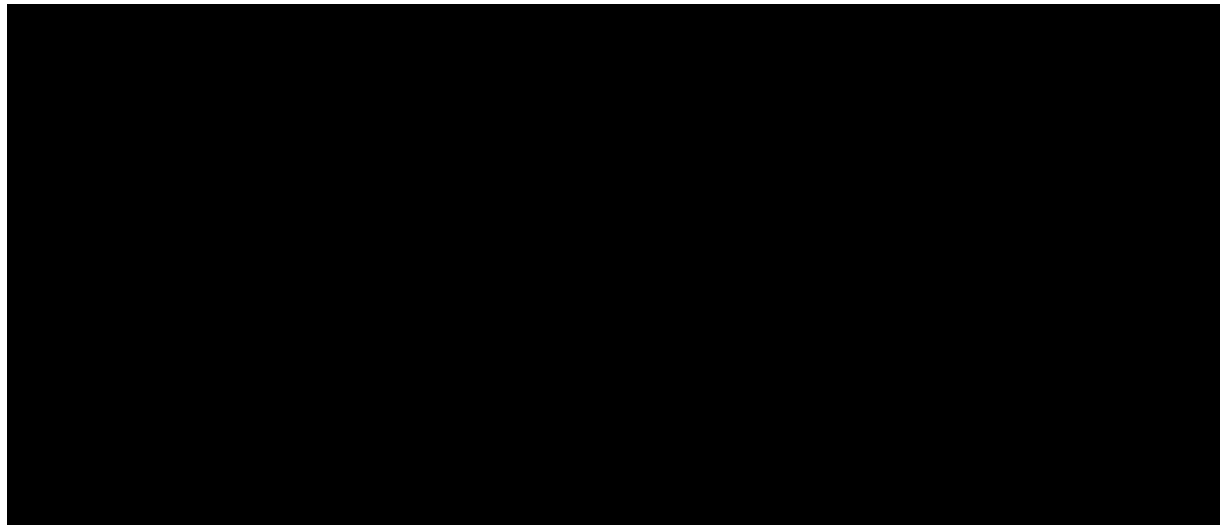
El motor de plantillas más usado en php y que utilizaremos es Twig.

Twig es un motor de plantillas desarrollado para el lenguaje de programación PHP y que nace con el objetivo de facilitar a los desarrolladores de aplicaciones web que utilizan la arquitectura MVC el trabajo con la parte de las vistas, gracias a que se trata de un sistema que resulta muy sencillo de aprender y capaz de generar plantillas con un código preciso y fácil de leer. Actualmente el código se distribuye bajo licencia BSD y es utilizado por el framework Symfony, aunque puede ser utilizado directamente con proyectos desarrollados en PHP en el que no interviene ese framework.

Twig tiene 2 etapas de desarrollo:

**Twig for Template Designers:** En esta etapa se desarrolla la plantilla o la estructura de la plantilla, la que va a contener las “directivas” o variables que van a ser reemplazadas dinámicamente por la información.

Un ejemplo de una plantilla en twig es la siguiente:



Si prestamos atención las diferencias que encontramos con una plantilla HTML tradicional son las siguiente líneas de sintaxis como:

```
{% for item in navigation %}  
{% endfor %}  
{{ a_variable }}
```

Estas son las expresiones propias de twig que le dan ese funcionamiento dinámico que introduce el concepto de “Motor de plantillas”, tal que, “a\_variable” es un valor dinámico, osea que puede variar, ser un número o un string o tal vez una foto. Esta información variable será inyectada en la plantilla más adelante.

### **Variables:**

La aplicación pasa variables a las plantillas para su manipulación en la plantilla. Las variables pueden tener atributos o elementos a los que también se puede acceder. La representación visual de una variable depende en gran medida de la aplicación que la proporciona.

Utilice un punto ( . ) para acceder a los atributos de una variable (métodos o propiedades de un objeto PHP o elementos de una matriz PHP):

```
{{ objeto.atributo }}
```

### **Estructuras de control:**

Una estructura de control se refiere a todas aquellas cosas que controlan el flujo de un programa - condicionales (es decir, if/ elseif/ else), bucles como el “for” entre otros. Las estructuras de control aparecen dentro de bloques.{% ... %}

Por ejemplo, para mostrar una lista de usuarios proporcionados en una variable llamada users, use la etiqueta for :

```
1 <h1>Members</h1>
2 <ul>
3   {% for user in users %}
4     <li>{{ user.username|e }}</li>
5   {% endfor %}
6 </ul>
```

**JavaScript en el proyecto:** JavaScript es un lenguaje de programación o de secuencias de comandos que te permite implementar funciones complejas en páginas web, cada vez que una página web hace algo más que sentarse allí y mostrar información estática para que la veas, muestra oportunas actualizaciones de contenido, mapas interactivos, animación de Gráficos 2D/3D, desplazamiento de máquinas reproductoras de vídeo, etc., puedes apostar que probablemente JavaScript está involucrado. Es la tercera capa del pastel de las tecnologías web estándar, dos de las cuales (HTML y CSS) ya son de conocimiento.

Javascript se suele utilizar también para realizar consultas asincrónicas a alguna API que provea información. Hoy por hoy es casi indispensable su utilización en los sistemas.

Symfony incluye una biblioteca de JavaScript puro, llamada Webpack Encore, que ayuda a que trabajar con CSS y JavaScript sea más sencillo. Se puede trabajar con esta librería o integrar los archivos JavaScript y Css de forma tradicional.

*Leer más información de:*

<https://symfony.com/doc/4.4/frontend.html#webpack-encore>

**¿Qué es un ORM?**

Object-Relational mapping, o lo que es lo mismo, mapeo de objeto-relacional, es un modelo de programación que consiste en la transformación de las tablas de una base de datos, en una serie de entidades que simplifiquen las tareas básicas de acceso a los datos del programador

- Ventajas

- Facilidad y velocidad de uso
- Abstracción de la base de datos usada.
- Seguridad de la capa de acceso a datos contra ataques.

- Desventajas

- En entornos con gran carga poner una capa más en el proceso puede mermar el rendimiento.
- Aprender el nuevo lenguaje del ORM.