



Universidad de
Oviedo



FRAMEWORKS WEB

Enol García González
Universidad de Oviedo
1 de diciembre de 2025

CONTENIDOS

1 Introducción

2 Javascript

3 Python

4 Java

5 Kotlin

6 C#

DOS TIPOS DE FRAMEWORKS

- **Orientados a controladores.** La pieza fundamental es el controlador. El controlador es un elemento que se encarga de definir como se responde ante cada solicitud. En este tipo de frameworks es común tener un motor de plantillas para independizar el código de la visualización.
- **Orientados a componentes.** Es el enfoque contrapuesto al de los controladores. En este tipo de frameworks nos olvidamos de las peticiones. Estos frameworks se caracterizan por centrarse en el diseño de partes de la web reutilizables, llamados componentes. Los componentes concentran la funcionalidad y visualización. Los componentes no tienen que ser un único fichero (pueden ser varios).

RESUMEN DE FRAMEWORKS

Framework	Tipo	Motor	Carácter de escape
JAVASCRIPT			
Express	Controladores	Swig	{ ... }
NextJS (React)	Componentes		{}{ ... }}
Vue.js	Componentes		{}{ ... }}
Angular	Componentes		{}{ ... }}
PYTHON			
Django	Controladores	Integrado	Sentencias de control: { % ... % } Expresiones: {}{ ... }}
Flask	Controladores	Jinja2	Sentencias de control: { % ... - % } Expresiones: {}{ ... }}

RESUMEN DE FRAMEWORKS

Framework	Tipo	Motor	Carácter de escape
JAVA			
Java Server Faces	Componentes		
Spring	Controladores	*	*
Quarkus	Controladores	*	*
Micronaut	Controladores	*	*
KOTLIN			
Micronaut	Controladores	*	*
Ktor	Controladores	*	*
C#			
ASP.NET	Controladores	Integrado	@...

JAVASCRIPT

- **Express.** Es el más básico y minimalista. Es de los pocos orientados a controladores en NodeJS
- **NextJS.** Es el framework más característico que utiliza React. Recuerdo: React define los componentes en forma de función. Los componentes se pueden utilizar en otro componente con la etiqueta que genera.
- **Vue.js** y **Angular**. Son análogos a React. Cambian la forma de definir los componentes, pero al final son igual. Generan una nueva etiqueta que se puede utilizar en otros componentes. De estos tenemos un ejemplo.

PYTHON

- **Django** y **Flask**. Son los dos frameworks característicos de Python.
- Los dos de controladores. Hay que definir una función que defina como responder a las peticiones.
- En Django se registran las nuevas peticiones en un fichero que configura las rutas.
- En Flask se hace como en Spring, con anotaciones.
- De estos dos frameworks también hay un ejemplo.

- **JSF** y **Spring** ya los conocemos. JSF es el representante de componentes y Spring es el representante de controladores.
- Otros dos ejemplo de controladores son **Quarkus** y **Micronaut**. Ambos funcionan igual que Spring –orientado a controladores–, y tratan de competir con él por ser más ligeros, pero no tienen tanto éxito.
- Spring, Quarkus y Micronaut pueden ser configurados con muchos motores de plantillas: Thymeleaf, Handlebars.java, Apache Velocity, Apache Freemaker, Pebble.

THYMELEAF

```
1 <table>
2   <thead>
3     <tr>
4       <th th:text="#{msgs.headers.name}">Name</th>
5       <th th:text="#{msgs.headers.price}">Price</th>
6     </tr>
7   </thead>
8   <tbody>
9     <tr th:each="prod: ${allProducts}">
10      <td th:text="${prod.name}">Oranges</td>
11      <td th:text="${#numbers.formatDecimal(prod.price, 1, 2)}">0.99</td>
12    </tr>
13  </tbody>
14 </table>
```

Figura: Fuente: Thymeleaf.org

HANDLEBARS.JAVA

```
{{#if busy}}
<h4>{{name}} is busy.</h4>
{{else}}
<h4>{{name}} is not busy.</h4>
{{/if}}
```

Figura: Fuente: Baeldung.com

APACHE VELOCITY

```
<table class="gridtable">
  <tr>
    <th>Serial #</th>
    <th>Product Name</th>
    <th>Price</th>
  </tr>
  #foreach( $product in $products )
  <tr>
    <td>$count)</td>
    <td>$product.getName()</td>
    <td>$product.getPrice()</td>
  </tr>
```

Figura: Fuente: Baeldung.com

APACHE FREEMARKER

```
<#list statuses>
    <ul>
        <#items as status>
            <li>${status}</li>
        </#items>
    </ul>
<#else>
    <p>No statuses available</p>
</#list>
```

Figura: Fuente: Baeldung.com

PEBBLE

```
{% for article in articles %}  
    <h3>{{ article.title }}</h3>  
    <p>{{ article.content }}</p>  
{% else %}  
    <p> There are no articles. </p>  
{% endfor %}
```

Figura: Fuente: pebbletemplates.io

KOTLIN

- Kotlin es un lenguaje que surgió en 2016 para aprovechar las ventajas de Java y de Javascript.
- Este lenguaje cubre la necesidad de un lenguaje de tipado estático y de un lenguaje reactivo.
- Para Kotlin encontramos dos frameworks: **Ktor** y **Micronaut**. Ambos pueden utilizar los mismos motores de plantillas que los frameworks de Java.

- Para C# el framework de referencia es **ASP.NET** incluido en el pack de desarrollo .NET de Microsoft.
- Es un framework de controladores que trae integrado el motor de plantillas.