

OPERACIONES
ARITMÉTICAS,
CONDICIONES Y
COMPARACIONES





1.

OPERADORES Y OPERACIONES


OPERACIONES ARITMÉTICAS

- ▶ Con OGNL (o SpEL) tenemos los operadores aritméticos habituales
- ▶ $+$, $-$, $*$, $/$, $\%$, $^$
- ▶ $\${4 + 0.2}$, $\${3 - 0.12}$, $\${4 * 3}$, $\${4 / 3}$,
 $\${5 \% 3}$, $\${4 / 3.0}$, $\${4^3}$, $\${-(-2)}$

OPERACIONES TEXTUALES

- ▶ También tenemos alguno de los operadores textuales más habitual
- ▶ + para la concatenación
- ▶ | ... | para la sustitución de literales
- ▶ `${'Hola ' + 'mundo'}`,

|Mi nombre es `${nombre}`|



¡Ojo! Esta expresión rompe un poco con el esquema habitual visto hasta ahora.

OPERADORES RELACIONALES

- ▶ También tenemos los habituales
- ▶ $<$, $<=$, $>$, $>=$, $==$, $!=$
- ▶ Versión textual: lt, le, gt, ge, eq, ne

OPERADORES **BOOLEANOS**

- ▶ También tenemos los habituales
- ▶ and, or, not, !



2.

CONDICIONES Y COMPARACIONES

CONDICIONES Y COMPARACIONES

- ▶ Usando los operadores anteriores podemos comparar valores y usarlos en expresiones condicionales.
- ▶ Una de las formas más habituales es mediante los atributos **th:if** y **th:unless**.

```
<div th:if="${not #lists.isEmpty(lista)}">
    <p th:text="|Nombre del producto: ${lista[0].nombre}|">nombre</p>
</div>
<div th:unless="${not #lists.isEmpty(lista)}"> ← funciona como un
    <p>No hay productos disponibles en la lista</p>else
</div>
```


EXPRESIONES **CONDICIONALES**

- ▶ En ocasiones, podemos resumir el uso de *th:if* y *th:unless* en una sola expresión.
- ▶ El operador ternario `? :` es la solución
- ▶ `${condición} ? valor verdadero : valor falso`

```
<p th:text="${not #lists.isEmpty(lista)} ?  
  |Nombre del producto: ${lista[0].nombre}| :  
  'No hay productos disponibles en la lista'">nombre</p>
```

OPERADOR ELVIS

- ▶ El operador elvis (?:), utilizado en las llamadas expresiones por defecto, es como el anterior, pero sin un valor si la condición es verdadera.

```
<p th:text="*{descripcion}?: 'Si quiere más información sobre nuestro  
productos, no dude en contactar con nosotros'">texto</p>
```

TOKEN NO-OP

- ▶ Representado por el guión bajo (_)
- ▶ La idea detrás de este token es que, tras evaluar una expresión no se haga nada.
- ▶ Si lo usamos con *th:text*, utilizará el texto de *prototipado* como valor por defecto.

```
<p th:text="*{descripcion}?: _">
```

Si quiere más información sobre nuestro productos, no dude en
contactar con nosotros

```
</p>
```



3.

ELEMENTO
TH:BLOCK

ELEMENTO **TH-BLOCK**

- ▶ *th:block* es un contenedor de atributos Thymeleaf.
- ▶ Estos atributos serán procesados, y posteriormente desaparecerá el bloque, pero no su contenido.
- ▶ Útil en varios contextos, como aplicación de condiciones a un grupo de elementos

ELEMENTO TH-BLOCK

```
<table>
  <th:block th:each="user : ${users}">
    <tr>
      <td th:text="${user.login}">...</td>
      <td th:text="${user.name}">...</td>
    </tr>
    <tr>
      <td colspan="2"
th:text="${user.address}">...</td>
    </tr>
  </th:block>
</table>
```

Nos permite crear dos filas por usuario sin usar una etiqueta HTML adicional, como *div*, que no sería correcta en este contexto.