

**id + class**

# El problema

Al usar nombres de etiquetas HTML, se afecta a todas aquellas etiquetas que existan en el documento.

```
div {  
  background-color: orange;  
}
```

En este ejemplo, todos los div del sitio web tendrán un color de fondo naranja.

# La solución

Se puede utilizar la misma etiqueta HTML múltiples veces y solo afectar la apariencia de aquellos que queremos, usando atributos como id o class.

# Utilidad

Los id y class permiten definir elementos usando nombres que podemos crear nosotros mismos.

```
<body id="page-services">
```

```
<div class="panel-message">
```

# Ejemplo

Si utilizamos un ul para la navegación de nuestro sitio web, podríamos escribir:

```
<ul id="nav-main">
```

En lugar de escribir propiedades para la etiqueta `ul`, las escribiríamos para `nav-main`.

De este modo, no se afectarán todos los `ul` del sitio, sino solo el de este id.

# Diferencias entre id y class

¿Por qué hay dos?

# Los id son únicos

Es decir...

- Cada elemento puede tener solo un id.
- Cada documento puede tener solo un elemento con ese id.

# Los class no son únicos

Es decir...

- Puedes usar el mismo class en múltiples elementos.
- Puedes usar múltiples class en un mismo elemento.



# ¿Y usar ambos?

Sí vale!

```
<li id="comment-27299" class="comment-item">
```

# id vs. class

No hay diferencias entre id y class en cuanto a propiedades o funciones de CSS. Todo lo que puede tener un id, lo puede tener un class.

En cuanto a CSS, no hay diferencia.

# Javascript

En Javascript sí hay diferencia.

Las acciones de js se aplican generalmente a un elemento específico en el documento.

Para poder afectar correctamente a un único elemento, js se basa en el id de este.

# No sobreusar

En lugar de:

```
<ul>  
  <li class="item-ingrediente"> ... </li>  
  <li class="item-ingrediente"> ... </li>  
</ul>
```

Prueba:

```
<ul class="list-ingredientes">  
  <li> ... </li>  
  <li> ... </li>  
</ul>
```

# No sobreusar

Es recomendable, en la medida de lo posible, tratar de usar selectores CSS en lugar de modificar el HTML.

Para esto se pueden aprovechar los selectores de tipo, de parent, adyacentes, de atributo, etc.

# Especificidad

# Espe... qué?

La especificidad se refiere a qué tan específicos o no son los selectores que usamos en CSS.

# ¿Cómo así?

Hemos visto que se puede dar estilo a un elemento mencionando el nombre del elemento mismo, a través de una agrupación, de una anidación, usando un class, un id, o incluso con el selector universal.

CSS posee niveles de especificidad a fin de mantener jerarquías.



# Orden de especificidad

Este es un orden ascendente del nivel de especificidad:

Universal selectors

Type selectors

Class selectors

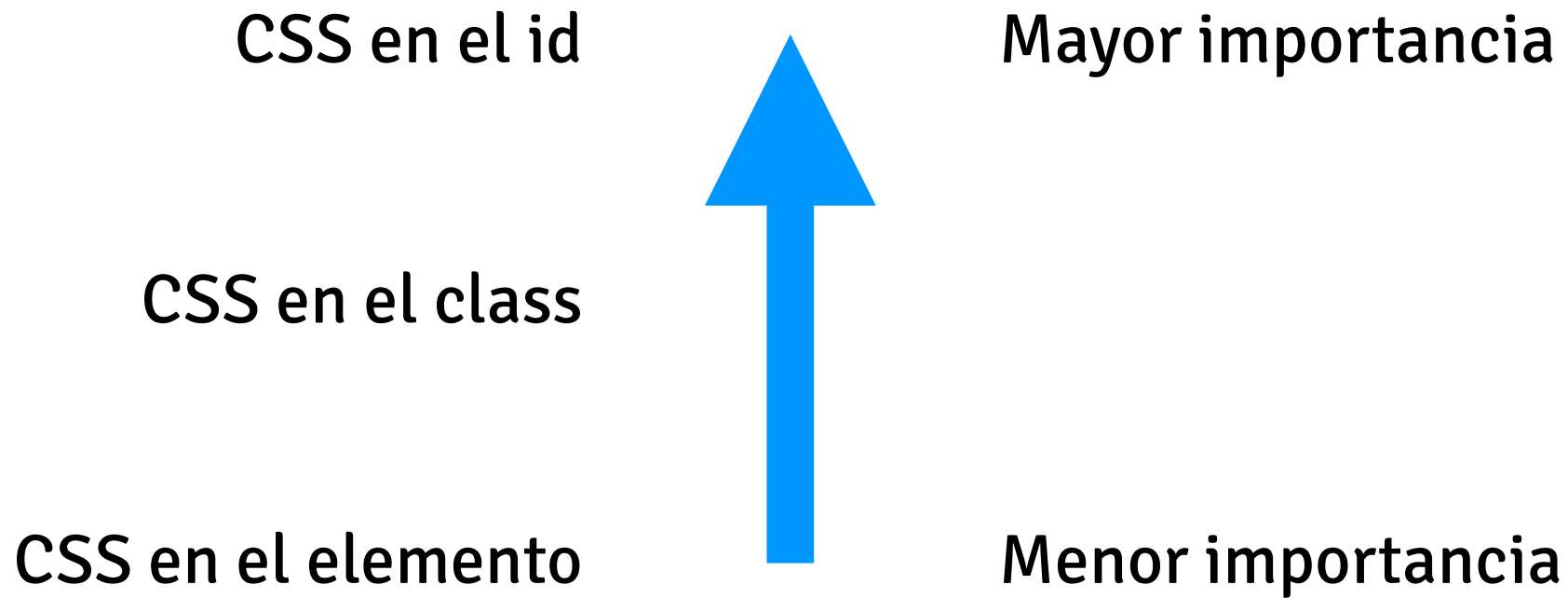
Attributes selectors

Pseudo-classes

ID selectors

Inline style

# Especificidad

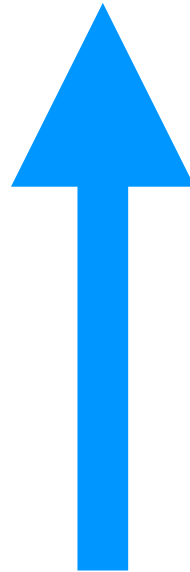


# Especificidad

CSS en el id

CSS en el class

CSS en el elemento



```
#panel-destacado {  
  background-color: blue;  
}
```

```
.panel-info {  
  background-color: green;  
}
```

```
div {  
  background-color: red;  
}
```

# Ejemplo de id y class

```
<ul id="list-ingredientes">  
  <li>250g de harina</li>  
  <li>10 gramos de levadura</li>  
  <li>185 cl de agua</li>  
  <li class="opcional">3 cucharadas de aceite</li>  
  <li class="opcional">5g de sal</li>  
</ul>
```

```
#list-ingredientes li {  
  border-left: 2px solid red;  
  color: #000;  
  margin: 4px 0;  
}
```

```
#list-ingredientes .opcional {  
  border-left: 2px solid #eee;  
  color: #444;  
  padding-left: 6px;  
}
```