**REPORTE TÉCNICO**

**Exploración de Técnicas CSS, Prevalencias, Frameworks y Animaciones.**

**ALUMNO**

**Daniel Barreto López**

**MODULO**

**Sistemas de información Web**

**CONTENIDO**

[**Introducción** 3](#_Toc193028923)

[**Desarrollo** 3](#_Toc193028924)

[**1. Técnicas para combinar selectores de CSS y uso de herencias** 3](#_Toc193028925)

[1.1. Combinación de selectores: 3](#_Toc193028926)

[1.2. Herencia en CSS: 5](#_Toc193028927)

[**2. Prevalencias entre selectores CSS** 5](#_Toc193028928)

[2.1. Especificidad: 5](#_Toc193028929)

[2.2. Orden de aparición: 6](#_Toc193028930)

[**3. Frameworks de CSS** 6](#_Toc193028931)

[**4. Ejemplos de animaciones con CSS** 6](#_Toc193028932)

[**Conclusión** 7](#_Toc193028933)

[**Referencias** 7](#_Toc193028934)

# **Introducción**

Este reporte técnico documenta las exploraciones realizadas sobre técnicas avanzadas de CSS, incluyendo la combinación de selectores, la herencia, la prevalencia entre selectores, frameworks de CSS populares y la implementación de animaciones utilizando únicamente CSS. El objetivo es proporcionar una comprensión clara de estos conceptos y su aplicación en el desarrollo web moderno.

# **Desarrollo**

## **1. Técnicas para combinar selectores de CSS y uso de herencias**

### 1.1. Combinación de selectores:

* **Selectores de tipo:** Permiten seleccionar elementos por su etiqueta HTML (p, div).

Ejemplo:

|  |
| --- |
| /\* Aplica estilos a todos los elementos <p> \*/  p {  color: blue;  font-size: 16px;  } |

* **Selectores de clase:** Seleccionan elementos basados en su clase (clase).

Ejemplo:

|  |
| --- |
| css |
| /\* Aplica estilos a todos los elementos con la clase "destacado" \*/  .destacado {  background-color: yellow;  font-weight: bold;  } |

|  |
| --- |
| html |
| <p class="destacado">Este texto está destacado.</p> |

* **Selectores de ID:** Apuntan a elementos específicos usando su atributo id (id).

Ejemplo:

|  |
| --- |
| css |
| /\* Aplica estilos al elemento con el ID "titulo-principal" \*/  #titulo-principal {  color: red;  font-size: 24px;  } |

|  |
| --- |
| html |
| <h1 id="titulo-principal">Título Principal</h1> |

* **Selectores de atributo:** Permiten seleccionar elementos con atributos específicos ([type="text"]).

Ejemplo:

|  |
| --- |
| css |
| /\* Aplica estilos a todos los elementos <input> con el atributo type="text" \*/  input[type="text"] {  border: 2px solid green;  padding: 5px;  } |

|  |
| --- |
| html |
| <input type="text" placeholder="Escribe algo..."> |

* **Selectores descendientes:** Combinan selectores para apuntar a elementos dentro de otros (div p).

Ejemplo:

|  |
| --- |
| css |
| /\* Aplica estilos a todos los elementos <p> dentro de un <div> \*/  div p {  color: purple;  } |

|  |
| --- |
| html |
| <div>  <p>Este párrafo está dentro de un div.</p>  </div>  <p>Este párrafo no está dentro de un div.</p> |

* **Selectores de hijos directos:** Seleccionan solo los hijos directos de un elemento (div > p).

Ejemplo:

|  |
| --- |
| css |
| /\* Aplica estilos a los elementos <li> que son hijos directos de un <ul> \*/  ul > li {  list-style-type: square;  } |

|  |
| --- |
| html |
| <ul>  <li>Elemento 1</li>  <li>Elemento 2</li>  <ol>  <li>Este no es un hijo directo de ul.</li>  </ol>  </ul> |

* **Selectores adyacentes:** Apuntan a elementos que están inmediatamente después de otro (h1 + p).

Ejemplo:

|  |
| --- |
| css |
| /\* Aplica estilos al primer <p> que está inmediatamente después de un <h2> \*/  h2 + p {  color: orange;  } |

|  |
| --- |
| html |
| html  Copy  <h2>Título</h2>  <p>Este párrafo está justo después de un h2.</p>  <p>Este párrafo no está justo después de un h2.</p> |

* **Selectores generales de hermanos:** Seleccionan elementos hermanos que comparten el mismo padre (h1 ~ p).

Ejemplo:

|  |
| --- |
| css |
| /\* Aplica estilos a todos los <p> que son hermanos de un <h2> \*/  h2 ~ p {  font-style: italic;  } |

|  |
| --- |
| html |
| <h2>Título</h2>  <p>Este párrafo es hermano de h2.</p>  <p>Este también es hermano de h2.</p>  <div>Este no es un párrafo.</div>  <p>Este también es hermano de h2.</p> |

### 1.2. Herencia en CSS:

* Algunas propiedades CSS, como color, **font-family** y **font-size,** se heredan automáticamente de los elementos padres a los hijos.
* Otras propiedades, como **margin,** **padding** y **border,** no se heredan.
* La herencia puede forzarse utilizando el valor **inherit**(por ejemplo, **color: inherit;**).

## **2. Prevalencias entre selectores CSS**

La prevalencia en CSS se determina mediante dos factores principales: especificidad y orden de aparición.

### 2.1. Especificidad:

* Es un valor que determina qué regla CSS se aplica cuando hay conflictos. Se calcula de la siguiente manera:
* **Inline styles:** 1,0,0,0
* **IDs:** 0,1,0,0
* **Clases, atributos y pseudo-clases:** 0,0,1,0
* **Elementos y pseudo-elementos:** 0,0,0,1
* **Selectores universales y combinadores:** 0,0,0,0

### 2.2. Orden de aparición:

* Si dos reglas tienen la misma especificidad, la última regla definida en el archivo CSS prevalece.

## **3. Frameworks de CSS**

Los frameworks de CSS son herramientas que facilitan el desarrollo web al proporcionar estilos predefinidos, componentes reutilizables y sistemas de grid. Algunos de los frameworks más populares incluyen:

* **Bootstrap:** Ofrece un sistema de grid, componentes predefinidos y utilidades para crear diseños responsivos rápidamente.
* **Tailwind CSS:** Un framework de utilidades que permite construir diseños directamente en el HTML usando clases de utilidad.
* **Foundation:** Un framework responsive que incluye componentes y herramientas para crear sitios web y aplicaciones.
* **Bulma:** Basado en Flexbox, es ligero y fácil de usar.
* **Materialize:** Basado en Material Design de Google, ofrece componentes y estilos predefinidos.

## **4. Ejemplos de animaciones con CSS**

A continuación, se presentan tres ejemplos de animaciones creadas únicamente con CSS:

|  |
| --- |
| **Ejemplo 1:** Animación de cambio de color |

@keyframes changeColor {

0% { background-color: red; }

50% { background-color: yellow; }

100% { background-color: green; }

}

div {

width: 100px;

height: 100px;

animation: changeColor 3s infinite;

}

|  |
| --- |
| **Ejemplo 2:** Animación de rotación |

@keyframes rotate {

0% { transform: rotate(0deg); }

100% { transform: rotate(360deg); }

}

div {

width: 100px;

height: 100px;

background-color: blue;

animation: rotate 2s linear infinite;

}

|  |
| --- |
| **Ejemplo 3:** Animación de movimiento |

@keyframes move {

0% { transform: translateX(0); }

50% { transform: translateX(200px); }

100% { transform: translateX(0); }

}

div {

width: 100px;

height: 100px;

background-color: orange;

animation: move 4s ease-in-out infinite;

}

# **Conclusión**

Las técnicas avanzadas de CSS, como la combinación de selectores, la herencia y las animaciones, permiten crear diseños web dinámicos y eficientes. Además, los frameworks de CSS simplifican el desarrollo al proporcionar herramientas y componentes predefinidos. Comprender la especificidad y la prevalencia entre selectores es crucial para evitar conflictos y garantizar que los estilos se apliquen correctamente.

# **Referencias**

1. MDN Web Docs. (2023). "CSS Selectors". Disponible en: <https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Selectors>
2. MDN Web Docs. (2023). "CSS Inheritance". Disponible en: <https://developer.mozilla.org/en-US/docs/Web/CSS/inheritance>
3. MDN Web Docs. (2023). "CSS Specificity". Disponible en: <https://developer.mozilla.org/en-US/docs/Web/CSS/Specificity>
4. Bootstrap. (2023). "Bootstrap Documentation". Disponible en: <https://getbootstrap.com/docs/>
5. Tailwind CSS. (2023). "Tailwind CSS Documentation". Disponible en: <https://tailwindcss.com/docs>
6. MDN Web Docs. (2023). "CSS Animations". Disponible en: <https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Animations>