# UNIVERSIDAD NACIONAL DE ASUNCIÓN FACULTAD POLITÉCNICA

# MAESTRÍA EN INGENIERÍA EN ELECTRÓNICA CON ÉNFASIS EN TECNOLOGÍA DE LA INFORMACIÓN

Módulo: SISTEMA DE INFORMACIÓN WEB

**TRABAJO 02: Explorando CSS** 

#### Alumna:

• Torres Ibarrola, Mónica Beatriz

#### **Docentes:**

- Julio Cesar Mello Roman
- Marcos Benítez

Marzo - 2025

**1. Introducción** Este documento presenta un reporte técnico de las exploraciones realizadas en el marco de la tarea asignada. Se detallan los objetivos, metodología empleada, etc. Además, se incluyen las referencias utilizadas y se adjuntan los ejercicios desarrollados en un archivo de formato zip.

# 2. Objetivos

- Documentar el proceso de exploración.
- Analizar los datos obtenidos.
- Presentar hallazgos relevantes.
- Generar conclusiones basadas en los resultados obtenidos.

# 3. Metodología

Se llevaron a cabo exploraciones mediante diversas técnicas, incluyendo:

- Investigación sobre técnicas de combinación de selectores de CSS y la utilización de herencia.
- 2. Análisis de la prevalencia entre selectores CSS.
- 3. Exploración de frameworks de CSS.
- 4. Desarrollo de ejemplos utilizando animaciones CSS sin uso de JavaScript.

#### 4. Desarrollo de Actividades

# 4.1 Investigación sobre combinación de selectores de CSS y la utilización de herencia

Los selectores en CSS permiten aplicar estilos a elementos específicos de una página web, son la clave para apuntar a elementos específicos dentro del documento HTML y aplicarles estilos. La combinación de selectores permite un control más preciso del estilo. Algunas técnicas incluyen:

• <u>Selectores descendientes:</u> Aplican estilos a elementos dentro de otros elementos, sin importar su nivel de anidamiento.

### Ejemplo:

```
div p {
  color: blue;
}
```

En este caso, todos los párrafos () que se encuentren en cualquier nivel dentro de un <div>> recibirán el estilo de color azul.

 <u>Selectores hijos</u>: Aplican estilos solo a los elementos hijos directos del elemento contenedor.

#### Ejemplo:

```
ul > li {
  list-style-type: none;
}
```

Aquí, solo los elementos que sean hijos inmediatos del serán afectados, excluyendo aquellos que estén más profundamente anidados.

• Selectores de hermanos adyacentes y generales: Aplican estilos a elementos que comparten el mismo padre.

### **Hermanos Adyacentes (+):**

Selecciona el elemento que sigue inmediatamente después de otro en el mismo nivel del DOM.

### Ejemplo:

```
h1 + p {
  margin-top: 0;
}
```

Este estilo se aplicará al párrafo que sigue de manera inmediata a un <h1>.

### **Hermanos Generales (~):**

Selecciona todos los elementos que comparten el mismo padre y aparecen después del elemento especificado, sin requerir que sean adyacentes.

#### Ejemplo:

```
h1 ~ p {
  color: gray;
}
```

En este ejemplo, todos los párrafos que aparecen después de un <h1> dentro del mismo contenedor recibirán el color gris.

 Herencia: La herencia en CSS es el mecanismo por el cual ciertos estilos definidos en un elemento padre se transmiten automáticamente a sus elementos hijos. Esto permite que se definan estilos de manera global y se mantenga la coherencia visual sin necesidad de repetir reglas para cada elemento.

#### **Propiedades que se Heredan Automáticamente:**

Propiedades relacionadas con la tipografía y el color, como color, font-

family, y line-height, se heredan de forma predeterminada. Esto significa que si defines el color del texto en el elemento <body>, la mayoría de los elementos secundarios adoptarán ese color a menos que se especifique lo contrario.

#### **Propiedades No Heredadas:**

Algunas propiedades, como margin, padding y border, no se heredan, lo que permite tener un mayor control sobre el espaciado y el diseño de cada componente de manera individual.

# Ventajas de la Herencia:

- Mantenibilidad: Permite definir estilos en un contenedor y que estos se propaguen a todos los elementos hijos, reduciendo la repetición de código.
- Consistencia: Ayuda a mantener un diseño uniforme en toda la página web, asegurando que los elementos compartan las mismas características básicas a menos que se especifique una excepción.

#### 4.2 Explicación de la prevalencia entre selectores CSS

La prevalencia entre selectores CSS se determina mediante un concepto llamado especificidad, que es una medida de la "fuerza" de una regla CSS. Cuando dos o más reglas apuntan al mismo elemento y definen estilos para la misma propiedad, la regla con mayor especificidad será la que se aplique. La prioridad se determina con la siguiente jerarquía (4 niveles):

#### 1. Estilos en línea:

Las reglas definidas directamente en el atributo style del elemento tienen la mayor prioridad.

### Ejemplo:

Texto

#### 2. Selectores de ID:

Cada vez que se utiliza un selector de ID (por ejemplo, #header), se asigna una puntuación alta.

#### Ejemplo:

```
#menu {color: blue;}
```

# 3. Selectores de clases, atributos y pseudo-clases:

Los selectores de clase (por ejemplo, active), selectores de atributos (por ejemplo [type="text"]) y pseudo-clases (por ejemplo, hover) tienen una puntuación intermedia.

# Ejemplo:

```
highlight { background-color: yellow; }
```

# 4. Selectores de elementos y pseudo-elementos:

Los selectores que apuntan directamente a un elemento (por ejemplo, p, h1) y pseudo-elementos (por ejemplo, before) tienen la puntuación más baja.

# Ejemplo:

```
p {font-size: 16px;}
```

#### Ejemplo Práctico de Especificidad:

```
p {
  color: blue;
}
.highlight {
  color: green;
}
#intro {
  color: red;
}
```

#### Cálculo de Especificidad:

Para p:

Especificidad = 0,0,0,1 (un selector de elemento)

Para .highlight:

Especificidad = 0,0,1,0 (una clase)

Para #intro:

Especificidad = 0,1,0,0 (un ID)

En este caso, la regla #intro tiene la mayor especificidad, por lo tanto, el color del párrafo será *rojo*.

### 4.3 Investigar sobre framework de CSS

Un framework de CSS es una biblioteca de estilos preconstruidos que ofrece soluciones comunes para el diseño web, tales como sistemas de rejilla (grids), componentes UI (botones, formularios, menús) y utilidades para manejar espacios, colores y tipografías. Su principal objetivo es agilizar el desarrollo al evitar la creación de estilos desde cero y proporcionar una base sólida y probada para la construcción de sitios web.

# Ventajas del Uso de Frameworks de CSS

#### Desarrollo Rápido:

Permiten implementar diseños responsivos y modernos sin tener que programar cada estilo individualmente, reduciendo el tiempo de desarrollo.

# Consistencia y Estándares:

Al usar un conjunto de estilos y componentes uniformes, se asegura una apariencia coherente en todas las páginas del proyecto.

#### Compatibilidad y Pruebas:

Los frameworks populares están ampliamente probados en múltiples navegadores y dispositivos, lo que minimiza problemas de compatibilidad.

### Documentación y Comunidad:

Los principales frameworks cuentan con documentación extensa y comunidades activas, lo que facilita el aprendizaje y la resolución de problemas.

#### **Principales Frameworks de CSS**

#### 1. Bootstrap

#### Descripción:

Uno de los frameworks más populares y ampliamente adoptados. Ofrece un sistema de rejilla flexible, numerosos componentes predefinidos (como botones, modales, formularios) y plugins JavaScript para interactividad.

#### Ventajas:

Gran comunidad, abundante documentación y fácil integración.

#### 2. Tailwind CSS

## Descripción:

Un framework basado en utilidades que permite una personalización muy granular. En lugar de ofrecer componentes predefinidos, Tailwind proporciona clases que se pueden combinar para crear diseños únicos sin tener que sobrescribir estilos.

#### Ventajas:

Flexibilidad y un enfoque altamente personalizable que evita el estilo "predefinido" característico de otros frameworks.

#### 3. Foundation

#### Descripción:

Similar a Bootstrap, Foundation es un framework robusto que se centra en el modularidad y la flexibilidad. Ofrece un sistema de rejilla, componentes UI y una arquitectura adaptable para crear proyectos escalables.

# ♣ Ventajas:

Enfoque en la accesibilidad y personalización, ideal para proyectos complejos.

#### 4. Bulma

### Descripción:

Basado en Flexbox, Bulma es un framework moderno y ligero que facilita la creación de interfaces responsivas sin depender de JavaScript. Su sintaxis es clara y está orientada a un diseño limpio y modular.

### Ventajas:

Fácil de aprender, diseño moderno y enfoque en la simplicidad.

En el caso de elegirlos se deberá tener en cuenta ciertos aspectos:

- ♣ Si necesitas algo rápido con componentes listos 
  → Bootstrap
- ♣ Si prefieres un diseño ultra personalizable 
  → Tailwind CSS
- **♣** Para un framework accesible y escalable → Foundation
- ♣ Para un diseño limpio y ligero → Bulma

# Referencias:

- ♣ MDN Web Docs. "Selectors in CSS". Disponible en: [https://developer.mozilla.org/en-US/docs/Web/CSS/CSS\_Selectors]
- ♣ W3Schools. "CSS Specificity". Disponible en: [https://www.w3schools.com/css/css\_specificity.asp]
- Tailwind CSS Rapidly build modern websites without ever leaving your HTML.
- **♣** Bootstrap · The most popular HTML, CSS, and JS library in the world.
- ♣ Bulma. "Bulma: The Modern CSS Framework." Bulma.io, 2024, <a href="https://bulma.io.">https://bulma.io.</a>