

Investigación y explicación

Dilan Alexander Robayo Tavera

Angelica María Triana Guarnizo

Centro de Diseño y Metrología

Ficha: 2848938

Bogotá D.C

2024

Figura 2.....	<b><i>¡Error! Marcador no definido.</i></b>
Figura 3.....	<b><i>¡Error! Marcador no definido.</i></b>
Figura 4.....	<b><i>¡Error! Marcador no definido.</i></b>
Figura 5.....	<b><i>¡Error! Marcador no definido.</i></b>
Figura 6.....	<b><i>¡Error! Marcador no definido.</i></b>
Figura 7.....	<b><i>¡Error! Marcador no definido.</i></b>
Figura 8.....	<b><i>¡Error! Marcador no definido.</i></b>
Figura 9.....	<b><i>¡Error! Marcador no definido.</i></b>
Figura 10.....	<b><i>¡Error! Marcador no definido.</i></b>
Figura 11.....	<b><i>¡Error! Marcador no definido.</i></b>
Figura 12.....	<b><i>¡Error! Marcador no definido.</i></b>
Figura 13.....	<b><i>¡Error! Marcador no definido.</i></b>
Figura 14.....	<b><i>¡Error! Marcador no definido.</i></b>
Figura 15.....	<b><i>¡Error! Marcador no definido.</i></b>

1.2 Investigue y explique a continuación que significa cada línea del código en

el siguiente cuadro:

Línea	código	Descripción
1	<code>&lt;!DOCTYPE html&gt;</code>	Esta línea indica el tipo de documento que se está utilizando.  <code>&lt;!DOCTYPE html&gt;</code> declara que este documento es un documento HTML5.
2	<code>&lt;html&gt;</code>	Esta etiqueta abre el elemento html, que es el contenedor principal de todo el contenido HTML de la página.
3	<code>&lt;head&gt;</code>	Esta etiqueta abre el elemento head, que contiene metadatos sobre el documento, como su título, enlaces a hojas de estilo, y scripts.
4	<code>&lt;title&gt;&lt;/title&gt;</code>	Dentro del head, esta línea define el título de la página que aparecerá en la pestaña del navegador.
5	<code>&lt;/head&gt;</code>	Esta etiqueta cierra el elemento head.
6	<code>&lt;body&gt;</code>	Esta etiqueta abre el elemento body, que contiene el contenido visible de la página web, como texto, imágenes, formularios, etc.

7	<code>&lt;form&gt;</code>	Esta etiqueta abre un elemento form, que es un contenedor para elementos de formulario que permiten a los usuarios enviar datos a un servidor.
8	<code>&lt;label&gt;&lt;/label&gt;</code>	La etiqueta label crea una etiqueta para un campo de entrada. Aquí, "Usuario:" es el texto de la etiqueta. La etiqueta <code>&lt;br&gt;</code> se utiliza para insertar un salto de línea, haciendo que el siguiente elemento se muestre en una nueva línea.
9	<code>&lt;input&gt;</code>	La etiqueta input crea un campo de entrada donde los usuarios pueden escribir texto. <code>type="text"</code> especifica que el campo es para entrada de texto. El <code>&lt;br&gt;</code> inserta otro salto de línea.
10	<code>&lt;label&gt;&lt;/label&gt;</code>	De nuevo, la etiqueta label crea una etiqueta, esta vez para el campo de contraseña. "Contraseña:" es el texto de la etiqueta. El <code>&lt;br&gt;</code> añade un salto de línea.
11	<code>&lt;input&gt;</code>	La etiqueta input crea otro campo de entrada, pero esta vez <code>type="password"</code> especifica que el campo es para entrada de

		contraseñas, lo que hará que los caracteres se oculten mientras se escriben.
<b>12</b>	<code>&lt;/form&gt;</code>	Esta etiqueta cierra el elemento form.
<b>13</b>	<code>&lt;/body&gt;</code>	Esta etiqueta cierra el elemento body.
<b>14</b>	<code>&lt;/html&gt;</code>	Esta etiqueta cierra el elemento html, finalizando el documento HTML.

**2.2** Realice captura de pantalla de la página y edite la imagen para señalar donde se encuentran las estructuras de HTML5 de la página. Por ejemplo, indique cual es el footer.

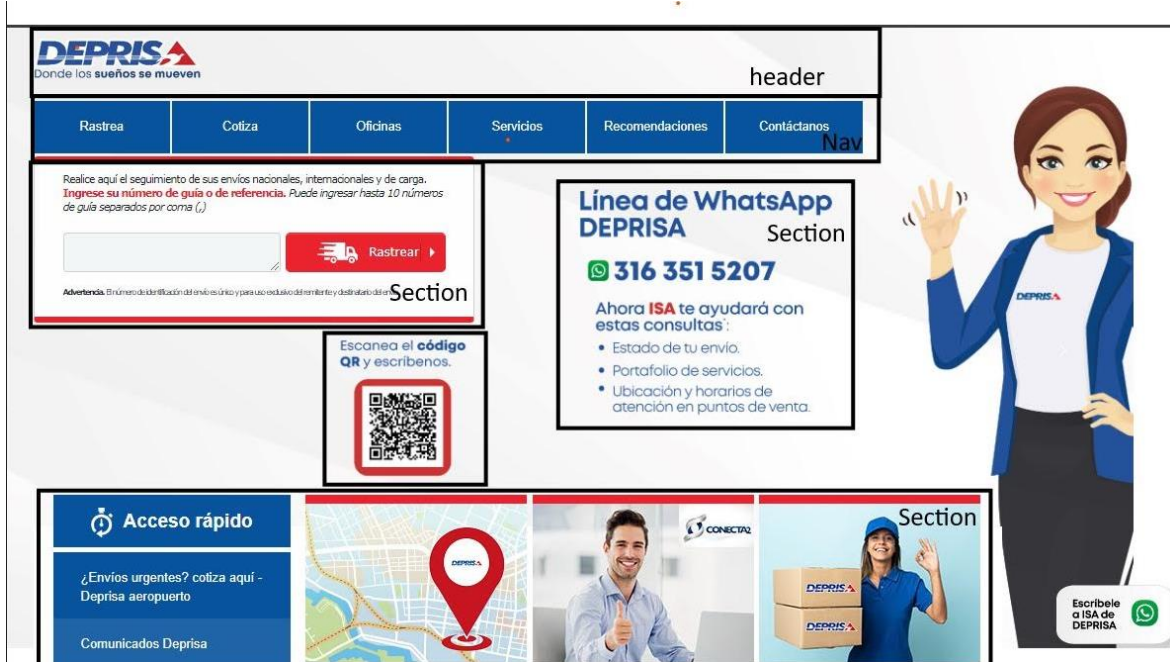


Figura 1

3. ¿Qué es un selector en CSS y para qué sirve? De ser necesario puede contextualizar con un ejemplo.

Un selector en CSS es un patrón utilizado para seleccionar y aplicar estilos a elementos HTML específicos. Los selectores permiten a los desarrolladores aplicar estilos de manera precisa y específica a los elementos de una página web, mejorando así la presentación y el diseño del contenido.

## Tipos de selectores en CSS

1. **Selector de Elemento:** Selecciona todos los elementos de un tipo específico .

A screenshot of a code editor with a dark theme. The editor has a tab labeled 'css' on the left and a button labeled 'Copiar código' on the right. The code inside the editor is a CSS rule for the 'p' element, setting the color to blue. The code is: 

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

```
css Copiar código  
  
p {  
  color: blue;  
}
```

*Figura 2*

Este selector aplica el color azul a todos los párrafos (<p>) en el documento.

2. **Selector de Clase:** Selecciona elementos que tienen una clase específica.

```
css Copiar código

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

*Figura 3*

Este selector aplica un fondo amarillo a todos los elementos con la clase highlight.

3. **Selector de ID:** Selecciona un elemento único con un ID específico.

```
css Copiar código

#header {
  font-size: 24px;
}
```

*Figura 4*

Este selector aplica un tamaño de fuente de 24 píxeles al elemento con el ID header.

4. **Selector Universal:** Selecciona todos los elementos en el documento.



```
CSS Copiar código

* {
  margin: 0;
  padding: 0;
}
```

*Figura 5*

Este selector elimina los márgenes y el relleno de todos los elementos.

5. **Selectores de Atributo:** Selecciona elementos en función de un atributo o valor de atributo.

```
CSS Copiar código

input[type="text"] {
  border: 1px solid #000;
}
```

*Figura 6*

Este selector aplica un borde negro a todos los campos de entrada de tipo texto (<input type="text">).

### **Ejemplo contextualizado**

## Título Principal

Este es un párrafo introductorio.

Este es otro párrafo introductorio destacado.

Este es un párrafo regular.

Este es un párrafo dentro de un contenedor.

## Explicación

**Selector de Elemento:** Este selector aplica color negro y un tamaño de fuente de 16 píxeles a todos los párrafos (<p>).

```
css Copiar código

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

**Selector de Clase:** Este selector aplica color verde a todos los elementos con la clase intro.

```
css Copiar código

.intro {
  color: green;
}
```

**Selector de Clase combinada:** Este selector aplica negrita a todos los elementos con la clase highlight.

```
css Copiar código

.highlight {
  font-weight: bold;
}
```

**Selector de ID:** Este selector aplica un tamaño de fuente de 32 píxeles y color azul al elemento con el ID main-title.

```
css Copiar código

#main-title {
  font-size: 32px;
  color: blue;
}
```

**Selector de Clase dentro de un contenedor:** Este selector aplica color rojo a los párrafos (<p>) que están dentro de un elemento con la clase container.

```
css Copiar código

.container p {
  color: red;
}
```

4.6 Describa en que consiste cada uno de los siguientes atributos de la etiqueta link:

```
<link rel="stylesheet" type="text/css" href="estilo.css" media="screen">
```

**rel**

- **Descripción:** Define la relación entre el documento actual y el recurso vinculado.
- **Valor en el ejemplo:** stylesheet.
- **Significado:** Indica que el documento vinculado es una hoja de estilo CSS.

## **type**

- **Descripción:** Especifica el tipo de contenido del recurso vinculado.
- **Valor en el ejemplo:** text/css.
- **Significado:** Indica que el recurso vinculado es una hoja de estilo en formato CSS (Cascading Style Sheets). Aunque este atributo es opcional y muchas veces implícito para hojas de estilo, se puede incluir para mayor claridad.

## **href**

- **Descripción:** Especifica la URL del recurso vinculado.
- **Valor en el ejemplo:** estilo.css
- **Significado:** Indica la ruta del archivo CSS que se está vinculando. En este caso, el archivo estilo.css contiene las reglas de estilo que se aplicarán al documento HTML.

## **media**

- **Descripción:** Especifica para qué dispositivo (o tipo de dispositivo) está diseñado el documento vinculado.

- **Valor en el ejemplo:** screen
- **Significado:** Indica que la hoja de estilo es para dispositivos de pantalla, como monitores, tablets y smartphones. Otros valores comunes para este atributo incluyen print (para impresión) y all (para todos los dispositivos).

**4.7** ¿Qué pasaría si el archivo con extensión? css cambia de nombre a prueba CSS?, ¿Cuál podría ser una solución para corregirlo?

**RTA:** Ya no será reconocido como un archivo **CSS** por los navegadores web y otros programas que manejan **CSS**. Esto causará que los estilos definidos en ese archivo no se apliquen a la página HTML.

**Razón del Problema:** Los navegadores web utilizan la extensión de archivo para determinar el tipo de contenido. Al cambiar el nombre de estilo.css a prueba CSS (sin la extensión .css), el navegador no podrá identificar correctamente el tipo de archivo y no lo procesará como una hoja de estilo.

**Solución:**

Para solucionar este problema, es importante mantener la extensión .css al final del nombre del archivo. Si necesitas cambiar el nombre del archivo, asegúrate de que aún termine con .css. Por ejemplo, si deseas renombrar estilo.css a algo diferente, usa un nombre como pruebaCSS.css.

**Aquí te muestro los pasos para corregir el problema:**

- **Renombra el archivo:** Cambia el nombre del archivo a pruebaCSS.css.
- **Actualiza la referencia en el HTML:** Modifica el valor del atributo href en la etiqueta <link> para que apunte al nuevo nombre del archivo.

**8. Mencione algunos editores que maneje JavaScript.**

### **8.1 Visual Studio Code (VS Code)**

- **Descripción:** Es uno de los editores de código más populares entre los desarrolladores. Ofrece una amplia gama de extensiones que facilitan el desarrollo en JavaScript, incluyendo autocompletado, depuración, integración con sistemas de control de versiones, y más.

- **Características:** Resaltado de sintaxis, terminal integrada, depurador, extensiones para mejorar la experiencia de desarrollo (como ESLint, Prettier, etc.).

## 8.2 Sublime Text

- **Descripción:** Un editor de texto ligero y rápido que es muy popular por su velocidad y simplicidad. Soporta múltiples lenguajes, incluyendo JavaScript, y permite la instalación de paquetes para ampliar su funcionalidad.
- **Características:** Múltiples cursores, panel de comandos rápido, personalización a través de plugins.

## 8.3 WebStorm

- **Descripción:** Un IDE desarrollado por JetBrains, especializado en el desarrollo web. Ofrece soporte completo para JavaScript, HTML, CSS, y frameworks como Angular, React, y Vue.js.
- **Características:** Depuración avanzada, integración con sistemas de control de versiones, herramientas para trabajar con bases de datos y servidores.



## 8.4 Atom

- **Descripción:** Un editor de texto de código abierto desarrollado por GitHub. Es altamente personalizable y cuenta con una gran cantidad de paquetes que pueden añadirse para mejorar el desarrollo en JavaScript.
- **Características:** Integración con Git y GitHub, autocompletado inteligente, temas personalizables.

## 8.5 Brackets

- **Descripción:** Un editor de código diseñado para desarrolladores web, creado por Adobe. Es especialmente útil para trabajar con HTML, CSS y JavaScript, y ofrece características únicas como la edición en vivo.
- **Características:** Vista previa en vivo, edición en línea, soporte para preprocesadores como Sass y LESS.

## 8.6 Notepad++

- **Descripción:** Un editor de texto simple y ligero que soporta varios lenguajes de programación, incluyendo JavaScript. Es menos avanzado que otros editores, pero es útil para ediciones rápidas y trabajos ligeros.
- **Características:** Resaltado de sintaxis, búsqueda avanzada, múltiples documentos abiertos en pestañas.

## 8.7 Eclipse (con el plugin Wild Web Developer)

- **Descripción:** Un IDE extensible que, con el plugin Wild Web Developer, ofrece soporte para JavaScript y otros lenguajes web. Es robusto y ampliamente utilizado en entornos corporativos.
- **Características:** Depurador, integración con sistemas de control de versiones, herramientas de modelado y bases de datos.

**11.** ¿Por qué cada navegador al ejecutar el código anterior muestra diferentes aspectos en la interfaz?

Las diferencias en cómo los navegadores muestran una interfaz al ejecutar el mismo código JavaScript se deben a varias razones, relacionadas principalmente con la interpretación del código, la implementación del motor de renderizado, y el soporte de estándares web. Aquí te explico las razones principales:

## 1.1. Diferencias en los Motores de Renderizado

- Cada navegador utiliza un motor de renderizado diferente para interpretar y mostrar el HTML, CSS, y JavaScript. Estos motores no siempre implementan los estándares web de la misma manera, lo que puede resultar en diferencias en la apariencia de una página web.
- **Ejemplos de motores de renderizado:**
  - Chrome usa Blink.
  - Firefox usa Gecko.
  - Safari usa WebKit.
  - Edge (nueva versión basada en Chromium) también usa Blink.

## 1.2. Implementación de JavaScript

- Aunque JavaScript es un estándar (ECMAScript), cada navegador implementa su propio motor de JavaScript para ejecutar el código. Estos motores, como V8 en Chrome y Edge, SpiderMonkey en Firefox, y JavaScript Core en Safari, pueden tener ligeras diferencias en cómo interpretan y ejecutan el código JavaScript.

- Estas diferencias pueden afectar el rendimiento y, en algunos casos, el comportamiento del código.

### **1.3. Compatibilidad y Soporte de Estándares Web**

- Los navegadores no siempre implementan nuevos estándares web al mismo tiempo o de la misma manera. Algunas características pueden estar soportadas en un navegador y no en otro, lo que puede llevar a inconsistencias en la interfaz.
- Por ejemplo, propiedades avanzadas de CSS o nuevas APIs de JavaScript podrían comportarse de manera diferente o no estar soportadas en navegadores más antiguos.

### **1.4. Manejo de CSS**

- El CSS es interpretado de manera ligeramente diferente por cada navegador, lo que puede llevar a variaciones en la forma en que se renderizan los elementos de la interfaz.

- Esto incluye diferencias en el modelo de caja (box model), cómo se calculan los márgenes, padding, y bordes, o cómo se manejan las fuentes y el texto.

### **1.5. Diferencias en el Manejo del DOM**

- El DOM (Document Object Model) es la representación estructural de la página web en la memoria del navegador. Diferentes navegadores pueden tener variaciones en cómo manejan y manipulan el DOM, lo que puede afectar la actualización dinámica del contenido de la página.

### **1.6. Comportamiento de las Funciones de JavaScript**

- Algunas funciones y APIs de JavaScript pueden tener ligeras diferencias de comportamiento entre navegadores. Por ejemplo, el manejo de eventos, las animaciones o las funciones de manipulación de fechas pueden no comportarse exactamente igual en todos los navegadores.

### **1.7. Optimización y Caching**

- Los navegadores pueden optimizar y cachear el código JavaScript de manera diferente, lo que a veces puede llevar a diferencias en el tiempo de carga y en cómo se presenta la interfaz.}

### **1.8. Configuraciones y Extensiones del Navegador**

- Las configuraciones específicas del usuario o las extensiones instaladas pueden afectar cómo un navegador renderiza una página web. Por ejemplo, bloqueadores de anuncios, configuraciones de privacidad, o temas personalizados pueden alterar la apariencia o funcionalidad de una página.

### **1.9. Versiones del Navegador**

- Diferentes versiones del mismo navegador pueden manejar el código JavaScript y CSS de manera diferente, especialmente si las versiones más antiguas no soportan los últimos estándares o tienen bugs específicos.

### **1.10. Resolución de Pantalla y Dispositivos**

- Aunque esto no está directamente relacionado con el navegador, diferentes dispositivos y resoluciones de pantalla pueden influir en cómo se muestra la interfaz, y los navegadores pueden aplicar diferentes escalas o ajustes según el dispositivo.

**12.** Consulte las fuentes necesarias y explique en el cuadro que significa cada sección del código del punto 9:

Línea	código	Descripción
1	<code>&lt;!DOCTYPE html&gt;</code>	Declara el tipo de documento y define la versión del HTML. En este caso, <code>&lt;!DOCTYPE html&gt;</code> indica que se está utilizando HTML5.
2	<code>&lt;html&gt;</code>	Etiqueta de apertura que indica el inicio del documento HTML.
3	<code>&lt;head&gt;</code>	Sección del documento HTML que contiene metadatos sobre la página, como su título, enlaces a archivos CSS, scripts, etc.
4	<code>&lt;title&gt;&lt;/title&gt;</code>	Etiqueta dentro del <code>&lt;head&gt;</code> que define el título de la página web. Este título se muestra en la pestaña del navegador. En este caso, está vacío.
5	<code>&lt;/head&gt;</code>	Cierra la sección <code>&lt;head&gt;</code> .
6	<code>&lt;body&gt;</code>	Inicia el cuerpo del documento HTML. Todo el contenido visible de la página web (como texto, imágenes, botones, etc.) se coloca dentro de esta etiqueta.

7	<b>&lt;script&gt;</b>	Etiqueta que se utiliza para incluir código JavaScript dentro de un documento HTML.
8	<b>function popup () {</b>	Declara una función en JavaScript llamada popup. Esta función será ejecutada cuando sea llamada.
9	<b>alert ("Hola gente")</b>	Dentro de la función popup, se utiliza alert() para mostrar un cuadro de alerta con el mensaje "Hola gente". Esta función detiene la ejecución del código hasta que el usuario cierre el cuadro de alerta.
10	<b>}</b>	Cierra la definición de la función popup.
11	<b>&lt;/script&gt;</b>	Cierra la sección de código JavaScript.
12	<b>&lt;button type="button" onclick="popup()"&gt;Probar Funcion&lt;/button&gt;</b>	Define un botón en la página. El atributo type="button" especifica que este es un botón estándar. El atributo onclick="popup()" indica que al hacer clic en el botón, se ejecutará la función popup. El texto "Probar función" es lo que se mostrará en el botón.



13	<code>&lt;/body&gt;</code>	Cierra el cuerpo del documento HTML.
14	<code>&lt;/html&gt;</code>	Cierra el documento HTML.

13.5 Ejecute el archivo .html. Si todo se ha realizado correctamente se muestra lo siguiente luego de dar clic en el botón:



14. ¿Cuáles son las etiquetas y atributos en el archivo html que permite conectar o enlazar archivos externos css y .js respectivamente para que funcione?

**RTA:** Para enlazar archivos CSS, utilizas la etiqueta `<link>` con el atributo href en el `<head>` del HTML: `<link rel="stylesheet" href="estilos.css">`. Para conectar archivos JavaScript, usas la etiqueta `<script>` con el atributo src, generalmente justo antes del cierre del `<body>`: `<script src="script.js"></script>`.

**15.** Para finalizar, comente las conclusiones o conocimientos adquiridos en el instructivo.

### **Conclusión**

Como conclusión he aprendido que el uso de etiquetas HTML y la vinculación de archivos externos como CSS y JAVASCRIPT son fundamentales para el diseño y la funcionalidad de una pagina web. Las etiquetas '`<link>`' y '`<script>`' permiten conectar hojas de estilo y script , respectivamente, mejorando la presentación y la interactividad de la página. Esto resalta la importancia de comprender como estas tecnologías trabajan juntas para crear experiencias web dinámicas y fáciles de entender.