**PROCESO DE GESTIÓN DE FORMACIÓN PROFESIONAL INTEGRAL**

**FORMATO GUÍA DE APRENDIZAJE**

**1. IDENTIFICACIÓN DE LA GUIA DE APRENDIZAJE**

* Denominación del Programa de Formación: **ANALISIS Y DESARROLLO DE SOFTWARE**.
* Código del Programa de Formación: **228118**
* Nombre del Proyecto Formativo: **CONSTRUCCIÓN DE SOFTWARE A LA MEDIDA, INTEGRANDO LAS TECNOLOGÍAS ORIENTADAS A LOS SERVICIOS**
* Fase del Proyecto: **PLANEACION**
* Actividad de Proyecto Formativo: **ESTRUCTURAR PROCESOS LOGICOS PARA LA CONSTRUCCIÓN DE ALGORITMOS**
* Competencia: **MODELADO DE LOS ARTEFACTOS DEL SOFTWARE.**
* Resultados de Aprendizaje:

- 01 ELABORAR LOS ARTEFACTOS DE DISEÑO DEL SOFTWARE SIGUIENDO LAS PRÁCTICAS DE LA METODOLOGÍA SELECCIONADA.

-02 ESTRUCTURAR EL MODELO DE DATOS DEL SOFTWARE DE ACUERDO CON LAS ESPECIFICACIONES DEL ANÁLISIS.

* Duración de la Guía de Aprendizaje (48 horas):

**2. PRESENTACIÓN**

HTML, CSS y JavaScript son los tres pilares fundamentales del desarrollo web. HTML, o Lenguaje de Marcado de Hipertexto, es el componente básico que estructura y organiza el contenido en una página web. CSS, o Hojas de Estilo en Cascada, complementa a HTML al proporcionar las herramientas necesarias para estilizar y darle formato visual a ese contenido, definiendo colores, fuentes, distribuciones y otros aspectos de diseño. Por último, JavaScript es un lenguaje de programación que añade interactividad y dinamismo a las páginas web, permitiendo la creación de funcionalidades avanzadas como formularios dinámicos, animaciones y la actualización de contenidos sin necesidad de recargar la página. Juntos, estos tres lenguajes forman la base de cualquier sitio web moderno, trabajando en conjunto para crear experiencias de usuario ricas y atractivas.



**Ambiente requerido** polivamente 3

**Estrategias o técnicas didácticas activas**: trabajo colaborativo

**Materiales de formación**: TV, COMPUTADORAS, INTERNET

**Material de apoyo**: ***figura 1***

**Duración de la actividad**: 30 minutos

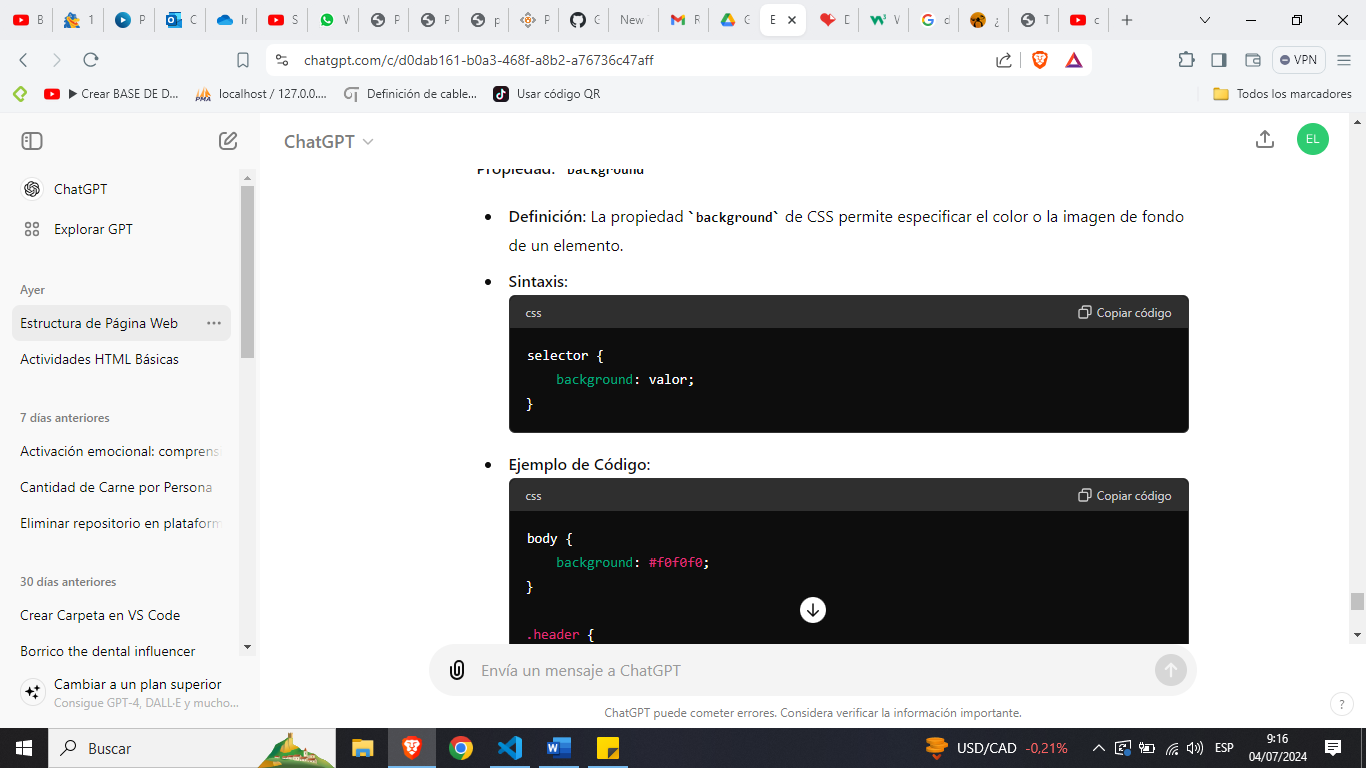
**Descripción de la actividad 2.4:** Cada aprendiz seleccionará un grupo de términos y propiedades de CSS para investigar. Utilizarán recursos en línea como **MDN Web Docs** y **W3Schools** para obtener información detallada sobre cada término o propiedad.

**Creación del Glosario**

Los aprendices crearán una entrada de glosario para cada término o propiedad investigada, incluyendo una definición, un ejemplo de código y una breve explicación del uso.

Las entradas del glosario se organizarán alfabéticamente.

Ejemplo:



**Presentación y Discusión**

Los aprendices presentarán sus términos y propiedades al grupo.

Se fomentará la discusión y el intercambio de ejemplos y experiencias.

Aplicación Práctica

Los aprendices aplicarán los términos y propiedades aprendidas en un proyecto práctico.

Se les asignará la tarea de diseñar una página web simple utilizando las propiedades CSS del glosario.

**Ambiente requerido** polivalente 3

**Estrategias o técnicas didácticas activas**: trabajo colaborativo

**Materiales de formación**: TV, COMPUTADORAS, INTERNET

**Material de apoyo**: ***inteligencia artificial, w3school,MDN web docs***

**Duración de la actividad**: 5 horas

**GLOSARIO**

**TERMINOS:**

1. **@rule (Media Query** Las reglas @ en CSS, como @media, permiten aplicar estilos basados en características específicas del dispositivo, como tamaño de pantalla, orientación, etc.

Ejemplo:

@media (max-width: 768px) {

.menu {

display: none;

}

}

1. **Animación**: Especifica cómo cambiará una propiedad CSS en el tiempo, utilizando @keyframes para definir los estados de la animación.

Ejemplo:

.animacion {

animation: mover 2s infinite alternate;

}

@keyframes mover {

from {

transform: translateX(0);

}

to {

transform: translateX(100px);

}

}

1. **Declaración**: Combinación de una propiedad y su valor dentro de un bloque de reglas CSS, como color: red;.

.caja {

background-color: #f0f0f0;

padding: 10px;

bordes: 1px solid #ccc;

}

1. **Especificidad**: Un valor numérico que determina qué conjunto de reglas CSS se aplicará a un elemento en caso de conflicto.

Ejemplo:

. contenido #seccion .articulo p {

color: black;

}

p {

color: gray;

}

1. **Flexbox**: Un modelo de diseño unidimensional que permite distribuir el espacio entre elementos de una manera más eficiente y predecible.

Ejemplo:

.contenedor {

display: flex;

justify-content: space-between;

}

1. **Grid**: Un sistema de diseño bidimensional que permite organizar elementos en filas y columnas, proporcionando un control más preciso sobre la disposición de los elementos.

Ejemplo:

.grid-container {

display: grid;

grid-template-columns: 1fr 2fr;

gap: 10px;

}

1. **Herencia**: El mecanismo por el cual ciertos estilos aplicados a un elemento se transmiten automáticamente a sus elementos hijos.

Ejemplo:

.padre {

font-family: Arial, sans-serif;

}

.hijo {

/\* Hereda la propiedad de font-family del elemento padre \*/

font-size: 14px;

}

1. **Pseudo-clase**: Un selector que permite seleccionar elementos que están en un estado específico, como :hover, :focus, :first-child, etc.

Ejemplo:

a:hover {

text-decoration: underline;

}

1. **Pseudo-elemento**: Un selector que permite estilizar partes específicas de un elemento, como ::before y ::after, que crean contenido adicional antes o después del contenido real del elemento.

Ejemplo:

.aviso::before {

content: "¡Advertencia!";

color: red;

}

1. **Propiedad y Valor**: Una propiedad CSS es una característica específica que se desea cambiar (por ejemplo, color, font-size), y un valor es la asignación específica de esa propiedad (por ejemplo, red, 16px).

Ejemplo:

h1 {

color: red;

}

.texto-grande {

font-size: 24px;

}

1. **Selector**: Un patrón que se utiliza para seleccionar los elementos a los que se aplicarán estilos CSS, como selectores de etiqueta (p), de clase (.clase), de ID (#id), entre otros.

Ejemplo:

p {

color: blue;

font-size: 16px;

}

.dest

acado {

font-weight: bold;

}

#titulo {

text-align: center;

}

1. **Transición**: Un efecto suave que se aplica a una propiedad CSS cuando cambia de un estado a otro, controlado por transition y especificado en transition-property, transition-duration, transition-timing-function, y transition-delay.

Ejemplo:

.caja {

transition: background-color 0.3s ease;

}

.caja:hover {

background-color: lightblue;

}

**Propiedades:**

1**.color**

* **Definición**: Define el color del texto.
* **Ejemplo**: color: blue;

2.**background-color**

* **Definición**: Define el color de fondo de un elemento.
* **Ejemplo**: background-color: #f0f0f0;

3.**font-family**

* **Definición**: Define la familia de fuentes para el texto.
* **Ejemplo**: font-family: Arial, sans-serif;

4.**font-size**

* **Definición**: Define el tamaño de la fuente.
* **Ejemplo**: font-size: 16px;

5.**font-weight**

* **Definición**: Define el grosor o peso de la fuente.
* **Ejemplo**: font-weight: bold;

6.**font-style**

* **Definición**: Define el estilo de la fuente (normal, itálica, oblicua).
* **Ejemplo**: font-style: italic;

7.**line-height**

* **Definición**: Define la altura de línea dentro de un elemento de texto.
* **Ejemplo**: line-height: 1.5;

8.**text-align**

* **Definición**: Define la alineación horizontal del texto dentro de un elemento.
* **Ejemplo**: text-align: center;

9.**text-decoration**

* **Definición**: Define la decoración del texto (subrayado, tachado, etc.).
* **Ejemplo**: text-decoration: underline;

10.**text-transform**

* **Definición**: Define cómo se transforma el texto (mayúsculas, minúsculas, capitalización).
* **Ejemplo**: text-transform: uppercase;

11.**letter-spacing**

* **Definición**: Define el espacio adicional entre caracteres de texto.
* **Ejemplo**: letter-spacing: 1px;

12.**word-spacing**

* **Definición**: Define el espacio adicional entre palabras.
* **Ejemplo**: word-spacing: 2px;

13.**padding**

* **Definición**: Define el espacio interno entre el borde y el contenido de un elemento.
* **Ejemplo**: padding: 10px;

14.**margin**

* **Definición**: Define el espacio externo alrededor de un elemento.
* **Ejemplo**: margin: 20px;

15.**border**

* **Definición**: Define el borde de un elemento (ancho, estilo y color).
* **Ejemplo**: border: 1px solid #ccc;

16.**border-radius**

* **Definición**: Define la curvatura de las esquinas de un borde.
* **Ejemplo**: border-radius: 5px;

17.**width**

* **Definición**: Define el ancho de un elemento.
* **Ejemplo**: width: 300px;

18.**height**

* **Definición**: Define la altura de un elemento.
* **Ejemplo**: height: 200px;

19.**max-width**

* **Definición**: Define el ancho máximo que puede tener un elemento.
* **Ejemplo**: max-width: 100%;

20.**max-height**

* **Definición**: Define la altura máxima que puede tener un elemento.
* **Ejemplo**: max
* -height: 80vh;

21.**min-width**

* **Definición**: Define el ancho mínimo que puede tener un elemento.
* **Ejemplo**: min-width: 200px;

22.**min-height**

* **Definición**: Define la altura mínima que puede tener un elemento.
* **Ejemplo**: min-height: 100px;

23.**display**

* **Definición**: Define cómo se muestra un elemento (bloque, línea, en línea, etc.).
* **Ejemplo**: display: inline-block;

24.**position**

* **Definición**: Define el método de posicionamiento de un elemento.
* **Ejemplo**: position: relative;

25.**top, right, bottom, left**

* **Definición**: Define la posición de un elemento posicionado en relación con su contenedor padre.
* **Ejemplo**: top: 10px;

26.**float**

* **Definición**: Define si un elemento flota hacia la izquierda o la derecha dentro de su contenedor.
* **Ejemplo**: float: left;

27.**clear**

* **Definición**: Define si un elemento debe estar a la izquierda, a la derecha o ambos lados de elementos flotantes.
* **Ejemplo**: clear: both;

28.**overflow**

* **Definición**: Define cómo se maneja el contenido que desborda los límites de su contenedor.
* **Ejemplo**: overflow: hidden;

29.**background-image**

* **Definición**: Define la imagen de fondo de un elemento.
* **Ejemplo**: background-image: url('imagen.jpg');

30.**background-position**

* **Definición**: Define la posición inicial de la imagen de fondo.
* **Ejemplo**: background-position: center top;

31.**background-size**

* **Definición**: Define el tamaño de la imagen de fondo.
* **Ejemplo**: background-size: cover;

32.**background-repeat**

* **Definición**: Define cómo se repite una imagen de fondo.
* **Ejemplo**: background-repeat: no-repeat;

33.**box-shadow**

* **Definición**: Define la sombra alrededor de un elemento.
* **Ejemplo**: box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);

34.**opacity**

* **Definición**: Define la opacidad de un elemento (valores entre 0 y 1).
* **Ejemplo**: opacity: 0.5;

36.**z-index**

* **Definición**: Define el orden de apilamiento de un elemento posicionado.
* **Ejemplo**: z-index: 100;

37.**cursor**

* **Definición**: Define el tipo de cursor cuando el mouse está sobre un elemento.
* **Ejemplo**: cursor: pointer;

38.**pointer-events**

* **Definición**: Define si un elemento debe responder a eventos del mouse.
* **Ejemplo**: pointer-events: none;

39.**transform**

* **Definición**: Aplica transformaciones a un elemento (rotación, escala, traslación, inclinación).
* **Ejemplo**: transform: rotate(45deg);

40. **transition**

* **Definición**: Define una transición suave entre diferentes estados de un elemento.
* **Ejemplo**: transition: opacity 0.3s ease-in-out;

41.**animation**

* **Definición**: Define una animación para un elemento.
* **Ejemplo**: animation: mover 2s infinite alternate;

42,**flex-direction**

* **Definición**: Define la dirección en la que se colocan los elementos flexibles en un contenedor.
* **Ejemplo**: flex-direction: row-reverse;

43.**justify-content**

* **Definición**: Define cómo se distribuyen los elementos a lo largo del eje principal de un contenedor flexible.
* **Ejemplo**: justify-content: space-between;

44.**align-items**

* **Definición**: Define cómo se alinean los elementos a lo largo del eje transversal de un contenedor flexible.
* **Ejemplo**: align-items: center;

45.**flex**

* **Definición**: Define la capacidad de un elemento flexible para expandirse para llenar el espacio disponible.
* **Ejemplo**: flex: 1;

46.**grid-template-columns**

* **Definición**: Define las columnas de una cuadrícula CSS.
* **Ejemplo**: grid-template-columns: 1fr 2fr;

47.**grid-template-rows**

* **Definición**: Define las filas de una cuadrícula CSS.
* **Ejemplo**: grid-template-rows: auto 100px;

48.**grid-gap**

* **Definición**: Define el espacio entre las filas y las columnas de una cuadrícula CSS.
* **Ejemplo**: grid-gap: 10px;

49.**grid-column**

* **Definición**: Define la ubicación de un elemento dentro de una cuadrícula CSS en términos de columnas.
* **Ejemplo**: grid-column: 1 / span 2;

50.**grid-row**

* **Definición**: Define la ubicación de un elemento dentro de una cuadrícula CSS en términos de filas.
* **Ejemplo**: grid-row: 2 / 3;

51.**list-style-type**

* **Definición**: Define el estilo del marcador de una lista.
* **Ejemplo**: list-style-type: disc;

**3.3 Actividades de apropiación:**

**Descripción de la actividad 1**: Crea una página web simple sobre ti mismo, Incluye una breve biografía, una foto tuya, tus hobbies y enlaces a tus redes sociales.

Utiliza etiquetas HTML básicas y más avanzadas que consideres para el desarrollo de tu pagina .

**MI BIOGRAFIA**

<!DOCTYPE html>

<html lang="es">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Sobre Mí</title>

    <style>

        body {

            font-family: Arial, sans-serif;

            margin: 0;

            padding: 0;

            background-color: #84c46b70;

            color: #333;

        }

        .container {

            width: 80%;

            margin: 0 auto;

            padding: 20px;

        }

        header {

            text-align: center;

            padding: 50px 0;

        }

        header img {

            border-radius: 50%;

            width: 150px;

            height: 150px;

        }

        header h1 {

            margin: 20px 0;

        }

        section {

            margin: 40px 0;

        }

        .social-links a {

            margin: 0 10px;

            text-decoration: none;

            color: #1a73e8;

        }

        .social-links a:hover {

            text-decoration: underline;

        }

    </style>

</head>

<body>

    <div class="container">

        <header>

            <img src="Juan.jpeg" alt="1">

            <h1>Juan Esteban Durango Salazar</h1>

        </header>

        <section>

            <h2>Biografía</h2>

            <p>Hola, mi nombre es Juan Esteban Durango Salazar, tengo 18 años. Nací el 19 de enero del 2006. Soy un aprendiz del SENA y vivo en Caucasia-Antioquia. Me apasiona la astronomía, medicina natural, reparación de dispositivos móviles. Vengo de Buenaventura Valle del Cauca. </p>

        </section>

        <section>

            <h2>Hobbies</h2>

            <p>En mi tiempo libre, disfruto de jugar videojuegos, salir con mis amigos, y visitar a mi padrino. Estas actividades me ayudan a relajarme y mantenerme creativo. </p>

        </section>

        <section>

            <h2>Redes Sociales</h2>

            <div class="social-links">

                <a href="https://www.facebook.com/juan.ds.5059?mibextid=ZbWKwL" target="\_blank">Facebook</a>

                <a href="https://wa.me/573135257397" target="\_blank">Whatsapp</a>

                <a href="https://www.instagram.com/juanes1.0?igsh=MTRtZGxoYzdiNnRmZw==" target="\_blank">Instagram</a>

            </div>

        </section>

    </div>

</body>

</html>

**Ambiente requerido**: polivamente 3

**Estrategias o técnicas didácticas activas**: Aprendizaje por proyectos

**Materiales de formación**: Computadores, visual studio code, Computadores, visual studio code

**Material de apoyo**: https://www.w3schools.com/

**Evidencias de aprendizaje:** Pagina web

**Instrumentos de evaluación**:

**Duración de la actividad**: 6 horas.

**3.4 Actividades de Transferencia el Conocimiento:** Para la siguiente actividad se trabaja el **CASO DE ESTUDIO** para el desarrollo de un sistema de información para la empresa de transporte público intermunicipal *“TransMarimba”*.

**Nota:** Esta actividad se deberá realizar de manera individual. No olvide al finalizar la actividad subir la carpeta con las evidencias de la página web realizada.

**CASO DE ESTUDIO:** La empresa **TransMarimba**, ubicada en la ciudad de Medellín, presta servicio de transporte intermunicipal dentro del departamento de Antioquia. Se requiere realizar un aplicativo para la venta y reserva de tiquetes.

Usted ha sido contratado para realizar la página web (FrontEnd), actividad para lo cual deberá tener en cuenta:

La empresa **necesita hacer saber a sus cliente**s los servicios que ofrece: transporte de pasajeros, los destinos que maneja y envío de encomiendas.

Texto, Logotipo

Descripción generada automáticamente

Para TransMarimba también es importante que los usuarios puedan conocer la empresa, sus políticas, su misión, su visión, en otras palabras que muestren toda la información de la empresa. Adicional, que se muestre la información de contacto y se cuente un formulario de contacto desde donde los usuarios puedan enviar directamente sus PQRSF (Peticiones, Quejas, Reclamos, Solicitudes, Felicitaciones).

**Actividad:**

De acuerdo a los requisitos solicitados por la empresa “TransMarimba”.

1. Realice en Paint, un editor de imágenes o en una hoja, un bosquejo de la estructura que debería tener la página web.
2. Utilizando las etiquetas de HTML 5 y CSS 3 y teniendo en cuenta los requisitos de la empresa, realice una página web, la cual deberá contener como mínimo:

* Un menú con listas No ordenas
* Inserción de una imagen
* Inserción de un vídeo
* Inserción de un mapa
* Un formulario de Contacto

Ambiente requerido: polivante 3

**Estrategias o técnicas didácticas activas**: caso de uso

**Materiales de formación: Computadores**, visual studio code, paint, hojad de bloc, lapiz

**Material de apoyo:** https://www.w3schools.com/

**Evidencias de aprendizaje: pagina web, mapa de navegación**

**Instrumentos de evaluación: lista de chequeo**

**Duración de la actividad: 12 horas**

**4. PLANTEAMIENTO DE EVIDENCIAS DE APRENDIZAJE PARA LA EVALUACIÓN EN EL PROCESO FORMATIVO.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Fase del proyecto formativo** | **Actividad del proyecto formativo** | **Actividad de Aprendizaje** | **Evidencias de Aprendizaje** | **Criterios de Evaluación** | **Técnicas e Instrumentos de Evaluación** |
| **Plenacion** | CONSTRUIR EL PROTOTIPO DEL SISTEMA DE INFORMACIÓN DE ACUERDO AL ANÁLISIS DE LAS CARACTERÍSTICAS FUNCIONALES Y DE CALIDAD. | Elaborar el prototipo navegable del software aplicando estandares de usabilidad y accesibilidad | Evidencias de Conocimiento:  Evaluación de conocimientos sobre interfaz gráfica de usuario, usabilidad, accesibilidad  Evidencias de Desempeño:  Participa en las diferentes actividades pertinentemente y asertivamente.  Evidencias de Producto:  Maquetación de Interfaz gráfica en HTML | Propone la interfaz gráfica de usuario.  •Construye el mapa de navegación cumpliendo reglas de usabilidad y accesibilidad.  •Elabora prototipos según los requisitos del software | **Evaluación**  **Creación de pagina web**  **Exposicion Mapa de Navegación** |
|  |  |  |  |  |  |

**5. GLOSARIO DE TÉRMINOS**

1. **Etiqueta (Tag)**:
   * Elemento del lenguaje HTML utilizado para definir la estructura y contenido de un documento web. Ejemplo: <p>, <div>, <a>.
2. **Atributo (Attribute)**:
   * Propiedad adicional que se puede agregar a una etiqueta HTML para definir sus características. Ejemplo: href en <a href="url">.
3. **Elemento (Element)**:
   * Combinación de una etiqueta de apertura, contenido y una etiqueta de cierre. Ejemplo: <p>Este es un párrafo.</p>.
4. **Anidamiento (Nesting)**:
   * Práctica de colocar elementos HTML dentro de otros elementos. Ejemplo: <div><p>Texto</p></div>.
5. **DOM (Document Object Model)**:
   * Representación en árbol del documento HTML que permite a los lenguajes de programación interactuar y modificar su estructura y contenido.
6. **Atributo Global (Global Attribute)**:
   * Atributo que puede ser utilizado en cualquier elemento HTML. Ejemplo: class, id, style.
7. **Metadatos (Metadata)**:
   * Información sobre los datos del documento, usualmente ubicada en la sección <head>. Ejemplo: <meta charset="UTF-8">.
8. **Formularios (Forms)**:
   * Estructuras de entrada de datos del usuario definidas con la etiqueta <form>. Incluyen elementos como <input>, <textarea>, <button>.
9. **Enlace (Link)**:
   * Elemento <a> utilizado para crear hipervínculos a otras páginas o recursos.
10. **Comentario (Comment)**:
    * Texto dentro de <!-- --> que no es visible en el navegador y es utilizado para dejar notas en el código.

### Glosario de Términos de CSS

1. **Selector**:
   * Parte de una regla CSS que indica a qué elementos HTML se aplicarán los estilos. Ejemplo: p, .clase, #id.
2. **Propiedad (Property)**:
   * Característica del estilo que se va a aplicar a un elemento HTML. Ejemplo: color, font-size, margin.
3. **Valor (Value)**:
   * Especificación del estilo que se aplicará a la propiedad. Ejemplo: red, 16px, auto.
4. **Regla (Rule)**:
   * Conjunto de selectores, propiedades y valores. Ejemplo: p { color: red; font-size: 16px; }.
5. **Declaración (Declaration)**:
   * Combinación de una propiedad y su valor. Ejemplo: color: blue;.
6. **Hoja de Estilos (Stylesheet)**:
   * Archivo .css que contiene reglas CSS para definir estilos de un documento HTML.
7. **Cascada (Cascade)**:
   * Mecanismo que determina qué estilos se aplican cuando hay múltiples reglas CSS para un mismo elemento.
8. **Especificidad (Specificity)**:
   * Medida que determina qué reglas CSS se aplican sobre otras en caso de conflicto. Ejemplo: selectores de ID tienen mayor especificidad que selectores de clase.
9. **Box Model**:
   * Modelo que describe cómo se renderizan los elementos en CSS, incluyendo content, padding, border y margin.
10. **Flexbox**:
    * Módulo de diseño CSS que permite la creación de layouts flexibles y eficientes. Utiliza contenedores y elementos flexibles.
11. **Grid Layout**:
    * Sistema de diseño CSS basado en una cuadrícula de filas y columnas, permitiendo un control preciso del layout.
12. **Pseudo-clase (Pseudo-class)**:
    * Palabra clave añadida a un selector que especifica un estado especial del elemento seleccionado. Ejemplo: :hover, :active.
13. **Pseudo-elemento (Pseudo-element)**:
    * Palabra clave añadida a un selector que permite definir estilos a una parte específica del elemento seleccionado. Ejemplo: ::before, ::after.
14. **Media Query**:
    * Técnica CSS para aplicar estilos condicionales basados en características del dispositivo, como el tamaño de la pantalla. Ejemplo: @media (max-width: 600px) { ... }.

**6. REFERENTES BILBIOGRÁFICOS**

**HTML**

Mozilla Developer Network (MDN) HTML Documentation

**La guía de referencia completa y tutoriales proporcionados por MDN**.

W3Schools HTML Tutorial

**Tutoriales interactivos y ejemplos fáciles de seguir.**

HTML.com

**Ofrece tutoriales y guías sobre HTML.**

CSS

Mozilla Developer Network (MDN) CSS Documentation

**Documentación completa y guías para aprender CSS.**

W3Schools CSS Tutorial

**Tutoriales interactivos y ejemplos prácticos.**

CSS-Tricks

**Blog, artículos y ejemplos avanzados sobre CSS.**

JavaScript

Mozilla Developer Network (MDN) JavaScript Documentation

**Documentación y tutoriales completos sobre JavaScript.**

W3Schools JavaScript Tutorial

**Tutoriales básicos y avanzados sobre JavaScript.**

Eloquent JavaScript

**7. CONTROL DEL DOCUMENTO**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Nombre** | **Cargo** | **Dependencia** | **Fecha** |
| **Autor (es)** | **Elvis Vergara H** | **Instructor** | **CTGA** | **04-06-2024** |

**8. CONTROL DE CAMBIOS** (diligenciar únicamente si realiza ajustes a la guía)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Nombre** | **Cargo** | **Dependencia** | **Fecha** | **Razón del Cambio** |
| **Autor (es)** |  |  |  |  |  |