



Resumen T1 - T4

1.Diseño/2.Prototipado/3.HTML/4.CSS

Tema 1: Diseño de la Interfaz

El diseño de la interfaz es fundamental en la creación de sitios web, ya que determina cómo los usuarios interactúan con el contenido. Este proceso implica no solo el aspecto visual, sino también la usabilidad y la accesibilidad del sitio. A continuación, se detallan los aspectos más relevantes del diseño de la interfaz web.

1.1 Introducción al Diseño Web

El diseño web presenta diferencias clave con respecto al diseño gráfico tradicional debido a factores únicos:

- **Canal de transmisión (Internet):** La web presenta limitaciones tecnológicas, como la velocidad de conexión y el ancho de banda.
- **Audiencia global y diversa:** Los usuarios pueden acceder a los sitios web desde cualquier parte del mundo y utilizando distintos dispositivos.

Herramientas de los Visitantes

Para ofrecer una experiencia óptima, se deben considerar las características de las herramientas utilizadas por los visitantes:

- **Navegador:** El navegador permite visualizar solo una parte del contenido al inicio, siendo importante optimizar el diseño para que la información relevante esté en el área visible.
- **Tamaño de la pantalla y resolución:** Varían en función del dispositivo, lo que requiere un diseño adaptable.

Objetivos del Diseño Web

Para lograr una experiencia satisfactoria para el usuario, el diseño debe:

1. **Ofrecer acceso rápido al contenido:** La navegación debe ser intuitiva y permitir al usuario encontrar la información fácilmente.
2. **Garantizar una experiencia visual atractiva:** Esto ayuda a fidelizar a los usuarios y mejorar su percepción del sitio.

Usabilidad y Accesibilidad

- **Usabilidad:** Consiste en diseñar el sitio web para que sea fácil, eficiente y satisfactorio de utilizar.
- **Accesibilidad:** Busca que el contenido sea accesible para todos los usuarios, independientemente de sus capacidades físicas, cognitivas o técnicas. El W3C define la accesibilidad web como el diseño inclusivo que permite a personas con discapacidades utilizar los sitios web.

1.1.1 El Ciclo de Vida del Diseño Web

El proceso de diseño web se basa en el **Diseño Centrado en el Usuario (DCU)**, que sigue las siguientes fases:

1. **Análisis de Requisitos:** Comprender las necesidades del usuario y los objetivos del sitio.
2. **Especificación de Contenidos y Estructura de la Información:** Organizar el contenido de manera coherente para facilitar la navegación.
3. **Desarrollo del Diseño Visual:** Se crea un prototipo que puede incluir:
 - **Bocetos:** Dibujos simples de baja fidelidad.
 - **Wireframes:** Esquemas más detallados que muestran la disposición de los elementos.
 - **Prototipos:** Representaciones visuales que incluyen gráficos y colores.
4. **Pruebas de Evaluación:** Ciclo iterativo en el que se evalúa el diseño mediante pruebas con usuarios.
5. **Codificación y Maquetación Web:** Convertir los prototipos en una versión operativa utilizando tecnologías como HTML, CSS y JavaScript.
6. **Redacción de Contenidos:** Asegurarse de que el contenido sea relevante y cumpla con los estándares SEO.
7. **Test Final y Mantenimiento:** Pruebas de usabilidad, accesibilidad, promoción SEO y actualización del sitio web.

1.1.2 Perfiles Profesionales en el Diseño Web

El desarrollo de un sitio web involucra a diferentes perfiles, que incluyen:

- **Diseñadores de Interacción (IxD):** Se centran en la facilidad de uso y la eficiencia de la navegación.
- **Diseñadores de Interfaces (UI):** Organizan los elementos funcionales de la página.

- **Diseñadores de Experiencia de Usuario (UX):** Aseguran que la experiencia de navegación sea placentera y coherente con la marca.
- **Desarrolladores Web:** Implementan la maquetación y el comportamiento del sitio web utilizando tecnologías como HTML, CSS y JavaScript.
- **Especialistas en SEO:** Optimizan el contenido para mejorar el posicionamiento en motores de búsqueda.

1.2 Elementos de Diseño

1.2.1 El Color

El color es un elemento fundamental en el diseño web, ya que puede guiar, destacar o persuadir al usuario:

- **Modelos de Color:**
 - **RGB (Red, Green, Blue):** Utilizado para la visualización en pantallas.
 - **HSL (Hue, Saturation, Lightness):** Define el color en función de la tonalidad, saturación e iluminación.
- **Tipos de Diseño de Color:**
 - **Monocromático:** Usa diferentes tonalidades de un solo color.
 - **Colores vecinos:** Colores adyacentes en el círculo cromático que crean armonía.
 - **Colores complementarios:** Colores opuestos en el círculo cromático que generan contraste.

1.2.2 Texturas

Las texturas añaden profundidad y realismo al diseño web, haciendo que los usuarios conecten con el contenido. Existen diferentes tipos:

- **Sutiles:** Ayudan a crear una atmósfera sin distraer al usuario.
- **Llamativas:** Destacan y pueden dirigir la atención hacia un elemento específico.

1.2.3 Tipografía

La tipografía no solo afecta la apariencia del sitio web, sino también su legibilidad y accesibilidad.

- **Tipos de Letras:**
 - **Serif:** Tienen pequeños adornos al final de los trazos (ej. Times New Roman).
 - **Sans Serif:** Letras sin adornos, con un aspecto más moderno (ej. Arial).
 - **Monoespaciadas:** Todas las letras tienen el mismo ancho (ej. Courier).
- **Jerarquía Tipográfica:** Se deben definir tamaños y estilos coherentes para títulos, párrafos y botones.

1.2.4 Iconos

Los iconos son representaciones gráficas que deben ser simples, comprensibles y consistentes con el estilo general del sitio.

1.2.5 Composición del Sitio Web

La composición visual se refiere a la disposición de los elementos en la interfaz. Los principios de diseño incluyen:

- **Equilibrio:** Distribución equitativa del peso visual.
- **Contraste:** Diferencias visuales que destacan ciertos elementos.
- **Jerarquía Visual:** Orden de importancia de los elementos según la forma en que se lee la página (normalmente de izquierda a derecha y de arriba hacia abajo).

1.3 Tecnologías y Herramientas del Diseño Web

1.3.1 Editores HTML

Los editores HTML son herramientas que permiten la creación y edición de código web, facilitando tareas como la gestión de plantillas, la validación del código y el autocompletado. Ejemplo: **Visual Studio Code**.

1.3.2 Navegadores Web

Los navegadores son esenciales para la visualización de los sitios web. Cada navegador puede interpretar HTML y CSS de manera diferente, por lo que es importante:

- **Validar el código:** Usar herramientas como [Acid Test](#) para comprobar si el navegador cumple con los estándares web.
- **Depurar código JavaScript:** Utilizar herramientas para desarrolladores que permiten modificar y probar el código en tiempo real (tecla **F12**).

Enlaces de interés

- **Teoría del color:** [Referencia rápida](#)
- **Generadores de paletas de color:** [Adobe Color](#)
- **Google Fonts:** [Biblioteca de fuentes](#)

Leyenda de Siglas, Elementos y Nomenclaturas en Diseño de la Interfaz

A

- **Accesibilidad:** La práctica de hacer que las páginas web sean utilizables por personas con diversas capacidades, incluyendo discapacidades visuales, auditivas, motrices o cognitivas. Implica el uso de buenas prácticas como el uso correcto de etiquetas HTML, descripciones de imágenes (**alt**), y teclas de acceso rápido.
- **Análisis de requisitos:** Primera fase del diseño web, en la que se identifican las necesidades del usuario y los objetivos del proyecto para definir el alcance y los elementos esenciales del sitio web.
- **Arquitectura de la información:** Organización y estructuración de la información en un sitio web para que sea intuitiva y fácil de navegar, mejorando la experiencia del usuario.

B

- **Boceto:** Dibujo inicial de baja fidelidad que sirve para plasmar ideas de diseño de manera rápida y sencilla. Los bocetos ayudan a visualizar la disposición de los elementos antes de crear wireframes o prototipos más detallados.

C

- **Cabecera (header):** Sección de la interfaz que generalmente contiene el logo, el título del sitio y la navegación principal. Es un componente clave en el diseño de la estructura de la página.
- **Ciclo de vida del diseño web:** Proceso completo desde el análisis de requisitos hasta la implementación y mantenimiento de un sitio web, incluyendo diseño, pruebas y lanzamiento.
- **Color:** Elemento esencial del diseño web utilizado para evocar emociones, guiar la atención del usuario y establecer la identidad visual. Los colores pueden organizarse en paletas para definir colores primarios, secundarios y neutros.
- **Contraste:** Diferencia visual entre dos o más elementos en una interfaz, que puede ayudar a resaltar información importante y mejorar la legibilidad.

D

- **Diseño Centrado en el Usuario (DCU):** Enfoque de diseño que pone al usuario en el centro del proceso, buscando satisfacer sus necesidades y expectativas en todas las fases del desarrollo del sitio web.
- **Diseño gráfico:** Parte del diseño web que se ocupa de la creación de elementos visuales, como logotipos, iconos, colores y tipografías.
- **Diseño de interacción:** Enfocado en cómo los usuarios interactúan con la interfaz, asegurándose de que los elementos sean accesibles, fáciles de usar y respondan adecuadamente a las acciones del usuario.

E

- **Equilibrio:** Principio de diseño que implica distribuir los elementos visuales de manera armónica para que la página se sienta estable y organizada. Puede ser simétrico o asimétrico.
- **Estructura de la información:** Organización de los contenidos del sitio web en categorías o secciones para facilitar la navegación y la comprensión por parte del usuario.

F

- **Fidelidad:** Nivel de detalle o realismo en las representaciones de un diseño, como prototipos o maquetas. Puede ser baja (bocetos) o alta (prototipos interactivos).

G

- **Guía de estilos:** Documento que recopila todas las decisiones de diseño de un sitio web, incluyendo tipografías, colores, iconos y espaciado. Asegura la coherencia visual y ayuda a todos los miembros del equipo a seguir un mismo estándar.

H

- **HTML (HyperText Markup Language):** Lenguaje de marcado utilizado para definir la estructura y el contenido de las páginas web. Es la base sobre la cual se construyen todas las interfaces de usuario en la web.
- **Heurísticas de usabilidad:** Conjunto de principios utilizados para evaluar la usabilidad de una interfaz, como la consistencia, el control del usuario y la retroalimentación.

I

- **Iconografía:** Uso de iconos como elementos visuales en el diseño de la interfaz. Los iconos deben ser simples y representativos, proporcionando una pista visual para la navegación o acciones.
- **Iteración:** Proceso de refinamiento continuo en el diseño web, donde se realizan pruebas, se recogen comentarios y se mejoran los prototipos hasta obtener el resultado deseado.

J

- **Jerarquía visual:** Organización de los elementos en la página para mostrar su importancia relativa. Los elementos más importantes son visualmente más destacados, mediante el uso de tamaño, color o posición.

L

- **Ley de proximidad:** Principio del diseño que indica que los elementos cercanos entre sí se perciben como relacionados. Se usa para agrupar información y facilitar la comprensión de la interfaz.

M

- **Maqueta (Mockup):** Representación visual de fidelidad media del diseño de una página web, que muestra la apariencia de los elementos sin ser completamente funcional.
- **Mood board:** Herramienta de diseño que consiste en un collage de imágenes, colores y conceptos que ayudan a capturar la estética y el "look and feel" del proyecto.

N

- **Navegabilidad:** Capacidad del usuario para moverse por el sitio de manera intuitiva y efectiva, encontrando la información o funcionalidad que busca con facilidad.

P

- **Paleta de colores:** Conjunto de colores seleccionados para ser utilizados en un sitio web. Los colores se eligen de manera que combinen y creen una identidad visual coherente.
- **Prueba de usabilidad:** Evaluación del diseño mediante la observación de usuarios reales mientras interactúan con el prototipo o sitio web. Sirve para identificar problemas y mejorar la experiencia de usuario.
- **Prototipo:** Representación de alta fidelidad de un diseño que simula la experiencia de usuario final. Puede ser navegable y permite evaluar la interacción y el flujo de navegación.

R

- **Responsive design (Diseño responsivo):** Técnica de diseño web que busca adaptar el contenido a diferentes tamaños de pantalla y dispositivos, utilizando media queries y diseños fluidos.
- **Retroalimentación (Feedback):** Respuesta visual o sonora que proporciona la interfaz en respuesta a una acción del usuario, como un cambio de color al hacer clic en un botón.

S

- **SEO (Search Engine Optimization):** Optimización del contenido y la estructura del sitio web para mejorar su visibilidad en los motores de búsqueda.
- **Simplicidad:** Principio de diseño que busca reducir la complejidad de la interfaz para hacerla fácil de entender y usar. Consiste en eliminar elementos innecesarios y enfocarse en la funcionalidad esencial.

T

- **Tipografía:** Arte y técnica de organizar letras y texto. En diseño web, implica elegir fuentes, tamaños y estilos adecuados para asegurar la legibilidad y transmitir la identidad visual.
- **Testing:** Proceso de evaluación de la interfaz para identificar errores o problemas de usabilidad. Incluye pruebas con usuarios, pruebas automatizadas y revisiones de accesibilidad.

U

- **Usabilidad:** Medida de la facilidad y eficiencia con la que los usuarios pueden interactuar con la interfaz web. Una buena usabilidad mejora la satisfacción del usuario y reduce la tasa de abandono.
- **UX (User Experience):** Experiencia del usuario, que abarca todos los aspectos de la interacción del usuario con el sitio web. El objetivo es proporcionar una experiencia positiva y significativa.
- **UI (User Interface):** Interfaz de usuario, enfocada en los elementos visuales con los que el usuario interactúa, como botones, menús y formularios.

V

- **Visibilidad:** Principio de diseño que asegura que la información importante esté claramente visible y accesible para los usuarios. Mejora la experiencia de usuario al destacar acciones relevantes y contenido clave.

Tema 2: Prototipado y Guías de Estilo

El prototipado y la creación de guías de estilo son etapas fundamentales en el desarrollo web, ya que permiten definir la estructura visual y la coherencia del diseño antes de la implementación final. A continuación, se detallan los aspectos clave del proceso.

2.1 Prototipado como Referente Visual

El prototipado es el proceso de creación de representaciones preliminares de la interfaz web para evaluar la disposición de los elementos y la usabilidad. Permite explorar diferentes alternativas y detectar problemas antes de entrar en la fase de desarrollo.

2.1.1 Tipos de Prototipos

1. Esbozo:

- Representación inicial de baja fidelidad, generalmente realizado a mano.
- Sirve para capturar ideas rápidamente y realizar cambios sobre la marcha.

2. Wireframes:

- Esquemas de baja fidelidad que muestran la estructura básica de la página web.
- Incluyen la organización del contenido, la jerarquía de los elementos y los componentes interactivos.
- **Qué no contienen los wireframes:**
 - No se especifican colores, tipografías o gráficos complejos.
 - La interactividad avanzada y los efectos visuales no están presentes.

Herramientas para crear wireframes:

- **Balsamiq Wireframes:** Una herramienta sencilla para esbozos rápidos.
- **Moqups:** Plataforma web para wireframes colaborativos.
- **Diagrams.net (Draw.io):** Herramienta gratuita para diagramas y esquemas.

3. Wireflows:

- Extensión de los wireframes que muestra el flujo de navegación según las acciones del usuario.
- Se representan en forma de árbol de navegación, destacando las conexiones entre las diferentes pantallas.

4. Maquetas (Mockups):

- Representaciones de fidelidad media que incluyen una primera propuesta de diseño visual, con imágenes, colores y tipografías.
- No suelen ser interactivas, se utilizan para evaluar la estética y la disposición de los elementos.

5. Prototipos:

- Representaciones de alta fidelidad que simulan la experiencia de usuario final.
- Son navegables y permiten evaluar el flujo de navegación y la interactividad.
- El diseño visual es cercano al producto final, incluyendo todos los detalles gráficos.

Herramientas recomendadas para prototipos:

- **Sketch:** Herramienta estándar en la industria del diseño web.
- **Figma:** Ideal para trabajo colaborativo y diseño basado en la nube.
- **Marvel y Canva:** Plataformas que ofrecen plantillas y opciones para prototipos rápidos.

2.1.2 Beneficios del Prototipado

- **Detección temprana de problemas:** Permite identificar carencias en la estructura y la usabilidad.
- **Iteración rápida:** Se pueden hacer modificaciones de manera ágil en etapas tempranas del desarrollo.
- **Colaboración efectiva:** Facilita la comunicación entre diseñadores, desarrolladores y clientes.

2.2 Guías de Estilo

Las guías de estilo son documentos que recopilan las decisiones de diseño para asegurar la coherencia visual y funcional en un proyecto web. Establecen estándares que todos los diseñadores y desarrolladores deben seguir.

2.2.1 Componentes de una Guía de Estilo

1. Uso del Logo:

- Especificar el diseño del logo y sus variaciones para diferentes tamaños y fondos.
- Definir restricciones sobre su uso (colores permitidos, proporciones, etc.).

2. Paleta de Colores:

- La paleta debe incluir los colores primarios, secundarios y neutros que serán utilizados en el sitio.
- **Versiones claras y oscuras:** Mejora la accesibilidad y adapta la visualización a diferentes contextos (modo oscuro).
- **Combinaciones de colores aceptadas:** Evitar combinaciones que dificulten la lectura o causen fatiga visual.

3. Tipografía:

- Definir las familias tipográficas y jerarquías (tamaños y pesos) para encabezados, párrafos, botones y formularios.
- **Buenas prácticas:** Usar fuentes web seguras (ej. Arial, Verdana) o tipografías accesibles desde plataformas como Google Fonts.

4. Espaciado y Cuadrículas:

- Establecer las reglas para el espaciado entre elementos y la disposición en la cuadrícula.
- Garantizar la consistencia en márgenes, relleno y alineación de los elementos.

5. Iconografía:

- Decidir si se usarán iconos de bibliotecas comunes o personalizados.
- Definir reglas para el uso de iconos (monocromáticos, con relleno, tamaños estándar).

6. Ilustraciones e Imágenes:

- Especificar el estilo visual de las ilustraciones y cómo deben editarse las imágenes (filtros, contraste, sombras).
- Incluir pautas para seleccionar y utilizar imágenes de repositorios de fotos.

7. Elementos de la Interfaz:

- Definir el diseño de botones, campos de formularios, menús, cuadros de diálogo y otros elementos interactivos.
- **Estados de los elementos:** Normal, hover, activo y con foco (en el caso de los botones).

2.2.2 Importancia de la Guía de Estilo

- **Coherencia visual:** Garantiza que el sitio web mantenga una identidad visual consistente.
- **Facilita la colaboración:** Todos los miembros del equipo tienen un marco de referencia para seguir.
- **Escalabilidad del diseño:** Las guías de estilo permiten extender el diseño a nuevos componentes o secciones del sitio sin perder la consistencia.

2.2.3 Ejemplos y Recursos

- Muchas empresas hacen públicas sus guías de estilo para promover la familiarización con su marca. Por ejemplo, Guía de Diseño de Material de Google o [Ant Design de Alibaba](#).

2.2.4 Herramientas para Crear Guías de Estilo

Existen diversas herramientas para confeccionar guías de estilo, desde simples plantillas en HTML/CSS hasta plataformas específicas de diseño:

1. **HTML/CSS:** Crear la guía manualmente, lo que permite un alto grado de personalización.
2. **Software especializado:**
 - **Sketch y Figma:** Amplia variedad de plantillas para diseñar interfaces.
 - **Canva:** Herramienta web con plantillas predefinidas para crear guías de estilo rápidamente.

Plantillas HTML/CSS Recomendadas

- **Style Guide Template:** Una plantilla básica en HTML que incluye las opciones más comunes como logotipos, colores y tipografías.
- **Style Guide Boilerplate:** Alternativa que muestra patrones y estilos básicos para un proyecto web. Requiere un servidor con soporte PHP.

Enlaces de interés

- **Wireframes y herramientas:** [Wireflow](#)
- **Guías de estilo en Figma:** [Ejemplo de guía](#)
- **Plantillas HTML/CSS para guías de estilo:** [Style Guide Template](#)

Leyenda de Siglas, Elementos y Nomenclaturas en Prototipado y Guías de Estilo

A

- **Accesibilidad:** La práctica de diseñar interfaces web para que sean utilizables por personas con diferentes capacidades, incluyendo discapacidades visuales, auditivas o motoras. En prototipado y guías de estilo, implica garantizar que los elementos sean legibles y navegables para todos.
- **Alto nivel de fidelidad:** En prototipado, se refiere a maquetas o prototipos que presentan detalles muy cercanos a la apariencia final del producto, incluyendo interacciones y elementos visuales completos.

B

- **Baja fidelidad:** Prototipos simples que capturan la estructura básica de la interfaz sin detalles visuales complejos. Los bocetos en papel y wireframes son ejemplos comunes de baja fidelidad.
- **Boceto:** Representación inicial y simplificada de un diseño, realizada generalmente en papel o con herramientas digitales básicas, que muestra las ideas generales del layout y la disposición de los elementos.

C

- **Color primario:** Los colores dominantes en la paleta de colores de una guía de estilo, que se utilizan para representar la identidad visual de la marca.
- **Color secundario:** Colores adicionales que complementan el color primario y se usan para resaltar elementos secundarios o proporcionar contrastes.

D

- **Diseño colaborativo:** Proceso de creación en el que múltiples diseñadores y miembros del equipo contribuyen al desarrollo de un prototipo o guía de estilo, trabajando juntos para optimizar la interfaz.

E

- **Esbozo:** Una representación rápida y simple de un diseño, utilizada en las primeras etapas para explorar ideas de manera ágil. Suele ser de baja fidelidad y no incluye detalles visuales o gráficos.
- **Espaciado:** En la guía de estilo, el espaciado define la distancia entre elementos, incluyendo márgenes, rellenos y espacios entre líneas de texto. Es clave para crear una interfaz limpia y organizada.

F

- **Figma:** Herramienta popular de diseño y prototipado que permite la creación de interfaces y colaboraciones en tiempo real. Ideal para diseñadores que trabajan en equipo.

G

- **Guía de estilo:** Documento que recopila todas las decisiones de diseño visual de un proyecto web. Incluye la paleta de colores, tipografías, iconos, espaciado y otros aspectos clave para mantener la coherencia del diseño.

H

- **High-fidelity (Alta fidelidad):** Prototipos detallados y realistas que se acercan mucho al producto final en términos de apariencia y funcionalidad.

I

- **Iconografía:** Uso de iconos en el diseño de la interfaz para proporcionar pistas visuales o representar acciones. Las guías de estilo suelen incluir un conjunto específico de iconos para mantener la coherencia.
- **Iteración:** Proceso de repetición y refinamiento continuo en el diseño, donde se prueban y mejoran prototipos con base en comentarios y pruebas.

L

- **Layout:** Distribución o disposición de los elementos en la página web, incluyendo bloques de contenido, imágenes y botones.

M

- **Maqueta (Mockup):** Representación visual de fidelidad media o alta que muestra cómo se verá una página web. Incluye detalles gráficos y tipográficos, pero no suele ser interactiva.
- **Mood board:** Colección de imágenes, colores y conceptos visuales que representan la dirección estética de un proyecto. Se utiliza para comunicar la visión del diseño al equipo y al cliente.

N

- **Navegación:** Manera en la que los usuarios se desplazan por el sitio web. En el prototipado, se evalúan diferentes flujos de navegación para optimizar la experiencia del usuario.

P

- **Paleta de colores:** Conjunto de colores seleccionados para el diseño de la interfaz. Incluye colores primarios, secundarios, neutros y sus variantes, utilizados para crear una identidad visual coherente.
- **Prototipo:** Versión preliminar del diseño que permite probar la funcionalidad y la interacción de la interfaz. Los prototipos pueden ser de baja o alta fidelidad, dependiendo de la etapa del desarrollo.
- **Pruebas de usabilidad:** Evaluaciones que se realizan con usuarios reales para identificar problemas de diseño en un prototipo o producto. Los resultados ayudan a iterar y mejorar la interfaz.

R

- **Responsive design (Diseño responsivo):** Técnica de diseño web que permite que el contenido se adapte a diferentes dispositivos y tamaños de pantalla, garantizando una experiencia de usuario coherente.

S

- **Sketch:** Herramienta de diseño utilizada para crear wireframes, maquetas y prototipos de alta fidelidad. Es una de las herramientas más populares en la industria del diseño de interfaces.
- **Semántica:** En diseño web, se refiere a la utilización de etiquetas y componentes que tienen un significado específico para mejorar la accesibilidad y la estructura del contenido.
- **Simetría:** Técnica en diseño visual que implica distribuir elementos de manera equilibrada para lograr una apariencia armoniosa y ordenada.

T

- **Tipografía:** Conjunto de decisiones relacionadas con la elección de fuentes, tamaños, pesos y estilos de texto. En una guía de estilo, se define una jerarquía tipográfica para mantener la consistencia visual.

U

- **UI Kit:** Conjunto de componentes visuales predefinidos (botones, formularios, menús) que se utilizan en la creación de interfaces, permitiendo una implementación más rápida y coherente.
- **UX (User Experience):** Enfoque en la calidad de la interacción y la percepción del usuario con un sitio web. El objetivo es proporcionar una experiencia de usuario fluida y satisfactoria.

W

- **Wireframe:** Esquema de baja fidelidad que muestra la estructura y disposición de los elementos de una página web, sin incluir detalles visuales o estéticos. Se utiliza para planificar la arquitectura de la información y el flujo de navegación.
- **Wireflow:** Combinación de wireframes y flujos de navegación, que muestra cómo se conectan las diferentes pantallas y los caminos que puede tomar el usuario.

Z

- **Z-Index:** Aunque es un término más relacionado con CSS, se refiere a la capa de superposición de elementos en una interfaz. Los valores más altos significan que el elemento se mostrará por encima de otros con valores más bajos.

Tema 3: El Lenguaje HTML

HTML es un lenguaje de marcado que se utiliza para definir la estructura y el contenido de las páginas web. Su uso se complementa con tecnologías como CSS (para el diseño visual) y JavaScript (para la funcionalidad interactiva). Actualmente, la versión utilizada es **HTML5**, que introduce mejoras significativas para manejar multimedia, mejorar la semántica y optimizar la experiencia del usuario.

3.1 Elementos y Estructuras Básicas

HTML5 ha introducido numerosas etiquetas nuevas que permiten crear páginas más accesibles y organizadas. Las mejoras más relevantes incluyen:

- **Etiquetas multimedia:** `<audio>`, `<video>` y `<canvas>` para gestionar contenido multimedia sin necesidad de complementos externos.
- **Interactividad mejorada:** Soporte para arrastrar y soltar (drag and drop) y formularios avanzados.
- **Web semántica:** Nuevas etiquetas que ayudan a los motores de búsqueda y a los dispositivos de asistencia (como lectores de pantalla) a entender mejor el contenido.

3.1.1 Estructura de un Documento HTML

Un documento HTML tiene una estructura específica que incluye una declaración del tipo de documento, elementos de metadatos y contenido. La estructura típica es:

```
html
Copiar código
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Título de la Página</title>
</head>
<body>
  <header>
    <h1>Encabezado Principal</h1>
  </header>
  <main>
    <p>Este es un párrafo de ejemplo en el cuerpo principal del documento.</p>
  </main>
  <footer>
    <p>© 2024 Mi Sitio Web</p>
  </footer>
</body>
</html>
```

- `<head>` : Contiene metadatos, el título y enlaces a archivos externos (CSS, scripts).
- `<body>` : Incluye todo el contenido visible en la página, como textos, imágenes y enlaces.

3.1.2 Conceptos Básicos

1. **Atributos:** Proporcionan información adicional a las etiquetas. Por ejemplo, ``.
2. **Nidificación de elementos:** La organización de los elementos en HTML se hace colocando unas etiquetas dentro de otras, respetando un orden jerárquico.

3. **Elementos vacíos:** Algunos elementos, como `` o `
`, no contienen contenido y se autocierran.

3.2 Contenidos HTML

3.2.1 Encabezados y Párrafos

Los encabezados (`<h1>` a `<h6>`) se utilizan para marcar los títulos y subtítulos, organizando el contenido en niveles de importancia.

- **Buenas prácticas:**
 - Un solo `<h1>` por página para definir el tema principal.
 - Mantener la coherencia en el uso de niveles de encabezado para evitar confusión.
- **Uso de `<hgroup>`:** Agrupa varios encabezados, destacando el principal.

Los párrafos se definen con `<p>`, y deben ser utilizados para separar bloques de texto, lo que facilita la lectura y el mantenimiento del código.

3.2.2 Listas

- **Listas no ordenadas (``):** Los elementos no tienen un orden específico. Cada elemento de la lista se indica con ``.
- **Listas ordenadas (``):** Los elementos están numerados automáticamente.
- **Listas de definición (`<dl>`):** Forman pares de términos (`<dt>`) y descripciones (`<dd>`).

Ejemplo de lista anidada:

```
html
Copiar código
<ul>
  <li>Fruta
    <ul>
      <li>Manzana</li>
      <li>Plátano</li>
    </ul>
  </li>
  <li>Verdura</li>
</ul>
```

3.2.3 Otros Elementos de Texto

- **Formato básico:** Uso de ``, `<i>`, `<u>` para negritas, cursivas y subrayado.
- **Formato semántico:** `` y `` para enfatizar contenido, mejorando la accesibilidad y el SEO.

3.2.4 Imágenes

Para insertar imágenes, se utiliza la etiqueta ``, con atributos como `src` para la fuente y `alt` para la descripción:

```
html
Copiar código

```

Consideraciones importantes:

- **Tamaño de las imágenes:** Redimensionar adecuadamente para evitar cargas lentas.
- **Derechos de autor:** Asegurarse de tener los permisos necesarios para utilizar imágenes.

3.2.5 Hipervínculos

Los enlaces (`<a>`) son fundamentales para la navegación web. Ejemplo de uso:

```
html
Copiar código
<a href="https://ejemplo.com" target="_blank">Visita nuestro sitio</a>
```

3.2.6 Tablas

Las tablas se crean con etiquetas como `<table>`, `<tr>`, `<th>` y `<td>`. Ejemplo básico:

```
html
Copiar código
<table>
  <tr>
    <th>Nombre</th>
    <th>Edad</th>
  </tr>
  <tr>
    <td>Juan</td>
    <td>28</td>
  </tr>
</table>
```

3.2.7 Incrustación de Contenido

HTML permite incrustar multimedia como videos (`<video>`), audios (`<audio>`) y otros elementos interactivos (`<iframe>`).

3.3 Estructura Semántica

El uso de etiquetas semánticas en HTML5, como `<header>`, `<nav>`, `<article>`, y `<footer>`, mejora la accesibilidad y ayuda a los motores de búsqueda a entender mejor la estructura de la página.

Ejemplo de una página con estructura semántica:

```
html
Copiar código
<header>
  <h1>Título del Sitio</h1>
  <nav>
    <ul>
      <li><a href="#inicio">Inicio</a></li>
      <li><a href="#servicios">Servicios</a></li>
    </ul>
  </nav>
</header>
<main>
  <article>
    <h2>Artículo Principal</h2>
    <p>Contenido del artículo...</p>
  </article>
</main>
<footer>
  <p>Contacto: contacto@ejemplo.com</p>
</footer>
```

3.3.1 Validación de Documentos HTML

El W3C ofrece una herramienta para validar la sintaxis HTML y detectar errores: [W3C Validator](#).

3.4 Formularios HTML

Los formularios permiten recoger datos del usuario. El elemento `<form>` sirve como contenedor y debe incluir:

1. **Elemento** `<input>` : Para campos de texto, correo electrónico, contraseñas, etc.
 - Atributos comunes: `type`, `name`, `placeholder`.
2. **Elemento** `<select>` : Para crear menús desplegables con opciones `<option>`.
3. **Elemento** `<textarea>` : Campos de texto de varias líneas.
4. **Elemento** `<button>` : Para crear botones que envían el formulario o ejecutan scripts.

Ejemplo de formulario simple:

```
html
Copiar código
<form action="/enviar" method="post">
  <label for="nombre">Nombre:</label>
  <input type="text" id="nombre" name="nombre" required>
  <button type="submit">Enviar</button>
```

```
</form>
```

Legenda de Siglas, Elementos y Nomenclaturas en HTML

A

- **<a> (Anchor):** Elemento para crear enlaces de hipertexto. Utiliza el atributo `href` para especificar la URL de destino.

```
html
Copiar código
<a href="https://example.com">Enlace a Example</a>
```

- **<article>** : Representa un contenido independiente y autocontenido en un documento, como una entrada de blog o una noticia.
- **<aside>** : Define un contenido que está relacionado con el contenido principal, como barras laterales, citas o bloques informativos.
- **attribute (Atributo):** Proporciona información adicional sobre un elemento HTML. Los atributos se escriben dentro de la etiqueta de apertura y tienen un formato `nombre="valor"`, como en ``.

B

- **<body>** : Contenedor principal del contenido visible de una página web. Todo el contenido mostrado en la página se encuentra dentro de esta etiqueta.
- **<button>** : Crea un botón interactivo en la página. Se puede usar para enviar formularios o ejecutar funciones JavaScript.

C

- **<canvas>** : Permite dibujar gráficos en una página web mediante scripts, generalmente JavaScript. Se usa para crear gráficos, animaciones y juegos.
- **<class> (Clase):** Atributo utilizado para agrupar elementos que comparten un estilo común. Se define con un nombre de clase en el HTML y se estiliza en CSS.
- **<code>** : Representa un fragmento de código de computadora, estilizado de forma predeterminada para diferenciarlo del texto normal.
- **<col> y <colgroup>** : Se usan en tablas para aplicar estilos a columnas enteras. `<colgroup>` define un grupo de una o más columnas, y `<col>` estiliza columnas individuales.

D

- **<div>** : Elemento genérico de contenedor de bloque. Se utiliza para agrupar contenido y aplicar estilos o scripts específicos.
- **<dl>, <dt>, <dd>** : Etiquetas para listas de definición. `<dl>` es el contenedor principal, `<dt>` representa un término, y `<dd>` su definición.

```
html
Copiar código
<dl>
  <dt>HTML</dt>
  <dd>Lenguaje de marcado para crear páginas web.</dd>
</dl>
```

E

- **** : Indica un énfasis en el texto. Por defecto, el navegador lo muestra en cursiva. Tiene un significado semántico, ayudando a los lectores de pantalla a interpretar el contenido.
- **Elementos vacíos**: Elementos HTML que no tienen contenido, como **** o **
**. Se escriben solo con la etiqueta de apertura.

F

- **<form>** : Define un formulario para recoger datos de usuario. Puede incluir varios elementos de entrada como campos de texto, botones y menús desplegables.

```
html
Copiar código
<form action="/submit" method="post">
  <input type="text" name="nombre" placeholder="Escribe tu nombre">
  <button type="submit">Enviar</button>
</form>
```

H

- **<head>** : Contenedor para los metadatos del documento, como enlaces a hojas de estilo, scripts y el título del documento.
- **<header>** : Sección de encabezado que contiene títulos, navegación o logotipos. Puede usarse en la página principal o dentro de secciones como **<article>**.
- **<h1>** a **<h6>** : Etiquetas de encabezado, que van del nivel 1 al 6 en términos de importancia, siendo **<h1>** el más importante.
- **href (Hypertext Reference)**: Atributo que se utiliza en la etiqueta **<a>** para definir la URL de un enlace.

I

- **** : Inserta una imagen en la página web. Requiere los atributos **src** (fuente) y **alt** (texto alternativo).

```
html
Copiar código
```

```

```

- **<input>**: Crea un campo interactivo en un formulario. El atributo **type** define el tipo de entrada, como texto, contraseña o botón de envío.
- **id**: Atributo que identifica un elemento único en la página. Se utiliza para estilos específicos en CSS o para manipulación en JavaScript.

L

- **<label>**: Asociado a un elemento de formulario para mejorar la accesibilidad. Se usa para vincular un texto descriptivo con un campo de entrada.
- **** (**List Item**): Elemento de una lista, utilizado dentro de **** (lista no ordenada) o **** (lista ordenada).
- **<link>**: Se usa para enlazar recursos externos, como hojas de estilo CSS o iconos.

M

- **<meta>**: Proporciona metadatos sobre el documento, como el conjunto de caracteres o la configuración para dispositivos móviles.

```
html
Copiar código
<meta charset="UTF-8">
```

- **method**: Atributo de **<form>** que define cómo se envían los datos del formulario. Puede ser **get** o **post**.

N

- **<nav>**: Elemento que contiene enlaces de navegación. Normalmente se usa para definir un conjunto de enlaces a otras secciones del sitio.
- **name**: Atributo utilizado en elementos de formulario para identificar el campo cuando se envían los datos.

O

- **** (**Ordered List**): Crea una lista ordenada, donde cada elemento **** se numerará automáticamente.

P

- **<p>**: Representa un párrafo de texto.
- **<pre>** (**Preformatted Text**): Muestra el texto exactamente como se escribe, respetando los espacios en blanco y los saltos de línea.

Q

- `<q>` : Representa una cita breve en línea. Generalmente, los navegadores muestran el contenido de la cita entre comillas.

R

- `readonly` : Atributo para campos de entrada que impide al usuario modificar el valor, pero permite que se copie.

S

- `<script>` : Incrusta o enlaza scripts de programación (JavaScript) en la página web.
- `<section>` : Define una sección temática en la página. Es un contenedor semántico que agrupa contenido relacionado.
- `` : Elemento en línea que sirve para aplicar estilos específicos o manipular con scripts. Es similar a `<div>` , pero para contenido en línea.
- `src` (**Source**): Atributo usado en elementos como `` y `<script>` para definir la fuente del recurso.
- `submit` : Tipo de `<input>` o `<button>` que se usa para enviar un formulario.

T

- `<table>` : Crea una tabla en la página web. Utiliza etiquetas adicionales como `<tr>` para filas, `<td>` para celdas y `<th>` para encabezados de columna.
- `<textarea>` : Proporciona un campo de entrada de texto multilínea.
- `<title>` : Define el título del documento, que aparece en la pestaña del navegador.

U

- `` (**Unordered List**): Crea una lista no ordenada. Los elementos `` se marcan con viñetas.

V

- `<video>` : Incrusta un video en la página. Se puede especificar una fuente con el atributo `src` o con etiquetas `<source>` anidadas.

W

- `width` y `height` : Atributos que definen el ancho y alto de elementos como `` , `<canvas>` , `<video>` , entre otros.

Y

- `<yield>` : No es un elemento HTML estándar. Este término no se usa en HTML

Tema 4: Hojas de Estilo CSS

CSS (Cascading Style Sheets) es un lenguaje utilizado para controlar la presentación de documentos HTML. Permite definir cómo se muestran los elementos en una página web, separando el contenido (HTML) de la presentación (CSS). Las ventajas principales incluyen:

- **Accesibilidad:** Mejora la experiencia del usuario al hacer que el contenido sea más fácil de usar.
- **Flexibilidad:** Permite adaptar el diseño para diferentes dispositivos y tamaños de pantalla.
- **Compatibilidad:** CSS es un estándar soportado por todos los navegadores modernos.
- **Reutilización:** Facilita el mantenimiento y la gestión de estilos mediante archivos CSS externos.

4.1 Inclusión de CSS en un Documento HTML

Hay varias maneras de incluir CSS en un documento HTML:

1. **CSS en línea:** Se aplica directamente en el atributo `style` de un elemento HTML.

```
html
Copiar código
<p style="color: blue; font-size: 16px;">Texto en azul</p>
```

2. **CSS interno:** Se utiliza la etiqueta `<style>` dentro del `<head>` del documento.

```
html
Copiar código
<head>
  <style>
    body { background-color: lightgrey; }
    h1 { color: blue; }
  </style>
</head>
```

3. **CSS externo:** Se enlaza un archivo CSS separado usando la etiqueta `<link>`.

```
html
Copiar código
<head>
  <link rel="stylesheet" href="styles.css">
</head>
```

Recomendación: Utilizar CSS externo para separar completamente la presentación del contenido y mejorar el rendimiento.

4.2 Estructura Básica de CSS

Una regla CSS está compuesta por:

- **Selector:** Indica el elemento al que se aplicará el estilo.

- **Declaración:** Define las propiedades y valores para aplicar al selector. Las declaraciones están compuestas por pares `propiedad: valor`.

Ejemplo:

```
css
Copiar código
p {
  color: red;
  font-size: 14px;
}
```

Tipos de Selectores

1. **Selector de tipo:** Aplica estilos a todas las etiquetas de un tipo específico, como `p` o `h1`.
2. **Selector de clase:** Utiliza un nombre de clase para aplicar estilos a elementos que tienen el atributo `class`. Se define con un punto (`.`).

```
css
Copiar código
.miClase {
  color: green;
}
```

3. **Selector de ID:** Selecciona un elemento por su atributo `id`, utilizando un símbolo de almohadilla (`#`). Solo se aplica a un único elemento.

```
css
Copiar código
#miID {
  background-color: yellow;
}
```

4. **Selectores combinados y descendientes:** Permiten aplicar estilos a elementos según su contexto jerárquico.

```
css
Copiar código
div p {
  color: blue; /* Aplica estilo a los <p> dentro de un <div> */
}
```

5. **Pseudoclases y Pseudoelementos:**

- **Pseudoclases:** Aplican estilos a un estado específico de un elemento, como `:hover` o `:focus`.
- **Pseudoelementos:** Permiten estilizar una parte específica de un elemento, como `::before` o `::after`.

4.3 Propiedades CSS

4.3.1 Propiedades de Estilo de Texto y Fuentes

- **Tipografía:**
 - `font-family`: Define la fuente a utilizar. Ejemplo: `font-family: Arial, sans-serif;`.
 - `font-size`: Controla el tamaño de la fuente, con unidades como `px`, `em` o `%`.
 - `font-weight`: Establece el grosor del texto, con valores como `normal`, `bold` o números (`100` a `900`).
 - `color`: Cambia el color del texto. Se puede especificar por nombre (`red`), en formato RGB (`rgb(255,0,0)`), hexadecimal (`#ff0000`), o HSL (`hsl(0, 100%, 50%)`).

4.3.2 Unidades de Medida

Las unidades en CSS pueden ser absolutas o relativas:

- **Unidades absolutas:** `px`, `cm`, `mm`.
- **Unidades relativas:** `em`, `rem`, `%`, que se calculan en función de otro valor, como el tamaño de fuente del elemento padre.

También existen unidades relativas al viewport, como `vw` (ancho del viewport) y `vh` (alto del viewport), que se adaptan a las dimensiones de la pantalla.

4.4 El Modelo de Caja en CSS

En CSS, todos los elementos se tratan como cajas que constan de cuatro componentes principales:

1. **Contenido:** El área que muestra el texto o la imagen.
2. **Relleno (`padding`):** Espacio entre el contenido y el borde.
3. **Borde (`border`):** El contorno de la caja.
4. **Margen (`margin`):** Espacio fuera del borde que separa el elemento de otros.

Las propiedades `width` y `height` determinan el tamaño del contenido, pero el tamaño total de la caja también incluye el padding y el borde.

Ejemplo:

```
css
Copiar código
div {
  width: 200px;
  padding: 10px;
  border: 1px solid black;
```



```
margin: 20px;  
}
```

Tipos de Elementos HTML

1. **Elementos de bloque:** Ocupan todo el ancho disponible y empiezan en una nueva línea (`<div>`, `<p>`).
2. **Elementos en línea:** Ocupan solo el espacio necesario y no interrumpen el flujo del texto (``, `<a>`).

4.5 Posicionamiento de Elementos

CSS ofrece varias técnicas para controlar la posición de los elementos:

1. **Posicionamiento Estático (`static`):** Es el valor predeterminado. Los elementos siguen el flujo normal del documento.
2. **Posicionamiento Relativo (`relative`):** Desplaza el elemento en relación con su posición original.
3. **Posicionamiento Absoluto (`absolute`):** El elemento se posiciona en relación al contenedor más cercano con `position` distinto de `static`.
4. **Posicionamiento Fijo (`fixed`):** El elemento permanece en una posición fija en la ventana, incluso al hacer scroll.
5. **Posicionamiento Adhesivo (`sticky`):** Cambia de `relative` a `fixed` según el desplazamiento del usuario.

Flotación (`float`) y Limpieza (`clear`)

- `float`: Permite que un elemento flote a la izquierda o derecha, con el contenido circundante fluyendo alrededor.
- `clear`: Se utiliza para evitar que los elementos floten a su alrededor.

4.6 Selectores Avanzados y Prioridad

- **Selectores de Atributos:** Seleccionan elementos en función de sus atributos (`[type="text"]`).
- **Selectores de Hijos (`>`), Adjuntos (`+`), y Generales (`-`):** Permiten aplicar estilos en función de la relación jerárquica entre elementos.

Especificidad y Herencia

- **Especificidad:** Determina cuál estilo se aplica cuando varias reglas afectan al mismo elemento. Depende del tipo de selectores utilizados.
- **Herencia:** Algunas propiedades CSS se heredan de los elementos padres, como el color o la fuente.

4.7 Validación y Herramientas

- **Validadores:** El W3C ofrece herramientas de validación de CSS para comprobar la corrección del código.
- **Extensiones y herramientas de desarrollo:** Programas como Visual Studio Code permiten verificar la sintaxis y encontrar errores.

Legenda de Siglas, Elementos y Nomenclaturas en CSS

A

- **absolute:** Tipo de posicionamiento en el que el elemento se coloca en relación con el contenedor más cercano que no tenga `position: static`. Se usa para crear diseños complejos y superponer elementos.
- **align-items:** Propiedad de Flexbox que alinea los elementos secundarios dentro del contenedor en el eje transversal (vertical por defecto). Valores comunes: `flex-start`, `flex-end`, `center`, `baseline`, `stretch`.
- **animation:** Define las animaciones para un elemento. Se especifican los pasos y la duración para crear efectos de transición en el diseño.

B

- **background-color:** Establece el color de fondo de un elemento. Puede usar nombres de colores, valores RGB, hexadecimales o HSL.
- **background-image:** Define una imagen de fondo para un elemento. Se puede utilizar una URL para cargar la imagen desde un recurso externo.
- **border:** Especifica el ancho, estilo y color del borde de un elemento. Ejemplo: `border: 1px solid black;`.
- **border-radius:** Redondea las esquinas de los bordes de un elemento. Valores más grandes crean bordes más redondeados.

C

- **clear:** Se utiliza para controlar el comportamiento de los elementos flotantes (`float`). Puede tener valores como `left`, `right` o `both`.
- **color:** Define el color del texto de un elemento. Se puede especificar en diferentes formatos como nombres de color (`red`), RGB (`rgb(255,0,0)`), hexadecimal (`#ff0000`), o HSL (`hsl(0, 100%, 50%)`).
- **content:** Se utiliza en pseudoelementos (`::before` y `::after`) para generar contenido que no está en el HTML. Ejemplo: `content: "Nota: ";`.
- **cursor:** Cambia el cursor del ratón cuando pasa sobre un elemento. Ejemplos: `pointer`, `default`, `text`.

D

- **display:** Controla cómo se presenta un elemento en la página. Valores comunes:
 - `block`: Ocupa todo el ancho disponible.
 - `inline`: Ocupa solo el espacio necesario.

- `none` : Oculta el elemento.

F

- **flex**: Propiedad de Flexbox que define cómo crecen, se reducen o el tamaño de los elementos secundarios dentro de un contenedor flex.
- **flex-direction**: Establece la dirección de los elementos secundarios en un contenedor Flexbox. Puede ser `row`, `row-reverse`, `column`, o `column-reverse`.
- **float**: Permite que un elemento flote a la izquierda o derecha dentro de su contenedor, y los elementos circundantes fluirán alrededor.
- **font-family**: Define la fuente a usar para un elemento de texto. Se puede especificar una lista de fuentes, en la que el navegador seleccionará la primera que esté disponible.
- **font-size**: Establece el tamaño del texto. Puede usar unidades absolutas (`px`, `pt`) o relativas (`em`, `%`).
- **font-weight**: Especifica el grosor de la fuente. Valores comunes son `normal`, `bold`, y números entre `100` y `900`.

H

- **height**: Define la altura de un elemento. Puede ser una unidad absoluta (`px`) o relativa (`%`).
- **hover**: Pseudoclase que aplica estilos a un elemento cuando el usuario pasa el ratón por encima.

I

- **inherit**: Valor que indica que una propiedad CSS debe ser heredada del elemento padre.

J

- **justify-content**: Propiedad de Flexbox que alinea los elementos secundarios a lo largo del eje principal (horizontal por defecto). Valores comunes: `flex-start`, `flex-end`, `center`, `space-between`, `space-around`.

L

- **left**: Se usa en posicionamiento absoluto o relativo para colocar un elemento en relación con el borde izquierdo de su contenedor.
- **line-height**: Controla la altura de la línea en elementos de texto. Puede afectar el espacio entre líneas de texto.

M

- **margin**: Espacio exterior que rodea a un elemento. Puede especificarse en los cuatro lados (`margin-top`, `margin-right`, `margin-bottom`, `margin-left`).
- **max-width / min-width**: Establece el ancho máximo o mínimo que puede tener un elemento.

O

- **opacity:** Define la transparencia de un elemento. Un valor de `1` significa completamente opaco, y `0` es completamente transparente.
- **overflow:** Controla qué sucede si el contenido de un elemento es más grande que el área del contenedor. Valores comunes: `visible`, `hidden`, `scroll`, `auto`.

P

- **padding:** Espacio interior entre el contenido de un elemento y su borde. Similar a `margin`, pero afecta el espacio dentro del borde.
- **position:** Define el tipo de posicionamiento para un elemento. Valores comunes:
 - `static`: Posición predeterminada.
 - `relative`: Posición relativa a su ubicación original.
 - `absolute`: Posición en relación al contenedor más cercano.
 - `fixed`: Posición fija en la pantalla.

R

- **right:** Se usa en posicionamiento absoluto o relativo para colocar un elemento en relación con el borde derecho de su contenedor.
- **rem:** Unidad de medida relativa basada en el tamaño de la fuente raíz (`<html>`).

T

- **text-align:** Alinea el texto dentro de un elemento. Valores comunes son `left`, `right`, `center`, `justify`.
- **text-decoration:** Controla los efectos decorativos sobre el texto, como `underline` (subrayado) o `line-through` (tachado).
- **top:** Se usa en posicionamiento absoluto o relativo para colocar un elemento en relación con el borde superior de su contenedor.

V

- **visibility:** Determina si un elemento es visible o no. Los valores son `visible` (predeterminado) y `hidden`.
- **vertical-align:** Alinea elementos en línea o elementos de bloque con `display: inline-block`.
- **vh / vw:** Unidades de medida relativas al viewport (ventana del navegador). `vh` es el 1% de la altura del viewport, y `vw` es el 1% del ancho.

W

- **width:** Establece el ancho de un elemento.
- **white-space:** Controla cómo se maneja el espacio en blanco en el contenido. Valores: `normal`, `nowrap`, `pre`.
- **word-wrap:** Permite que las palabras se dividan y ajusten dentro del contenedor.

Z

- **z-index:** Controla la superposición de elementos. Los valores más altos significan que el elemento se mostrará por encima de otros con valores más bajos.