

PROGRAMA DE ESTUDIOS

DESARROLLO DE SISTEMAS DE INFORMACIÓN

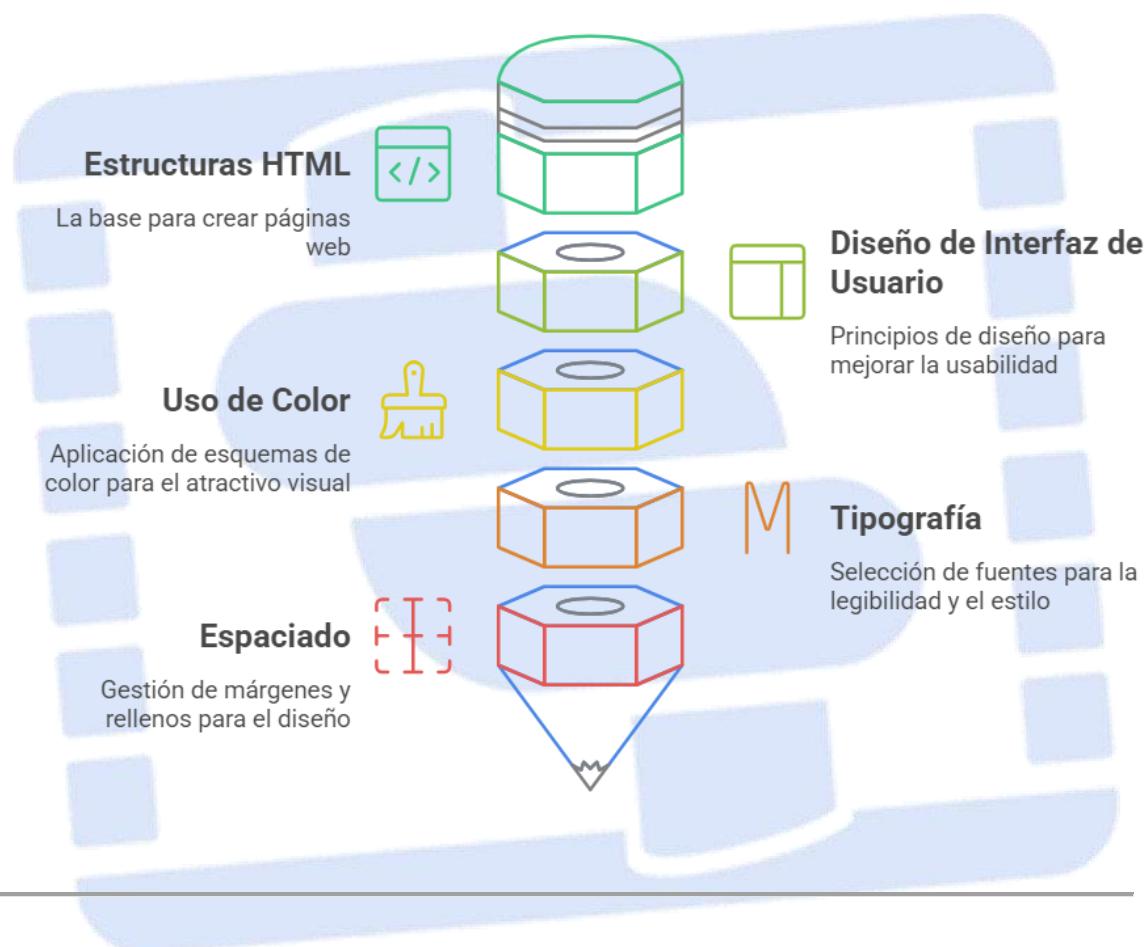
**COMUNITY MANAGER Y
MARKETING DIGITAL**

Tema

**INTRODUCCIÓN A CSS3 Y DISEÑO DE
INTERFACES**

¡Error! No se encuentra el origen de la referencia. En esta sesión se explorarán los fundamentos de **CSS3**, un pilar esencial en el desarrollo Front-End, que permite transformar estructuras HTML básicas en interfaces visualmente atractivas y funcionales. Además, se abordarán conceptos clave de **diseño de interfaces**, enfocados en el uso de colores, tipografías y espaciados para mejorar la experiencia del usuario.

Mejorando el Desarrollo Front-End con CSS3



¿Qué es CSS3 y para qué se utiliza?

CSS3 (Cascading Style Sheets, Nivel 3) es el lenguaje de hojas de estilo utilizado para describir la presentación de un documento HTML. CSS3 define cómo se mostrarán los elementos en pantalla, incluyendo aspectos como colores, fuentes, tamaños, márgenes, bordes, alineación, animaciones y mucho más.

¿Para qué se utiliza CSS3?

- **Estilizar contenido HTML:** Controla la apariencia visual de textos, imágenes, botones y otros elementos.
- **Diseño adaptable (responsive design):** Permite crear interfaces que se ajustan a diferentes tamaños de pantalla (móviles, tabletas, PC).
- **Animaciones y transiciones:** Facilita la creación de efectos visuales sin necesidad de JavaScript.
- **Separación de contenido y diseño:** Mejora la organización del código, permitiendo mantener el HTML limpio y enfocado en la estructura, mientras que el CSS gestiona el estilo.



Estructura básica de una regla CSS

Una **regla CSS** está compuesta por tres partes principales: **selector**, **propiedad** y **valor**.

↗ Sintaxis básica:



```
selector {  
    propiedad: valor;  
}
```

- **Selector:** Indica a qué elemento(s) HTML se aplicará el estilo.
- **Propiedad:** Especifica la característica del estilo que se desea modificar (como color, tamaño de fuente, etc.).
- **Valor:** Define cómo se aplicará la propiedad, por ejemplo, un color específico o un tamaño de fuente.

 Ejemplo simple:



```
h1 {  
    color: blue;  
    font-size: 24px;  
}
```

En este caso:

- **Selector:** `h1` (aplicará los estilos a todos los encabezados de nivel 1).
- **Propiedades:** `color` y `font-size`.
- **Valores:** `blue` para el color del texto y `24px` para el tamaño de la fuente.

Selectores en CSS3

Los **selectores** permiten identificar y aplicar estilos a uno o varios elementos HTML. Existen varios tipos de selectores básicos:

Selector de elemento (tag selector): Aplica estilos a todas las instancias de un tipo de etiqueta HTML.

```
/*Este ejemplo aplica el color negro a todos los párrafos (<p>)*/
p {
    color: black;
}
```

Selector de clase (.clase): Permite aplicar estilos a uno o varios elementos que comparten la misma clase.

```
/* Se aplicará un fondo amarillo a cualquier elemento que tenga
class="highlight"*/
.highlight {
    background-color: yellow;
}
```

Selector de ID (#id): Aplica estilos a un único elemento identificado por su atributo `id`.

```
/*Solo el elemento con id="titulo-principal" tendrá este tamaño de fuente.*/
#titulo-principal {
    font-size: 32px;
}
```

Propiedades y Valores en CSS3

Las **propiedades** definen el aspecto visual de los elementos, mientras que los **valores** determinan cómo se aplican esas propiedades. Algunos ejemplos comunes incluyen:

- **color:** Define el color del texto.



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

- **font-size**: Ajusta el tamaño de la fuente.



```
h1 {  
    font-size: 28px;  
}
```

- **background-color**: Establece el color de fondo de un elemento.



```
div {  
    background-color: lightgray;  
}
```

- **margin y padding**: Controlan el espacio exterior (margin) e interior (padding) del contenido.



```
.box {  
    margin: 20px;  
    padding: 10px;  
}
```

Formas de Aplicar CSS

Existen tres formas principales de aplicar CSS a un documento HTML:

¿Qué método de CSS se debe usar?

Estilos en línea

Adecuado para cambios rápidos y específicos en un solo elemento.

Estilos internos

Ideal para documentos HTML únicos con estilos coherentes.

Estilos externos

Mejor para sitios web grandes con múltiples páginas que requieren estilos consistentes.

1 Estilos en línea (Inline Styles)

Se aplican directamente sobre el elemento HTML utilizando el atributo `style`.

```
● ● ●  
<p style = "color: green; font-size: 16px;">Texto en verde</p>
```

- **Ventajas:** Fácil para cambios rápidos.
- **Desventajas:** Difícil de mantener en proyectos grandes.

2 Estilos internos (Internal CSS)

Se definen dentro de una etiqueta `<style>` ubicada en la sección `<head>` del documento HTML.

```
● ● ●  
<head>  
  <style>  
    h1 { color: navy; }  
    p { font-family: Arial; }  
  </style>  
</head>
```

- **Ventajas:** Útil para estilos específicos de una sola página.
- **Desventajas:** No es óptimo para proyectos con muchas páginas.

3 Estilos externos (External CSS)

Se crean en un archivo separado con la extensión `.css`, que luego se enlaza al HTML mediante la etiqueta `<link>`.



```
<head>
  <link rel="stylesheet" href="estilos.css">
</head>
```

- **Ventajas:** Facilita la gestión de estilos en proyectos grandes, permitiendo reutilizar la misma hoja de estilos en múltiples páginas.
- **Desventajas:** Depende de la correcta carga del archivo externo.

Conceptos Básicos de Diseño de Interfaces

El **diseño de interfaces (UI)** busca mejorar la apariencia y la experiencia del usuario al interactuar con un sistema. Algunos principios clave incluyen:

🎨 Uso de Colores

- Los colores comunican emociones y mejoran la legibilidad.
- Se recomienda utilizar una paleta de colores coherente para mantener la identidad visual.
- Considerar el contraste para garantizar la accesibilidad.

✍ Tipografías

- La **legibilidad** es fundamental: elegir fuentes claras y fáciles de leer.
- Combinar máximo 2-3 tipos de fuentes para mantener un diseño limpio y profesional.
- Controlar el tamaño, peso (negrita), interlineado y espaciado entre letras.

Espaciado (Spacing)

- El **espaciado adecuado** entre elementos mejora la organización visual y reduce la sobrecarga cognitiva del usuario.
- Uso de **márgenes** (espacio externo) y **relleno** (padding) para separar secciones y destacar contenido importante.

Conclusión

En esta sesión, hemos aprendido que **CSS3** es la herramienta clave para transformar una estructura HTML básica en una interfaz visualmente atractiva y funcional. Conocer la estructura de las reglas CSS, los distintos selectores y formas de aplicar estilos es esencial para diseñar páginas web eficientes. Además, la comprensión de principios básicos de diseño de interfaces, como el uso de colores, tipografías y espaciado, contribuye a mejorar significativamente la experiencia del usuario.

En la siguiente sesión, profundizaremos en técnicas avanzadas de CSS, incluyendo el diseño responsive y la creación de layouts flexibles.



INSTITUTO
KHIPU