

# ITSQMET

INSTITUTO TECNOLÓGICO SUPERIOR QUITO METROPOLITANO





### Desarrollo de Aplicaciones de Internet

Ing. Julio Rosero







### Introducción a la clase

- Manejo de bordes CSS
- Manejo de Fuentes e Íconos con CSS
- Grid layout





#### Objetivos de la clase:

- Conocer las diferentes propiedades de borde disponibles en CSS.
- Aprender a establecer el ancho, el estilo y el color de los bordes.
- Aprender a incorporar íconos utilizando fuentes de íconos, como Font Awesome o Material Icons.





### MANEJO DE BORDES CSS





- Los bordes juegan un papel importante en el diseño web.
- Con CSS, podemos controlar el ancho, estilo y color de los bordes.
- En esta clase, aprenderemos a manejar los bordes con CSS para crear diseños atractivos y personalizados.





- La propiedad border permite establecer los atributos de los bordes de un elemento.
- Podemos definir el ancho, estilo y color del borde en una sola declaración.

```
.elemento {
border: 1px solid #000;
}
```





- Existe una gran cantidad de efectos que pueden ser aplicados en bordes, la siguiente dirección presenta una hoja de documentación con diferentes características referentes a bordes:
- https://developer.mozilla.org/en-US/docs/Web/CSS/border







## MANEJO DE FUENTES E ÍCONOS EN CSS





Las fuentes y los iconos juegan un papel fundamental en el diseño web.

Mediante el uso de CSS, se pueden gestionar las fuentes de texto y utilizar íconos para enriquecer las experiencias visuales.

CSS permite controlar el tipo de fuente, tamaño, peso y estilo del texto.

Se utilizan propiedades como font-family, font-size, font-weight y font-style para gestionar las fuentes.





#### Manejo de Fuentes

```
.elemento {
  font-family: Arial, sans-serif;
  font-size: 16px;
  font-weight: bold;
  font-style: italic;
}
```





#### **Fuentes personalizadas**

Es posible incorporar fuentes personalizadas mediante el uso de @font-face.

Esta técnica permite utilizar fuentes que no están presentes en la mayoría de los sistemas.

Se requieren archivos de fuente en formatos como .ttf, .woff o .woff2.





#### **Fuentes personalizadas**

```
@font-face {
    font-family: 'MiFuentePersonalizada';
    src: url('ruta-de-la-fuente/mifuente.woff2') format('woff2'),
        url('ruta-de-la-fuente/mifuente.ttf') format('truetype');
}
.elemento { font-family: 'MiFuentePersonalizada', sans-serif;}
```





Los iconos son elementos gráficos que representan acciones o conceptos de manera visual.

Es posible utilizar fuentes de íconos o imágenes de íconos para incorporarlos en los diseños.







































Se pueden utilizar fuentes de íconos populares, como Font Awesome o Material Icons.

Estas fuentes ofrecen una amplia variedad de íconos listos para utilizar en los proyectos.

Solo es necesario agregar la fuente y asignar la clase correspondiente al elemento.





Para lograr dicha implementación es necesarios incluir las hojas de estilos del repositorio a usar

<link rel="stylesheet" href="estilos.css">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/fontawesome/5.15.3/css/all.min.css">





Cada ícono se mostrará con la etiqueta <i>

```
<div class="iconos">
     <i class="fas fa-heart"></i>
     <i class="fas fa-star"></i>
     <i class="fas fa-envelope"></i>
</div>
```





Por último en la hoja de estilos .css, se procede a dar características de los íconos como:

- Tamaño
- Margen, etc

```
.container {
 text-align: center;
 margin-top: 100px;
.iconos {
 font-size: 2rem;
.iconos i {
 margin: 10px;
```





Algunos ejemplos de estos repositorios de íconos los podemos encontrar en los siguientes enlaces:

- Font Awesome: <a href="https://fontawesome.com">https://fontawesome.com</a>
- Material Icons: <a href="https://fonts.google.com/icons">https://fonts.google.com/icons</a>
- Ionicons: <a href="https://ionicons.com">https://ionicons.com</a>
- Feather Icons: https://feathericons.com





### GRID LAYOUT







 Grid Layout es un sistema de diseño en CSS (Cascading Style Sheets) que permite crear diseños de cuadrícula en una página web.

 Grid Layout ofrece un enfoque más poderoso y flexible para organizar y distribuir elementos en una página.





#### ¿Cómo funciona Grid Layout?

- Grid Layout se basa en la creación de una cuadrícula bidimensional compuesta por filas y columnas.
- Cada elemento HTML puede colocarse dentro de una celda de la cuadrícula, permitiendo un control preciso sobre su posición y tamaño.





#### **Ejemplo**

```
CSS
.grid-container {
  display: grid;
  grid-template-columns: 1fr 1fr 1fr;
  grid-gap: 10px;
.item {
  background-color: #eaeaea; padding:
  20px; text-align: center;
```





# **PRÁCTICA**









### GRACIAS



