



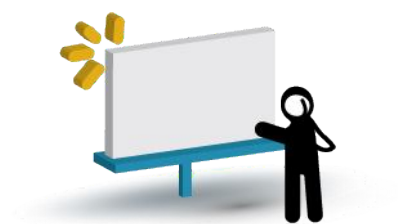
UTN.BA FACULTAD
REGIONAL
BUENOS AIRES
SECRETARÍA DE EXTENSIÓN UNIVERSITARIA FRBA UTN

**Centro de
e-Learning**

Programador Web Inicial Front- End Developer



www.sceu.frba.utn.edu.ar/e-learning



Presentación de la Unidad:

En las unidades anteriores, pudimos experimentar nuestro primer acercamiento a CSS: algunas de sus propiedades, la forma de vincularlo y cómo a través de este lenguaje podemos darle algunos estilos al sitio.

En esta unidad, profundizaremos un poco más en este mundo del estilo. Aprenderemos más formas de ubicar elementos y cómo poder tener un poco más de control sobre los elementos y sus estilos, lo que nos permitirá usar la propiedad adecuada para lograr un estilo en particular y no tener que recurrir a combinaciones complejas.



Objetivo:

Unidad 2: Que el alumno logre aplicar las propiedades de estilos CSS a la estructura del sitio.



BLOQUES TEMÁTICOS:

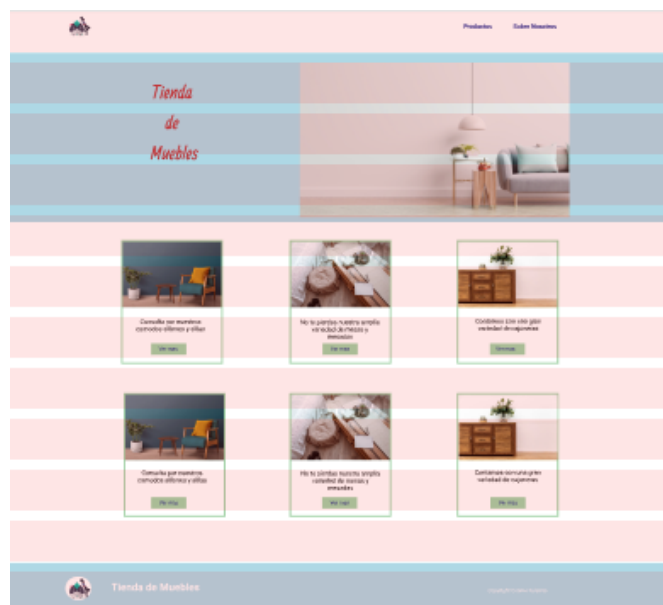
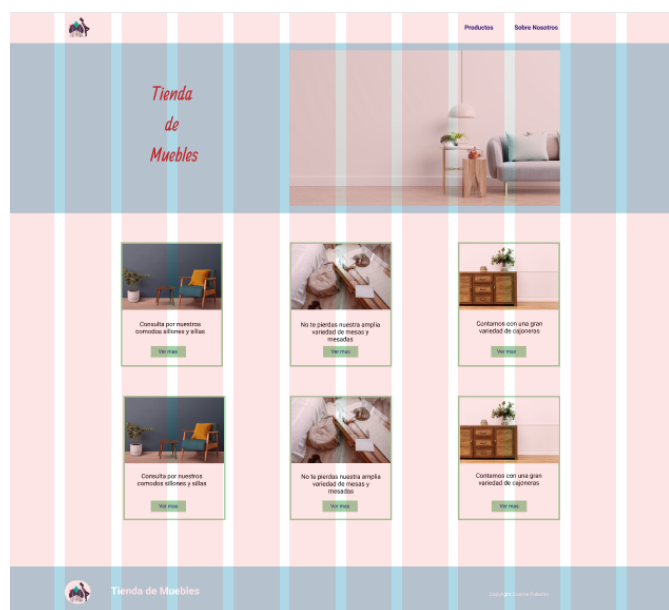
1. Grid: Ubicando elementos en forma de grillas
2. Más propiedades CSS:
 - 2.1- Profundizando en background
 - 2.2- Sombreado
 - 2.3- Degradado de color
3. Pseudoclases y Pseudoelementos
4. En Resumen
5. Bibliografía

1. Grid: Ubicando elementos en forma de grillas

En las anteriores unidades, pudimos ver que no hay una única propiedad que nos permite ubicar elementos. Tener varias opciones nos permite no solamente utilizar la que más nos guste, sino también poder combinar estas propiedades en el sitio, utilizando la más conveniente para cada elemento.

Siguiendo este lineamiento, vamos a agregar una manera más para definir el posicionamiento de elementos: **Grid**.

Grid es una propiedad de CSS que nos sirve para posicionar elementos en ambos ejes (vertical y horizontal) haciendo que la pantalla se comporte como una grilla y tomando como parámetros filas y columnas en donde ubicar a cada componente. En la siguiente imagen, podemos ver una forma gráfica donde “fraccionamos” la pantalla en filas y columnas para ir ubicando los elementos.





Pero... ¿cómo hacemos para utilizar Grid?

Para utilizarlo, primero necesitamos invocarlo utilizando la propiedad **display:grid** en el elemento padre; es decir, en el elemento que contiene a aquellos elementos a ubicar. Con eso, podemos acceder a todas las propiedades de Grid, tanto en el elemento padre como en los elementos hijos.

Propiedades de Grid y lo que nos permiten hacer

- **grid-template-columns:** Esta propiedad establece la cantidad y el ancho de las columnas en que se va a “dividir” el elemento padre. Puede utilizar cualquier unidad de medida para definirlo (% , px, vh, vw) e incluso podemos utilizar una unidad que nos presenta grid, llamada **fr** (fracción).
La unidad **fr** representa una fracción del espacio disponible en el contenedor de la cuadrícula, es decir que se expanden y se encogen de acuerdo al espacio disponible.
- **grid-template-rows:** Esta propiedad establece la cantidad y el ancho de las filas en que se va a “dividir” el elemento padre. Al igual que **grid-template-columns**, podemos utilizar cualquier unidad de medida para definirlo (% , px,vh,vw) e incluso también la unidad **fr**.
- **grid-column:** Esta propiedad define qué columnas ocupa cada elemento de la grilla. Para ello, utilizamos números enteros donde enumeramos las columnas de izquierda a derecha.
- **grid-row:** Esta propiedad define qué filas ocupa cada elemento de la grilla. Para ello, utilizamos números enteros donde enumeramos las filas de izquierda a derecha.
- **grid-area:** Esta propiedad “unifica” las propiedades **grid-column** / **grid-row**.
- **grid-template:** Esta propiedad “unifica” las propiedades **grid-template-column** / **grid-template-row**.

2. Más propiedades CSS

En las unidades anteriores, pudimos ver algunas propiedades generales de CSS que nos permitían darle algunos estilos a los elementos. En este punto, estaremos viendo unas propiedades un poco más específicas que nos van a permitir dar estilos más controlados a los elementos. ¡Empecemos!

2.1- Profundizando en background

Cuando hablamos de background, aprendimos que tenemos la posibilidad de establecer una imagen de fondo o incluso un color. Pero eso no es lo único que podemos hacer:

- **background-size:** Esta propiedad nos permite indicar el tamaño de la imagen. Puede tomar los valores de unidad que se desee (px, porcentaje, rem, em) y también puede tomar los siguientes valores preestablecidos:
 - **cover:** escala el tamaño de la imagen al mayor tamaño posible para que ocupe el total del contenedor. Si la imagen es mayor al contenedor, la recorta vertical u horizontalmente para que no quede un espacio vacío.
 - **contain:** escala la imagen al mayor tamaño posible sin recortarla ni estirla.
- **background-position:** Esta propiedad nos permite definir la posición inicial de la imagen de fondo que especificamos con **background-image** o **background**. Puede tomar los valores de unidad, porcentajes e incluso medidas como centímetros. A su vez, nos permite indicar valores específicos de coordenadas (top-left-right-bottom).
- **background-attachment:** Esta propiedad nos permite definir si la posición de la imagen de fondo será fija o se desplazará con su bloque contenedor. Puede tomar los siguientes valores:
 - **scroll:** nos permite hacer que la imagen de fondo se mueva dentro de la pantalla junto al bloque que la contiene.
 - **fixed:** nos permite definir que la imagen de fondo quede fija en la pantalla y no se mueva con el bloque contenedor.



2.2- Sombreado

Para poder darle propiedades de sombreado a los elementos, tenemos la propiedad **box-shadow**. Esta propiedad le da un sombreado alrededor del marco del elemento y podemos determinar el ancho del difuminado, su color y el desplazamiento del mismo. Para ello, tenemos que colocar la definición de estas características de la siguiente forma:

box-shadow: [posX] [posY] [desenfoque] [tamaño]
[color];

posX: colocamos el valor en unidades (px o porcentaje) para indicar la sombra que se aplica en el eje horizontal.

posY: colocamos el valor en unidades (px o porcentaje) para indicar la sombra que se aplica en el eje vertical.

desenfoque: colocamos la cantidad de sombra borrosa.

tamaño: colocamos el tamaño de sombra a los lados.

color: colocamos el color de la sombra.

2.3-Degradado de color

En la unidad 3 del módulo 1 de este curso, vimos la posibilidad de definir colores planos: utilizando las distintas unidades, podíamos indicarle el color exacto que necesitamos. Pero, ¿qué pasa cuando necesitamos un color que sea un degradé del mismo o un conjunto de varios colores? Para ello existe lo que se llama **gradiente**.

Nosotros también podemos definir en CSS una gradiente de colores, una leve transición entre varios colores dentro de una misma figura. Este gradiente puede ser de forma lineal o radial.

Para ello, al momento de definir el color, tendremos que colocar dentro de la propiedad que recibe el color la palabra **linear-gradient** o **radial-gradient** según sea el tipo de transición de colores; luego colocamos los colores en orden, con el porcentaje donde le indicamos qué proporción de espacio queremos que ocupe.

También, de manera adicional, podemos indicarle al **linear-gradient** la inclinación y a **radial-gradient** la pronunciación de la curvatura.

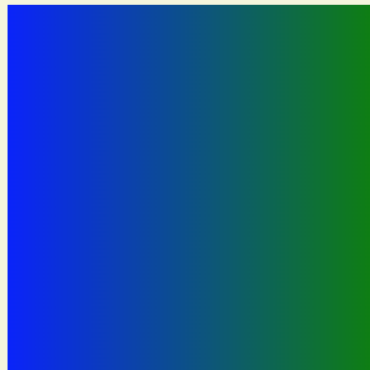
Ejemplos de linear-gradient



```
background: linear-gradient(blue, green);
```



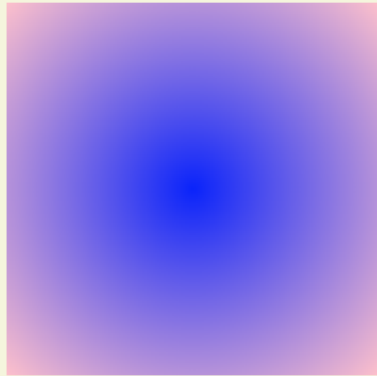
```
background: linear-gradient(90deg,blue, green);
```



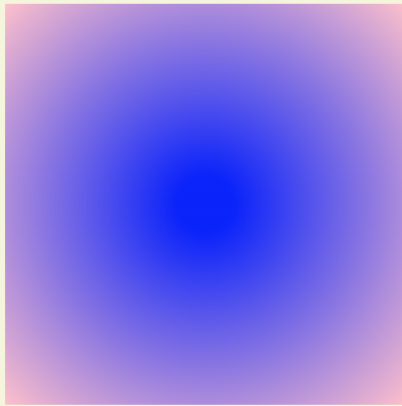
Ejemplos de radial-gradient



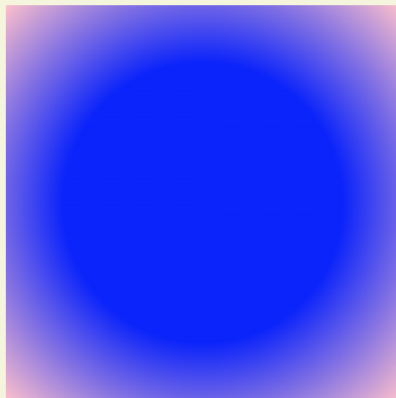
```
background: radial-gradient(blue, pink);
```



```
background: radial-gradient(blue 10%, pink);
```



```
background: radial-gradient(blue 50%, pink);
```



3. Pseudoclases y Pseudoelementos



Las pseudoclases y pseudoelementos son palabras claves ubicadas junto a los selectores que nos permiten generar algunos “efectos” ante ciertas acciones. De esta forma, empezamos a generar un tipo de interacción entre el CSS y el usuario, especificando el estilo del elemento ante ciertos estados de los elementos.

Pseudoclases

Estas palabras claves nos permiten indicar el estado de un elemento al tener el cursor sobre el mismo. Se colocan luego del nombre del selector con dos puntos y el nombre de la pseudoclase.

La sintaxis es:

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

Algunas de sus propiedades:

- **:hover**

Este tipo de pseudoclase funciona cuando el cursor pasa por encima del elemento que representa al selector. Hay que tener en cuenta que este selector no va a funcionar en celulares ya que allí no tenemos cursor.

- **:active**

Este tipo de pseudoclase funciona cuando se activa el elemento que representa al selector. Esta activación suele definirse al presionar el elemento. Hay que tener en cuenta que se va a producir este efecto al pulsar el elemento pero dejará de producirse el efecto al dejar de presionar el elemento.

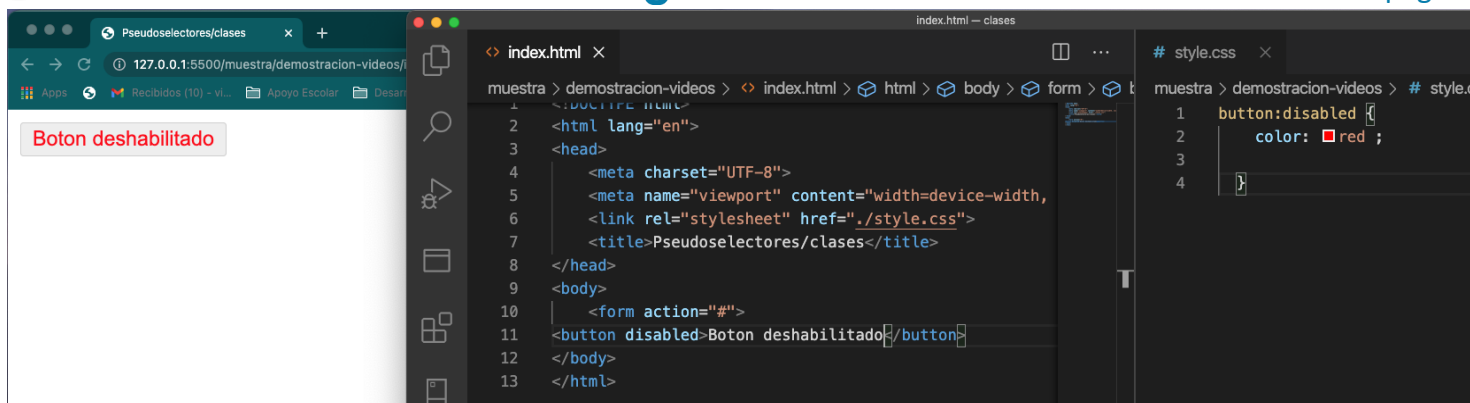
Este tipo de pseudoclases se suele utilizar en etiquetas <a> o <button>

- **:disable**

Este tipo de pseudoclase funciona en cualquier elemento deshabilitado.

Esto quiere decir, que cuando tenemos algún elemento como por ejemplo un botón el cual no queremos que el usuario pueda clicar, podemos deshabilitar el mismo y con esta pseudoclase darle las propiedades de css que necesitemos.

Ejemplo



<https://developer.mozilla.org/es/docs/Web/CSS/Pseudo-classes>

Pseudoelementos

Estas palabras claves nos permiten seleccionar una parte del elemento y añadirle estilos. A diferencia de las pseudoclasas, debemos colocar dos veces dos puntos.

La sintaxis es:

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

Algunas de las propiedades son:

- **::first-line**

Este pseudoelemento afecta solo a la primera línea de un bloque de texto.

- **::before**

Este pseudoelemento nos permite agregarle estilo antes del elemento.

- **::after**

Este pseudoelemento nos permite agregarle estilo después del elemento.



4. En Resumen

En esta unidad, pudimos aprender no solamente nuevas propiedades que nos permiten ubicar elementos en forma de grillas, sino también que aprendimos a dar algunos “efectos” en los elementos.

También aprendimos cómo modificar los fondos, manejar las imágenes que colocamos de fondo y el sombreado de los elementos.

Por último, tenemos que tener en cuenta que CSS nos ofrece una amplia variedad de posibilidades y propiedades que nos permiten hacer un sin fin de estilos sobre los elementos.

Esto nos permite tener más de una forma de hacer lo mismo, generando que haya más de una solución correcta.



5. Bibliografía

[https://developer.mozilla.org/es/docs/Web/CSS/CSS_Grid_Layout/
Basic_Concepts_of_Grid_Layout](https://developer.mozilla.org/es/docs/Web/CSS/CSS_Grid_Layout/Basic_Concepts_of_Grid_Layout)

<https://developer.mozilla.org/es/docs/Web/CSS/background-size>

<https://www.eniun.com/atributos-propiedades-colores-css/>

<https://developer.mozilla.org/es/docs/Web/CSS/background-attachment>

<https://developer.mozilla.org/es/docs/Web/CSS/box-shadow>

<https://developer.mozilla.org/es/docs/Web/CSS/Pseudo-classes>

<https://developer.mozilla.org/es/docs/Web/CSS/Pseudo-elements>

