DGTIC UNAM
DIRECCIÓN GENERAL DE CÓMPUTO Y DE TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN

Jornadas de
Visibilidad
Web UNAM 2022

Tailwindcss sesión: 1

Edgar Vargas Zermeño
talos@unam.mx

Visibilidad Web UNAM

Visibilidad Web UNAM

1. ¿Por qué usar Tailwindcss?

- La forma como trabajamos
- CSS convencional
- ATOMIC CSS
- Cómo funciona Tailwindcss

2. Apariencia básica

- Implementación con CDN
- Texto, colores y bordes
- Ancho y alto de elementos
- Imágenes y gradientes

3. Maquetación con Flexbox

- Documentación oficial
- Compilación de estilos
- Sistema de columnas
- Formularios
- Directiva @apply

4. Maquetación con Grid Layout

- Conceptos básicos
- Grid Layout en Tailwind
- Plantilla de columnas
- Maquetación responsiva

5. Configuraciones de Tailwindcss

- Archivo tailwind.config.js
- Colores personalizados
- Fuentes personalizadas

DGTIC UNAM
DIRECCIÓN GENERAL DE CÓMPUTO Y DE TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN

ENTREGA FINAL

Se entregarán los archivos en las 5 carpetas del curso (una por sesión), cada carpeta consta de subdirectorios “ejercicio” y “practica”.

Pasos:

1. Envío de correo **gmail** al instructor*
2. Asignación de carpeta **Drive** por parte del instructor
3. Notificación de acceso a carpeta personal
4. Subida de archivos (con límite al viernes 2 de diciembre 23:59 hrs)

*De no contar con un correo gmail se enviará correo con carpeta comprimida bajo el mismo plazo

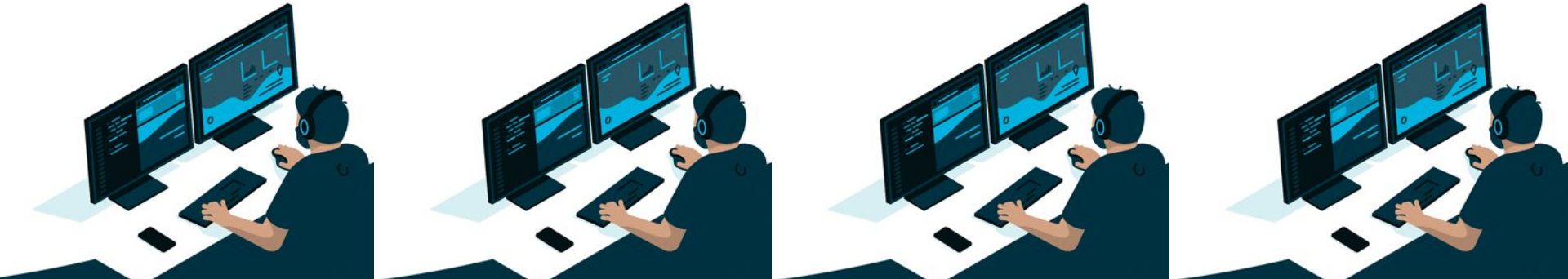
SEPARATION OF CONCERNS (***SoC***)

El **Modelo-Vlsta-Controlador(*MVC*)** que diferencía el desarrollo en backend y frontend, trae el concepto de **separación de intereses** (***SoC***), principio de diseño que enfoca el trabajo en **intereses** respecto del código para la construcción y mantenimiento de una interfaz de usuario.

SEPARATION OF CONCERNS (***SoC***)

Separar trae muchos beneficios pero no podemos negar que llega a ser tedioso "brincar" del **HTML** a las reglas escritas en **CSS** para sincronizar la apariencia.

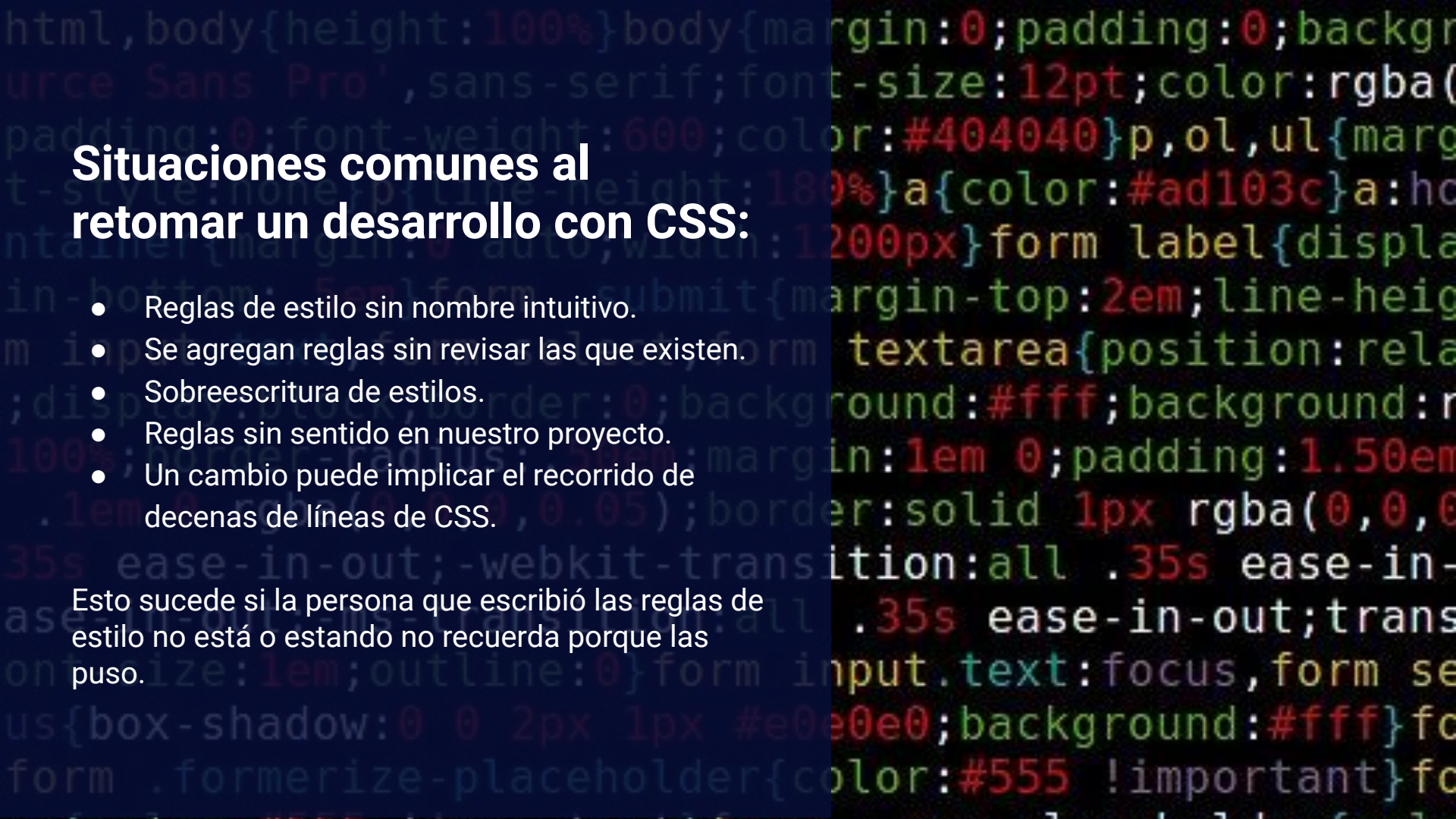
Esto se vuelve más complicado cuando varios desarrolladores necesitan actualizar, modificar o depurar una aplicación de forma simultánea.



Situaciones comunes al retomar un desarrollo con CSS:

- Reglas de estilo sin nombre intuitivo.
- Se agregan reglas sin revisar las que existen.
- Sobreescritura de estilos.
- Reglas sin sentido en nuestro proyecto.
- Un cambio puede implicar el recorrido de decenas de líneas de CSS.

Esto sucede si la persona que escribió las reglas de estilo no está o estando no recuerda porque las puso.



¿Agrego más estilos?

Para un desarrollo en constante evolución significa que se han de escribir muchas reglas adicionales.

Sobre todo en equipos de desarrollo donde cada colaborador resuelve como mejor le parece.

Los archivos CSS podrían convertirse en un desastre. Los desarrolladores no se sienten cómodos al eliminar reglas de estilo porque no están seguros de qué componentes se verán afectados en el proyecto.

Atomic CSS

Atomic CSS es un término acuñado en 2013 por **Thierry Koblentz**, aparece en su artículo de ***Smashing Magazine*** titulado ***Challenging CSS Best Practices*.**

Koblentz habla de lo doloroso e intrincado que se torna manejar CSS tradicional y presenta una nueva forma de implementar diseño junto con el lenguaje de marcas, le da el adjetivo de ***game-changer***.

Thierry Koblentz

Utility-first

Tradicionalmente cada regla CSS representa un elemento de la interfaz de usuario, describe su apariencia.

Los frameworks bajo el principio ***Utility-first*** le dan la vuelta a esta idea al proporcionarnos una “caja de herramientas” con **clases de utilidad, *clases de un solo propósito*** que podemos usar para aplicarlas a los elementos HTML y diseñar componentes.

Thierry Koblentz

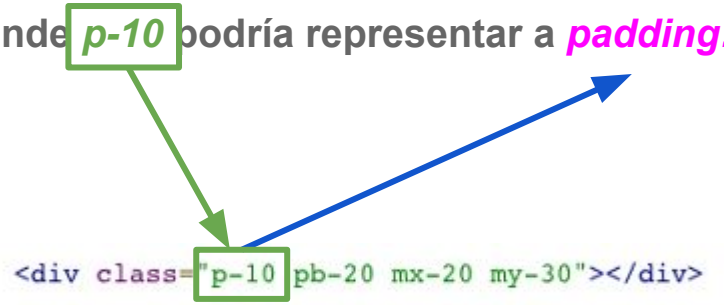
Utility-first

CSS convencional

```
// .css
.item{
  padding: 10px 10px 20px;
  margin: 20px 30px;
}

// .html
<div class="item"></div>
```

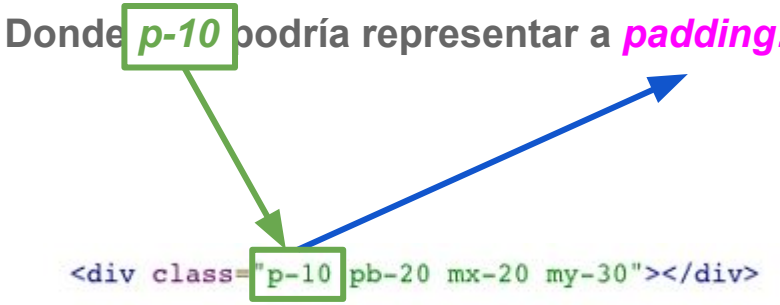
Donde *p-10* podría representar a ***padding: 10px;***



Utility-first

Al aplicar estilos predefinidos a los elementos se simplifica el proceso de dar apariencia, tenemos un código más claro y se proporcionan muchas opciones de diseño adicionales que contribuyen a crear contenido personalizado.

Las clases de utilidad pueden parecer extrañas al principio, pero después de usarlas se aprecian sus beneficios.



Utility-first - Frameworks

La idea de un enfoque basado en la utilidad para la denominación de clases ya lleva un tiempo existiendo, pero tomó impulso hasta el lanzamiento de **Tailwindcss**.

tailwindcss

Expressive CSS

Basscss

Atomizer

Windi CSS

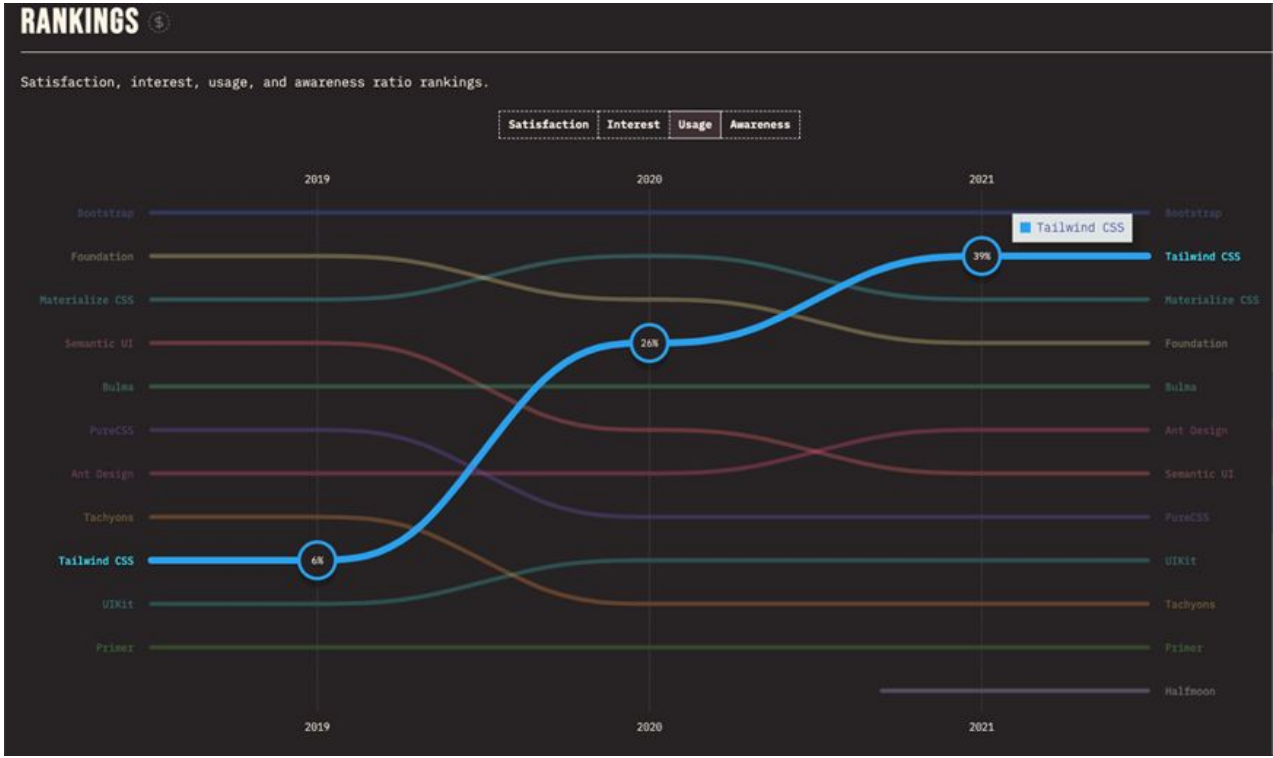
TACHYONS

CSS for designing faster than the speed of light

shed.css

Tailwindcss

Tailwindcss ganó mucho terreno en dos años (2019 - 2021) según la encuesta anual ***State of CSS 2021***

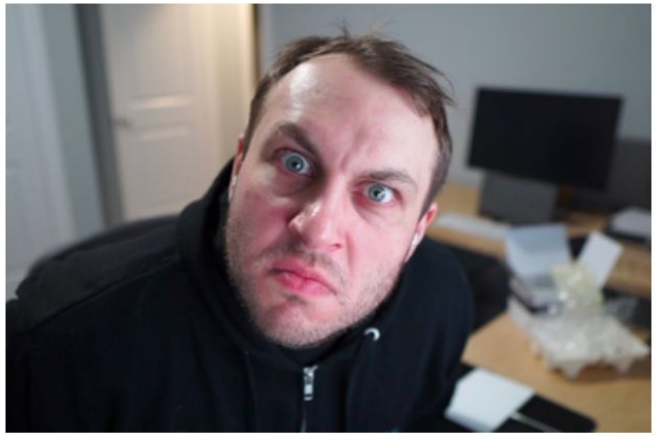


https://2021.stateofcss.com/es-ES/technologies/css-frameworks/

Tailwindcss

Tailwindcss se basa en ***Utility-first*** y es un proyecto CSS iniciado por Adam Wathan. El primer alfa se realizó en 2017 y la primera versión estable fue liberada en 2019.

Su nombre se eligió pensando en el factor que te hace ir más rápido, en alusión a la frase naval "viento en popa" que coloquialmente usamos como “viento a favor”.



Adam Wathan

Tailwindcss - Características

Tailwindcss establece un conjunto de directivas que permiten la creación de un vínculo estrecho entre el estilo y el contenido. Algunas de sus características:

- Código predecible, mantenible y escalable.
- Reduce la abstracción.
- Evita la redundancia de reglas.
- Detiene el crecimiento sin control de CSS.
- Omite las reglas sin usar.
- Mejora de almacenamiento en caché
- Se favorece la compresión gzip.

Tailwindcss - Instalación

Se puede instalar en 3 contextos:

- CLI
- Framework de desarrollo
- CDN (no compilado)

Tailwindcss rastrea los archivos y directorios que indiquemos, busca los nombres de clase utilizados y genera los estilos correspondientes en un archivo CSS estático.

DGTIC UNAM
DIRECCIÓN GENERAL DE CÓMPUTO Y DE TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN

Práctica 1: Mi galería

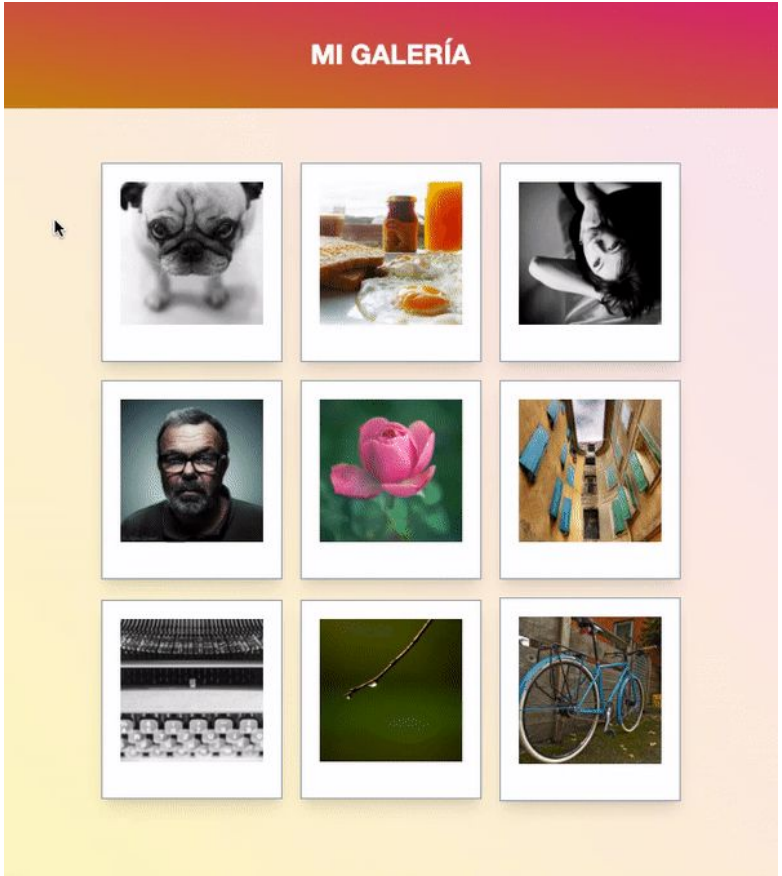
Genera una galería con diversas imágenes, deben tener el aspecto visual llamado polaroid.

Nuestro ejemplo utiliza fotos del portal dzoom.org.es. Si es tu deseo puedes usarlas para el ejercicio, son 14 y basta que cambies el número en el nombre de la imagen.

https://www.dzoom.org.es/wp-content/uploads/2011/08/insp-cuadradas-1.jpg

Pasos:

1. La práctica debe quedar como se ilustra.
2. Genera la estructura HTML
3. Coloca solo una imagen y ensaya las clases en la imagen antes de replicar.
4. Guarda tu práctica con tu nombre en el title.



Visibilidad Web UNAM

DGTIC UNAM
DIRECCIÓN GENERAL DE CÓMPUTO Y DE TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN

Edgar Vargas Zermeño

talos@unam.mx

2022

Visibilidad Web UNAM

DGTIC UNAM
DIRECCIÓN GENERAL DE CÓMPUTO Y DE TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN

Jornadas de
Visibilidad
Web UNAM 2022

Tailwindcss sesión: 2

Edgar Vargas Zermeño
talos@unam.mx

Visibilidad Web UNAM

Visibilidad Web UNAM

1. ¿Por qué usar Tailwindcss?

- La forma como trabajamos
- CSS convencional
- ATOMIC CSS
- Cómo funciona Tailwindcss

2. Apariencia básica

- Implementación con CDN
- Texto, colores y bordes
- Ancho y alto de elementos
- Imágenes y gradientes

3. Maquetación con Flexbox

- Documentación oficial
- Compilación de estilos
- Sistema de columnas
- Formularios
- Directiva @apply

4. Maquetación con Grid Layout

- Conceptos básicos
- Grid Layout en Tailwind
- Plantilla de columnas
- Maquetación responsiva

5. Configuraciones de Tailwindcss

- Archivo tailwind.config.js
- Colores personalizados
- Fuentes personalizadas

Instalación de Node

Antes de iniciar debemos asegurarnos de que tenemos instalado **Node.js** en nuestro equipo, pues la compilación de **Tailwindcss** la implementamos con **NPM** a través de la línea de comandos. Se recomienda utilizar la terminal desde el editor Visual Studio Code.

1. Abra una terminal o Ventana de Comandos y ejecute: ***node -v.*** La respuesta arrojará la versión de node y hará patente que lo tenemos instalado.

```
$ node -v
v17.4.0
```

2. Después ejecute: ***npm -v.*** Arrojará la versión del manejador de paquetes de Node.

```
$ npm -v
8.3.2
```

Si los comandos no arrojaron ningún resultado, instale **Node.js** desde el sitio oficial https://nodejs.org/es/download/. Al terminar ejecuta los comandos del punto 1 y 2.

Directivas de Tailwindcss

Las capas válidas son:

Capa	Uso
`base`	Se utiliza para modificar un estilo base aplicado a elementos HTML simples.
`components`	Se utiliza para definir estilos basados en clases.
`utilities`	Se utiliza para clases pequeñas de propósito único que siempre deben tener prioridad sobre cualquier otro estilo.

La sintaxis utilizada es la siguiente:

```
@tailwind base;
@tailwind components;
@tailwind utilities;

@layer base {
  //definición de estilos de elementos HTML
}
@layer components{
  //definición de estilos de clases personalizadas
}
@layer utilities {
  //definición de clases de propósito único
}
```

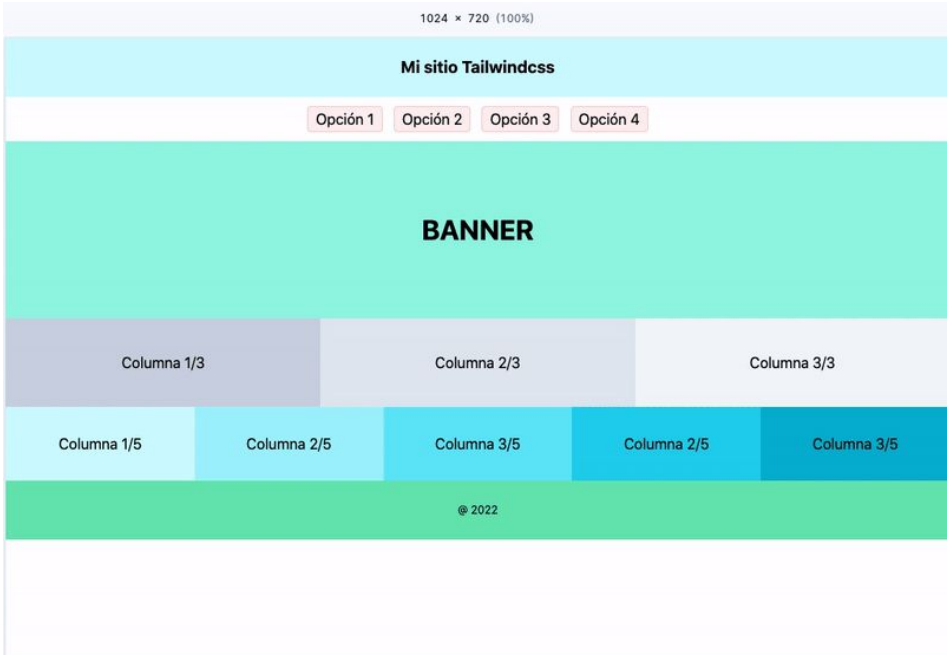
DGTIC UNAM
DIRECCIÓN GENERAL DE CÓMPUTO Y DE TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN

Práctica 2: Estructura HTML responsiva

Ahora crea la estructura de una página web responsiva con Tailwindcss, compila utilizando los archivos de configuración.

Pasos sugeridos:

1. Analiza la imagen de ejemplo, la práctica debe quedar como se ilustra.
2. Configura tu proyecto para utilizar la compilación de Tailwindcss.
3. Genera la estructura base HTML con etiquetas semánticas.
4. Asigna las clases Tailwindcss correspondientes a cada sección.
5. Genera una columna para el banner, tres y cinco columnas respectivamente para el resto del contenido. Ponles el color que desees.
6. Una vez terminado guarda tu proyecto para entregarlo el viernes.



Visibilidad Web UNAM

DGTIC UNAM
DIRECCIÓN GENERAL DE CÓMPUTO Y DE TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN

Edgar Vargas Zermeño

talos@unam.mx

2022

Visibilidad Web UNAM

DGTIC UNAM
DIRECCIÓN GENERAL DE CÓMPUTO Y DE TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN

Jornadas de
Visibilidad
Web UNAM 2022

Tailwindcss sesión: 3

Edgar Vargas Zermeño
talos@unam.mx

Visibilidad Web UNAM

Visibilidad Web UNAM

1. ¿Por qué usar Tailwindcss?

- La forma como trabajamos
- CSS convencional
- ATOMIC CSS
- Cómo funciona Tailwindcss

2. Apariencia básica

- Implementación con CDN
- Texto, colores y bordes
- Ancho y alto de elementos
- Imágenes y gradientes

3. Maquetación con Flexbox

- Documentación oficial
- Compilación de estilos
- Sistema de columnas
- Formularios
- Directiva @apply

4. Maquetación con Grid Layout

- Conceptos básicos
- Grid Layout en Tailwind
- Plantilla de columnas
- Maquetación responsiva

5. Configuraciones de Tailwindcss

- Archivo tailwind.config.js
- Colores personalizados
- Fuentes personalizadas

Grid Layout Module CSS

Para entender cómo funciona **Grid Layaut** imagina que tienes dos tabiques y los colocas uno a la izquierda del otro, dado esto veamos los conceptos:

Líneas

Nunca las vas a ver, existen de forma imaginaria, no las puedes contar sino hasta que pones elementos y dependiendo del número de ellos, será el número de líneas que existirán. Pensando en nuestros dos tabiques, tendrás 3 líneas:

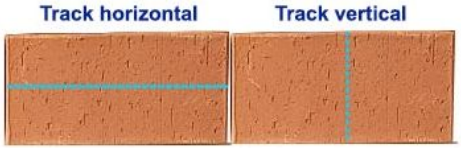
Primera línea al costado del tabique uno, segunda línea en medio de ambos tabiques, tercera línea al final del tabique dos, si pones otro tabique se generará una cuarta línea. Por decir así, siempre habrá una constante:

número de líneas = número de elementos + 1

Grid Layout Module CSS

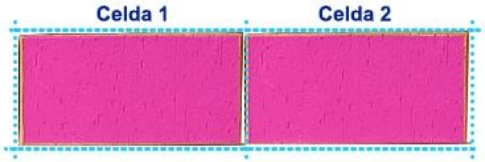
Track

Los tracks son el espacio comprendido entre dos líneas consecutivas en un solo sentido (vertical u horizontal). Según nuestro modelo imaginario, el espacio ocupado por cada tabique de izquierda a derecha (track horizontal), o de arriba a abajo (track vertical).



Celda

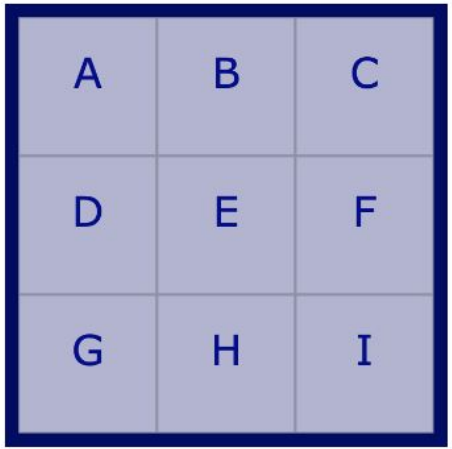
Es un espacio definido entre dos líneas horizontales consecutivas y dos líneas verticales consecutivas. En la intersección de dos líneas horizontales y dos líneas verticales se ubica una celda.



Una celda tiene el tamaño de 1x1 en nuestra rejilla (un tanto de ancho X un tanto de alto)

Grid Layout Module CSS

**Rejilla implicita,
ancho de contenedor de 300px
y tracks de 100px en lo horizontal y vertical**



Eso quedaría expresado así:

```
display: grid;
grid-template-columns: 100px 100px 100px;
grid-template-rows: 100px 100px 100px;
```

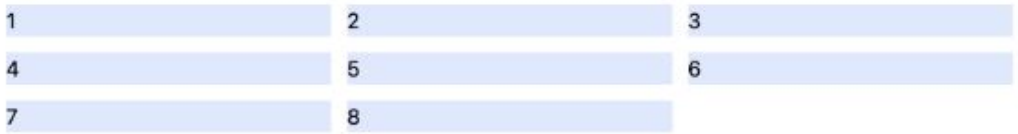
Grid Layout Module CSS

Para utilizar este módulo en Tailwindcss basta con indicar en el contenedor la clase *grid*, el número de columnas con grid-cols y a continuación el número deseado:

grid-cols-3

Si fueran filas sería *grid-rows-3*.
Para espaciar usamos *gap-3*.

```
<div class="grid grid-cols-3 gap-3 m-3">
  <section class="bg-indigo-100">1</section>
  <section class="bg-indigo-100">2</section>
  <section class="bg-indigo-100">3</section>
  <section class="bg-indigo-100">4</section>
  <section class="bg-indigo-100">5</section>
  <section class="bg-indigo-100">6</section>
  <section class="bg-indigo-100">7</section>
  <section class="bg-indigo-100">8</section>
</div>
```



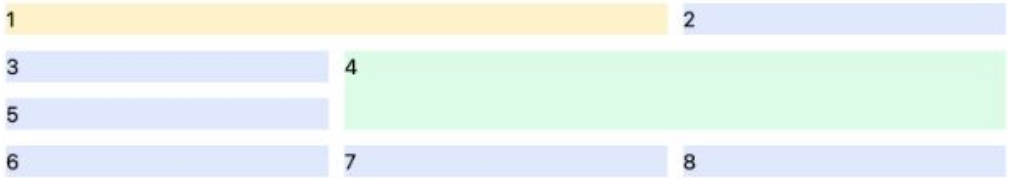
Grid Layout Module CSS

Para que un elemento abarque varios tracks agregamos *col-span*, si fuese una columna con una área de 2 celdas horizontales escribimos en el elemento deseado:

col-span-2

El elemento 4 del ejemplo muestra además *row-span-2* para abarcar dos filas.

```
<div class="grid grid-cols-3 gap-3 m-3">
  <section class="bg-amber-100 col-span-2">1</section>
  <section class="bg-indigo-100">2</section>
  <section class="bg-indigo-100">3</section>
  <section class="bg-green-100 col-span-2 row-span-2">4</section>
  <section class="bg-indigo-100">5</section>
  <section class="bg-indigo-100">6</section>
  <section class="bg-indigo-100">7</section>
  <section class="bg-indigo-100">8</section>
</div>
```

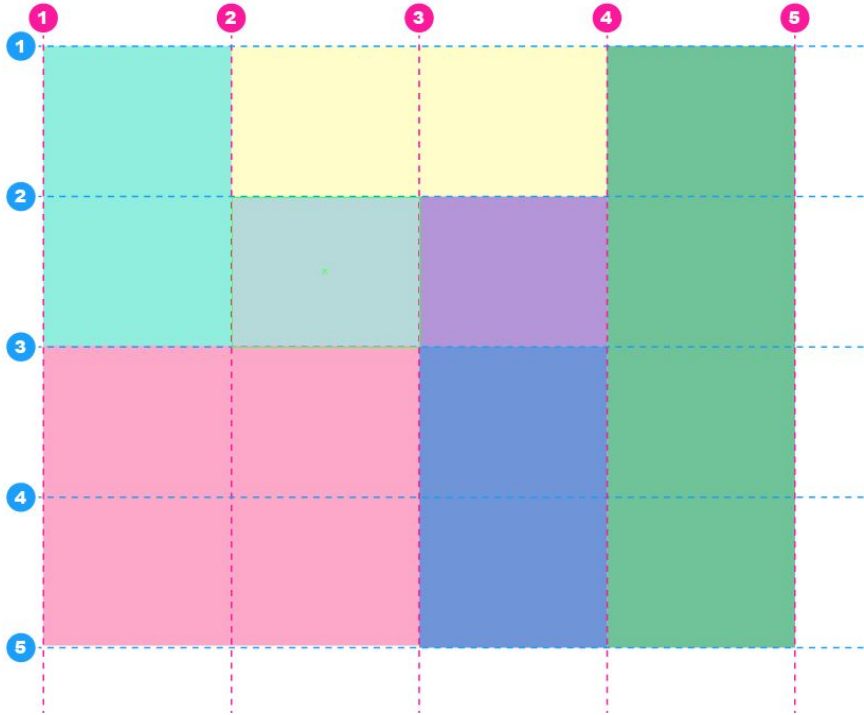


Práctica 3: Estructura Grid Layout

Crea una estructura con Grid.

Pasos sugeridos:

1. Analiza la imagen de ejemplo para reproducirla
2. Haz un proyecto Tailwindcss
3. Construye la estructura que creas necesaria
4. Aplica las clases de Tailwindcss y compila
5. Una vez terminado guarda tu proyecto para entregarlo.



DGTIC UNAM
DIRECCIÓN GENERAL DE CÓMPUTO Y DE TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN

Edgar Vargas Zermeño

talos@unam.mx

2022

Visibilidad Web UNAM

DGTIC UNAM
DIRECCIÓN GENERAL DE CÓMPUTO Y DE TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN

Jornadas de
Visibilidad
Web UNAM 2022

Tailwindcss sesión: 4

Edgar Vargas Zermeño
talos@unam.mx

Visibilidad Web UNAM

Visibilidad Web UNAM

1. ¿Por qué usar Tailwindcss?

- La forma como trabajamos
- CSS convencional
- ATOMIC CSS
- Cómo funciona Tailwindcss

2. Apariencia básica

- Implementación con CDN
- Texto, colores y bordes
- Ancho y alto de elementos
- Imágenes y gradientes

3. Maquetación con Flexbox

- Documentación oficial
- Compilación de estilos
- Sistema de columnas
- Formularios
- Directiva @apply

4. Maquetación con Grid Layout

- Conceptos básicos
- Grid Layout en Tailwind
- Plantilla de columnas
- Maquetación responsiva

5. Configuraciones de Tailwindcss

- Archivo tailwind.config.js
- Colores personalizados
- Fuentes personalizadas

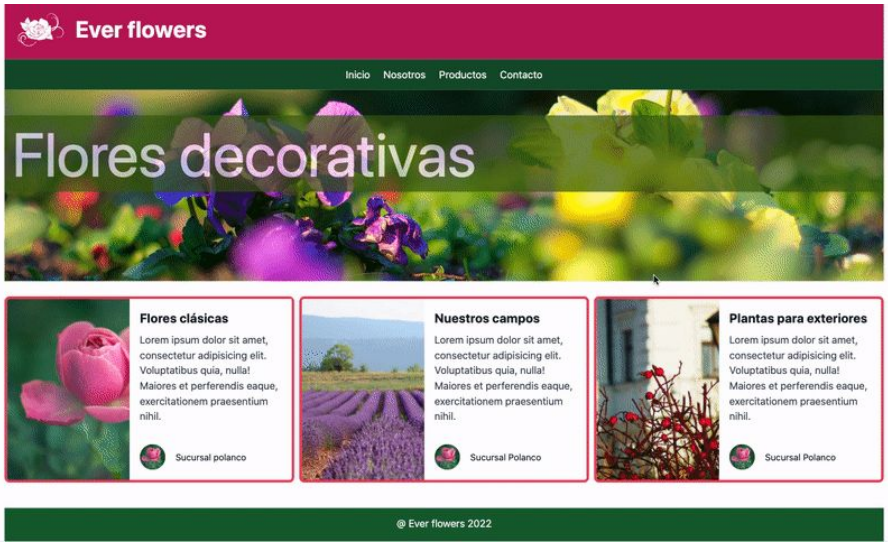
DGTIC UNAM
DIRECCIÓN GENERAL DE CÓMPUTO Y DE TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN

Práctica 4: Previo del proyecto final

Página estructurada con Grid.

Pasos sugeridos:

1. Genera un proyecto compilado de Tailwindcss.
2. Haz con Grid la estructura.
3. Analiza la imagen, la práctica debe quedar estructuralmente igual al ejemplo.
4. Asigna las clases Tailwindcss correspondientes a cada sección.
5. Sube tus ejercicios y práctica del día a tu carpeta de Drive



Visibilidad Web UNAM

DGTIC UNAM
DIRECCIÓN GENERAL DE CÓMPUTO Y DE TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN

Edgar Vargas Zermeño

talos@unam.mx

2022

Visibilidad Web UNAM

DGTIC UNAM
DIRECCIÓN GENERAL DE CÓMPUTO Y DE TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN

Jornadas de
Visibilidad
Web UNAM 2022

Tailwindcss sesión: 5

Edgar Vargas Zermeño
talos@unam.mx

Visibilidad Web UNAM

Visibilidad Web UNAM

1. ¿Por qué usar Tailwindcss?

- La forma como trabajamos
- CSS convencional
- ATOMIC CSS
- Cómo funciona Tailwindcss

2. Apariencia básica

- Implementación con CDN
- Texto, colores y bordes
- Ancho y alto de elementos
- Imágenes y gradientes

3. Maquetación con Flexbox

- Documentación oficial
- Compilación de estilos
- Sistema de columnas
- Formularios
- Directiva @apply

4. Maquetación con Grid Layout

- Conceptos básicos
- Grid Layout en Tailwind
- Plantilla de columnas
- Maquetación responsiva

5. Configuraciones de Tailwindcss

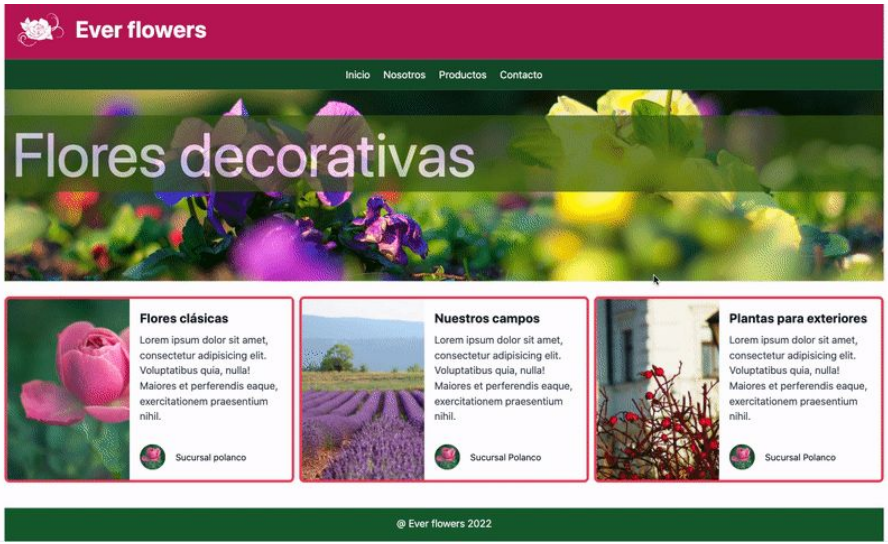
- Archivo tailwind.config.js
- Colores personalizados
- Fuentes personalizadas

Práctica 5: conclusión del proyecto final

Página estructurada con Grid.

Pasos sugeridos:

1. Continúa en tu proyecto de la sesión anterior.
2. Recuerda que la práctica terminada debe quedar estructuralmente igual al ejemplo.
3. El tema, las fuentes, los colores y las imágenes son libres.
4. Asigna las clases Tailwindcss a los elementos que vayas integrando.
5. Sube tu práctica del día a tu carpeta de Drive



DGTIC UNAM
DIRECCIÓN GENERAL DE CÓMPUTO Y DE TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN

Edgar Vargas Zermeño

talos@unam.mx

2022

Visibilidad Web UNAM