
# M4.258 - Eines HTML i CSS II aula 1

## PAC 3.

**Alumne: Gerard Vidal Gonzalez**

**Data: 31/12/2021**

# 1. BASE DEL PROJECTE

L'objectiu d'aquesta pràctica és reproduir el mateix projecte web fet a la PAC 2, amb la diferencia que s'inclourà i adaptarà als coneixements que hem adquirit al mòdul 4.2 de teoria.

Per aquest motiu, la base del projecte parteix del Boilerplate de la UOC (v3.3.0), obtingut a partir de l'enllaç https://github.com/uoc-advanced-html-css/uoc-boilerplate, amb les modificacions fetes a la PAC 2. Posteriorment es vincula la copia local de UOC Boilerplate al repositori GitHub amb enllaç https://github.com/LrMgic/M4.258-PAC3.

Executem a través del terminal el comandament d'instal·lació:

- *npm install*

Per a la instal·lació de tailwindcss es segueixen els passos indicats a la seva guia oficial d'instal·lació, URV: https://v2.tailwindcss.com/docs/installation. Per recomanació del professorat no s'utilitza la última versió, sinó la versió 2.

S'inicia la instal·lació executant a través del terminal:

- *npm install -D tailwindcss@2 postcss@latest autoprefixer@latest*

El qual ens ha ens modifica el fitxer *.postcssrc*, afegint el codi necessari per a que PostCSS compili el nou codi TailWind CSS que introduïm:

```
{
  "plugins": {
    "tailwindcss": {},
    "autoprefixer": {},
    "postcss-preset-env": {}
  }
}
```

Per finalitzar s'executa el codi següent:

- *npx tailwindcss init*

El qual genera el fitxer de configuració *tailwind.config.js*, que ens permetrà personalitza la configuració de TailWind CSS.

Es vincula el projecte publicat a GitHub amb la plataforma Netlify, la qual crea un entorn i una direcció web per al projecte.

URL publica: https://uoc-4-258-pac3-gerard.netlify.app/

## 2. PREPARACIÓ DE TAILWINDCSS

Per a poder utilitzar la llibreria TailWind CSS que hem instal·lat prèviament, adjuntarem les següents línies de codi a l'inici de l'arxiu main.scss:

```
// TAILWINDCSS START
@import "tailwindcss/base";
@import "tailwindcss/components";
```

I al final:

```
// TAILWINDCSS END
@import "tailwindcss/utilities";
```

S'utilitza la directiva @import en lloc de @tailwind per a evitar errors a la hora de capil·lar.

Després d'estudiar la teoria i alguns articles de recomanacions a l'hora de convertir el codi CSS a TailWind CSS es recomana partir del propi fitxer CSS i afegir un "@apply" i anar traduint cada una de les línies per a cada un dels elements, anant verificant que l' aspecte de la pàgina no es modifica.

Posant un exemple d'aquesta metodologia, partint del codi següent:

```
.main--contacta {
  background: $color1;
  justify-content: center;
  align-items: center;
  text-align: center;
}
```

S'introdueix l'"@apply" i es van comentant les línies ja convertides fins que només quedi la conversió:

```
.main--contacta {
  @apply bg-propi-1 items-center justify-center text-center;
  // background: $color1;
  // justify-content: center;
  // align-items: center;
  // text-align: center;
}
```

Un cop s'han traduït totes les línies, es substitueixen directament al fitxer .html.

En els casos que els estils predeterminats de TailWind CSS no s'adaptin al projecte, es creen les noves regles a través del fitxer main.css o s'adapten paràmetres en el fitxer *tailwind.config.js*.

# 3. CONVERSIÓ A TAILWINDCSS

Es decideix començar modificant els elements comuns que tenen tots el fitxers .html del projecte, que son el contenidors <header> i <footer> que es troben dintre del <body> de la pàgina.

Per a la traducció del element <header>, hem partit del component que dona com exemple la documentació de TailWind CSS, mostrat en l'enllaç següent:

https://tailwindui.com/components/marketing/sections/heroes

D'aquest codi s'ha esborrat tot el que corresponia al cos del document i a la barra de navegació minimitzada, ja que no entra dintre de l'abast d'aquesta PAC.

El codi sense modificar quedaria:

```
<div class="relative bg-white overflow-hidden">
    <div class="max-w-7xl mx-auto">
        <div class="relative z-10 pb-8 bg-white sm:pb-16 md:pb-20 lg:max-w-2xl lg:w-full lg:pb-28 xl:pb-32">
            <svg class="hidden lg:block absolute right-0 inset-y-0 h-full w-48 text-white transform translate-
x-1/2"
                fill="currentColor" viewBox="0 0 100 100" preserveAspectRatio="none" aria-hidden="true">
                <polygon points="50,0 100,0 50,100 0,100" />
            </svg>
            <div>
                <div class="relative pt-6 px-4 sm:px-6 lg:px-8">
                    <nav class="relative flex items-center justify-between sm:h-10 lg:justify-start"
                        aria-label="Global">
                        <div class="flex items-center flex-grow flex-shrink-0 lg:flex-grow-0">
                            <div class="flex items-center justify-between w-full md:w-auto">
                                <a href="#">
                                    <span class="sr-only">Workflow</span>
                                    <img class="h-8 w-auto sm:h-10"
                                        src="https://tailwindui.com/img/logos/workflow-mark-indigo-600.svg">
                                </a>
                                <div class="-mr-2 flex items-center md:hidden">
                                    <button type="button"
                                        class="bg-white rounded-md p-2 inline-flex items-center justify-center
text-gray-400 hover:text-gray-500 hover:bg-gray-100 focus:outline-none focus:ring-2 focus:ring-inset
focus:ring-indigo-500"
                                        aria-expanded="false">
                                        <span class="sr-only">Open main menu</span>
                                        <!-- Heroicon name: outline/menu -->
                                        <svg class="h-6 w-6" xmlns="http://www.w3.org/2000/svg" fill="none"
                                            viewBox="0 0 24 24" stroke="currentColor" aria-hidden="true">
                                            <path stroke-linecap="round" stroke-linejoin="round" stroke-
width="2"
                                                d="M4 6h16M4 12h16M4 18h16" />
                        </svg></button>div>div><div>
```

```
                    <div class="hidden md:block md:ml-10 md:pr-4 md:space-x-8">
                        <a href="#" class="font-medium text-gray-500 hover:text-gray-900">Product</a>
                        <a href="#" class="font-medium text-gray-500 hover:text-gray-900">Features</a>
                        <a href="#" class="font-medium text-gray-500 hover:text-gray-900">Marketplace</a>
                        <a href="#" class="font-medium text-gray-500 hover:text-gray-900">Company</a>
                        <a href="#" class="font-medium text-indigo-600 hover:text-indigo-500">Log in</a>
</div></nav></div></div></div></div></div>
```

S'adapta la icona i continguts utilitzats als del projecte i es comença amb la adaptació d'estils.

Al necessitar modificar el color de background, s'aprofita i s'introdueixen l'escala de colors propis en el fitxer *tailwind.config.js*:

```
module.exports = {
  purge: [],
  darkMode: false, // or 'media' or 'class'
  theme: {
    extend: {},
    colors: {
      propi: {
        1: 'rgb(34, 103, 230)',
        2: 'rgb(77, 81, 90)',
        3: 'rgba(39, 144, 185, 0.712)',
        4: 'rgb(214, 227, 255)',
        5: 'rgba(206, 212, 218, 1)',
        Text: '#282828',
        Text2: '#575757',
      },
      white: 'rgb(255, 255, 255)'
    }
      ...
```

Per exemple, en el cas que ens interessa, per a referir-nos al paràmetre de backgroun en el color terciari, introduirem *bg-propi-3*.

El codi del <header> queda de la següent manera:

```
<header>
    <div class="relative bg-propi-3">
        <div class="max-w-7xl mx-auto px-4 sm:px-6">
            <div class="flex justify-around items-center md:justify-start
md:space-x-10">
                <div class="flex justify-start items-stretch lg:w-0 lg:flex-1
min-h-full">
                    <a href="#" class="min-h-full">
                        <span class="sr-only">Dos mes Dos</span>
                        <img class="w-auto h-14 md:h-14 max-w-vw25"
src="./assets/images/logodos.jpg" alt="">
```

```
                    </a>
                </div>
                <div class="-mr-2 -my-2 md:hidden py-3">
                    <button type="button" class="bg-white rounded-md p-2
inline-flex items-center justify-center text-gray-400 hover:text-gray-500
hover:bg-gray-100 focus:outline-none focus:ring-2 focus:ring-inset
focus:ring-indigo-500" aria-expanded="false">
                        <span class="sr-only">Open menu</span>
                        <svg class="h-6 w-6"
xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24"
stroke="currentColor" aria-hidden="true">
                            <path stroke-linecap="round" stroke-
linejoin="round" stroke-width="2" d="M4 6h16M4 12h16M4 18h16" />
                        </svg>
                    </button>
                </div>
                <nav class="hidden md:flex items-center justify-end md:flex-1
lg:w-0">
                    <a href="./index.html"
                        class="whitespace-nowrap inline-flex items-center
justify-center px-4 py-4 shadow-sm text-base font-medium text-white bg-propi-
1 hover:bg-propi-1">Portada</a>
                    <a href="./nosaltres.html"
                        class="whitespace-nowrap text-base font-medium text-
white hover:text-gray-900 px-4 sm:px-6 lg:px-8">Qui som</a>
                    <a href="./noticies.html"
                        class="whitespace-nowrap text-base font-medium text-
white hover:text-gray-900 px-4 sm:px-6 lg:px-8">Noticies</a>
                    <a href="./contacta.html"
                        class="whitespace-nowrap text-base font-medium text-
white hover:text-gray-900 px-4 sm:px-6 lg:px-8">Contacta</a>
</nav></div></div></div></header>
```

Per al <footer>, partint del fitxer _footer.scss, es fa la traducció de la forma següent:

```
footer {
  background: rgba(39, 144, 185, 0.712);
  border: rgb(77, 81, 90) 1px solid;
  box-shadow: 5px 5px 15px 6px rgb(0 0 0 / 20%);
  color: white;
  line-height: 0.5em;
}

footer ul li {
  display: inline-block;
  margin: 0px 5px;
}
footer a {
```

```
  color: white;
  text-decoration: none;
}
```

Quedant de la següent manera:

```
.tw-shadow1 {
  box-shadow: 5px 5px 15px 6px $colorText2;
}
footer {
  @apply border-solid border-px border-propi-2 text-white tw-shadow1;
}
footer ul li {
  @apply inline-block my-0 mx-1;
}
footer a {
  @apply text-white no-underline;
}
```

S'afegeix una nova classe amb la ombra, la qual no es pot convertir a TailWind CSS. Posteriorment es declara a la línia del footer.

El codi final del <footer> queda de la següent manera:

```
<footer class="border-solid border-px border-propi-2 text-white
 tw-shadow1">
    <p class="p-0 min-w-0">2021 © Gerard y Vidal Gonzalez</p>
    <ul>
        <li class="inline-block my-0 mx-1"><a class="text-white no-
underline" href="*">Aviso legal</a></li>
        <li class="inline-block my-0 mx-1"><a class="text-white no-
underline" href="*">Política de privacidad</a>
        </li>
        <li class="inline-block my-0 mx-1"><a class="text-white no-
underline" href="*">Política de cookies</a></li>
    </ul>
</footer>
```

Els pàgines que es decideixen convertir son els corresponents als fitxers contacta.html, nosaltres.html i noticies.htm, en els quals s'ha utilitzat el mateix mètode que l'element <footer> amb la diferencia que els flitxers SCSS que corresponen a la mida mitjana/gran de pantalla, davant de cada element TailWind CSS, s'ha afegit un md:, el qual indica al navegador que només s'aplicarà a partir d'una mida mitjana de pantalla.

Durant la traducció del codi s'han completat el fitxer *tailwind.config.js* amb les mesures necessàries per adaptar el projecte.

# 4. EXTRACCIÓ DE COMPONENTS

Per a complir el requeriment de la PAC 3 de extreure com a mínim quatre components amb @apply, es procedeix de la següent manera:

- Els elements extrets es situen en el document main.scss, quedant el codi de la següent manera:

```
@layer components {
  .footer {
    @apply text-white bg-propi-3 w-full;
    box-shadow: 5px 5px 15px 6px $colorText2;
  }

  .nav-active {
    @apply whitespace-nowrap inline-flex items-center justify-center
px-4 py-4 shadow-sm text-base font-medium text-white bg-propi-1
hover:bg-propi-1;
  }

  .nav-inactive {
    @apply whitespace-nowrap text-base font-medium text-white
hover:text-gray px-4 sm:px-6 lg:px-8;
  }

  .form-control-trans {
    @apply border-propi-5 text-black appearance-none bg-clip-padding
bg-white rounded border-solid border block font-normal text-base
leading-normal py-1 px-3 w-auto min-w-vw30 max-w-vw50;
    transition: border-color 0.15s ease-in-out, box-shadow 0.15s ease-
in-out;
  }

  .tw-position {
    @apply w-full md:flex-auto md:object-cover md:max-h-vh45;
    object-position: 0% 25%;
  }
}
```

- L'element extret `.footer` es troba en tots els elements <footer>, el qual a part d'utilitzar-se el paradigma d'Atomic CSS i la llibreria d'utilitats TailWind CSS, s'utilitza un element CSS per a les sambres.
- Els elements extrets `.nav-active` i `.nav-inactive` formen part de l'element <header>, concretament els elements de la navbar <nav> que ens serveixen per navegar per les diferents pàgines, indicant a través d'aquest CSS en quina pàgina ens trobem. El codi quedaria de la forma següent:

```
<nav class="hidden md:flex items-center justify-end md:flex-1 lg:w-0">
    <a href="./index.html" class="nav-active">Portada</a>
    <a href="./nosaltres.html" class="nav-inactive">Qui som</a>
    <a href="./noticies.html" class="nav-inactive">Noticies</a>
    <a href="./contacta.html" class="nav-inactive">Contacta</a>
</nav>
```

- `.form-control-trans` es troba en el fitxer contacta.html i dona estil a tots es elements <input> i <textarea>, el qual a part d'utilitzar-se el paradigma d'Atomic CSS i la llibreria d'utilitats TailWind CSS, s'utilitza un element CSS per a les sambres.
- `.tw-position` dona l'estil i posició de les imatges del fitxer nosaltres.html.

# 5. COPILACIÓ DE PRODUCCIÓ

Un cop s'ha finalitzat la conversió a TailWind CSS, compilem el nou projecte a través del comandament de producció "*npm run build"*, en comptes del "*npm run dev"* que correspon a la execució de desenvolupament. Aquest canvi de metodologia ens farà minimitzar el volum de codi que compila per defecte TailWind i que no utilitzem en el nostre projecte.

Quan s'executa el comandament "*npm run build"* detectem que no s'executa de forma correcta i ens retorna el següent error per el terminal:.

```
× Build failed.
@parcel/optimizer-cssnano: Expected an opening square bracket.
```

Investigant el motiu de l'error ens trobem que no s'ha especificat el mètode "purge" correctament ja que no està especificat. Per aquest motiu, introduïm en el fitxer tailwind.config.js les següents especificacions de purga:

```
module.exports = {
  purge: [
    './src/**/*.html',
    './src/**/*.vue',
    './src/**/*.jsx',
  ],
 ...
```

Introduint aquesta especificació ens executa correctament la compilació per a producció, però al visualitzar el projecte, ens adonem que les pàgines no contenen els estils especificats a través de TailWind CS. Introduïm novament en el fitxer tailwind.config.js les següent especificació de "important" la qual facilitarà que TailWind reconegui el codi propi i no esborri de més:

```
module.exports = {
  important: '#app',
  purge: [
      ...
```

També, per a poder especificar quin son els elements que volem que TailWind entengui com a importants, nombrarem tots els element <body> de cada fitxer .html amb l'ID "app":

```
...
      <body id="app">
            ...
      </body>
</html>
```

Un cop s'ha especificat el codi anterior, s'executa novament en el terminal el comandament de compilació per a producció i comprovem que s'executa i es visualitza correctament tant en el projecte local com el pujat a Netlyfi.

## 6. PREGUNTES

- Quines diferències hi ha entre l'enfocament de tipus CSS semàntic (el que has fet servir a les altres PAC) i el CSS d'utilitats? Com ha afectat això el teu procés de desenvolupament? I el teu codi?

El CSS semàntic es molt bàsic i es te que programar cada punt, introduint més codi per a arribar al resultat desitjat. Per contra, utilitzant CSS d'utilitats es genera un codi complexa sense necessitat d'introduir tat de codi, però per el contrari, quan s'intenta fer alguna personalització que no està contemplat en la llibreria, es te que escriure molt més codi del que faria falta amb el CSS semàntic.

Durant el desenvolupament d'aquesta PAC, el CSS semàntic ha ajudat molt i ha minimitzat l'escriptura de codi (Sobretot CSS), però al partir d'un projecte existent "PAC 2", en el qual s'han utilitzat elements molt específics, tot el codi no coincideix amb el contemplat en les galeries de TailWind CSS, obligant a utilitzar el fitxer *tailwind.config.js* per a introduir les especificacions propies. També, no s'ha pogut convertir el 100% del codi a CSS d'utilitats, com per exemple el ombrejat o la posició d'imatges, la qual s'ha tingut de utilitzar CSS semàntic

- Quines diferències has trobat entre usar una llibreria de components i una llibreria d'utilitats?

La diferencia principal que he trobat es la sensació que deixa en utilitzar-les, ja que la llibreria de components facilita molt la creació de pàgines web, arribar a crear una pàgina bastant desenvolupada utilitzant poques línies e invertint poc temps, però per lo contrari no pots sortir de uns criteris específics i molt marcats, obligant a crear pàgines que s'assemblen molt entre elles.

Quan s'utilitza una llibreria d'utilitats, la quantitat de estils que es poden utilitzar i la personalització de les pàgines, augmenta molt. En aquest aspecte, utilitzar llibreries d'utilitat en lloc de llibreria de component, augmenta la llibertat de personalitzar, sense augmentar molt el temps que es tarda en crear una pàgina funcional.

També es té que tenir en compte que l'aprenentatge de utilitzar una llibreria de components es relativament fàcil, mentre que la llibreria d'utilitats es més complexa, encara que te una dinàmica més similar a l'escriptura semàntica.

- Quins components has decidit extreure i per què?

S’han extret els elements que es repetien de forma més recurrent i que podien reduir la quantitat de codi utilitzat. També s’ha extret el codi que no s’ha pogut convertir del tot a TailWind CSS.