

# Eines HTML i CSS II

P2.

Maria Planella Vall

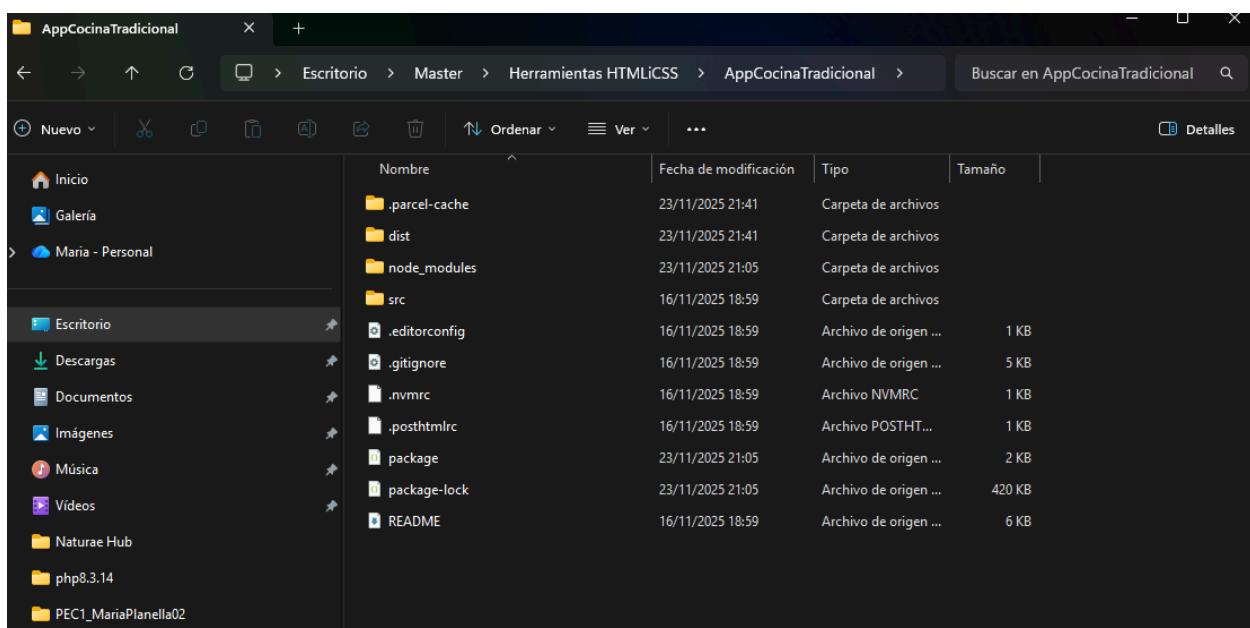
# 1. Introducción

En aquest treball hi ha el repte de disenyar una pàgina web de múltiples pantalles, partint de UOC Boilerplate. Aquesta pàgina web tracta de crear un portal on els usuaris puguin informar-se sobre menjars tradicionals, utilitzan diverses metodologies i noves funcions de SCSS per a facilitar l'estilització de les pàgines, Bootstrap també serà una eina important per ajudar a configurar les grids de la pàgina i afegir noves funcionalitats que aportin més informació atractiva per a l'usuari. En aquest cas la pàgina web tracta de menjars tradicionals del món, mostrant receptes de plats tradicionals i altres plats importants d'alguns països del món.

## 2. Procés de desenvolupament.

### 2.1 Instal·lació

En aquest apartat s'explicarà com s'ha fet la instal·lació tant del projecte de UOC Boilerplate com de les diverses dependències. En aquesta ocasió s'ha descarregat el ZIP del projecte de Github i s'han extret els arxius en un directori local.



Seguidament, s'ha creat un repositori a Github públic i connectar el repositori remot amb el repositori local. En aquest cas s'ha utilitzat Git per a conectar-los, primer és amb aquesta aplicació ficar-se en el directori del projecte local i incialitzar-lo amb la comanda següent.

```
git init
```

Es connecta amb la següent comanda.

```
git           remote      add          origin  
https://github.com/MariaPlanella02/AppCocinaTradicional.git
```

Després per a afegir tots els archius del projecte és amb la comanda següent.

```
git add .
```

I per últim fer el commit i el push per a carregar els archius al Github remot.

```
git commit -m "Primer commit"
```

```
Git push -u origin main
```

Per a facilitar la feina s'ha iniciat el projecte amb Github Desktop per a facilitar els commit i els push del projecte.

El següent pas es configurar les instalacions corresponents del projecte, al obrir el projecte desde VisualStudioCode, a la terminal s'escriu la comanda “npm install” per a instalar les dependències que ja ens proporciona el UOC-Boilerplate.

Seguidament per a provar que tot funcioni correctament s'executen les comandes de “npm run build” per a compilar el projecte i “npm run dev” per a executar-lo, aquesta última comanda ens dona un localhost que al obrir-lo al navegador ens nostra la pàgina web funcional.

## Dependències.

Una de les dependències externes que s'ha afegit es FontAwesome per a poder ficar icones que li donin una millor estetica a la pàgina web, per a instalar aquesta dependència es tan fàcil com executar a la terminal aquesta comanda “npm install @fortawesome/fontawesome-free” i per a saber si s'ha instal·lat correctament s'ha de mirar el package.json.

En aquest cas la configuració de Font Awesome ha estat diferent de la PAC anterior, ja que en aquest cas s'ha copiat la carpeta de “ebfonts”, aquesta es troba del directori “node\_modules/@fortawesome”, s'ha tingut que fer d'aquesta manera ja que sinó al intentar utilitzar les icones no trobava el directori, ja que “node\_modules” no es carrega al Github i es necessita per a carregar les icones. Per a poder utilitzar les icones s'ha creat un arxiu arrel al directori de styles guardat com a webfonts per a poder-ho utilitzar s'ha d'instanciar a \_dependencies.scss amb el @use.

```
// Font Awesome SCSS  
@use "@fortawesome/fontawesome-free/scss/fontawesome";  
@use "@fortawesome/fontawesome-free/scss/brands";  
@use "@fortawesome/fontawesome-free/scss/solid";  
@use "@fortawesome/fontawesome-free/scss/regular";
```

Per a poder instanciar qualsevol icona s'ha de buscar a <https://fontawesome.com/search?o=r> la icona corresponent i al escollir-la a la pàgina es proporciona el codi per a instanciar-la en el projecte, amb una estructura com aquest.

<i class="fa-solid fa-instagram"></i>

S'ha escollit aquesta dependència externa ja que Font Awesome proporciona icones intuïtives i clares, facilitant la comprensió de les persones per a interpretar diferents opcions que proporciona la pàgina web amb una sola icona, també facilita la estilització de dispositius més petits ja que únicament amb la icona ja s'enten el resultat. Al ser icones del mateix estil aporta una coherència i sincronia d'imatges dins de la pàgina web, diferents icones amb el mateix estil es veurà com a opcions ben integrades entre elles, però cadascuna amb la seva funcionalitat o significat diferent.

## Publicació.

Finalment, per a publicar el projecte s'utilitza Netlify, al tenir el GitHub remot és molt senzilla la publicació.



Primer esva a l'opció de “Add new project”, allí se li especificarà un nom a la pàgina web, per a poder crear la URL correctament, en aquest cas se li ha assignat el nom de “appmenjartradicinal”, Netlify compila el GitHub y publica la pàgina web de manera automàtica i cada vegada que detecta alguna modificació al projecta, compila i publica automaticament, fent així que el programador no s'hagi de preocupar d'aquesta feina. En l'últim apartat d'aquest document hi han els enllaços cap al GitHub i a la pàgina web publicada de Netlify.

## 2.2 Metodologia BEM

La metodologia que s'ha escollit ha sigut BEM ja que permet una reutilització del codi SCSS per a tot el projecte i aporta una gran claritat dins del SCSS per a diferenciar tots els estils, ajudarà a mesura que el projecte es torni més gran per a encapsular els estils de manera clara.

Per a instal·lar-ho s'utilitza la comanda explicada al enunciat `npm install --save-dev stylelint-scss stylelint-config-recommended-scss`. Seguidament a l'arrel del projecte es crea el fitxer “stylelintrc” on s'especificaran les regles que s'han de seguir per a complir amb la metodologia BEM.

En la imatge següent es mostres les següents normes que ha de complir el projecte per a que es compleixi la metodologia BEM, les opcions de “selector-class-pattern”, “scss/dollar-variable-pattern” i “scss/percent-placeholder-pattern” estan pensades per a que els arxius compleixin noms amb formats predeterminats, les regles que comencen amb `scss/` estan més centrades amb els arxius de SCSS per a que no s'utilitzin regles de SCSS incorrectes o descatalogades i així poder tenir els estils actualitzats a

les versions més noves i compatibles d'avui en dia, altres regles es centren en els estils de valors i altres de bones pràctiques d'estructura.

```
▀ .stylelintrc > ...
1  ↴ {
2    "extends": ["stylelint-config-standard-scss"],
3    "plugins": ["stylelint-scss"],
4    ↴ "rules": {
5      "selector-class-pattern": "^[a-z]([a-z0-9-]+)?(_[a-z0-9]+(-[a-z0-9]+)*)?(--[a-z0-9]+(-[a-z0-9]+)*)?$",
6
7      "scss/at-rule-no-unknown": true,
8      "scss/selector-no-redundant-nesting-selector": true,
9      "scss/dollar-variable-pattern": "^(a-z0-9-]+)$",
10     "scss/percent-placeholder-pattern": "^(a-z0-9-]+)$",
11     "scss/load-no-partial-leading-underscore": true,
12     "scss/at-import-partial-extension-disallowed-list": ["scss"],
13     "scss,double-slash-comment-empty-line-before": ["always"],
14     "scss,double-slash-comment-whitespace-inside": "always",
15
16     "color-hex-length": "short",
17     "alpha-value-notation": "percentage",
18     "hue-degree-notation": "angle",
19     "font-weight-notation": "numeric",
20     "max-nesting-depth": 3,
21     "selector-max-id": 0,
22     "declaration-no-important": true,
23     "no-duplicate-selectors": true,
24     "block-no-empty": true
25   }
26 }
27 }
```

I per a comprovar-ho s'utilitza la comanda de `npx stylelint "*/*.scss"` per a saber si els estils són correctes al inici han sortit algun errors, com esperar una línia en blanc abans d'una norma o algun canvi de unitats, però que es podien arreglar amb la comanda `npx stylelint "*/*.scss" --fix`, altres errors eren dons que s'havien de renombrar manualment per a que complisin les regles específicades.

## 2.3 Diseny i desenvolupament

Per a aquest projecte s'utilitza una metodologia de Desktop First, ja que al ja estar treballant i disenyant amb un ordenador agafa la configuració base com a la del ordenador i per a fer la pàgina web responsiva per a mòvils o tablets es modifica en els estils. Considero que al ja realitzar la opció de Desktop la primera, ja tractes les funcionalitats i la opció més complicada i quedaria adaptar-ho a altres dispositius de diferents tamanys, tindran la mateixa funcionalitat però amb estils adaptats a cada dispositiu.

En el cas de Bootstrap s'ha utilitzat de base en alguns components, però s'ha personalitzat mitjançant variables SCSS per adaptar-lo al projecte.

```

1  /* ===== */
2
3  /* Paleta de colores cocina */
4
5  /* ===== */
6
7  :root {
8      --brand: #D2691E;
9      --ink: #333;
10     --paper: #FAF3E0;
11     --border: #DEB887;
12     --accent: #6B8E23;
13     --highlight: #8B0000;
14     --cta-vibrant: color(display-p3 0.9 0.2 0.3);
15     --transparent: transparent;
16 }
17
18 /* ===== */
19
20 /* Variables Sass para cálculos y mixins */
21
22 /* ===== */
23
24 $brand: #D2691E;
25 $ink: #333;
26 $paper: #FAF3E0;
27 $border: #DEB887;
28 $accent: #6B8E23;
29 $highlight: #8B0000;
30 $transparent: transparent;
31
32 /* ===== */
33
34 /* Espaciados escalados para coherencia del proyecto */
35
36 /* ===== */
37
38 $space-1: 0.25rem;
39 $space-2: 0.5rem;
40 $space-3: 1rem;
41 $space-4: 2rem;
42

```

S'han especificat colors en el `:root` per a poder utilitzar la mateixa colorimetria en tot el projecte sense necessitat de repetir els colors en cada arxiu d'estil manualment, aquests mateixos colors s'han inicialitzat amb `$` per a poder-los utilitzar en altres seccions, com en càlculs o mixins i s'han especificat espaciats escalats per a donar una coherència d'espai entre els títols, les imatges, els subtítuls o els paràgrafs.

Altres funcionalitats de Bootstrap utilitzades són les classes utilitaries, són classes que aplicen una sola propietat CSS de forma ràpida, en aquest cas s'han utilitzat les classes per a crear grids, aquesta nomenclatura s'especifica en el HTML d'aquesta manera.

```
<div      class="hero__content      col-12      d-flex      flex-column
justify-content-center align-items-start">
```

Aquesta part de codi està extreta de `_home.scss` on s'especifica una grid en la pantalla principal controlant l'amplada de les columnes de la grid, aplicant-li propietats de flexbox, espaiats, alineacions i pes tipogràfic. Especificant els requisits de la grid, en l'arxiu d'estils de

\_home.scss s'especifiquen altres conceptes que indicar-los al HTML seria més mala praxis i complexio al codi innecessaria. Es combinen utilitats de Bootstrap amb estils personalitzats de SCSS com per exemple la imatge següent.

```
.hero {  
  min-height: 100vh;  
  padding: vars.$space-3 fn.to-rem(32);  
  color: var(--ink);  
  
  // animacion para canviar las imagenes de fondo  
  animation: hero-background 20s infinite alternate;  
  container-type: inline-size;  
  container-name: hero;  
  
  @supports (display: grid) {  
    &__grid {  
      display: grid;  
      grid-template-columns: 1fr 1fr;  
      align-items: center;  
      gap: vars.$space-3;  
    }  
  }  
}
```

El bloc .hero es construeix amb CSS grid, container queries per a facilitar la utilització i funcions propies de SCSS mentre aquest bloc s'aprofita d'utilitats de Bootstrap. Aquesta combinació permet rapidesa en el layout gràcies a Bootstrap i el control total dels components claus mitjançant estils propis.

S'utilitza @support per a saber si el navegador pot soportar la propietat de la grid abans d'aplicar-la, ajuda a evitar errors en navegadors antics, permet aplicar estils moderns sol si estan disponibles i es una forma de fer millores progresives sense trencar la compatibilitat. Així el estil comprova si si es pot aplicar o no la grid, si no es pot ho ignorarà.

La container querie permet aplicar estils en funció del tamany del contenidor pare, s'utilitza per a que els components siguin més adaptables i reutilitzables perquè s'ajusten segons l'espai disponible del contenedor, cón ideals per als layouts complexes amb grids com seria en aquest projecte, que utilitza diverses grids en diversos arxius i permeten un diseny verdaderament responsiu a nivell de component.

```
@container hero (max-width: 600px) {
  .hero__grid {
    grid-template-columns: 1fr;
  }

  .hero__title {
    font-size: fn.modular-scale(1);
  }

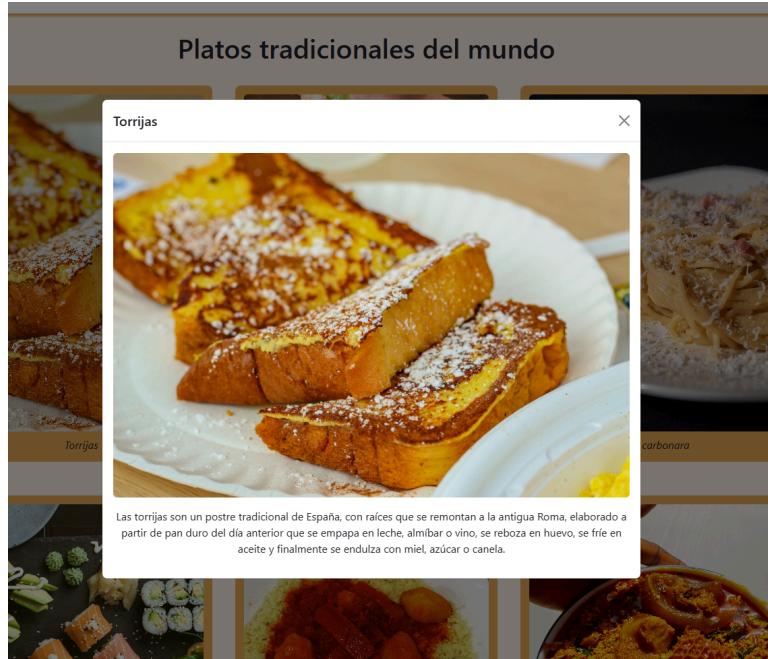
  .hero__description {
    font-size: fn.to-em(15);
    line-height: 1.6;
  }

  .hero__cta {
    font-size: fn.modular-scale(1);
    padding: fn.to-em(12);
    width: 100%;
    justify-content: center;
  }
}
```

Una altra característica implementada són les capes per cascada, s'han implementat en \_article.scss i \_galery.scss. Aquestes capes d'estil defineixen un ordre explícit en la cascada, cada @layer agrupa regles relacionades amb la base, components entre altres i s'apliquen en le ordre en que es declaren.

Ajudes amb la organització per a separar estils amb propòsits, t'ajuden a decidir quina capa té prioritat sense dependre sol de la especialitat, fan el projecte més fàcil de llegir i entendre, perquè queda clar quins estils són base, quins són de components i quins són els ajustos finals, també ajuden amb la compatibilitat amb Bootstrap ja que es poden ficar els seus estils en una capa i després crear una capa de @overrides per a personalitzar-los sense lluitar amb la especificitat.

Perúltim, en la galery.html s'ha utilitzat els modals que proporciona Bootstrap per a crear una pàgina més interactiva per a l'usuari, en aquest cas aquesta pàgina proporciona 6 plats tradicionals de diversos països del món i l'usuari al clicar a cada fitxa s'activarà el modal de Bootstrap, per a que es mostri el modal l'usuari ha de clicar a la fitxa de qualsevol plat i apareixerà una petita pantalla que mostrerà una imatge i una explicació sobre el plat.



Al afegir aquest modal aporta una interactivitat de l'usuari amb la pàgina web, així se li fa més atractiva i a futur es poden afegir més funcionalitats referent al modal, podent afegir redireccions desde la pantalla del modal cap a altres pantalles de la pàgina web o fer més interactiva la pàgina.

Per a poder incorporar els models de Bootstrap aquests s'afegeixen al HTML de la següent manera.

```
<figure class="gallery__item">
  
  <figcaption class="gallery__caption">Torrijas</figcaption>
</figure>
```

Primer en aquest cas s'ha creat una imatge amb un títol i en aquesta s'ha especificat data-bs-toggle per a especificar que quan l'usuari i cliqui s'activi com un toggle i es mostri el #galeryModal1.

Per a programar el modal es de la seüent manera.

```

<div class="modal fade" id="galleryModal1" tabindex="-1" aria-labelledby="galleryModalLabel1" aria-hidden="true">
  <div class="modal-dialog modal-lg modal-dialog-centered">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="galleryModalLabel1">Torrijas</h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Cerrar"></button>
      </div>
      <div class="modal-body">
        
        <p class="mt-3">Las torrijas son un postre tradicional de España, con raíces que se remontan a la antigua Roma, elaborado a partir de pan duro del día anterior que se empapa en leche, almíbar o vino, se reboza en huevo, se fríe en aceite y finalmente se endulza con miel, azúcar o canela.</p>
      </div>
    </div>
  </div>
</div>

```

Primer se li especifica que el modal està amagat i quan l'usuari hi interactua aques apareix en pantalla, s'han creat uns estils per a fer la aparició del model més atractiva, com una opacitat mès fosca al darrere del modal i que aquest sigui una fitxa blanca amb la amteixa imatge que en la galeria i amb una explicació. També per a tancar-lo s'ha creat un botó per a que l'usuari pugui tancar el modal, que en aquest cas seria únicament amagar-lo.

### 3. Enllaços

Pàgina Web: <https://appcuinatradicional.netlify.app/>

Repositori GitHub: <https://github.com/MariaPlanella02/AppCocinaTradicional>