

PEC 1

Herramientas HTML y CSS II.

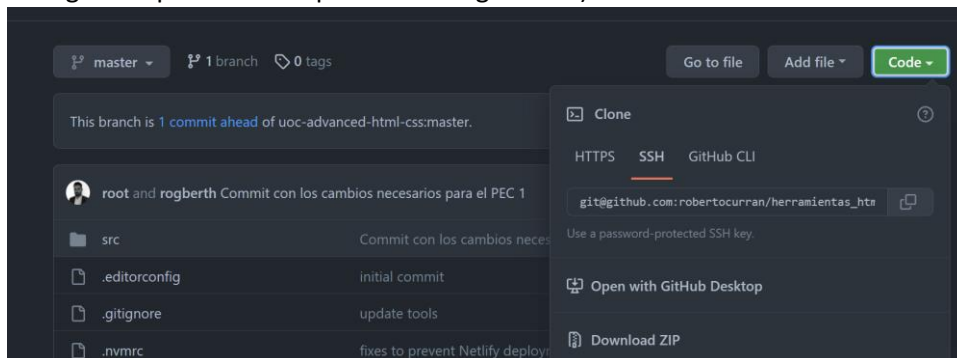
Trabajo realizado por Roberto Curran García

Enlace a la web: <https://luxury-duckanoo-f10108.netlify.app/>

Enlace al repositorio en GitHub: https://github.com/robertocurran/herramientas_html_css_II_pec1

Configuración inicial

El primer paso fue hacer un fork del repositorio en una de las opciones disponibles en el repositorio del UOC Boilerplate en Github. Seguidamente cloné mi fork en mi máquina local mediante SSH (ya configurado previamente para otras asignaturas)



El entorno de desarrollo elegido es VSCode con WSL Ubuntu (o en su defecto powershell para los comandos).

Para clonarlo se copia el enlace, en este caso de SSH de github y se utiliza el comando

```
git clone git@github.com:robertocurran/herramientas\_html\_css\_II\_pec1.git
```

Y Git descargará todos los archivos del repositorio en la carpeta donde se llame el mismo comando.

Primero tocaría instalar todas las dependencias del proyecto base con

```
npm install
```

Una vez hecho había que instalar dependencias externas que no vienen por defect, en este caso sólo he necesitado FontAwesome y Stylelint por lo que los comandos ya se han visto en los módulos de la asignatura:

```
npm install --save @fortawesome/fontawesome-free
```

```
npm install --save-dev stylelint
```

Y también la configuración base recomendada `npm install --save-dev stylelint-scss stylelint-config-recommended-scss`

Una vez hecho todo esto se podía comprobar que el proyecto hacía build de desarrollo y de producción con los comandos ya incluidos en “scripts” del package.json

```
npm run dev
```

```
npm run build
```

Metodologías y estilos elegidos

De las metodologías propuestas, elegí basar el diseño de la web en una combinación de BEM y ITCSS aunque sin utilizar la nomenclatura propuesta en la documentación de ITCSS llamada BEMIT ya que me recuerda a una suerte de notación húngara y creo que es innecesario contar con prefijos para las clases de CSS teniendo ya la propia nomenclatura de BEM y la estructura de carpetas de ITCSS que son suficientes para la clarificación del código y su organización.

No elegí OOCSS pues realmente BEM es una metodología que también es orientado a objetos por lo que me parece que se obtienen los mismos beneficios además de que con BEM se cuenta con la nomenclatura que si bien podría ser entendido como una limitación, me pareció interesante usarlo para este PEC 1.

SMACSS por otro lado me pareció muy versátil por sí solo, al ser una guideline más que una metodología concisa, en retrospectiva quizás debería haber elegido SMACSS debido puntos negativos que comentaré a continuación.

Debido a que el proyecto es una página única para un CV, realmente no se ocurría ninguna manera de hacerla interactiva sin que dejara de parecer un CV, por lo que finalmente es una página estática que si bien tiene varios elementos distintos al final estos elementos carecen de estados por lo que no llegué a utilizar la M de BEM: los Modifiers. Ciertamente es que no son exclusivamente para un estado en un contexto de interacción con el usuario ya que el modificador se puede utilizar también para distinguir entre elementos iguales, pero en diferentes situaciones o posiciones, pero debido a la sencillez del estilo de la web no llegué a necesitarlos.

También mencionar que debido a que el único Block es el CV en sí, todas las clases de los componentes vienen prefijadas con “cv_” por lo que podría parecer innecesario usar la nomenclatura BEM en este ejercicio específicamente, aunque de extenderlo ya podría volverse rentable su uso.

Otro problema de usar BEM y ITCSS es un conflicto con la propia nomenclatura de ambas metodologías, ya que para en ITCSS el nivel 4 del triángulo Elements no correspondería con los Elements de BEM sino con los elementos básicos de HTML. Por lo que hay que asociar los tres componentes de BEM con los niveles ITCSS de la siguiente manera:

BEM	ITCSS
Blocks	Objects
Elements	Components

Modifiers	Parte de los Components
-----------	-------------------------

A pesar de estos problemas considero que el resultado es un proyecto con las clases de CSS bastante claras y bien organizado.

Para asegurar que se cumpliera con estas metodologías elegidas, configuré Stylelint de la siguiente manera:

Primero añadí en “scripts” de package.json

```
"stylelint": "stylelint src/**/*.scss"
```

Para que stylelint se ejecutara a todos los ficheros .scss dentro de todas las carpetas contenidas en src.

Cree el archivo .stylelintrc.json y le añadí los siguientes datos:

```
{
  "extends": ["stylelint-config-recommended-scss"],
  "rules": {
    "selector-class-pattern": "^[a-z]([a-z0-9-]+)?(__([a-z0-9-]+)?(--([a-z0-9-]+)?)){0,2}$",
    "selector-id-pattern": "^$",
    "unit-disallowed-list": ["px", "rem"],
    "unit-allowed-list": ["vw", "vh", "em", "%", "fr"],
    "media-feature-name-value-allowed-list": {
      "orientation": "portrait"
    }
  }
}
```

En la que se cargan la configuración recomendada instalada anteriormente y se definen las reglas:

Regla para los nombres de las clases:

```
"selector-class-pattern": "^[a-z]([a-z0-9-]+)?(__([a-z0-9-]+)?(--([a-z0-9-]+)?)){0,2}$",
```

Esta expresión regular ya venía en los módulos de la asignatura de ejemplo por lo que fue sencillo añadirla

Reglar para los nombres de las IDs:

```
"selector-id-pattern": "^$",
```

Esta expresión es simplemente para forzar que no se utilicen IDs de HTML pues en BEM se recomienda usar clases.

Reglas de las unidades válidas y no válidas:

```
"unit-disallowed-list": ["px", "rem"],
```

```
"unit-allowed-list": ["vw", "vh", "em", "%", "fr"],
```

Esto no forma parte de las metodologías, pero es parte del estilo que uso en la que intento no utilizar píxeles pues son unidades absolutas. Tampoco quiero confundirme entre la unidad de tamaño relativo al texto “rem” y relativo al texto “em” por lo que impido el uso de la primera.

Finalmente, la última regla es:

```
"media-feature-name-value-allowed-list": {
  "orientation": "portrait"
```

}

En la que sólo se permite usar la media query de orientación “portrait” ya que utilizo un diseño de “desktop first” y añadido reglas CSS para móvil (pantallas verticales) después.

Diseño de la web

Como acabo de mencionar en el punto anterior, elijo hacer “Desktop first” añadiendo las reglas necesarias con las “media query” a posteriori para pantallas verticales.

El diseño de la web es sencillo y similar al de un documento CV en vertical, pero con una imagen de fondo en la página y el fondo del CV como color translúcido.

El CV es el único Block de BEM, su clase es `class="cv"` y el resto de Elements de BEM se pueden encontrar sus estilos definidos en la carpeta `6_components`.

De los Elements el principal es el de `class="cv__section"` que propone subdividir el documento CV en segmentos. En cada uno de estos “cv__section” habrá un contenido distinto y por lo tanto tendrá distintos Elements.

He utilizado el anidado permitido en SASS con una profundidad máxima de 1 como es recomendado en BEM para no conseguir estructuras complicadas. Principalmente he sobre escrito las reglas CSS de los elementos básicos de HTML (`h1`, `h2`, `p`, etc.) en caso de ser necesario, y he definido los valores por defecto para los elementos básicos de HTML en el archivo `3_generic/generic.scss`.

Debido a que ITCSS no coincidía con la estructura de ficheros y carpetas del UOC Boilerplate tuve que eliminar estas salvo el archivo `main.scss` que es donde ocurren los imports de todos los archivos `.scss` en el orden adecuado ya que CSS depende del orden de definición de las reglas y alterar esto tendría efecto en el resultado final de la web.

Para la dependencia externa de FontAwesome, para poder usar los iconos sólo tuve que añadir:

```
@import "npm:@fortawesome/fontawesome-free/css/all.css";
```

En `settings.scss`

Para los colores usados en la web creé un archivo “partial” de SASS que se encuentra en `1_settings/_colors.scss` donde se definen como variables.

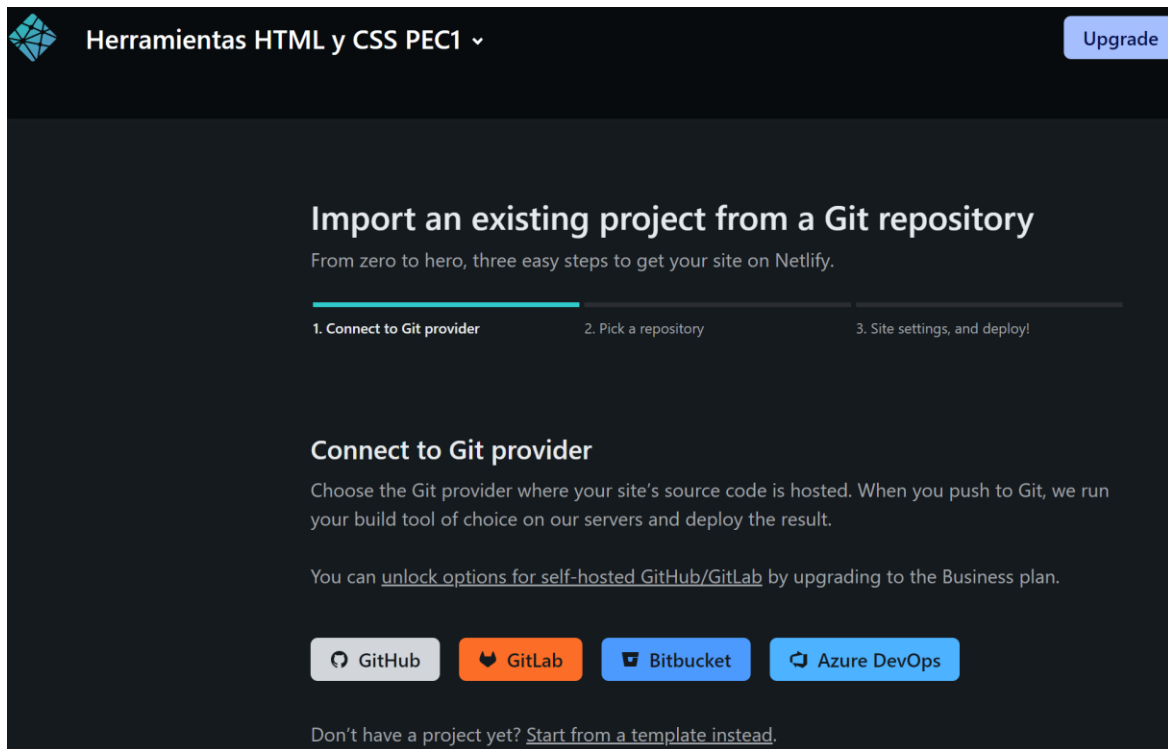
En `2_tools/tools.scss` defino un mixin para un tipo de “etiqueta” cuyo estilo se comparte entre dos Elements.

En el archivo `3_generic/generic.scss` sólo se definen las reglas para el elemento universal CSS *

Y finalmente no había necesidad de usar ninguna “utility” que corresponde con el último nivel del ITCSS debido de nuevo a la sencillez de la página.

Despliegue

Una vez todos los cambios están listos, se hace git commit y git push para empujar los cambios al repositorio remoto en github, el cual será utilizado por Netlify que permiter enlazar una cuenta de Github:



Una vez enlazada la cuenta de github y seleccionado el repositorio hay que elegir el branch y el comando para ejecutar la compilación del proyecto y su carpeta de destino, pero como estos datos ya se encuentran en package.json, Netlify los carga automáticamente.

Owner

Herramientas HTML y CSS PEC1 ▼

Branch to deploy

master ▼

Basic build settings

If you're using a static site generator or build tool, we'll need these

[Learn more in the docs](#) ↗

Base directory ⓘ

Build command ⓘ

npm run build

Publish directory ⓘ

dist

Show advanced

Deploy site


Una vez hecho click en “Deploy site” faltaría esperar a que termine el primer build para que sea posible acceder a la web cuyo enlace es autogenerado por Netlify:

luxury-duckanoo-f10108

- <https://luxury-duckanoo-f10108.netlify.app>

Deploys from [GitHub](#).

Last published at 8:28 PM.



Site settings Domain settings Favorite site

Finalmente se puede observar que la web carga correctamente

