

En esta práctica lo que más me ha costado ha sido conseguir que todo se viese como en las imágenes mockup de cada dispositivo, es decir, hacer un diseño responsive.

Como sugería la práctica, he decidido hacer un desarrollo *mobile first* y para ello he construido primero el html de cada página, después he desarrollado el css de dicha página para móvil, lo he retocado para tableta, y lo he dejado como se debía de ver en escritorios/pc (entre estas tres, el móvil y tableta compartían la mayoría, pero el pc no).

Fui trabajando de ésta manera, página por página, html, después css de cada dispositivo, hasta que terminé y retoqué algunos detalles.

Debo puntualizar que el *sticky* del nav decidí aplicarlo también al logo “IA al día”, simplemente me gustó como quedaba estéticamente, entonces al scrollear hacia abajo la página, también baja el logo.

Tanto la página 3 (en-el-cine) como la 4 (despacho-42), móvil y tablet comparten el mismo “grid”, por lo tanto el flujo de ítems html es muy parecido. En cambio en las páginas 1 (index) y 2 (aplicaciones) es distinto. En la página 1, tablet y pc comparten grid en las noticias (3, 1fr, es decir, tres por fila), y en móvil solo un elemento por fila. Luego, en la página 2, en móvil igual, un elemento por fila, pero en pc 3 elementos por fila y en tablet 2 elementos por fila. (para cambiar dichos elementos he usado `grid-template-columns` y en el html les he asignado una clase).

El resto de cambios en el responsive han sido simplemente sutiles en cuanto a tamaños, y para que, al hacerse pequeña la web, (es decir, pasar de más píxeles a menos) los elementos no estuviesen pegados a los bordes de la pantalla (he aumentado los `padding`s de elementos).

(He utilizado la clase `.card` para cada encapsulamiento de noticia, vídeo, aplicación o derivado)

En cuanto a etiquetas, un poco lo mismo que en prácticas anteriores, `time datetime` para fechas, `<q>` para las citas cortas, `<abbr>` para las abreviaturas (como IA), `<cite lang=en>` para un par de libros (que tienen su título en inglés), `article` para las entradas, un `nav` con una lista para el menú de navegación, y un `footer` con dos listas y al final un par de `small`s.

(y bueno, `p` para cada párrafo o línea de texto, headers desde `h1` hasta `h4` para los títulos, `div`s para contener estilo y encapsulamiento, `em` y `strong` para dar énfasis a palabras, `a` para los enlaces, `img` para algunas imágenes e `iframe` para los vídeos)

Al validar los html, tenía algún `article` sin header, entonces lo único que hice fue cambiar ese `article` por un `div`, los estaba usando para modificar el estilo, más que por semántica.

**LINK AL REPOSITORIO GITHUB:**

[Kaladin000/disDGV3.github.io](https://kaladin000.github.io/disDGV3.github.io/)

<https://kaladin000.github.io/disDGV3.github.io/>