

HTML Y CSS

# **Contrucción del sitio web “Al al día”**

## **PEC 3**

Alumno: Cecilia Sánchez Losada Flores

Junio, 2023.

## **Primera parte: Sección teórica**

### Pregunta 1

Sobre técnicas de posicionamiento y modos de diseño, responde a las siguientes preguntas:

1. Explica que es el normal document flow o normal layout flow (flujo de documento/diseño normal) y cómo se comportan los elementos de bloque en ese flujo.

La posición por defecto de los elementos contenidos en una página web **es el flujo normal de maquetación o normal layout flow**

El flujo normal es la forma en que el navegador presenta las páginas HTML por defecto cuando no se hace nada para controlar el diseño de la página.

Los elementos bloque en este flujo normal de maquetación **aparecen uno debajo de otro**, en contraste con los elementos en línea, que aparecen uno al lado del otro como las palabras individuales de un párrafo.

Empezar con un documento sólido, bien estructurado y legible en flujo normal es la mejor manera de empezar cualquier página web. Además, puesto que el flujo normal está diseñado para hacer un documento legible, al empezar de esta manera estás trabajando con el documento en lugar de luchar contra él mientras realizas cambios en la maquetación.

El flujo normal de maquetación (mencionado en el artículo de introducción a la maquetación) **es el sistema por el cual los elementos se colocan dentro de la ventana gráfica del navegador**. Por defecto, los elementos a nivel de bloque se disponen en la dirección de flujo del bloque, que se basa en el modo de escritura del padre (inicial: horizontal-tb). Cada elemento aparecerá en una nueva línea debajo del anterior, con cada uno separado por el margen que se haya especificado. En inglés, por ejemplo, (o en cualquier otro modo de escritura horizontal, de arriba abajo) los elementos de nivel bloque se disponen verticalmente.

Los elementos en línea se comportan de forma diferente. No aparecen en líneas nuevas, sino que se sitúan en la misma línea junto con cualquier contenido de texto adyacente (o envuelto) siempre que haya espacio para ello dentro de la anchura del elemento de nivel de bloque padre. Si no hay espacio, el contenido desbordante se desplazará a una nueva línea.

## 2. Explicad las diferencias básicas entre los módulos CSS Grid i Flebox.

Mientras que flexbox está diseñado para un **diseño unidimensional**, Grid Layout está diseñado para **dos dimensiones** - alinear cosas en filas y columnas.

Flexbox es el nombre abreviado del módulo CSS Flexible Box Layout, diseñado para facilitarnos la disposición de elementos en una dimensión, ya sea como fila o como columna. Para utilizar flexbox, se aplica display: flex al elemento padre de los elementos que se desea disponer; todos sus hijos directos se convierten entonces en elementos flex.

Si añadimos display: flex al elemento padre, los tres elementos se disponen ahora en columnas. Esto se debe a que se convierten en elementos flex y se ven afectados por algunos valores iniciales que flexbox establece en el contenedor flex. Se muestran en una fila porque la propiedad flex-direction del elemento padre tiene un valor inicial de row. Todos parecen estirarse en altura porque la propiedad align-items de su elemento padre tiene un valor inicial de stretch. Esto significa que los elementos se estiran hasta la altura del contenedor flexible, que en este caso está definido por el elemento más alto. Todos los elementos se alinean al principio del contenedor, dejando espacio extra al final de la fila.

Además de las propiedades que se pueden aplicar a un contenedor flexible, también hay propiedades que se pueden aplicar a los elementos flexibles. Estas propiedades, entre otras cosas, pueden cambiar la forma en que los elementos se flexionan, permitiéndoles expandirse o contraerse según el espacio disponible.

Como ejemplo sencillo, podemos añadir la propiedad flex a todos nuestros elementos hijo, y darle un valor de 1. Esto hará que todos los elementos crezcan y

llenen el contenedor, en lugar de dejar espacio al final. Si hay más espacio, los elementos se harán más anchos; si hay menos espacio, se harán más estrechos. Además, si añades otro elemento al marcado, los demás elementos se harán más pequeños para dejarle espacio; los elementos juntos seguirán ocupando todo el espacio.

## Pregunta 2

Sobre diseño responsivo, responde a las siguientes preguntas:

1. Explica los siguientes conceptos en relación con el diseño responsivo: breakpoint, mobile first y media query.

El diseño web responsivo (RWD) es un enfoque de diseño web para que las páginas web se visualicen bien en todos los tamaños y resoluciones de pantalla, garantizando al mismo tiempo una buena usabilidad. Es la forma de diseñar para una web multidispositivo.

El diseño web responsivo no es una tecnología independiente, sino un enfoque. Es un término utilizado para describir un conjunto de buenas prácticas utilizadas para crear un diseño que pueda responder a cualquier dispositivo que se utilice para ver el contenido.

Las **media queries** nos permiten ejecutar una serie de pruebas (por ejemplo, si la pantalla del usuario es mayor que una determinada anchura, o una determinada resolución) y aplicar CSS de forma selectiva para estilizar la página adecuadamente a las necesidades del usuario.

Es habitual aplicar una **media query** basada en el tamaño de la ventana gráfica para poder elegir el diseño para dispositivos con diferentes tamaños de pantalla.

Se pueden añadir varias **media queries** en una hoja de estilos para ajustar el diseño completo o partes del mismo a los distintos tamaños de pantalla.

Los puntos en los que se introduce una **media query** y se modifica el diseño se conocen como **breakpoints**.

El diseño mobile first, una forma de mejora progresiva, es un enfoque de desarrollo y diseño web que se enfoca en **la priorización del diseño y el desarrollo para dispositivos móviles** por encima del diseño y desarrollo para pantallas de escritorio. El fundamento detrás del enfoque "mobile-first" es proveer al usuario una buena experiencia para todos los tamaños de pantalla—empezando por una experiencia de usuario que funcione bien para dispositivos pequeños, para posteriormente, basada en dicha experiencia, continuar desarrollando para enriquecer la experiencia de usuario conforme el tamaño de pantalla es mayor. El enfoque "mobile-first" contrasta con el antiguo enfoque de diseño para escritorio en el que se pensaba primero en el diseño/desarrollo para tamaños de pantalla de escritorio y posteriormente se realizaban ajustes para conseguir que dichos diseños y desarrollos funcionasen correctamente en dispositivos móviles.

2. Explicad de qué manera el elemento picture puede ayudar a implementar soluciones para incluir imágenes responsivas dentro de un sitio web. Aportad algún ejemplo de código que apoye la explicación.

Para garantizar que los medios nunca sean más grandes que su contenedor responsive, se puede utilizar el siguiente enfoque:

```
img,  
imagen,  
vídeo {  
  max-width: 100%;  
}
```

Las imágenes responsivas, mediante el elemento <picture> y los atributos <img> srcset y sizes, permite servir imágenes orientadas a la ventana gráfica y la resolución del dispositivo del usuario. Por ejemplo, puede incluir una imagen

cuadrada para móviles, pero mostrar la misma escena que una imagen apaisada en escritorio.

El elemento `<picture>` permite proporcionar múltiples tamaños junto con "hints" (metadatos que describen el tamaño de pantalla y la resolución para los que la imagen es más adecuada), y el navegador elegirá la imagen más apropiada para cada dispositivo, asegurándose de que un usuario descargará un tamaño de imagen apropiado para el dispositivo que está utilizando. El uso de `<picture>` junto con `max-width` elimina la necesidad de dimensionar las imágenes con media queries. Permite dirigir imágenes con diferentes relaciones de aspecto a diferentes tamaños de ventana gráfica.

También puede dirigir imágenes utilizadas en diferentes tamaños, proporcionando así un recorte diferente o una imagen completamente diferente a diferentes tamaños de pantalla.

Al igual que `<video>` y `<audio>`, el elemento `<picture>` es una envoltura que contiene varios elementos `<source>` que proporcionan diferentes fuentes para que el navegador elija, seguidos del importantísimo elemento `<img>`. El código en `responsive.html` tiene este aspecto:

```
<picture>
  <source media="(max-width: 799px)" srcset="elva-480w-close-portrait.jpg" />
  <source media="(min-width: 800px)" srcset="elva-800w.jpg" />
  
</picture>
```

## **SEGUNDA PARTE: PROTOTIPO WEB.**

Notas y explicaciones sobre la construcción del sitio web “IA al día”.

Después de analizar los mockups y los requerimientos de la web, he decidido hacer la estructura general haciendo uso de Grid, para que los elementos estructurales generales de la web se ajusten a los diferentes breakpoints de manera más óptima y con menos repetición de código. Ha sido interesante revisitar la teoría sobre maquetación en base a cuadrícula, así que partiré de ella para levantar el sitio web.

El proyecto está conformado por cuatro archivos HTML en el mismo nivel de la carpeta de recursos (*/img*) y un directorio css que incluye el archivo de estilos general para todos los documentos con extensión .html.

Para la estructura general de los archivos HTML:

1. En la sección principal se encuentra el encabezado de la web, que incluye logo y los elementos de navegación. (<nav>)
2. La sección main, abraza el contenido específico de cada página.
3. La sección footer incluye el pie de página de la web.

Se han utilizado etiquetas <section> para estructurar cada sección diferenciada de cada página. Así mismo se han incorporado comentarios dentro de la estructura html para ubicar más fácilmente cada una de las diferentes secciones de acuerdo a su contenido.

El archivo de estilos (style.css) está dividido en secciones lógicas para su mayor entendimiento.

- La parte inicial del archivo alberga las reglas globales y generales que comparten todas las páginas.
- **Main structure:** En esta sección se encuentran las reglas de estilos correspondientes al logo.

- **Navigation Bar:** Aquí se encuentran todas las reglas relacionadas al menú de navegación.
- **Main content:** Incluye todo el contenido de las páginas. Las reglas para cada una de ellas están marcadas con sus nombres correspondientes:
  - App Page: Incluye las reglas de la página "Aplicaciones".
  - Films page: Incluye las reglas de la página "en el Cine".
  - Despacho 42: Incluye las reglas de la página "Despacho 42".
    - Form: incluye reglas relacionadas al formulario de la página de la noticia "Despacho 42"
- Footer incluye las reglas relacionadas al pie de página global de la web.

En cada apartado se incluyen algunos comentarios puntuales sobre reglas que generan un comportamiento específico e importante de señalar.

Se comparten reglas de estilo, para no repetir código, entre las diferentes páginas cuyo comportamiento se espera sea idéntico.

Para el diseño responsivo, se utilizan media queries con breakpoints especialmente diseñados para móvil, tablet y escritorio. Todos ellos se encuentran justo después de la regla de estilo que requiera un comportamiento específico en cada dimensión de la pantalla.

Para la entrega del prototipo final de la web **Al al día**, se usa Github pages, ya que github permite seguir el historial de versiones y genera una [URL](#) que se actualiza con cada cambio realizado en el repositorio.

Github Pages url:

- <https://cecisalof.github.io/MaterialesPEC3/>



