

Práctica 1 : HTML y CSS

La presente memoria tiene como objetivo documentar el proceso de diseño y creación de la página web que he diseñado. En esta memoria se describirán las decisiones tomadas en cuanto al diseño y estructura de la página, así como los aspectos técnicos relacionados con la implementación de la misma.

Estructura HTML de las páginas principales

Lo primero que veremos es cómo está diseñado el header, el cual es compartido por todas las páginas de la web. La estructura HTML que sigue es la siguiente :

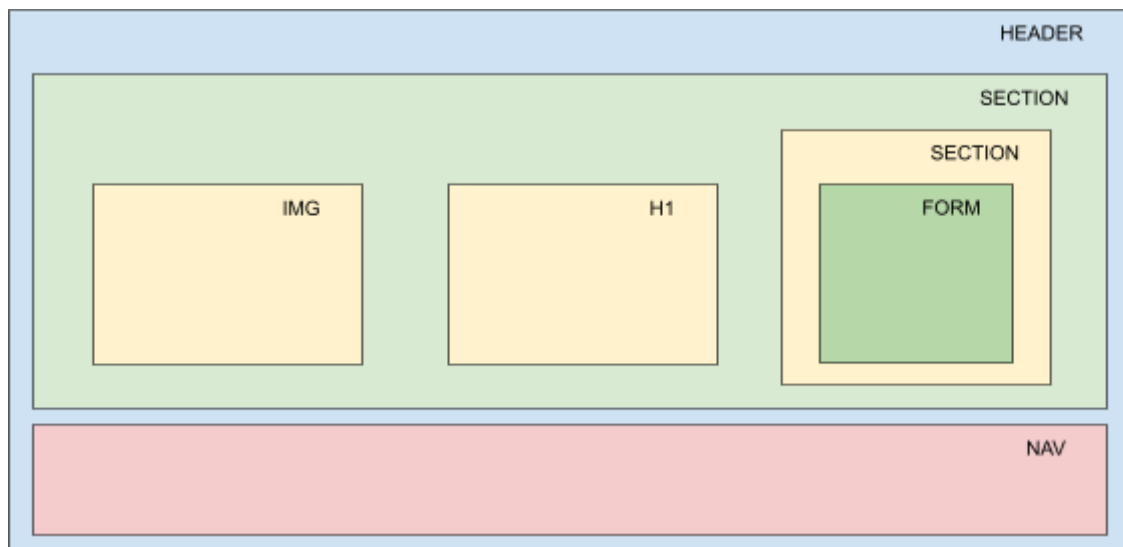


Fig 1 : Diseño HTML del HEADER de la página web

A continuación veremos el diseño HTML del main de la página principal (index.html).

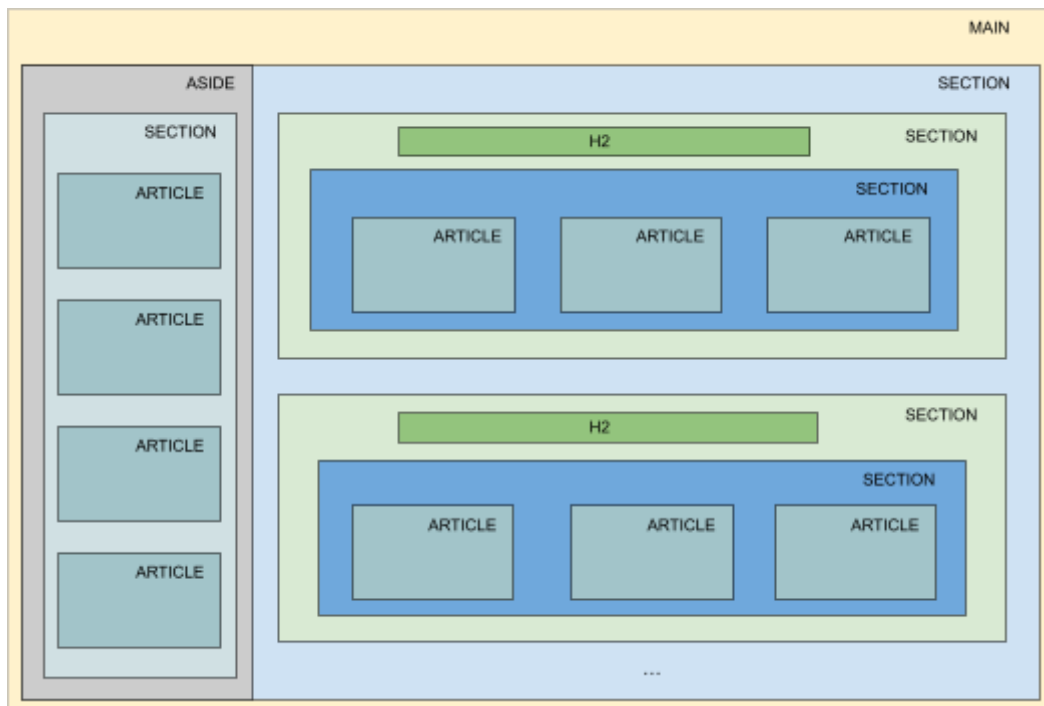


Fig 2 : Diseño HTML del MAIN de la página principal (index.html)

Por último, veremos cómo ha sido diseño HTML de la parte principal correspondiente a las películas (*pel1.html*, *pel2.html*, *pel3.html*)

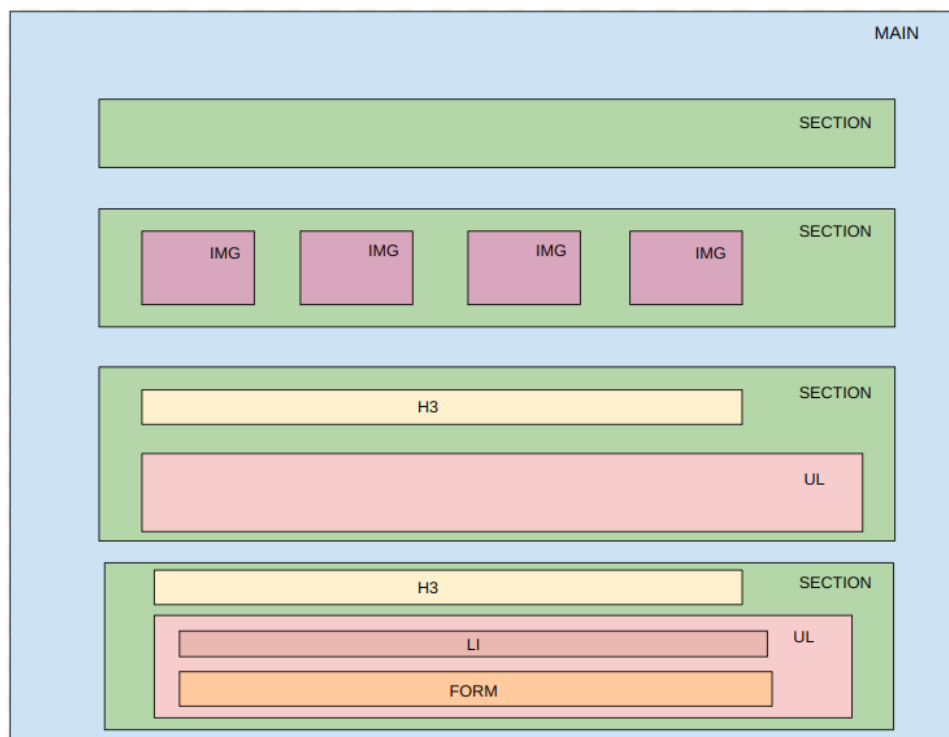


Fig 3 : Diseño HTML del MAIN de la página de una película (pel1.html)

FlexBox

En el diseño de la página web, se ha utilizado la técnica de FlexBox para ordenar los elementos y lograr un diseño flexible y adaptable. Flexbox es un modelo de diseño en CSS que permite organizar los elementos en un contenedor de manera más eficiente, ya que se ajusta automáticamente al tamaño de la pantalla y al contenido del contenedor.

Para implementar FlexBox, se debe crear un contenedor (que sea el que engloba los elementos que queremos alinear) con la propiedad **display : flex**. Por defecto, los elementos se dispondrán en horizontal, y en el caso que queramos que se ordenen de forma vertical, deberemos cambiarlo con la propiedad **flex-direction : column**. Para alinear los elementos tanto vertical como horizontalmente dentro del contenedor, disponemos de las propiedades **justify-content** (para alinear horizontalmente) y **align-items** (para alinear verticalmente) cuando **flex-direction : row**.

Diseño Responsive

La página principal (*index.html*) ha sido desarrollada para ser adaptable al tamaño de pantalla de un teléfono. Para ello hemos hecho uso de las media-queries, las cuales nos permiten adaptar el estilo y diseño de la página en función de las características de la pantalla y del uso de **o bien porcentajes, rem o vw** como medidas para que se ajusten al tamaño de la pantalla . En mi diseño he tomado un máximo de 1024px de anchura, es decir, que para anchos de pantalla menores a esta cantidad, se aplicaran todos los cambios de diseño.

Entre los cambios más significativos

- El main queda organizado de manera que ahora aparecen la sección de películas seguido de la sección de las noticias gracias a **flex-direction : column-reverse**.
- He cambiado a **flex-direction : column**, tanto en la cabecera, navegación y en la sección donde se encuentran las portadas de las películas, que ahora aparecen una debajo de otra.
- **flex-direction : row** en la sección de noticias, lo que significa que se colocan las noticias una a lado de la otra. Además he cambiado a **overflow-x : scroll** lo que supone que el desbordamiento del bloque se produce en el eje horizontal.
- Además he ajustado algunos *margin* y *padding* de algunos elementos.

Técnicas de Diseño

A continuación veremos algunas de las técnicas de diseño que he implementado :

- **Información Oculta (Información de Películas)** : Permite ocultar la información relacionada con la película hasta el momento en el que pasamos con el ratón por encima. Además la información aparecerá de forma progresiva. Podemos ver su uso tanto en el *index.html*, *estrenos.html* y *cartelera.html* :

Unset

```
.datos-pelicula {  
    ...  
    // Posicionar un elemento de forma relativa a su posición original sin  
    // afectar a la posición de los elementos adyacentes  
    position: relative;  
    // Movemos hacia arriba el elemento 250px de su posición original para que  
    // el bloque no se salga fuera del espacio correspondiente a la película  
    top: -250px;  
    // Ocultamos el contenido del bloque  
    opacity: 0;  
}  
  
// Los dos siguientes bloques se "ejecutan" cuando el ratón pase por encima  
.pelicula:hover .datos-pelicula {  
    // Reajustamos la ubicación a la original  
    top: 0px;  
    // Mostramos el contenido  
    opacity: 1;  
    // Se realiza una transición suave sobre la propiedad opacity durante 2.5s  
    transition: opacity 2.5s ease;  
}  
  
.pelicula:hover .foto-pelicula{  
    // Reajustamos el tamaño de la imagen  
    height: 50%;  
    // Se realiza una transición suave sobre el movimiento de la imagen que  
    // dura 0.7s cuando se reajusta al 50% (propiedad anterior)  
    transition: height 0.7s ease;  
    // Se ajusta el borde superior izquierdo de la imagen
```

```
border-top-left-radius: 10%;
}
```

- **Animación de un elemento** : Permite animar de manera progresiva la aparición de un elemento HTML al cargar la página. Podremos ver su uso en la página *información.html* :

Unset

```
.animacion {
// Permite que la animacion se muestre letra por letra
display: inline-block;

// Oculta el texto mientras se está cargando
overflow: hidden;

// Animacion de nombre aparecer, que dura 1.5s, se divide en 40 pasos y que
se inicia al cargar la página
animation: aparecer 1.5s steps(40) 0s forwards;
}

// Definimos la animación que establece que el ancho del elementos se
inicia en 40% hasta el 80%
@keyframes aparecer {
  from { width: 40%; }
  to { width: 80%; }
}
```

- **Desbordamiento de un elemento** : Permite establecer el comportamiento de desbordamiento horizontal o vertical de un elemento. Lo podemos encontrar en la página principal *index.html*.

Unset

```
.contenedor-principal {
...
// Se produce desbordamiento del elemento en el eje y, es decir, el
elemento que lo contiene es de menor tamaño y el contenido que se deborda se
oculta por debajo y para acceder a el hay que hacer scroll
overflow-y: scroll;
}
```

- **Selección de las columnas pares de una tabla :** Nos permite seleccionar cada elemento que es un hijo par o impar de su elemento padre. Podemos encontrar esta técnica en el documento *horarios.html*.

Unset

```
// Selecciona todas las filas pares de una tabla
tr:nth-child(even) {
    background-color: lightblue;
}

// Selecciona todas las filas impares de una tabla
tr:nth-child(odd) {
    background-color: rgba(101, 117, 144, 0.386);
}
```