# Memoria U1-A1

## 1. Introducción

El objetivo de este trabajo ha sido construir una maqueta de página web empleando HTML5, CSS y Bootstrap 5.  
La práctica se ha centrado en el uso del sistema de grid de Bootstrap, así como en el reordenamiento de elementos mediante las clases order y clases de columnas responsive; para realizar un diseño que se adapte a tres tamaños de pantalla diferentes.

El diseño final consiste en varias secciones diferenciadas: HEADER, MENU, HERO, MAIN, BANNER, EXTRA e IMAGE, cada una con un color de fondo para visualizar con claridad cómo se reparten los espacios y cómo se comportan al modificar el tamaño de la ventana.

## 2. Estructura del código HTML

### 2.1 Inclusión de recursos

En la cabecera (<head>) se enlaza Bootstrap 5.3.8 desde un CDN y, adicionalmente, se incluye un fichero externo styles.css con estilos propios. Esto permite heredar la potencia de Bootstrap (clases predefinidas, rejilla, tipografía, utilidades) y al mismo tiempo personalizar el aspecto.

### 2.2 Contenedor principal

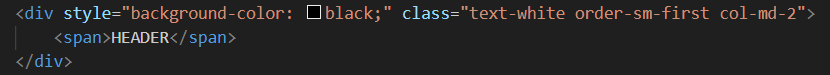
El <body> lleva la clase container de Bootstrap, que centra y aplica márgenes horizontales automáticos. Dentro se define un <main> con la clase row, que servirá como fila general de la que dependen todas las secciones.

### 2.3 Distribución de secciones

Cada bloque (HEADER, MENU, HERO, MAIN, BANNER, EXTRA, IMAGE) se implementa como un <div> con clases de Bootstrap que controlan:

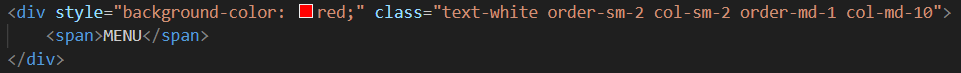
* Tamaño en rejilla: col-sm-2, col-md-10, etc.
* Orden de visualización: order-sm-first, order-md-last…
* Color de fondo: aplicado con style en línea para distinguirlos claramente.
* Color del texto: con la clase text-white.

Ejemplo: HEADER



Este bloque ocupa 2 columnas en pantallas medianas (md), se pinta de negro y se asegura de aparecer el primero en pantallas pequeñas gracias a order-sm-first.

Ejemplo: MENU



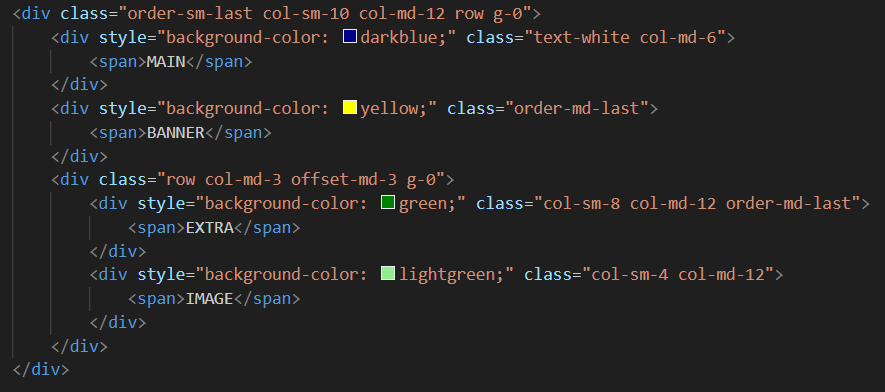
Aquí se observa el uso de dos órdenes distintos:

* En pantallas pequeñas (sm), ocupa 2 columnas y se muestra en segunda posición.
* En pantallas medianas (md), ocupa 10 columnas y pasa a la primera posición.

Con este enfoque, la disposición de los bloques cambia dinámicamente según el ancho de la pantalla, logrando un diseño totalmente responsive.

### 2.4 Sección compuesta (MAIN + BANNER + EXTRA + IMAGE)

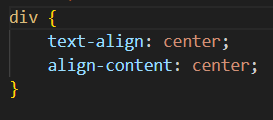
La parte más compleja del código es la que combina varias columnas dentro de una misma fila:



* El bloque global ocupa toda la anchura en md (12 columnas) y contiene varias subdivisiones.
* MAIN ocupa media pantalla (6 columnas).
* BANNER aparece después, con la posibilidad de colocarse al final en pantallas medianas.
* Finalmente, dentro de una sub-grid se colocan EXTRA e IMAGE, distribuidos 8/4 en pantallas pequeñas, y cada uno con el 100% de ancho en pantallas medianas.

## 3. Estilos CSS personalizados

El fichero styles.css es muy sencillo pero importante. Contiene:



Esto asegura que el texto dentro de cada bloque quede centrado horizontalmente (text-align: center) y que el contenido vertical se alinee de manera equilibrada (align-content: center). Aunque el uso de align-content tiene más efecto en contenedores flex, aquí sirve como ajuste general de presentación.

## 4. Retos y soluciones

1. Gestión de órdenes (order):  
   Fue necesario jugar con las clases order-sm-\* y order-md-\* para que la maqueta tuviera una estructura diferente según el tamaño de pantalla. Esta es la clave de la responsividad.
2. Anidación de filas:  
   Para ubicar EXTRA e IMAGE en distintas proporciones dependiendo del tamaño de pantalla, se anidó una fila (row) dentro de una columna principal.