

Índice General

- Descripción de la tarea..... 2
 - Requisitos.....3
 - Rúbrica de Evaluación..... 4
- Código HTML/CSS (Maquitación con Grid).....6
- Razonamiento..... 8
- Resultado..... 12
- Reflexión..... 12

LM

LM03 – UT03 – Tarea 05: Maquetación con Grid

El objetivo de esta tarea es diseñar **desde cero** una página web utilizando **CSS Grid** para obtener un resultado similar al prototipo mostrado en la imagen: cabecera superior, columna de anuncios a la izquierda y zona de artículos a la derecha.

Descripción de la tarea

Debes crear:

- Un archivo `index.html`.
- Un archivo `estilos.css`.

La página debe mostrar:

- Una **cabecera** en la parte superior.
- Una **columna de anuncios** en el lateral izquierdo.
- Una **zona de artículos** en el lateral derecho, con al menos dos artículos.



Requisitos

R1. Estructura general Organizar la página en tres zonas claras: cabecera, anuncios y contenido principal con artículos.

R2. Cabecera La cabecera debe ocupar todo el ancho y mostrar un mapa.

R3. Columna de anuncios La columna de anuncios debe situarse a la izquierda, con fondo rojo o similar, y contener al menos tres anuncios diferenciados.

R4. Zona de artículos La zona de artículos debe situarse a la derecha, con fondo verde o similar, y mostrar al menos dos artículos, cada uno en un bloque claramente separado.

R5. Espaciado y márgenes Debe existir espacio razonable entre columnas y entre bloques de contenido (márgenes, rellenos, separación entre elementos).

R6. Restricciones de maquetación No está permitido usar `position` (en ninguna variante), `float` ni tablas HTML para maquetar. La solución debe basarse en CSS Grid.

R7. Formato de entrega La entrega debe incluir exactamente:

- `index.html`
 - `estilos.css`
 - Las imágenes utilizadas
 - Un **PDF** con:
 - Breve explicación de las decisiones de diseño (cómo has organizado la rejilla y las zonas).
 - Capturas de pantalla del resultado final.
-

Rúbrica de Evaluación (0 – 3 – 6 – 9 – 10)

Nota	Descripción
0	No se puede visualizar la página, falta alguno de los archivos o no se usa Grid.
3	La estructura general es confusa; no se distinguen bien cabecera, anuncios y artículos o el uso de Grid es muy limitado/incorrecto.
6	Se reconocen cabecera, columna de anuncios y zona de artículos, pero hay problemas importantes de tamaño, alineación o estética. El PDF es muy escueto o incompleto.
9	La página se parece claramente al prototipo, las tres zonas están bien resueltas con Grid y el diseño es coherente. Solo hay detalles menores mejorables. El PDF explica razonablemente las decisiones tomadas.
10	Hay un aporte personal extra que se explica y se solicita claramente la nota

Código HTML/CSS (Maquitación con Grid)

```
1  <!DOCTYPE html>
2  <html lang="es">
3
4  <head>
5      <meta charset="UTF-8" />
6      <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7      <title>LM03_UT03_Tarea_05_CSS Grid</title>
8      <link rel="stylesheet" href="estilos.css" />
9  </head>
10
11 <body>
12
13     <header class="cabecera"></header>
14
15     <aside class="anuncios">
16         <article id="anuncio1">
17             <h2>Anuncio 1</h2>
18             <p>
19                 Este es el desarrollo del anuncio 1. Voy a poner varias palabras para
20                 que ocupe varias líneas.
21             </p>
22         </article>
23         <article id="anuncio2">
24             <h2>Anuncio 2</h2>
25             <p>
26                 Este es el desarrollo del anuncio 2. Voy a poner varias palabras para
27                 que ocupe varias líneas.
28             </p>
29         </article>
30         <article id="anuncio3">
31             <h2>Anuncio 3</h2>
32             <p>
33                 Este es el desarrollo del anuncio 3. Voy a poner varias palabras para
34                 que ocupe varias líneas.
35             </p>
36         </article>
37     </aside>
38
39     <section class="articulos">
40         <article id="articulo1">
41             <h2>Artículo 1</h2>
42             <div id="contenido1">
43                 
44                 <p>
45                     Este es el artículo 1. Voy a poner varias palabras para que ocupe varias líneas. Voy a poner varias
46                     palabras para que ocupe varias líneas. Voy a poner varias palabras para que ocupe varias líneas. Voy
47                     a poner varias palabras para que ocupe varias líneas.
48                 <br>
49                 <br>
50                     Este es el final del artículo 1. Voy a poner varias palabras para que ocupe varias líneas. Voy a
51                     poner varias palabras para que ocupe varias líneas. Voy a poner varias palabras para que ocupe
52                     varias líneas. Voy a poner varias palabras para que ocupe varias líneas.
53                 </p>
54             </div>
55         </article>
56
57         <article id="articulo2">
58             <h2>Artículo 2</h2>
59             <div id="contenido2">
60                 <p>
61                     Este es el artículo 2. Voy a poner varias palabras para que ocupe varias líneas. Voy a poner varias
62                     palabras para que ocupe varias líneas. Voy a poner varias palabras para que ocupe varias líneas.
63                 <br>
64                     Este es el final del artículo 1. Voy a poner varias palabras para que ocupe varias líneas. Voy a
65                     poner varias palabras para que ocupe varias líneas. Voy a poner varias palabras para que ocupe
66                     varias líneas. Voy a poner varias palabras para que ocupe varias líneas.
67                 </p>
68                 
69             </div>
70         </article>
71     </section>
72 </body>
73
74 </html>
```



```

1  /* Reinicio general: elimina márgenes/padding por defecto y asegura el alto completo. */
2  html,
3  body {
4      margin: 0;
5      padding: 0;
6      min-height: 100vh
7  }
8
9  body {
10     /* Contenedor principal como cuadrícula: define qué bloque ocupa cada área. */
11     display: grid;
12     /* Permite ajustar la estructura en la que queremos ver la página */
13     grid-template-areas:
14         "header header header"
15         "aside article article"
16         ;
17
18     grid-template-columns: 1fr 1fr 2fr;
19     /* Altura de las filas: la primera fija y las otras adaptables */
20     grid-template-rows: 200px 1fr;
21
22 }
23
24 .cabecera {
25     grid-area: header;
26     background-image: url("UT2 - TB banner.jpg");
27     background-size: cover;
28     background-position: center;
29     background-repeat: no-repeat;
30
31     height: 200px;
32     width: auto;
33 }
34
35 .anuncios {
36     grid-area: aside;
37     background-color: #f30a0a;
38     padding: 20px;
39 }
40 h2 {
41     color: #fbff00;
42     font-family: serif ;
43 }
44
45 p {
46     color: #ffffff;
47     font-weight: bold;
48     font-family: serif ;
49     font-size: 17px;
50 }
51
52 .articulos {
53     grid-area: article;
54     background-color: #04810e;
55     padding: 30px;
56     /* Flex column: apila los artículos verticalmente conservando su orden. */
57     display: flex;
58     flex-direction: column;
59
60 }
61
62 #contenido1, #contenido2{
63     border-style: solid;
64     border-width: 2px;
65     padding: 40px;
66     border-color: #fbff00 ;
67
68 }
69 #articulo1 {
70     margin-bottom: 20px;
71 }
72
73 #contenido1 p {
74     grid-area: parrafo1;
75 }
76 #contenido2 p {
77     grid-area: parrafo2;
78 }
79
80 #contenido1 img {
81     grid-area: imagen1;
82 }
83 #contenido2 img {
84     grid-area: imagen2;
85 }
86
87 #contenido1 {
88     display: grid;
89     grid-template-areas:
90         "imagen1 parrafo1 parrafo1 "
91         "imagen1 parrafo1 parrafo1"
92         "imagen1 parrafo1 parrafo1"
93         ;
94     grid-template-columns: 1fr 1fr 2fr;
95     grid-gap: 20px;
96 }
97
98 #contenido2 {
99     display: grid;
100     grid-template-areas:
101         "parrafo2 parrafo2 imagen2 "
102         "parrafo2 parrafo2 imagen2"
103         "parrafo2 parrafo2 imagen2"
104         ;
105     grid-template-columns: 2fr 1fr 1fr;
106     grid-gap: 20px;
107
108 }
109

```


Razonamiento

1- Primeramente quise darle a cada parte de los elementos su supuesto formato (colores, negrita, tamaño de letra, estructura, fondo).

2- Seguidamente empecé por los anuncios (elementos a la izquierda de la página), asignándole unos márgenes adecuados o razonables para que todo se viera con mejor formato:

```
1  .anuncios {
2      grid-area: aside;
3      background-color: #f30a0a;
4      padding: 20px;
5  }
```

3- Posteriormente, me puse con los bordes y el formato de los artículos, para ello utilice la etiqueta aside para contener a los dos artículos, y para colocar el nombre del artículo (Artículo 1, Artículo 2) fuera del borde, cree un identificador justo después del h1 (1 para cada contenido).

```
1  <section class="articulos">
2      <article id="articulo1">
3          <h2>Artículo 1</h2>
4          <div id="contenido1">
5              
6              <p>
7                  Este es el artículo 1. Voy a poner varias palabras para que ocupe varias líneas. Voy a poner varias
8                  palabras para que ocupe varias líneas. Voy a poner varias palabras para que ocupe varias líneas. Voy
9                  a poner varias palabras para que ocupe varias líneas.
10                 <br>
11                 <br>
12                 Este es el final del artículo

```

4- Luego, aplique a cada elementos que formaba la estructura (cabecera, anuncios, artículos) sus respectivos grid-area para poder utilizarlo más adelante en el body, donde utilice la propiedad grid y le di formato a las posiciones de cada elemento para que se distribuyeran por la pantalla, aplicando un grid-template-columns de 1-1-2:

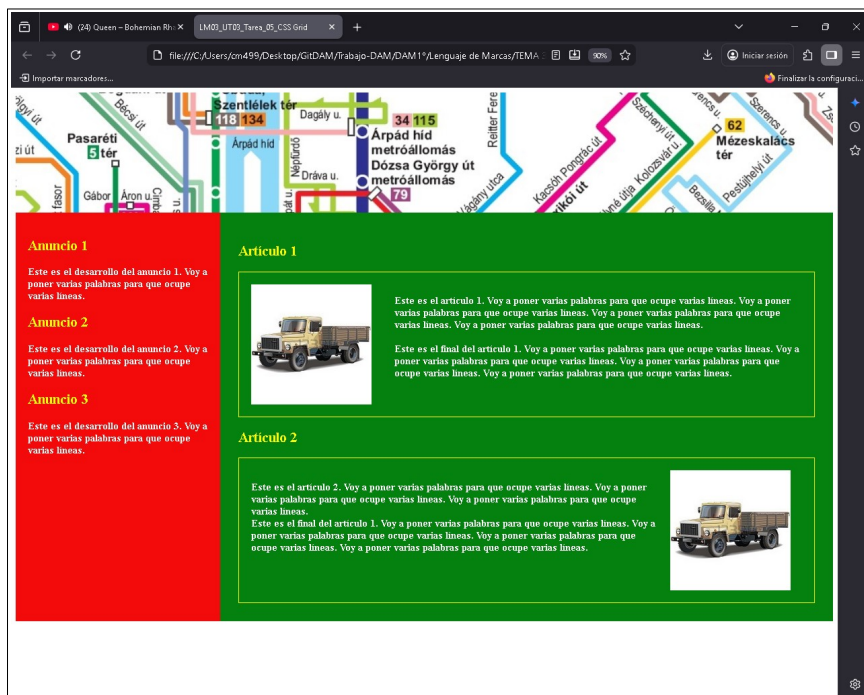
```
1  body {
2      /* Contenedor principal como cuadrícula: define qué bloque ocupa cada área. */
3      display: grid;
4      /* Permite ajustar la estructura en la que queremos ver la página */
5      grid-template-areas:
6          "header header header"
7          "aside article article"
8      ;
9
10     grid-template-columns: 1fr 1fr 2fr;

```


5- En los artículos, para que quedaran uno encima de otro, utilice la propiedad flex column (propiedad que estaba permitida ya que no decía nada sobre él, nada más sobre position y float).

```
1 .articulos {
2   grid-area: article;
3   background-color: #04810e;
4   padding: 30px;
5   /* Flex column: apila los articulos verticalmente conservando su orden. */
6   display: flex;
7   flex-direction: column;
8 }
9 }
```

6- Sin embargo, hubo varias complicaciones. Una vez terminado de dar forma a la página con grip, los elementos no ocupaban la pantalla completa y dejaban bordes blancos por fuera. Por lo que para arreglarlo, hice que el body y el html ocuparan la pantalla completa con la propiedad height y elimine los márgenes y padding que se habían generado por defecto para que no dejaran espacios por fuera.



```
1 /* Reinicio general: elimina márgenes/padding por defecto y asegura el alto completo. */
2 html,
3 body {
4   margin: 0;
5   padding: 0;
6   min-height: 100vh
7 }
```

7- Finalmente, para poder ver las imágenes de los artículos al lado, primeramente utilicé la propiedad flex row (primeramente se veía bien y funciono sin embargo luego de mucho tiempo me di cuenta que eran causa de un error) para poder ver las imágenes y les aplique un gap como margen

entre los elementos. Desafortunadamente, esto me trajo problemas al haber un pequeño margen blanco entre la imagen y los elementos de abajo (poco visible a la vista, pero si agrandabas la pantalla o el monitor es más grande se nota más).



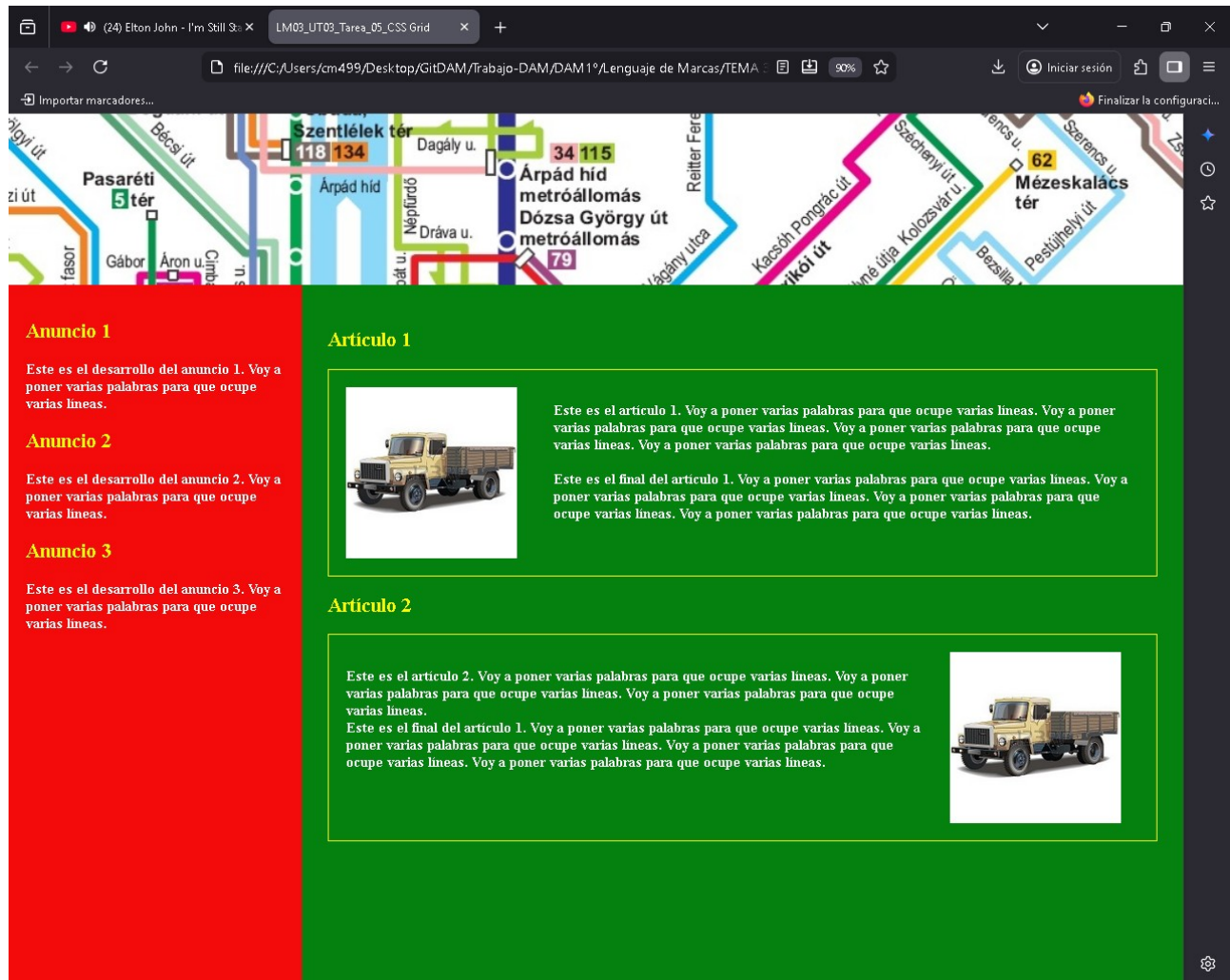
Después de varias horas, buscando varias respuestas y mucho sufrimiento, pensé en dejarlo así, no obstante, en el último momento me decante por realizarlo de nuevo utilizando otro grid para el texto y la imagen. Establecí a cada párrafo e imagen su determinado identificador para poder utilizarlo en su propio contenedor (contenido1,2).

```
1  #contenido1 p {
2    grid-area: parrafo1;
3  }
4  #contenido2 p {
5    grid-area: parrafo2;
6  }
7
8  #contenido1 img {
9    grid-area: imagen1;
10 }
11 #contenido2 img {
12   grid-area: imagen2;
13 }
14
15 #contenido1 {
16   display: grid;
17   grid-template-areas:
18     "imagen1 parrafo1 parrafo1 "
19     "imagen1 parrafo1 parrafo1"
20     "imagen1 parrafo1 parrafo1"
21   ;
22   grid-template-columns: 1fr 1fr 2fr;
23   grid-gap: 20px;
24 }
25
26 #contenido2 {
27   display: grid;
28   grid-template-areas:
29     "parrafo2 parrafo2 imagen2 "
30     "parrafo2 parrafo2 imagen2"
31     "parrafo2 parrafo2 imagen2"
32   ;
33   grid-template-columns: 2fr 1fr 1fr;
34
35   grid-gap: 20px;
36 }
```


8- Luego de tanto buscar y buscar el problema, encontré la una propiedad de grip (grid-template-rows), que la configura con los valores: grid-template-rows: 200px 1fr; (Esta propiedad permite establecer 200px para el header (coincide con el tamaño que le puse a la clase .cabecera) y 1fr para el resto (para que pudiera ocupar todo el espacio sobrante)). Finalmente los pude solucionar.

```
1  body {  
2    /* Contenedor principal como cuadrícula: define qué bloque ocupa cada área. */  
3    display: grid;  
4    /* Permite ajustar la estructura en la que queremos ver la página */  
5    grid-template-areas:  
6      "header header header"  
7      "aside article article"  
8      ;  
9  
10   grid-template-columns: 1fr 1fr 2fr;  
11   /* Altura de las filas: la primera fija y las otras adaptables */  
12   grid-template-rows: 200px 1fr;  
13  
14 }
```


Resultado



Reflexión

Al principio, la verdad parecía un trabajo fácil de realizar, luego al pasar el tiempo aparecieron varias complicaciones que me hicieron poner más empeño al asunto, pero la gota que colmo el vaso fue el incidente del flex row (dejaba un borde blanco entre la imagen y el contenido) que la verdad, era un detalle insignificante pero quise solucionarlo, estuve más de 2h buscando el fallo y ya lo iba a dejar así, hasta que creí que era cosa del flex y lo cambie por un grip. Finalmente encontré la solución. En conclusión, hubo un percance muy frustrante, pero todo se pudo solucionar.