



## EJERCICIOS UD3 – 3.4 – CSS GRID

1. Escribe el código HTML y CSS para definir una cuadrícula con tres columnas de igual tamaño utilizando la propiedad `grid-template-columns`. Luego, utiliza la propiedad `column-gap` para establecer un espacio de 20 píxeles entre las columnas y la propiedad `row-gap` para establecer un espacio de 10 píxeles entre las filas.

Además, da los estilos correspondientes para obtener la siguiente apariencia. Los elementos de la cuadrícula se deben adaptar en todo momento al tamaño de la pantalla:

|   |   |   |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |

2. Escribe el código necesario para obtener el siguiente resultado con Grid. Para ello utiliza las siguientes propiedades como mínimo: `grid-gap`, `grid template-areas` y `justify-items`

| Usando grid-areas |         |         |
|-------------------|---------|---------|
| celda 2           | celda 3 | celda 4 |
|                   | celda 5 | celda 6 |

3. Escribe el código necesario para obtener el siguiente resultado con Grid. Para ello utiliza las siguientes propiedades como mínimo: `grid-gap`, `grid template-areas` y `justify-items`

|         |                                                                             |
|---------|-----------------------------------------------------------------------------|
| celda 1 |                                                                             |
| celda 2 | celda 3:<br>contenido que marca el alto de la<br>parte central de la página |
| celda 4 |                                                                             |



4. Escribe el código necesario para obtener el siguiente resultado con Grid. Para ello utiliza las siguientes propiedades como mínimo: `grid-gap`, `grid template-areas` y `justify-items`



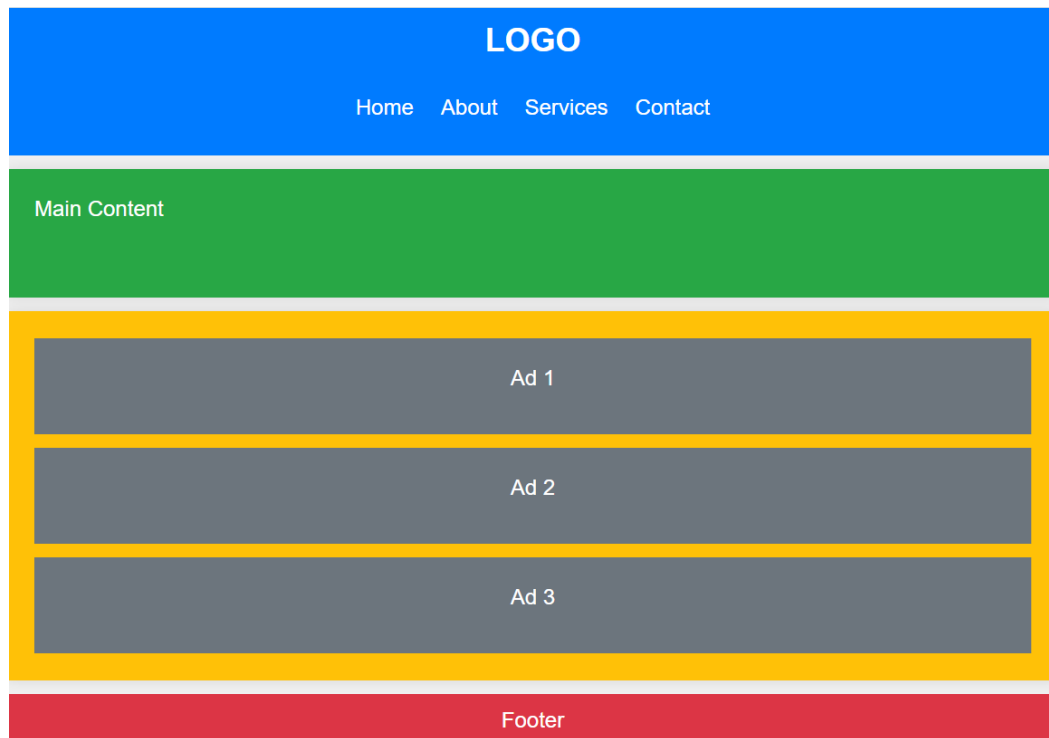
5. Dado el código HTML adjunto, escribe el código CSS correspondiente para obtener la siguiente salida **responsive** con GRID:

- Pantalla tamaño **desktop**:

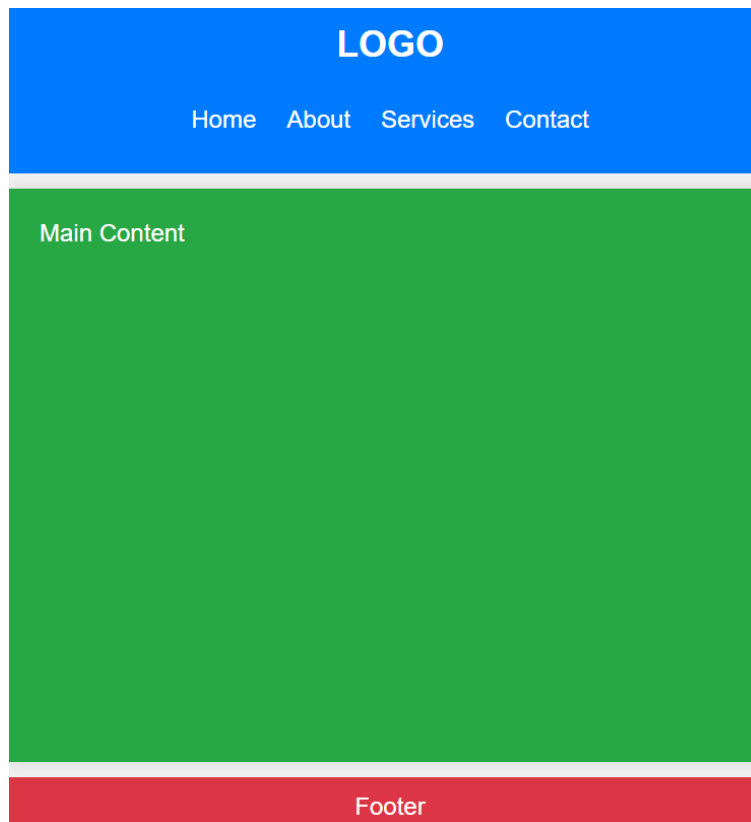




- Pantallas inferiores a 800px:



- Pantallas inferiores a 500px:





Las **instrucciones** son las siguientes:

- Utiliza una altura mínima de 100vh para el body.
- Utiliza como mínimo las siguientes propiedades de GRID:
  - grid-template-rows y grid-template-columns
  - unidades fr
  - gap
  - justify-content
- colores de fondo:
  - del body: #f0f0f0
  - header y nav: #007bff
  - main: #28a745
  - ad: #6c757d
  - footer: #dc3545

- Código **HTML**:

```
<body>
  <div class="header-nav">
    <header class="header">
      <div class="logo">LOGO</div>
    </header>
    <nav class="nav">
      <ul>
        <li><a href="#">Home</a></li>
        <li><a href="#">About</a></li>
        <li><a href="#">Services</a></li>
        <li><a href="#">Contact</a></li>
      </ul>
    </nav>
  </div>
  <div class="content">
    <main class="main">Main Content</main>
    <aside class="aside">
      <div class="ad">Ad 1</div>
      <div class="ad">Ad 2</div>
      <div class="ad">Ad 3</div>
    </aside>
  </div>
  <footer class="footer">Footer</footer>
</body>
```



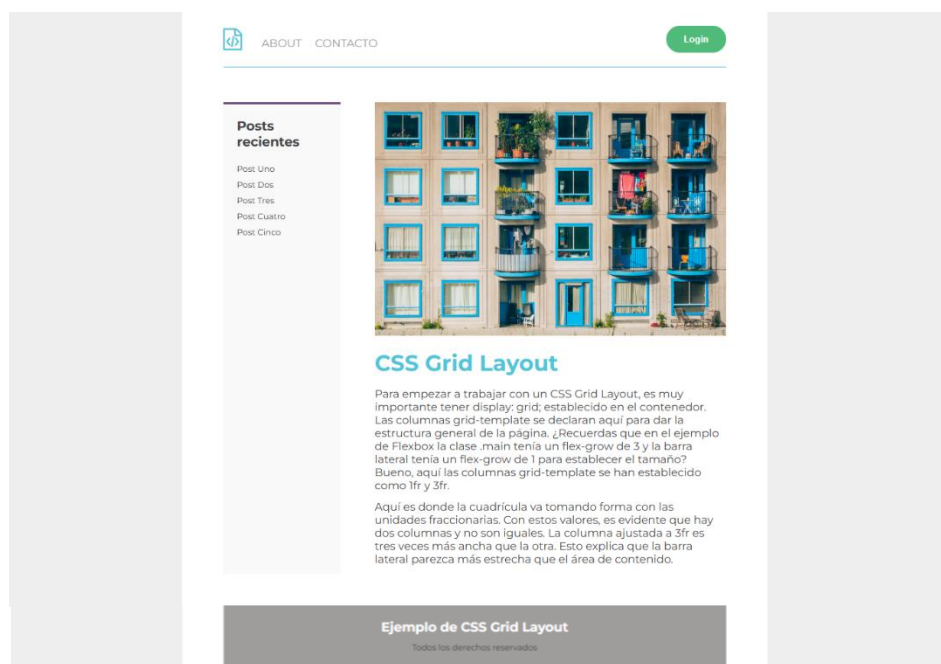
6. Dado el código HTML adjunto, escribe el código CSS correspondiente para obtener la siguiente salida **responsive** con GRID.

Las **instrucciones** son las siguientes:

- El ancho máximo de la clase `.container` es de 850px;
- Utiliza las propiedades `grid-template-columns` con las unidades `fr`, `grid-template-areas`.
- El tipo de letra es Montserrat.
- Además del blanco y negro, los colores son los siguientes:
  - Color de los párrafos: #434344;
  - Fondo del body: #eee;
  - Color de la letra del header: #9f9c9c;
  - Color de fondo del botón: #51bb7b;
  - Color del encabezado h2 y el logo: #50c6db;
  - Borde superior del sidebar: #693f7e;
  - Color de fondo del sidebar: #f9f9f9;
  - Color de fondo del footer: #9f9c9c

La apariencia de la interfaz es la siguiente para los siguientes tamaños de pantalla:

- Pantalla tamaño **desktop**:





- Pantallas inferiores a 500px:



El código HTML es el siguiente:

```
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Ejemplo-Interfaz-con-grid</title>
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/font-awesome@4.7.0/css/font-awesome.min.css">
  <link rel="stylesheet" type="text/css" href="styles-grid.css">
</head>
<body>
```



```
<div class="container">
  <header>
    <nav>
      <ul>
        <li class="logo"><a href="#"><i class="fa fa-file-code-o"
aria-hidden="true"></i></a></li>
        <li><a href="#">About</a></li>
        <li><a href="#">Contacto</a></li>
      </ul>
    </nav>
    <button>Login</button>
  </header>

  <aside class="sidebar">
    <h3>Posts recientes</h3>
    <ul>
      <li><a href="#">Post Uno</a></li>
      <li><a href="#">Post Dos</a></li>
      <li><a href="#">Post Tres</a></li>
      <li><a href="#">Post Cuatro</a></li>
      <li><a href="#">Post Cinco</a></li>
    </ul>
  </aside>

  <section class="main">
    
    <h2>CSS Grid Layout</h2>
    <p>Para empezar a trabajar con un CSS Grid Layout, es muy
importante tener display: grid; establecido en el contenedor. Las columnas
grid-template se declaran aquí para dar la estructura general de la
página. ¿Recuerdas que en el ejemplo de Flexbox la clase .main tenía un
flex-grow de 3 y la barra lateral tenía un flex-grow de 1 para establecer
el tamaño? Bueno, aquí las columnas grid-template se han establecido como
1fr y 3fr.</p>
    <p>Aquí es donde la cuadrícula va tomando forma con las unidades
fraccionarias. Con estos valores, es evidente que hay dos columnas y no
son iguales. La columna ajustada a 3fr es tres veces más ancha que la
otra. Esto explica que la barra lateral parezca más estrecha que el área
de contenido.</p>
  </section>
  <footer>
    <h3>Ejemplo de CSS Grid Layout</h3>
    <p>Todos los derechos reservados</p>
  </footer>
</div>
</body>
```