

## Práctica Entregable 1 – Programación WEB

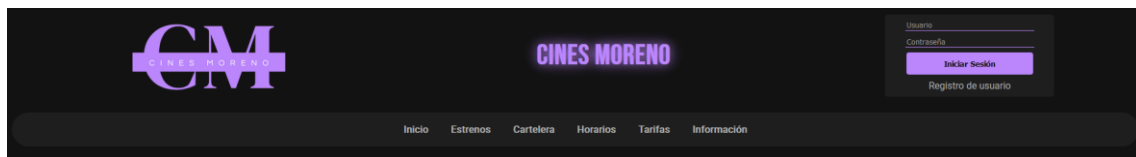
Juan Pedro Moreno Ruiz

La estructura principal de los documentos de la web está compuesta por un *section*, que recibe el nombre de "**caja\_\*\*\*\***", cada una de ellas es una *flexbox*, el método que he elegido para ordenar los elementos de la página por encima de otros como *grid* o *float*.

En algunos casos, he tenido que usar un *section* llamado "break", para que los elementos *flex* se coloquen debajo de los anteriores, el sitio donde más lo he usado ha sido en la cabecera, siendo este el código HTML, CSS y el resultado final:

```
<section class="break"></section>
```

```
.break {  
  flex-basis: 100%;  
  height: 0;  
}
```



El *break* ocupa todo el ancho de la página y no tiene ningún contenido, por lo que es invisible, pero obliga al índice a comenzar en la línea de abajo.

He elegido este método porque ya lo he usado en situaciones anteriores y ya estaba familiarizado con él, además de que es más sencillo controlar el lugar y orden de los elementos. A continuación, dos muestras de esto, la primera del HTML y la segunda del CSS relacionado:

```
<main>  
  <section class="caja_registrado">  
    <h2>Información de Contacto</h2>  
  
    <h3>Desarrollador</h3>  
    <p>Juan Pedro Moreno Ruiz</p>  
  
    <h3>Curso</h3>  
    <p>4º Doble Grado Informática y ADE</p>  
  
    <h3>Correo</h3>  
    <p>juanpemoreno@correo.ugr.es</p>  
  
    <h3>Teléfono</h3>  
    <p>6666666666</p>
```

```

        <h3>Dirección</h3>
        <p>C. Periodista Daniel Saucedo Aranda, s/n, 18014
Granada</p>

        <h3>Horario</h3>
        <p>De 9:00 a 21:00</p>
    </section>
</main>

```

```

.caja_registrado {
    background-color: #1E1E1E;
    border-radius: 25px;
    display: flex;
    flex-direction: column;
    flex-basis: 75%;
    margin-left: 10%;
    padding-left: 5%;
}

```

En algunos casos, he usado la misma clase en varios sitios, como en el ejemplo de la foto, que, aunque es del documento de contacto, usa la clase del documento que se muestra cuando alguien se ha registrado en la web, ya que el estilo que necesitaba era igual o muy parecido y repetirlo todo era muy redundante.

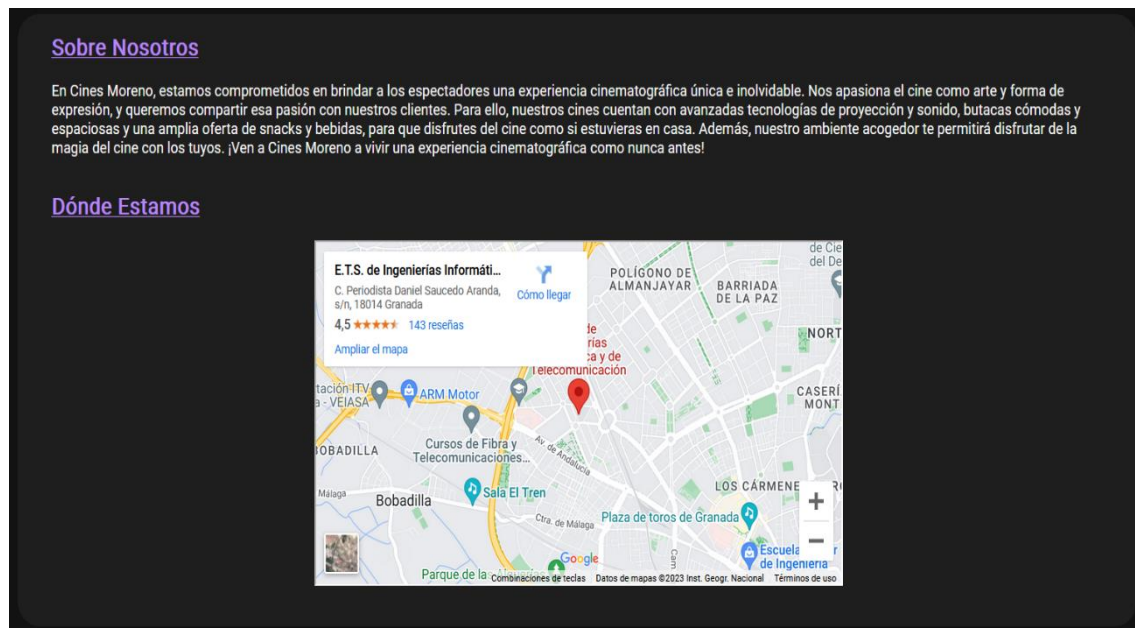
Este formato de solo una caja es aplicable a todos los documentos, menos al *index.html*, que está formado también por una **"barra\_noticias"** que era requisito de la práctica, en el fondo su estilo es muy parecido a las cajas, pero una propiedad extra llamada *overflow-y: scroll*, que añade una barra de scroll al elemento cuando sus hijos superan la altura del container, este es el código CSS de este elemento:

```

.barra_noticias {
    background-color: #1E1E1E;
    border-radius: 25px;
    height: calc(100vh - 200px - 25px);
    width: 20%;
    overflow-y: scroll;
    /*Con overflow-y la barra del scroll es vertical*/
    justify-content: center;
    text-align: center;
    padding-top: 2%;
}

```

Otro aspecto importante que me gustaría destacar es el mapa de Google Maps que he incrustado en el documento de Información:



Para hacer esto, solo he tenido que buscar la ubicación de la Escuela en Google Maps y al darle al botón de compartir, una de las opciones es copiar el código HTML:

```
<iframe src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d12712.801418078932!2d-3.6190105595703126!3d37.19547036020044!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0xd71fc55025f928b%3A0x4dbbca09efdcad08!2sE.T.S.%20de%20Ingenier%C3%ADas%20Inform%C3%A1tica%20y%20de%20Telecomunicaci%C3%B3n!5e0!3m2!1ses!2ses!4v1680126593673!5m2!1ses!2ses" loading="lazy" referrerpolicy="no-referrer-when-downgrade">
```

Con respecto al diseño responsive del documento *index.html*, la gran mayoría de las unidades de la web están expresadas con un porcentaje, lo que facilita mucho que el diseño se adapte a otros tipos de pantalla, un ejemplo de esto puede ser la "caja\_registro" mostrada anteriormente. Además de esto y para terminar de adaptar las partes, he usado código CSS que solo se tiene en cuenta si la pantalla es menor a 480px de ancho:

```
@media screen and (max-width: 480px) {  
  header h1 {  
    visibility: hidden;  
    display: none;  
  }  
  .enlace-imagen {  
    flex-basis: 50%;  
  }  
}
```

```

.login {
    flex-basis: 100%;
    margin-bottom: 3%;
}
.main_index {
    display: block;
}
.main_index>* {
    width: 100%;
    overflow-y: visible;
}
.caja_películas {
    margin-top: 3%;
    height: auto;
}
.barra_noticias {
    height: auto;
}
.barra_noticias article img {
    margin-top: 2%;
}
.foto_titular {
    width: 70%;
}
.titulares {
    margin-bottom: 2%;
}
.boton-flotante{
    position: fixed;
    bottom: 2%;
    right: 2%;
    padding: 5%;
    width: auto;
}
}

```

Lo más importante del código anterior, es que desaparecen los elementos *flex* para pasar a ser *block*, lo que hace que se coloquen uno debajo de otro, adaptándose mejor al formato de las pantallas de móvil. Además de esto, el nombre del cine definido con *H1*, desaparece y solo se muestra el logo. El resto de las propiedades son solo ajustes para que los elementos terminen de encajar en la pantalla.

Por último, he de decir que he seguido el diseño de la web inspirándome en las instrucciones que da Google en su web de *Material* (<https://m3.material.io/>)