

Receta de maquetación usando flexbox

1. Crear la estructura de capas en el archivo html, con una estructura jerárquica que represente todas las capas unas dentro de otras

```
<div id="contenedor">
  <div id="cabecera">cabecera</div>
  <div id="medio">
    <div id="menu">menu lateral</div>
    <div id="contenido">
    </div>
  </div>
  <div id="pie">pie</div>
</div>
```

2. Crear la capa contenedora de la anchura que queramos definir en pixeles y una altura que sea como mínimo de todo el alto de la pantalla, como margen automático para que quede centrada.

```
width:800px;
min-height: 100vh;
margin:auto;
```

3. Definir la disposición flex de los elementos en la capa contenedora en columnas, de manera que el primer y último elemento estén arriba y abajo del todo.

```
display: flex;
flex-direction:column;
justify-content:space-between;
```

4. Definir el tamaño de los elementos del eje flex de la capa contenedora, estableciendo la altura de los elementos fijos (cabecera y pie), y el resto de capas rellenen el espacio disponible (medio).

```
#cabecera{
  width: 100%;
  height: 100px;
}
#pie{
  width: 100%;
  height: 100px;
}
#medio{
  width: 100%;
  flex-grow: 1;
}
```

5. Si algunas de las capas dispuestas a lo largo del eje vertical de la capa contenedora tiene a su vez varias capas o elementos que queremos distribuir con flexbox, repetimos el proceso que hemos realizado en la capa contenedora para dicha capa pero con disposición en fila, y con la propiedad align-items a stretch para que los elementos ocupen todo el alto de la capa.

```
#medio{
  width: 100%;
  flex-grow: 1;
  display:flex;
  flex-direction:row;
  justify-content:space-between;
  align-items:stretch;
}
```

6. A su vez las capas que estén dentro de la capa medio tendrán que definirse con la anchura deseada.

7. Si además de toda la estructura de capas anterior queremos añadir algún elemento independiente que se coloque en alguna posición de la pantalla, podemos añadirlo en el archivo html, fuera de la capa contenedora, asignarle en el css la propiedad position con valor absolute o fixed (en función de si queremos que haga scroll con la página o esté fijo), y asignarle la posición con las propiedades left, top, right o bottom, indicando los píxeles desde el borde de la pantalla.

```
<div id="logo">LOGOTIPO</div>

#logo{
  position: fixed;
  top: 10px;
  right: 10px;
}
```

8. En el interior de cada capa podemos añadir el contenido que queramos, títulos, imágenes, enlaces, tablas, formularios etc... a los que podemos aplicar todos los estilos vistos: bordes, colores, float, tipos de letra, etc...