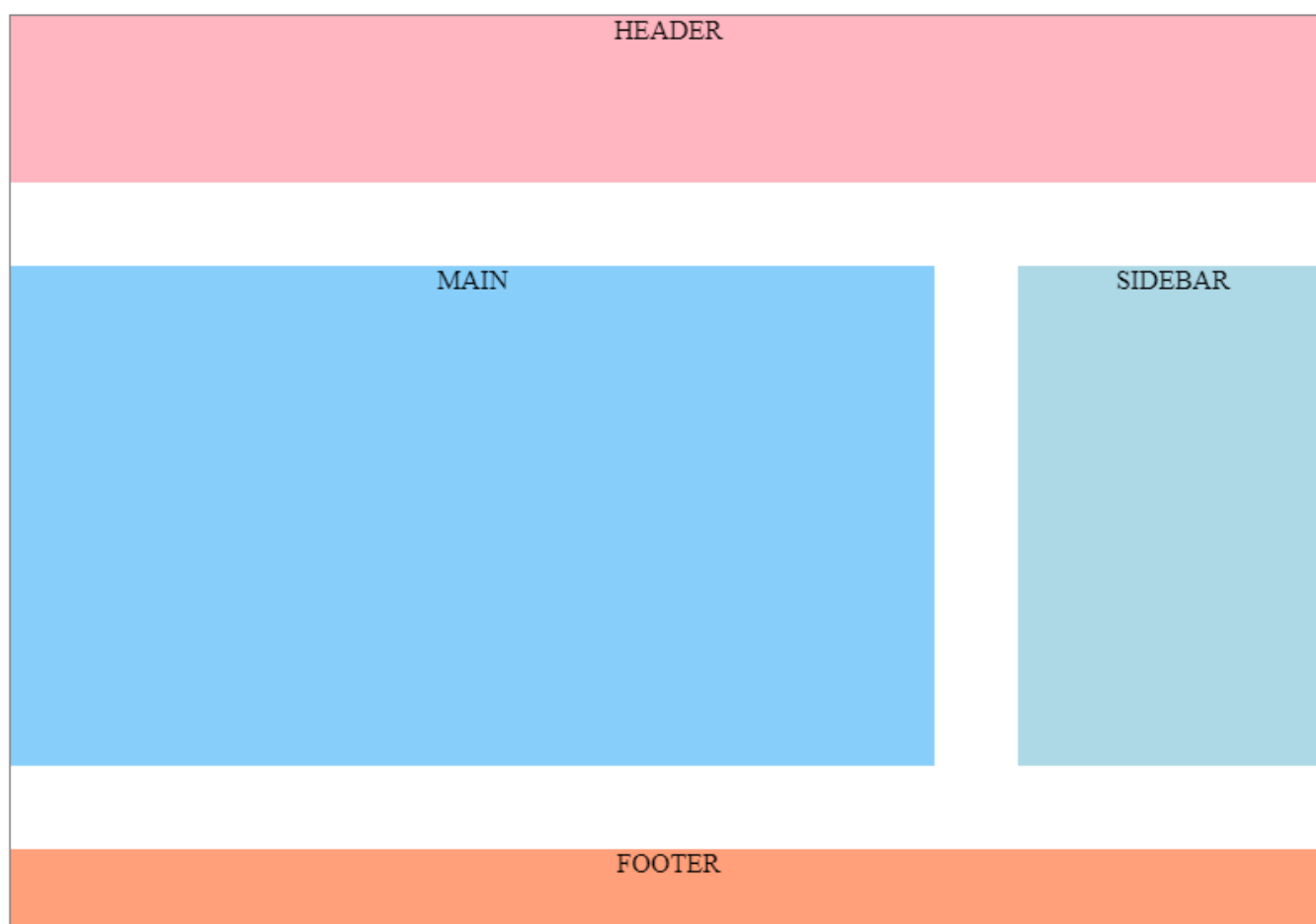


9. Ejemplo de uso Responsive de maquetacion en Flexbox y CSS Grid

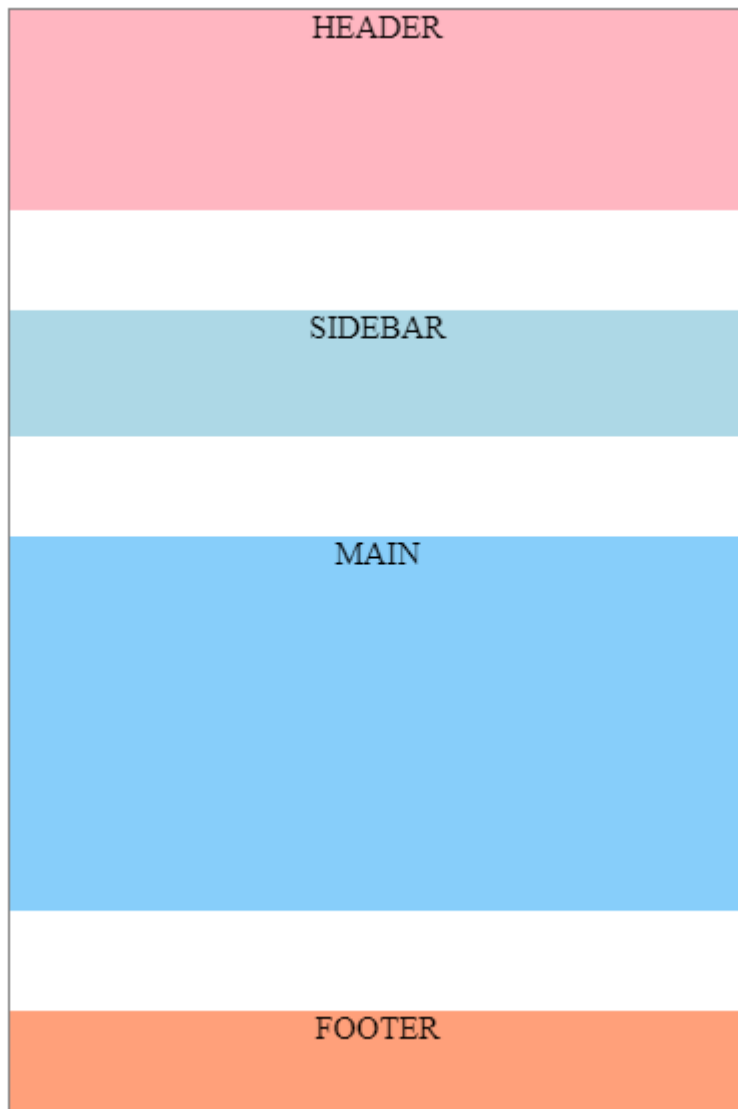
Tabla de contenidos

- **9. Ejemplo de uso Responsive de maquetacion en Flexbox y CSS Grid**
 - 9.1. Flexbox
 - 9.1.1. Posición de las secciones de la página
 - 9.1.2. Hacer Responsive una página
 - 9.2. CSS Grid
 - 9.2.1. Posición de las secciones de la página
 - 9.2.2. Hacer Responsive una página A continuación, veremos un ejemplo práctico para resolver tanto en Flexbox como en Grid, situaciones que se nos pueden dar a la hora de realizar un proyecto.

Realizaremos el siguiente diseño:



y lo haremos responsive con la ayuda de media queries



9.1. Flexbox

9.1.1. Posición de las secciones de la página

En Flexbox. Situamos todas las secciones, una debajo de otra:

```
<div class="container">
  <header>HEADER</header>
  <div class="main-and-sidebar-wrapper">
    <section class="main">MAIN</section>
    <aside class="sidebar">SIDEBAR</aside>
  </div>
  <footer>FOOTER</footer>
</div>
```

```
.container {
  border: 1px solid grey;
  margin: 0px auto;
  width: 80%;
}
```

```

    text-align: center;

    display: flex;
    flex-direction: column;
}

header{
    background-color: lightpink;
    height: 100px;
}

.main-and-sidebar-wrapper {
    height: 300px;
    margin: 50px 0px;

    display: flex;
    flex-direction: row;
}

.main {
    background-color: lightskyblue;

    flex: 3;
    margin-right: 50px;
}

.sidebar {
    background-color: lightblue;
    flex: 1;
}

footer {
    background-color: lightsalmon;
    height: 50px;
}

```

9.1.2. Hacer Responsive una página

```

@media (max-width: 600px) {
    .main-and-sidebar-wrapper {
        flex-direction: column;
    }

    .main {
        margin-right: 0px;
        margin-top: 50px;
        order: 1;
    }
}

```

Ejemplo maquetación responsive con media queries y Flexbox (Codepen)

9.2. CSS Grid

9.2.1. Posición de las secciones de la página

```
<div class="container">
  <header>HEADER</header>
  <section class="main">MAIN</section>
  <aside class="sidebar">SIDEBAR</aside>
  <footer>FOOTER</footer>
</div>
```

Observar como no hemos necesitado el wrapper para ajustar la parte central.

```
.container {
  border: 1px solid grey;
  margin: 0px auto;
  width: 80%;
  text-align: center;

  display: grid;
  grid-template-columns: 3fr 1fr;
  grid-template-areas:
    "header header"
    "main sidebar"
    "footer footer";
  grid-gap: 50px;
}

header {
  background-color: lightpink;
  height: 100px;

  grid-area: header;
}

.main {
  height: 300px;
  background-color: lightskyblue;
  grid-area: main;
}

.sidebar {
  background-color: lightblue;
  grid-area: sidebar;
}

footer {
  background-color: lightsalmon;
```

```
height: 50px;

grid-area: footer;
}
```

9.2.2. Hacer Responsive una página

En este caso, la mejor opción es redefinir el grid

```
@media (max-width: 600px) {
  .container {
    /* Redefine la cuadrícula en un solo diseño de columna. */
    grid-template-columns: 1fr;
    grid-template-areas:
      "header"
      "sidebar"
      "main"
      "footer";
  }
  .sidebar{
    height: 60px;
  }
  .main{
    height: 200px;
  }
}
```

[Ejemplo maquetación responsive con media queries y CSS Grid \(Codepen\)](#)