

# Diseño Fijo, Fluido y Responsivo

# Diseño Fijo

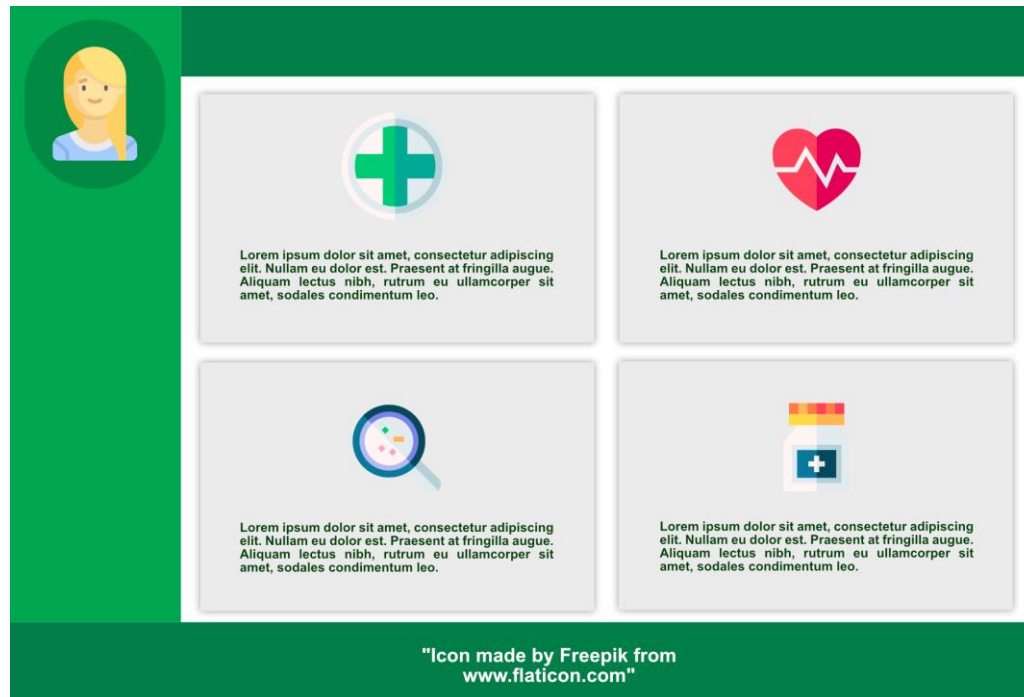
Los diseños fijos usan anchos de píxeles exactos, lo que significa que el tamaño de los componentes de la página será el mismo para todas las resoluciones.

Por ejemplo, generalmente hay algún tipo de "contenedor" utilizado para agrupar contenido relacionado. Luego, se aplica un ancho para evitar que el contenido se expanda por toda la página.

# Problemas

Supongamos que se tiene la siguiente página web, la cual es una página de 1000px de ancho.

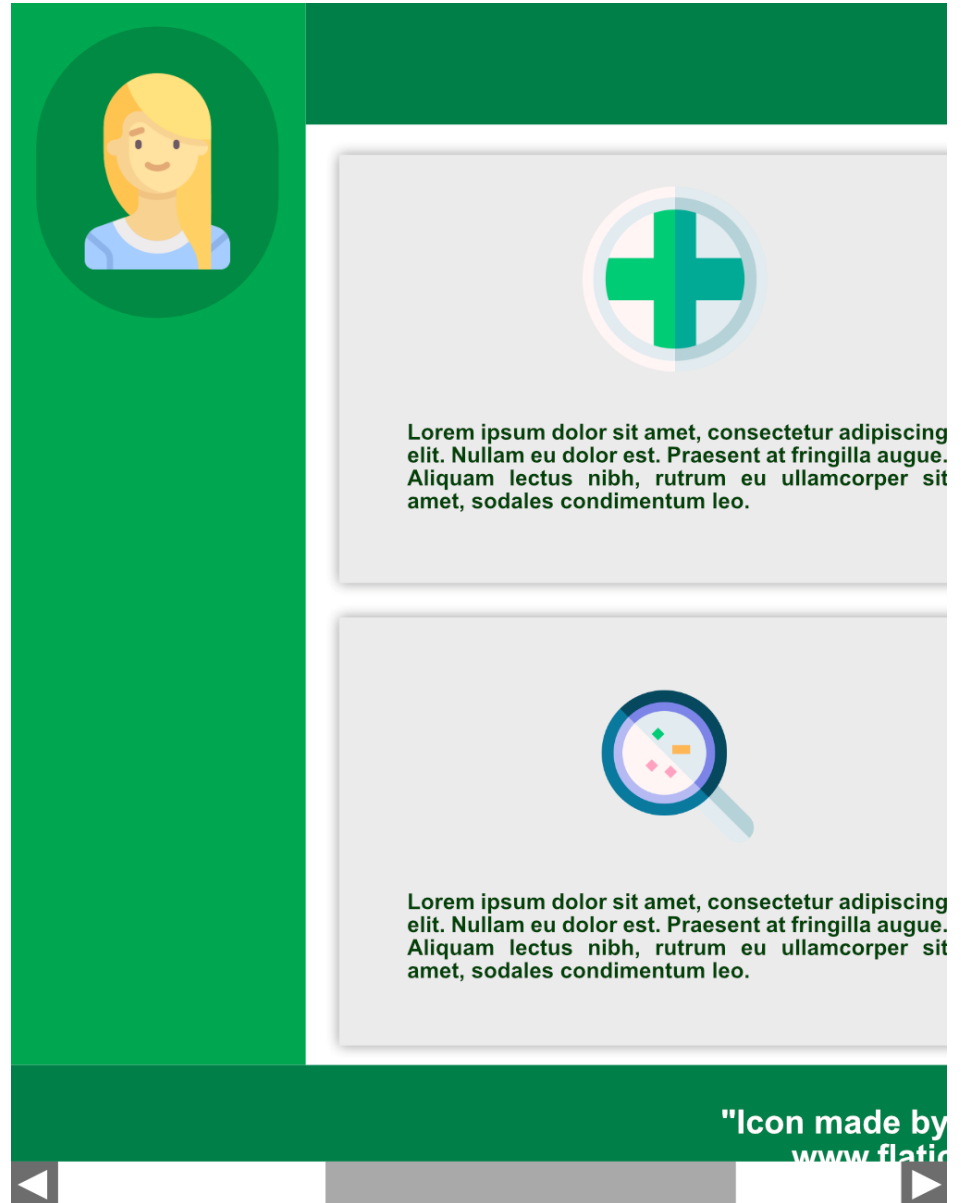
¿Qué pasaría si se ejecuta en una pantalla que tiene un ancho de 500px?



# Respuesta

Solamente se verá el 50% de la página.

Por último, aparecerá el temido scroll horizontal quitando la fluidez de navegación.



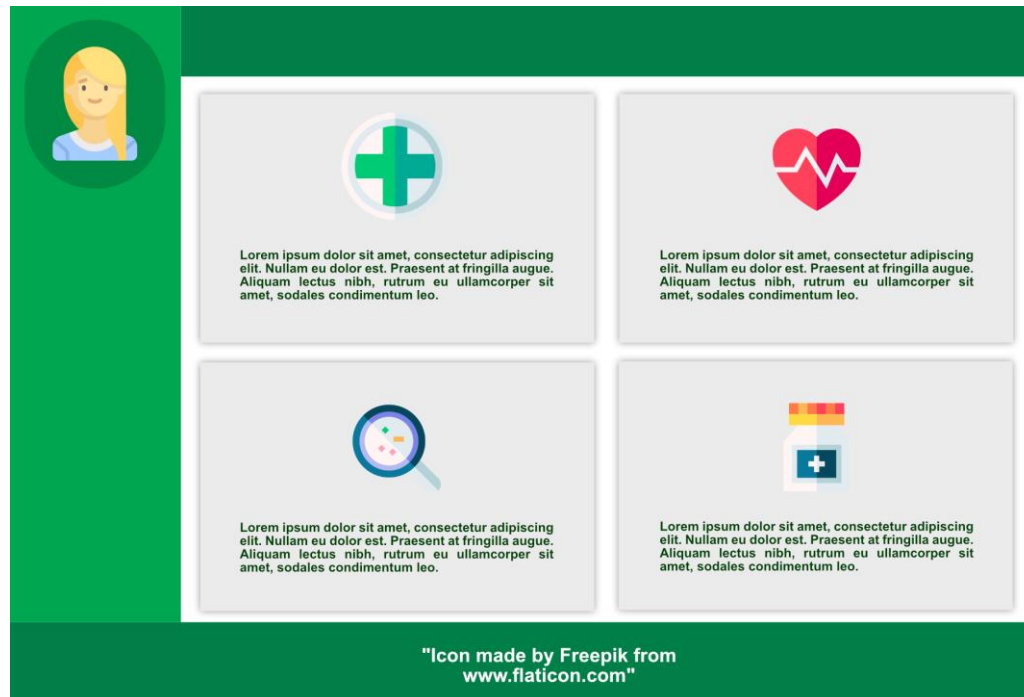
# Diseño Fluido

La mayoría de los componentes de la página en un diseño fluido se ajustan al tamaño de pantalla del usuario mediante el uso de anchos porcentuales en lugar de anchos fijos de píxeles. Los diseños fluidos también se denominan a veces diseño líquido. Para hacer que el elemento sea más flexible.

```
.content {  
    width: 50%;  
    margin: 0 auto;  
}
```

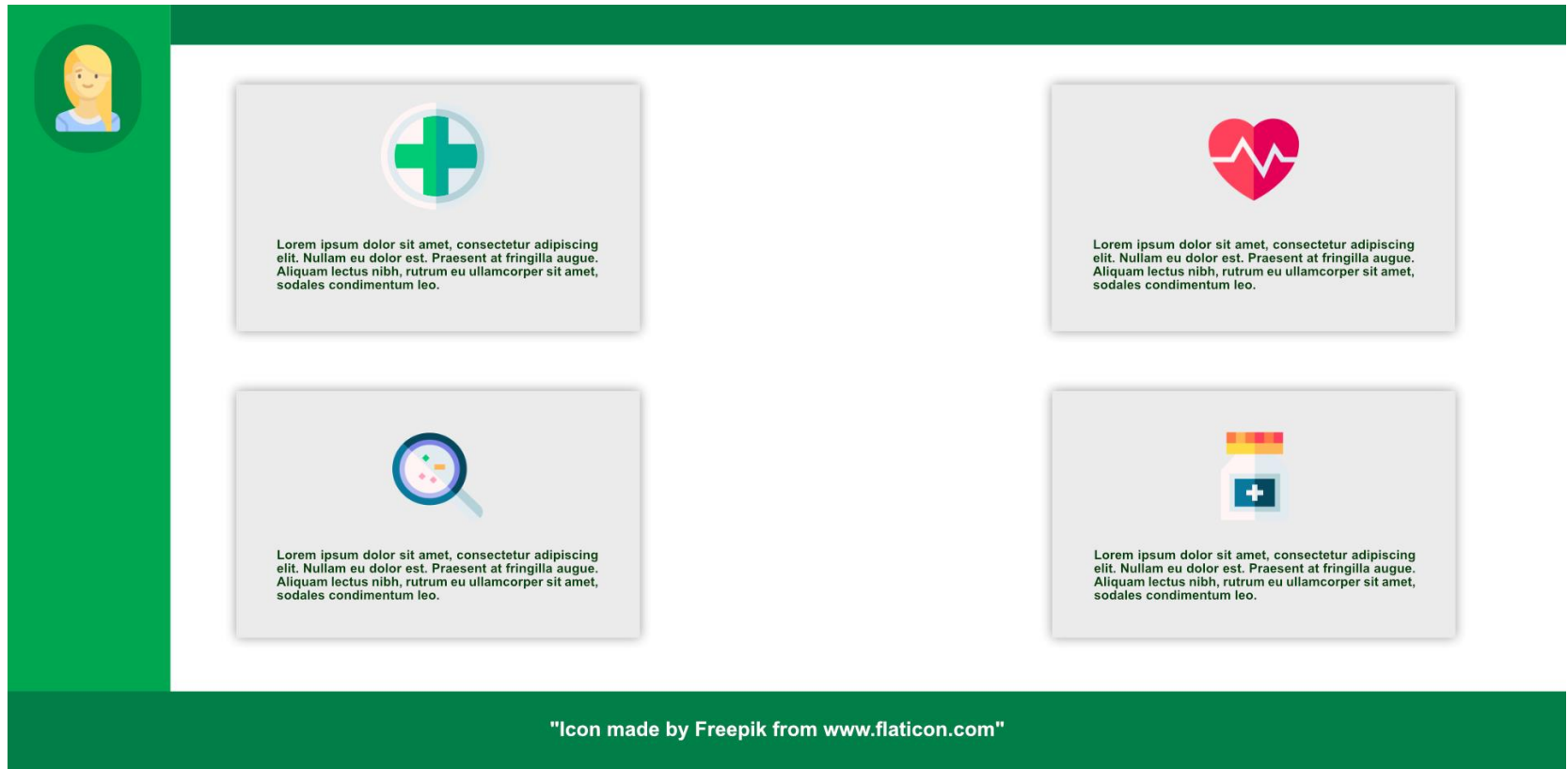
# Diseño Fluido

Esta técnica es excelente para resoluciones pequeñas, pero ¿qué sucede cuando la resolución de la pantalla es más ancha que el ancho deseado?



# Respuesta

El contenido mostrado se separará, pareciendo una página cada vez más vacía.



# ¿Soluciones para todos estos problemas?

Fusionar ambas técnicas.

Trabajar ciertos elementos de forma fija y otros porcentuales.

Pero también incluir en el caso de los porcentuales rangos **fijos** de anchura o altura (mínima/máxima).

```
.content {  
    max-width: 800px;  
    min-width: 200px;  
    width: 50%;  
    margin: 0 auto;  
}
```



## Propiedad min/max Width (Ancho)

- 1. min-width:** Define el ancho mínimo de un elemento. Si el contenido es más pequeño que el ancho mínimo, se aplicará el ancho mínimo. Si el contenido es mayor que el ancho mínimo, la propiedad min-width no tiene efecto.
- 2. max-width:** Define el ancho máximo de un elemento. Si el contenido es mayor que el ancho máximo, cambiará automáticamente el ancho del elemento. Si el contenido es más pequeño que el ancho máximo, la propiedad max-width no tiene efecto.

## Propiedad min/max Height (Altura)

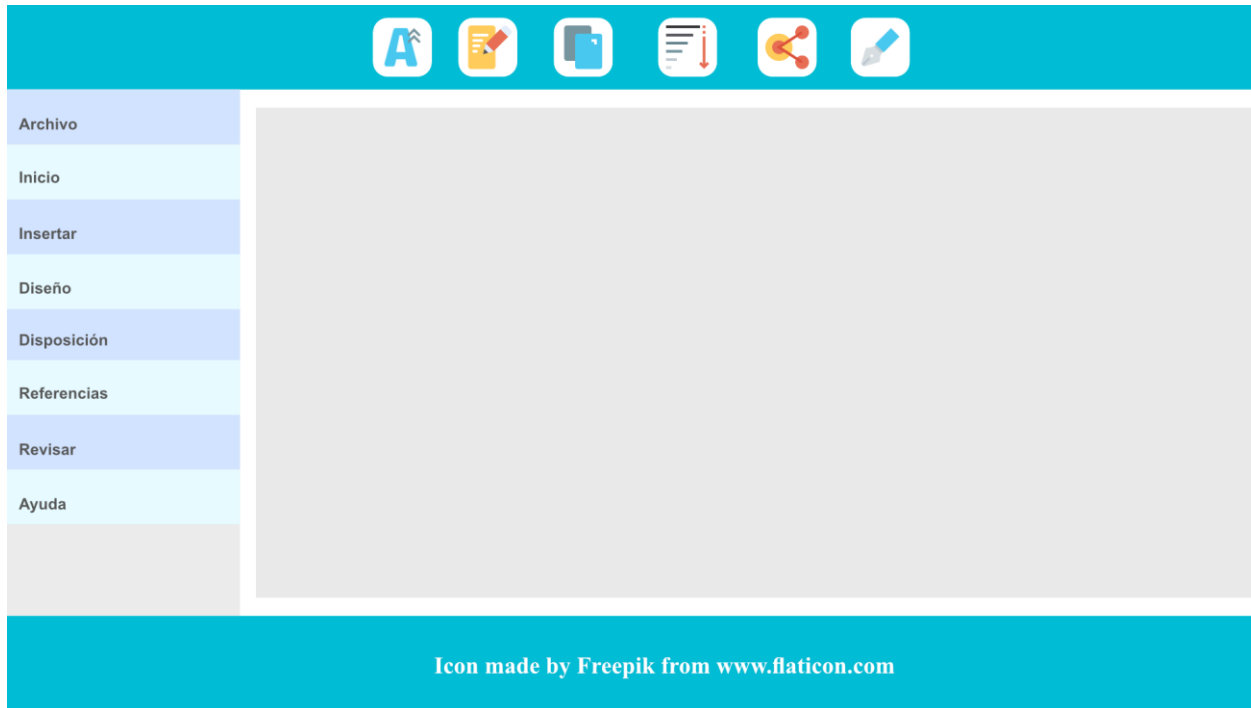
- 1. min-height:** Define el alto mínimo de un elemento. Si el contenido es más pequeño que el alto mínimo, se aplicará el alto mínimo. Si el contenido es mayor que el alto mínimo, la propiedad min-height no tiene efecto.
- 2. max-height:** Define el alto máximo de un elemento. Si el contenido es mayor que el alto máximo, cambiará automáticamente el alto del elemento. Si el contenido es más pequeño que el alto máximo, la propiedad max-height no tiene efecto.

# Observe la siguiente imagen

En un equipo de escritorio la página se ve así.

Pero ahora debe reproducirse en una table.

¿Qué técnica ocupas para que se siga viendo bien?



# Respuesta

Serán fluido todos los  
contenedores principales.  
¡Problema resuelto!



# Observe la siguiente imagen

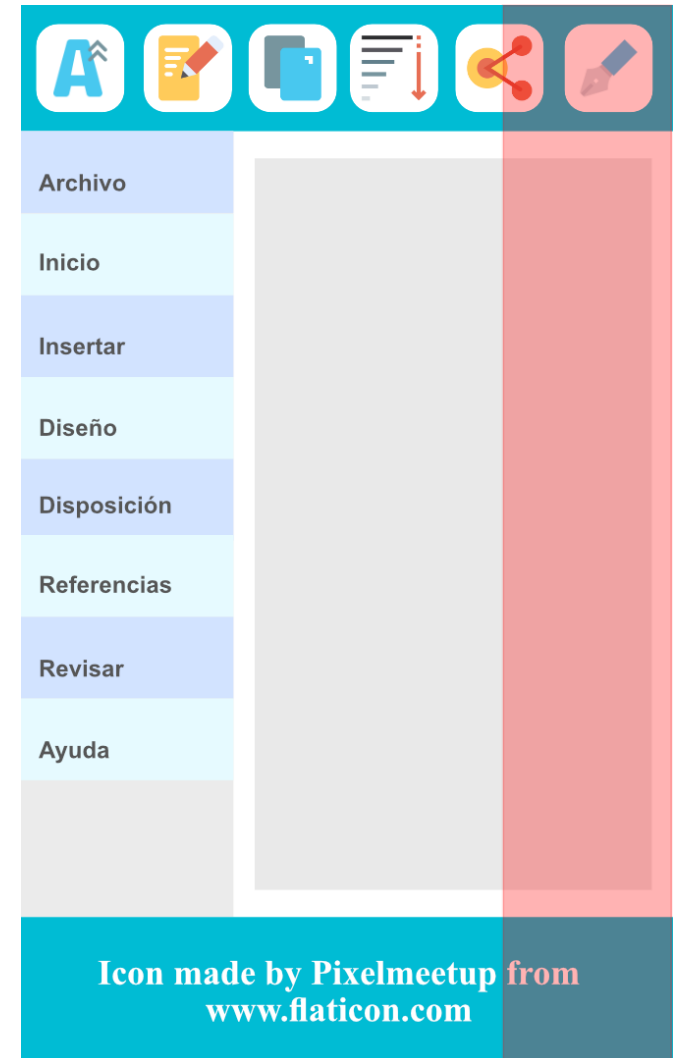
En una tablet se ve así.

Pero ahora debe reproducirse en un smartphone “1:2”. (El alto es el doble de ancho)

**El recuadro en rojo indicará el espacio que se deberá quitar.**

¿Qué técnica se ocupará para que se siga viendo bien?

**¿La técnica de diseño fijo y fluido no dan abasto?**



# Diseño Web Adaptable (Responsividad)

Filosofía de diseño y desarrollo cuyo objetivo es adaptar la apariencia de las páginas web al dispositivo que se esté utilizando para visitarlas.

Esta adaptación incluye la modificación de elementos visuales y estructuras de una página web para una coherente visualización.

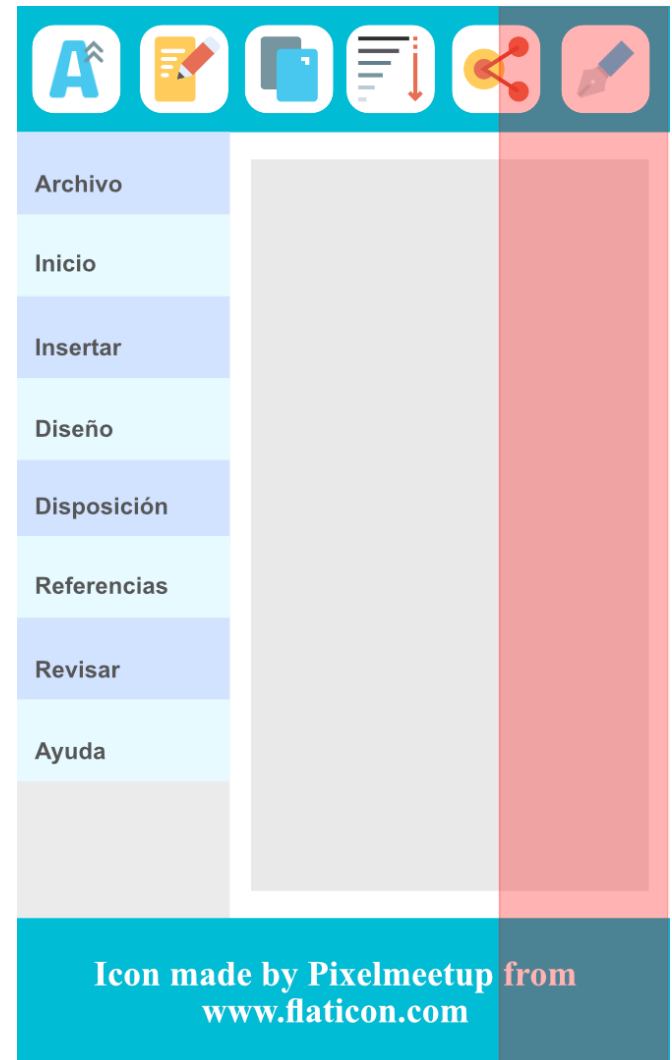
Para el problema anterior, ¿cómo se deberá adaptar el contenido?

# Respuesta

Hasta el momento tenemos problemas en el header y nav.

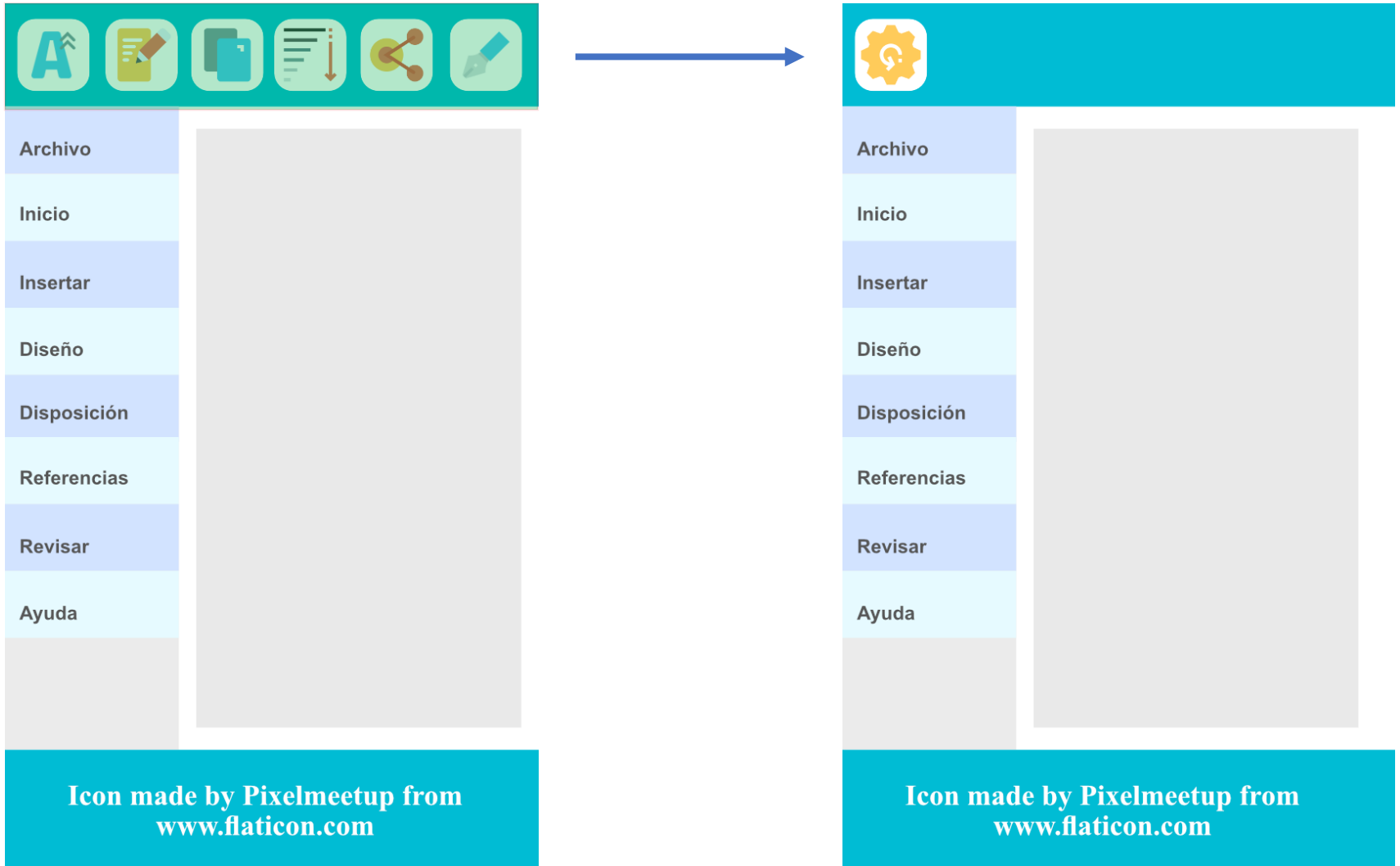
Ya no es posible reducir más.

**Solución:** Cambiarlos por un botón que al presionar se despliegue las opciones.



Ahora al presionar el botón de herramientas se desplegarán todos los elementos hacia abajo.

Pero el contenedor gris necesita más espacio, ¿De dónde podremos sacar más espacio?



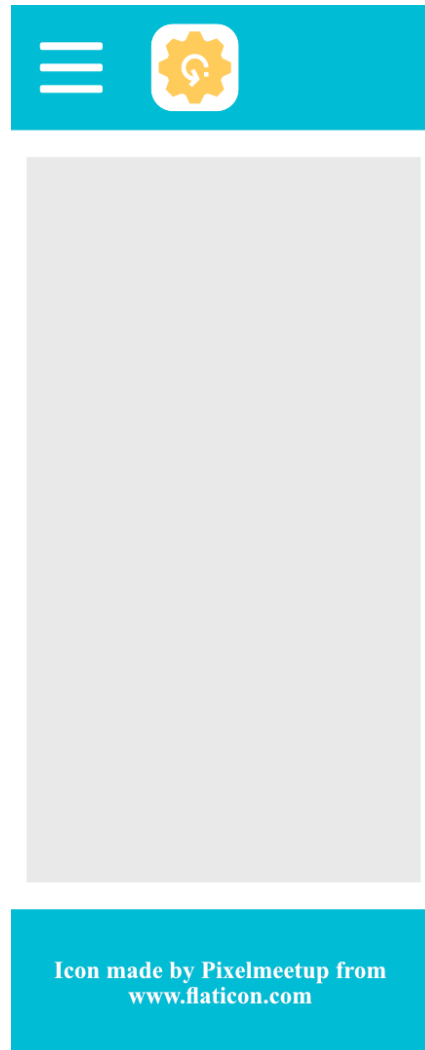


Moveremos el menú arriba también, dejando todo el espacio al cuadrante gris.



Por último el cuadrante gris ocupará el antiguo espacio del menú de navegación.

¡Listo! Página responsiva.



# ¿Cómo se crea una página responsiva?

## Media Queries


Las media queries (en español "consultas de medios") son útiles cuando deseas modificar tu página web o aplicación en función del tipo de dispositivo (como una impresora o una pantalla) o de características y parámetros específicos (como la resolución de la pantalla o el ancho del navegador).

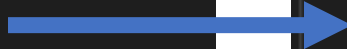
# Media Queries


Gracias a las media queries es posible alterar los valores de los selectores dependiendo de condiciones impuestas por el desarrollador.

Ejemplo:

Si el ancho de la pantalla es menor de 500px, altera algunos valores de la clase “.contenido” y agrega una nueva propiedad.

```
.contenido{  
  width: 500px;  
  height: 200px;  
  background-color:  teal;  
}
```



```
.contenido{  
  width: 300px;  
  background-color:  coral;  
  font-size: 10px;  
}
```

# Estructura

```
@media condición {  
  selector {  
    prop1: valor;  
    prop2: valor;  
    ...  
  }  
}
```

```
@media only screen and (max-width: 350px) {  
  .content {  
    max-width: 800px;  
    min-width: 200px;  
    width: 50%;  
    margin: 0 auto;  
  }  
}
```

# Estructura

**only:** palabra clave que evita que los navegadores más antiguos que no admiten consultas de medios con funciones de medios apliquen los estilos especificados. No tiene ningún efecto en los navegadores modernos.

```
@media only screen and (max-width: 350px) {  
  .content {  
    max-width: 800px;  
    min-width: 200px;  
    width: 50%;  
    margin: 0 auto;  
  }  
}
```

# Estructura

**screen:** Esta query está habilitada solamente para dispositivos que usen pantalla (Tablet/Smartphone/Etc.)

```
@media only screen and (max-width: 350px) {  
  .content {  
    max-width: 800px;  
    min-width: 200px;  
    width: 50%;  
    margin: 0 auto;  
  }  
}
```

# Estructura

**and:** Palabra clave para encadenar otra condición más  
Para este caso la condición dice:

Sólo es posible ejecutar si el dispositivo usa pantalla “y”  
tiene un ancho mayor o igual a 100px “y” tiene un ancho  
menor o igual a 350px.

Para nuevas condiciones agregue un nuevo “and”.

```
@media only screen and (min-width: 100px) and (max-width: 350px) {  
  .content {  
    max-width: 800px;  
    min-width: 200px;  
    width: 50%;  
    margin: 0 auto;  
  }  
}
```



## ¿Y Or?

Es solamente necesario separar cada condición por una coma.

Pero por elemento trabajaremos con condiciones simples.

En el módulo de JavaScript retomaremos el tema. 😊

Sólo es posible ejecutar si el dispositivo usa pantalla “y” tiene un ancho mayor o igual a 100px “o” tiene un ancho menor o igual a 350px.

```
@media only screen and (min-width: 100px), (max-width: 350px) {  
  .content {  
    max-width: 800px;  
    min-width: 200px;  
    width: 50%;  
    margin: 0 auto;  
  }  
}
```

# Condiciones más comunes

```
/*Para activar esta media query
| El ancho debe ser exactamente de 500px
*/
@media (width: 500px){}

/*Para activar esta media query
| El ancho debe mayor o igual a 500px
*/
@media (min-width: 500px){}

/*Para activar esta media query
| El ancho debe menor o igual a 500px
*/
@media (max-width: 700px){}

/*Misma lógica que los anteriores*/
@media (height: 500px){}
@media (min-height: 500px){}
@media (max-height: 500px){}
```

# Condiciones más comunes

```
/*Para activar esta media query
    El ratio debe ser exactamente 1:2
    Ej: 500px x 1000px
    Ancho x Alto
*/
@media (aspect-ratio: 1/2){}

/*Para activar esta media query
    El ratio debe ser mayor o igual 1:2
    Ej: 500px x 1000px
*/
@media (min-aspect-ratio: 1/2){}

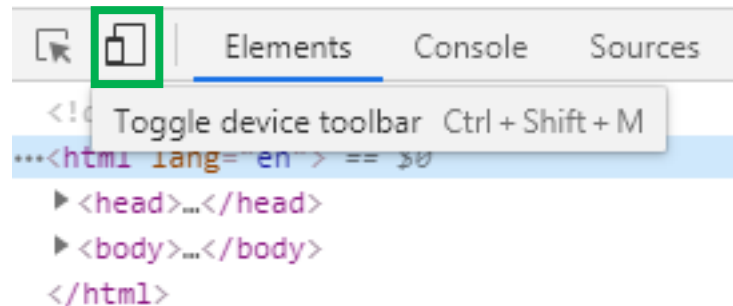
/*Para activar esta media query
    El ratio debe ser menor o igual 1:2
    Ej: 500px x 1000px
*/
@media (max-aspect-ratio: 1/2){}
```

Para mayor información sobre ratio consulte en :  
[https://en.wikipedia.org/wiki/Display\\_aspect\\_ratio](https://en.wikipedia.org/wiki/Display_aspect_ratio)  
[https://en.wikipedia.org/wiki/Display\\_resolution](https://en.wikipedia.org/wiki/Display_resolution)

# Observación

Para los siguientes ejercicios trabajaremos utilizando el visor de móviles que proporciona el navegador.

Si usas Chrome debes hacer click derecho -> Inspeccionar -> Seleccionar una pequeña imagen de un dispositivo móvil.



# Ejercicio

Escriba el siguiente código e indique qué sucede y por qué.  
Recordar usar el modo visor.

```
<body>  
</body>
```

```
@media only screen and (max-width: 1000px) {  
  body {  
    background-color: blue;  
  }  
}  
  
@media only screen and (max-width: 500px) {  
  body {  
    background-color: green;  
  }  
}  
  
@media only screen and (max-width: 100px) {  
  body {  
    background-color: red;  
  }  
}
```

# Respuesta

Dependiendo de la condición en la media query, se asignará las propiedades de color.

# Ejercicio

Diseñe una caja de 500px x 500px.

De color teal y en su interior un texto de color blanco.

Si el ancho de la pantalla llega a 500px.

Se deberá actualizar:

1. Dimensión: 400px x 400px.
2. Color fondo: Amarillo.
3. Color letras: Gris.
4. Tamaño de letra: 25px.
5. Texto se pone a la derecha.





Si el ancho de la pantalla llega a 400px.



1. Dimensión: 300px x 300px.
2. Color fondo: Coral.
3. Color letras: Blanco.
4. Tamaño de letra: 10px.
5. Texto debe estar centrado.

A teal square representing a design element. The text "Aquí un texto" is visible in the top-left corner of the square.

Aquí un texto

# Solución

```
.caja {  
  width: 500px;  
  height: 500px;  
  background-color:  teal;  
  color:  white;  
}  
  
@media only screen and (max-width: 500px) {  
  .caja {  
    width: 400px;  
    height: 400px;  
    background-color:  yellow;  
    color:  gray;  
    text-align: right;  
    font-size: 25px;  
  }  
}
```

```
@media only screen and (max-width: 400px) {  
  .caja {  
    width: 300px;  
    height: 300px;  
    background-color:  coral;  
    color:  white;  
    text-align: center;  
    font-size: 10px;  
  }  
}
```

```
<body>  
  <div class="caja">  
    Aquí un texto  
  </div>  
</body>
```



# Ejercicio

Diseñe 2 Divs.

Uno de 400x400 y el otro de 100x100.



Si la pantalla tiene un ancho de:

1000px: El primer div mide 300x300 y el segundo 200x200.

800px: El primer div mide 200x200 y el segundo 300x300.


600px: El primer div mide 100x100x y el segundo 400x400.


Pinte cada div de un color diferente.

**Recordar:** En una media query puede agregar más bloque CSS.

# Solución

```
<body>
  <div class="div-1"></div>
  <div class="div-2"></div>
</body>
```

```
.div-1 {
  width: 400px;
  height: 400px;
  background-color:  teal;
}

.div-2 {
  width: 100px;
  height: 100px;
  background-color:  coral;
}

@media only screen and (max-width: 1000px) {
  .div-1 {
    width: 300px;
    height: 300px;
  }

  .div-2 {
    width: 200px;
    height: 200px;
  }
}
```

```
@media only screen and (max-width: 800px) {
  .div-1 {
    width: 200px;
    height: 200px;
  }

  .div-2 {
    width: 300px;
    height: 300px;
  }
}

@media only screen and (max-width: 600px) {
  .div-1 {
    width: 100px;
    height: 100px;
  }

  .div-2 {
    width: 400px;
    height: 400px;
  }
}
```



# Ejercicio

Si la pantalla está en horizontal se debe pintar el fondo de body de color teal.

Si la pantalla está en vertical se debe pintar el fondo de body de color coral.

```
/*Se activa cuando el ancho es más grande que el alto*/  
    (orientation: landscape) {  
  
}  
/*Se activa cuando el alto es más grande que el ancho*/  
    (orientation: portrait) {  
  
}
```

# Solución

```
/*Se activa cuando el ancho es más grande que el alto*/  
@media only screen and (orientation: landscape) {  
  body {  
    background-color: teal;  
  }  
}  
/*Se activa cuando el alto es más grande que el ancho*/  
@media only screen and (orientation: portrait) {  
  body {  
    background-color: coral;  
  }  
}
```