



El futuro digital
es de todos

MinTIC



Diseño adaptativo

OPERADO POR:



Misión
TIC 2022

ruta de aprendizaje 1



Diseño adaptativo

Al comienzo de la era del internet, la única forma de acceder a una página web era a través de un ordenador, los cuales trabajan con pantallas o monitores. Por lo tanto el tamaño de la pantalla desde donde se vería nuestra web era una preocupación menor. Sin embargo en la actualidad tenemos una gran cantidad de dispositivos de diferentes tamaños, desde los cuales podemos ingresar a la web.

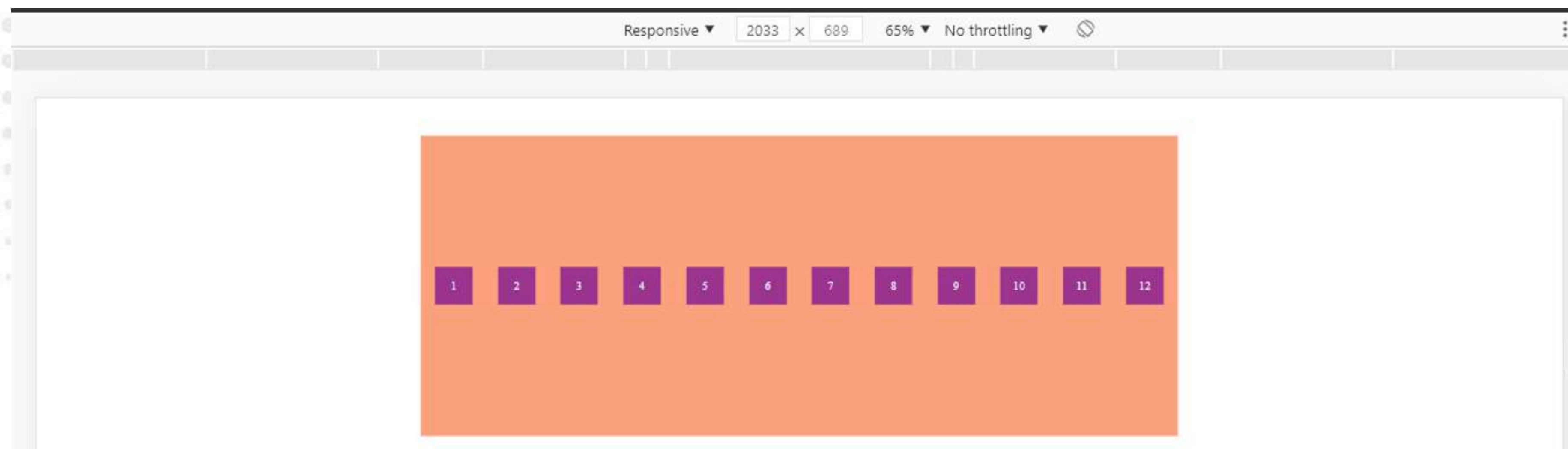
Por esta razón necesitamos aprender a utilizar la **adaptabilidad** en nuestras páginas, para que siempre funcionen visualmente y se adapten al dispositivo.



Debido a ese inconveniente se comenzó a pensar en un concepto llamado “responsive web design” o diseño adaptativo. Que consiste en utilizar ciertas técnicas para lograr que una página web adapte la distribución del contenido lo mejor posible al tamaño de la pantalla.



Anteriormente vimos la propiedad **Flexbox** en CSS, la cual nos permitía modificar la ubicación de los bloques o de los hijos dentro de un contenedor, en posiciones que se adecuaban al diseño que necesitábamos.

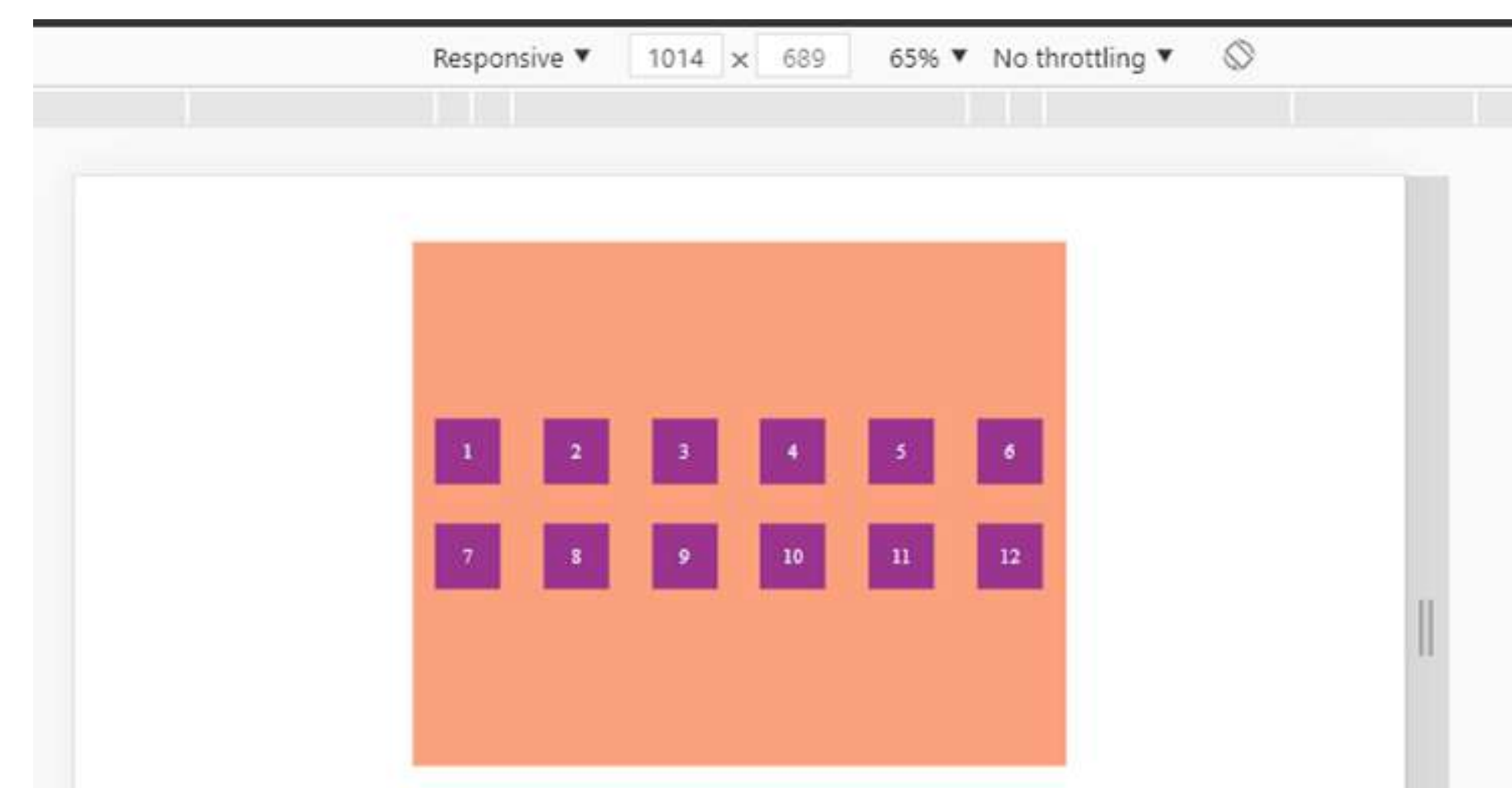
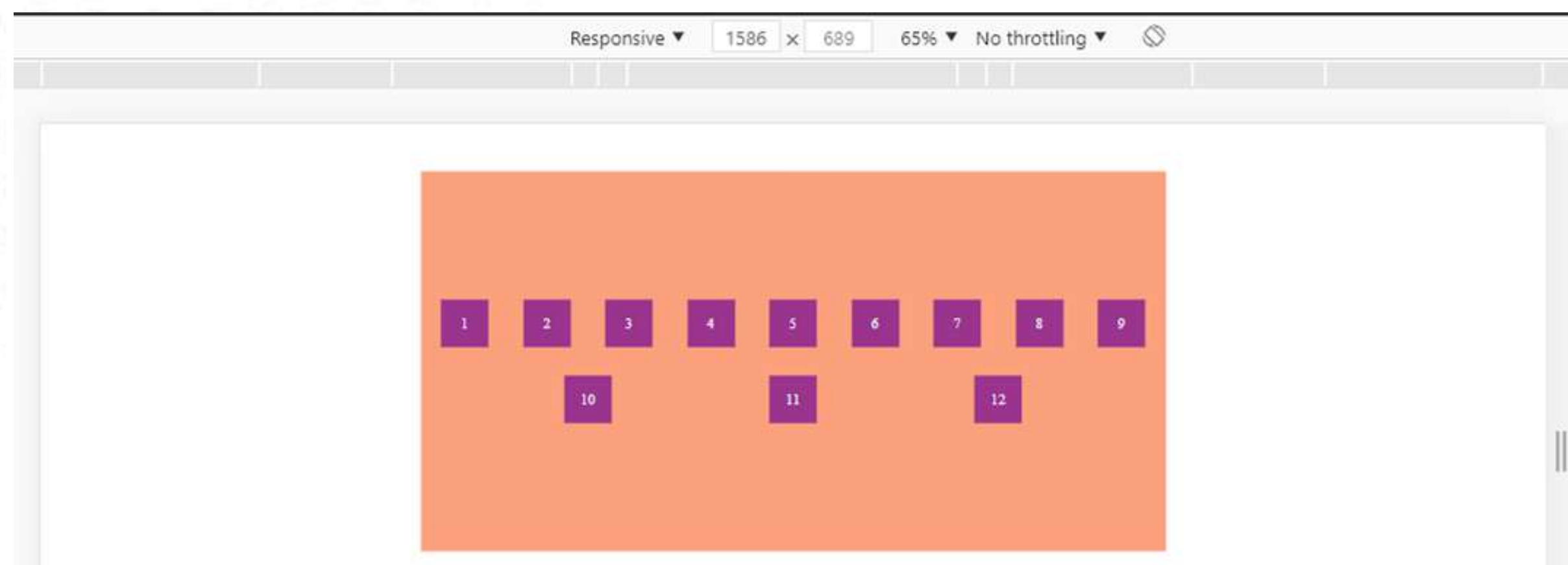


esta propiedad es el comienzo al **diseño adaptativo**.





Recordemos que dependiendo de la propiedad y cantidad de hijos que tengamos en nuestro contenedor, los bloques se reacomodaran, según el comportamiento de su padre. Si el padre es flexible entonces los hijos tomaran diferentes posiciones según las instrucciones que les entreguemos:



Para tener mas claridad en como la pantalla se comportara al cambiar su tamaño y esto como afecta el tamaño y el comportamiento de nuestros objetos tenemos que revisar un concepto simple





Píxeles (px)

Los píxeles es lo más cercano que encontraremos a un **tamaño fijo**, usualmente se considera que un píxel equivale a un punto de la pantalla del dispositivo utilizado para navegar, pero en realidad es una medida relativa a la resolución que tengamos en nuestro dispositivo.

Debido a esta relación directa entre la resolución y la unidad, se considera a los píxeles la medida más exacta para establecer una valor de tamaño de fuente.

Esta característica de “precisión”, hace que la medida tenga ventajas y desventajas:

- La principal ventaja que esto trae, es que el diseñador puede ser más exacto que con otras unidades.

- Sin embargo, esto hace que el tamaño no aumente o disminuya en base a alguna variación en el tamaño del espacio visual, lo que provoca que el texto se vea muy pequeño en monitores con resolución muy alta, o muy grande en pantallas pequeñas.



Puntos (pt)

Los puntos o points, son otra unidad de medida para establecer el tamaño de una fuente. Estos generalmente se utilizan para la **impresión en papel**, y basan su tamaño en las pulgadas. Cada pulgada equivale a 72 puntos.

Cuando se utiliza esta medida en **CSS**, varia mucho su presentación, ya que depende del navegador y de la pantalla donde se esté observando.

Los puntos son buenos al momento de crear plantillas de impresión con CSS, pero fuera de eso pueden causar más dolores de cabeza que soluciones. Es por esta razón, que estas unidades no son nada recomendadas para llevar a cabo un diseño adaptable.

Dentro de esta misma categoría también encontramos otras unidades como **“cm”** (centímetros), **“mm”** (milímetros), **“in”** (pulgadas) y **“pc”** (picas).



Porcentajes (%)

Los porcentajes son muy similares a los **ems**, sirven para asignar valores escalables y también asignan el valor en forma de cascada, por lo que su valor es relativo al valor del padre más cercano.

Si utilizamos un valor de **100%** estamos indicando que queremos el tamaño de fuente estándar.

Ems (em)

Una medida más adecuada para la funcionalidad que estamos buscando, son los **ems**. Los **ems** son una **medida escalable**, esto hace que sean ideales para lograr un diseño adaptable.

Un em es relativo al tamaño de la fuente que establecimos en el elemento contenedor o padre del texto.

En caso de no haber establecido ningún tamaño, entonces el em toma el valor estándar de 16 píxeles, medida que es usada por los navegadores para establecer el tamaño por defecto de las letras.

La desventaja de utilizar **ems** en nuestro diseño, es que su aplicación respeta el orden jerárquico, por lo que si tenemos un código algo completo, con frecuencia resulta difícil de calcular.



Como vimos en clases anteriores, una pagina web puede tener diferentes estructuras según el dispositivo en el cual estamos visualizando la pagina. Para esto necesitamos hacer uso de nuevas herramientas que permitan que nuestra pagina se adapte a su entorno.

Para indicarle si después de cierto tamaño es necesario que haga una modificación en el espació donde se están mostrando los elementos de la pagina.

Los media query, nos apoyan en la labor de mover nuestros elementos y modificar su apariencia, según los cambios en el entorno.



Responsive Web Design



Uso de viewport en la etiqueta metatag

La mayoría de los navegadores utilizados en dispositivos móviles, escalan las páginas HTML en base a un ancho fijo el cual las ajusta para que quepan en la pantalla que las despliega.

Para modificar esto y restablecer el valor que tomará, podemos hacer uso del **metatag viewport**, el cual indica al navegador que ancho debe tomar y deshabilita la escala inicial.

Este metatag debe ser incluido dentro de la etiqueta **“head”** y debe ser representado de la siguiente manera:



Con el código que mostramos en el ejemplo, estaríamos utilizando la anchura del dispositivo donde visualizamos la página para establecerlo como el nuevo **viewport** de nuestro sitio:

```
meta name="viewport" content="width=device-width, initial-scale=1.0"&gt;
```




Media query:

Son propiedades que nos permiten trabajar con las medidas que necesitamos y nos ayudan a modificar las posiciones usando como base vales fijos.

El truco está en **preveer** la manera en que nuestro sitio se desplegará en distintos dispositivos y tamaños de ventana, para ello debemos utilizar una estructura HTML **responsive**, que sea flexible y fácil de adaptar a distintos entornos, para que después de esto, baste con agregar las condiciones necesarias a la hoja de estilo, las cuales permitan el uso de distintos **layouts**.



Dentro de esta estructura, y a modo de ejemplo vamos a definir para un espacio máximo de **600 px** que modifique nuestra barra de navegación. Ahora somos nosotros los que definimos que tipos de cambios tendrá según el dispositivo.

```
<style>

@media (max-width: 600px) {

    .facet_sidebar {

        display: none;

    }

}

</style>
```

```
@media (max-width: 600px) {

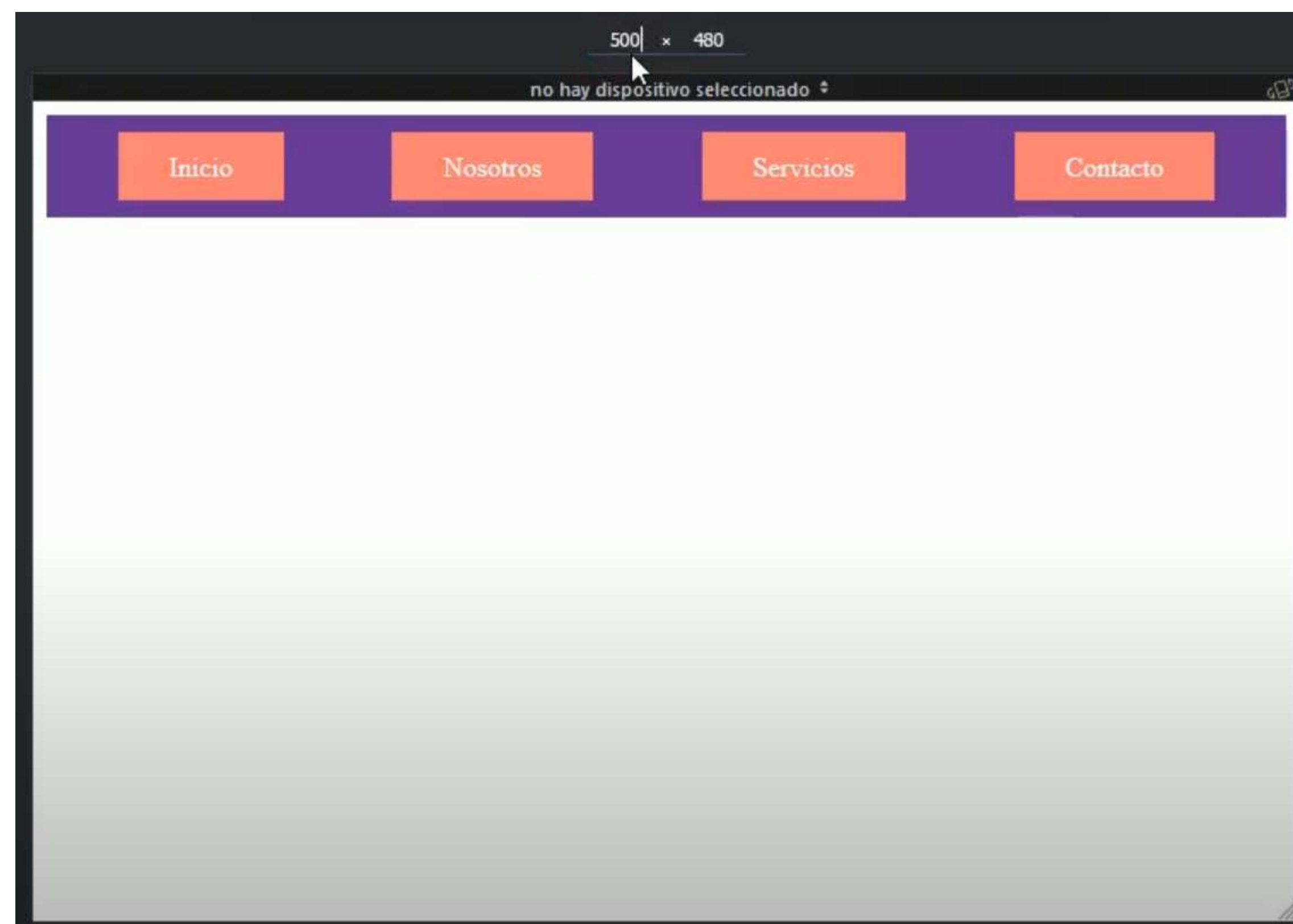
    }

}
```



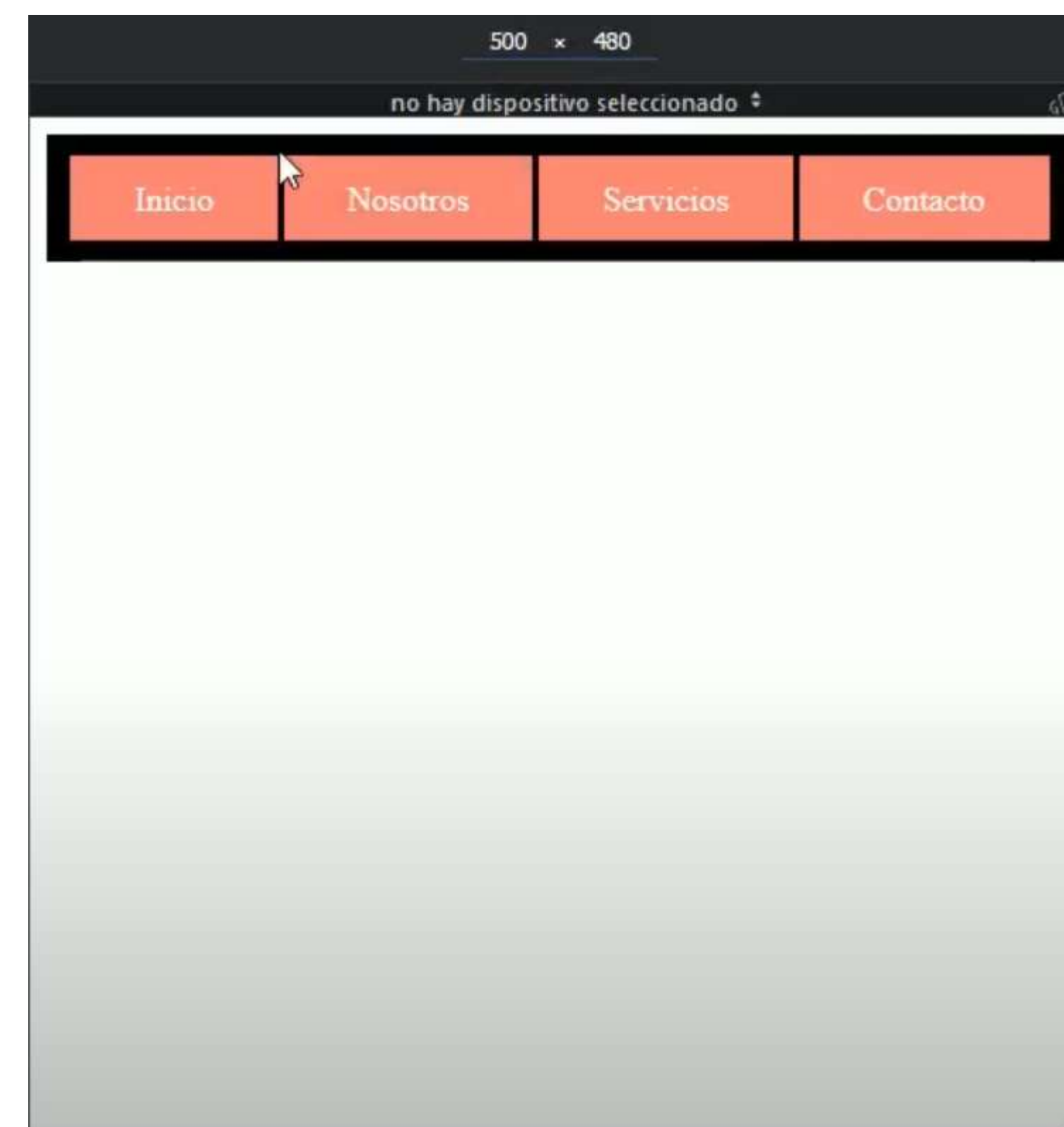
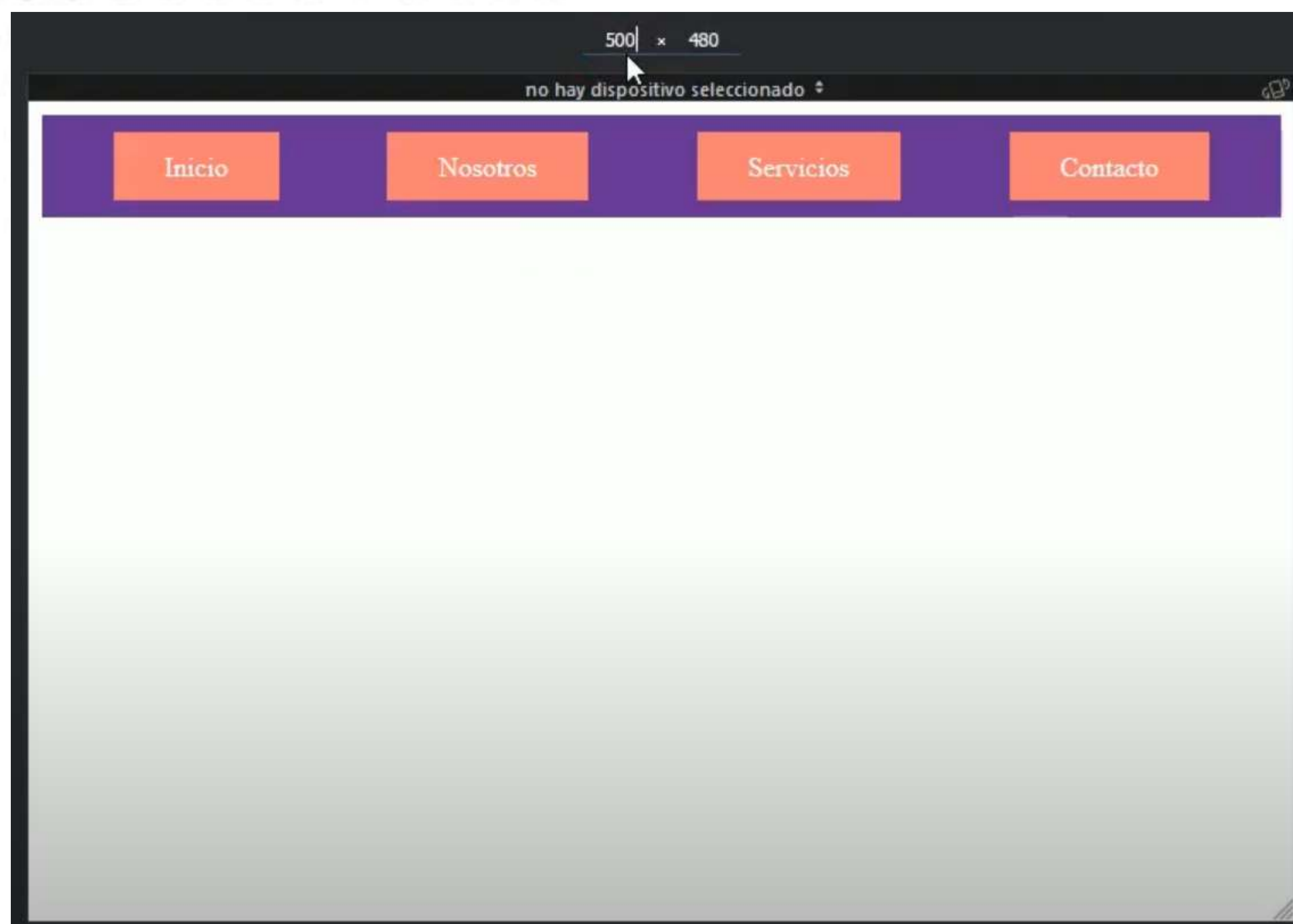

Por ejemplo, entre 0 y 600 px vamos a modificar el contendor o la etiqueta **nav**, vamos a definir que el fondo será negro una vez que bajemos a menos de 600px el fondo será de color negro:

```
@media (max-width: 600px) {  
    nav{  
        background: black;  
    }  
}
```





Si comparamos entre los dos valores, tendremos una diferencia con el diseño de la pagina, específicamente con su color de fondo.





Los demás estilos se mantienen igual, solo se modifica la propiedad que le indicamos modificar cuando se encuentra en un valor diferente en el tamaño de la página. Ahora si adicionamos otra condición que deberá ser cumplida cuando estemos con un tamaño menor a **600px**, vamos a indicarle que nos muestre el menú en columnas:

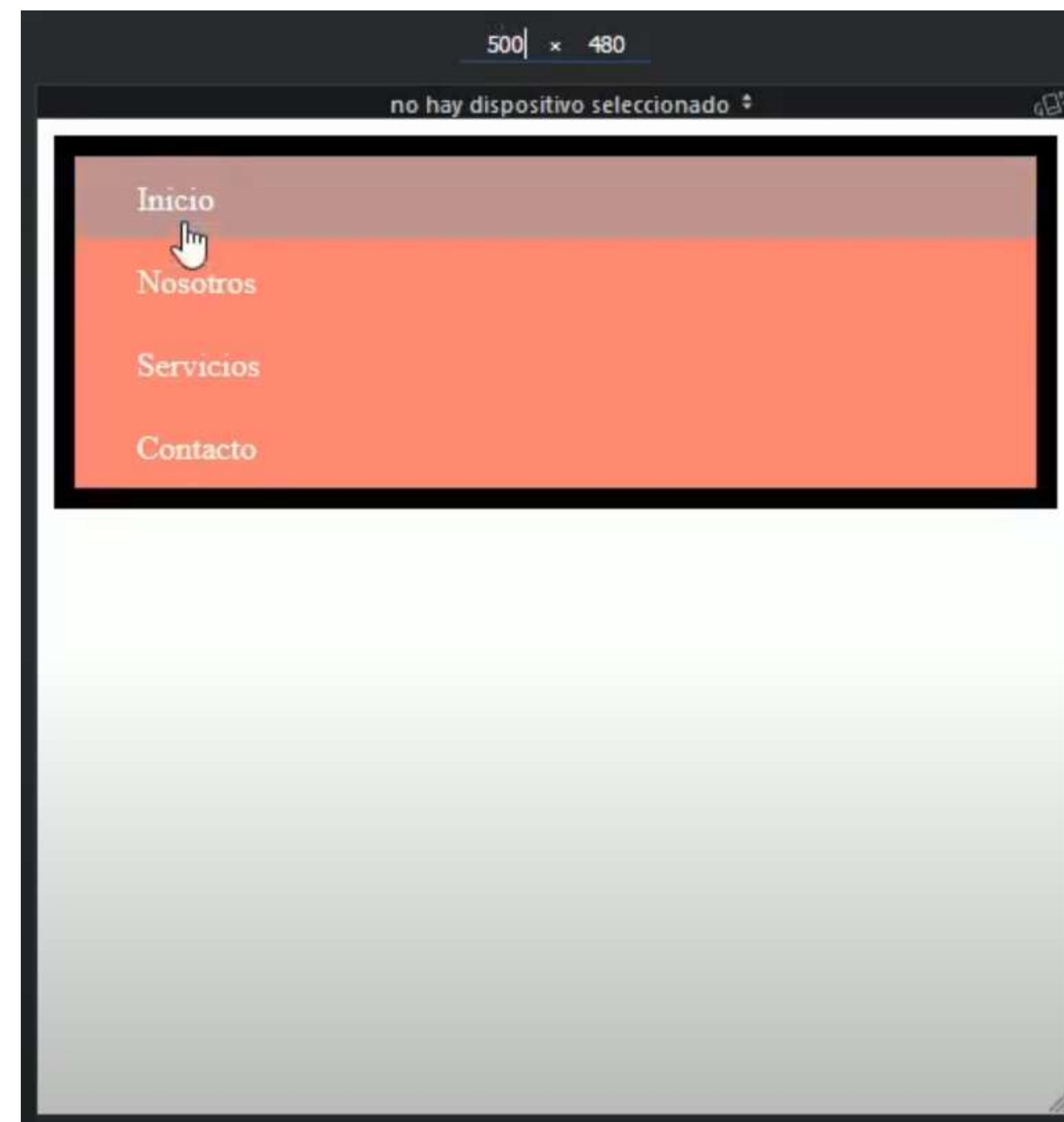
```
@media (max-width: 600px) {  
    nav{  
        background: black;  
        flex-direction: column;  
    }  
}
```

Para eso vamos a utilizar la propiedad **flex-direction: column;** y con esto cuando la pantalla comience a reducir su tamaño, tendremos un menú en columnas.



Vamos a ver que efectivamente el menú posiciona cada una de sus bloques uno debajo de otro.

Recordemos que es posible modificar la apariencia de estos bloques para que tengan una visualización mas separada, utilizando características vistas en clases anteriores.





Podemos hacer mas modificaciones dentro de la pagina, sin necesidad de generar mas secciones que utilicen la propiedad media, para eso solo escribimos la etiqueta de la zona de nuestra pagina que deseamos modificar y escribimos sus cambios o características nuevas.

Para este ejemplo vamos a utilizar la clase **.nav-enlace** y vamos a generar un **margin**:

```
@media (max-width: 600px) {  
    nav{  
        background: black;  
        flex-direction: column;  
    }  
    .nav-enlace{  
        margin: 10px 0;  
    }  
}
```



Así cuando tenemos una pantalla que tiene un tamaño menor al tamaño definido por defecto o esperado para la pagina web, el menos se adaptara a estos cambios y cambiara su estética:





Esta es la forma básica de adaptar detalles específicos según el tamaño de la pantalla, usar el **max-width** y **min-width** (puede que las necesidades nos requieran hacer algo más específico, pero con esto nos debería de valer).

También podemos utilizar estas propiedades para generar una estructura general, cuando la pagina sea abierta en un dispositivo, y con esto también controlar que ciertas cosas solo sean visibles bajo esa condición.

```
<select class="display-pequeno">  
<option>uno</option>  
<option>dos</option>  
<option>tres</option>  
</select>
```



El futuro digital
es de todos

MinTIC

GRACIAS

OPERADO POR:

