

DISEÑO WEB RESPONSIVO

- Crearé **paginas web** cuyo diseño o maquetación **cambie** según las características de los **distintos tipos de pantallas** en las que se muestren.

...una aproximación al **diseño web** que permite que nuestras página web se vean **correctamente** en una variedad de dispositivos y de tamaño de pantalla...**adaptando el layout al entorno de visualización...**

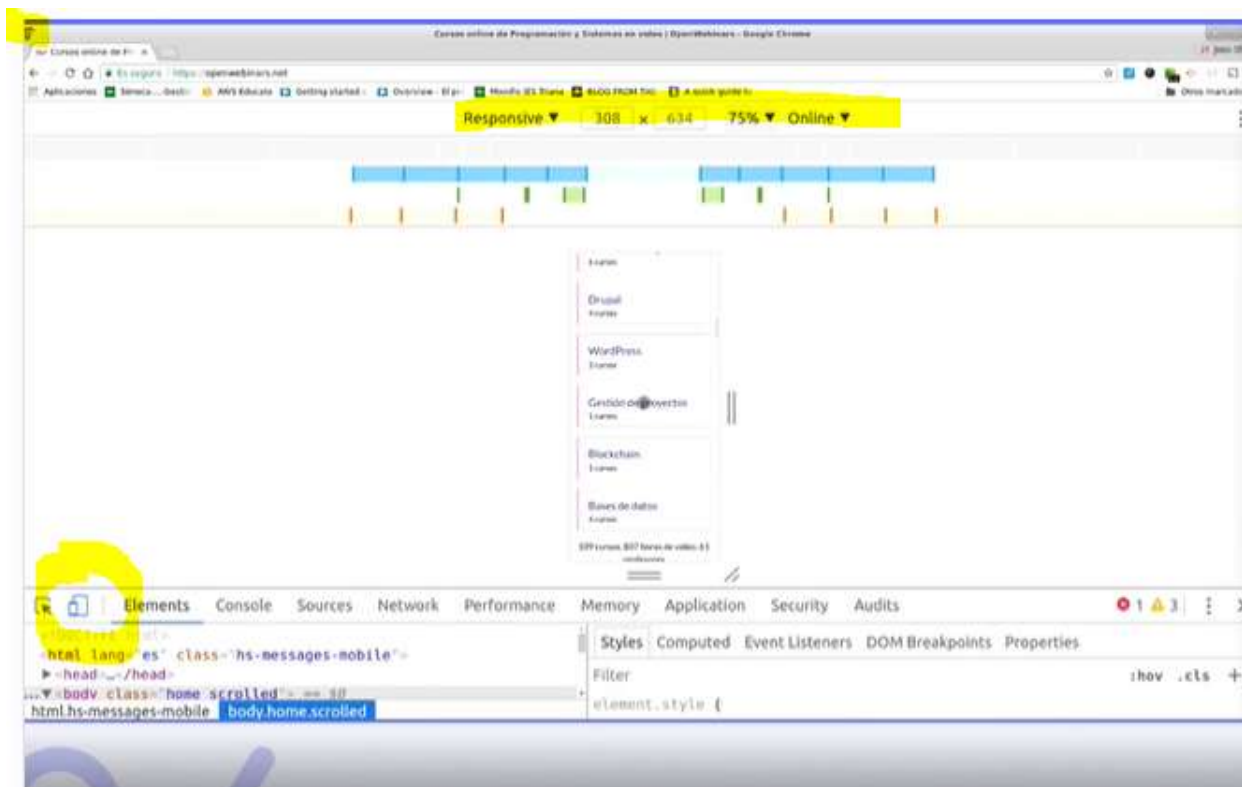
¿CÓMO?

CSS+
ETIQUETAS+
MEDIA QUERIES

(CSS3)

ENTORNO DE TRABAJO

Desde las herramientas de desarrolladores de google Chrome podemos comprobar cómo se adapta el diseño según evolucione la pantalla



Si nuestra página no tiene diseño responsive google hará un DEMOTED (degradar, última posición en las búsquedas).

REFERENCIAS

https://www.w3schools.com/html/html_responsive.asp

<https://web.dev/articles/responsive-web-design-basics?hl=es>

https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Responsive_Design

Viewport: Área de la pantalla en la que el navegador puede renderizar contenido, es decir, el espacio disponible para mostrar mi página web.

A partir de ahora en nuestro html pondremos ésta línea de código para que nuestro diseño sea responsivo.

```
<head>
...
<meta name="viewport" content="width=device-width, initial-
scale=1.0" />
...
</head>
```

Tipo de layout utilizado para el diseño responsivo. FLUID layout.

Ventaja: Se adapta al Viewport, al tamaño de lo que está viendo el usuario. La anchura de la página depende del tamaño del Viewport del usuario y se expresa en **porcentajes (%)**. **Recordar, siempre en porcentajes.**

Breakpoints: Es la anchura en la que se produce un cambio en el layout en respuesta a ciertas condiciones del dispositivo (media queries).

Una vez hemos establecido que usaremos diseños de tipo **Fluid** debemos recordar que el concepto de diseño responsivo, además de **adaptación** al tamaño implicaba el **cambio** en el diseño atendiendo al tamaño u otras características del dispositivo.

Típicos **breakpoints**

- <576px (pantallas pequeñas)
- 576px-768px (móviles apaisados)
- 768px-992px (tablets)
- 992px-1200px (desktop)
- >1200px (pantallas grandes)

Media-queries: Herramientas para realizar un diseño responsivo.

"módulo de CSS3 que nos permite adaptar la representación de contenido a las características del dispositivo"

Síntaxis de las Media-Queries

Las media queries son expresiones en las que indicamos un tipo de medio y una consulta en relación a las características del dispositivo como alto, ancho e incluso color:

Algo similar a lo siguiente:

```
@media mediatype [condiciones] {
```

```
.... /* Estilos para esas condiciones */ ....;;  
}
```

Los distintos tipos de valores que puedo tener para **mediatype** son:

- all
- **screen (el más usado)**
- tty (terminal)
- print (para impresoras)
- tv (para pantallas de televisión)
- projected (para proyectar contenidos)
- braille (para dispositivos para invidentes)
- etc...

En cuanto a las condiciones que puedo consultar:

- **width | min-width | max-width**
- height | min-height | max-height
- orientation (landscape / portrait)
- aspect-ratio | min-aspect-ratio | max-aspect-ratio
- color | min-color | max-color

Estas condiciones se pueden combinar y modificar utilizando cláusulas como:

- And
- Not
- All
- Only

Si juntamos todo podemos ver algunos ejemplos:

```
/* Estilos para todo tipo de pantallas con una anchura máxima  
de 576px*/
```

```
@media all and (max-width: 576px) {  
.....;  
}
```

```
/* Estilos para pantallas con al menos 992px de anchura y que estén apaisadas (más ancho que  
alto)*/
```

```
@media screen and (min-width: 992px) and (orientation: landscape) {  
.....;  
}
```

```
/* Estilos sólo para pantallas que tengan al menos 768px de anchura*/
```

```
@media only screen and (min-width: 768px) {
```

```
.....;  
}
```

Hojas de estilos diferentes

Podemos usar media-queries en la etiqueta link para seleccionar hojas de estilos diferentes atendiendo a las características del dispositivo.

Por ejemplo:

```
<!--Para pantallas de hasta 576px de ancho -->  
<link rel="stylesheet" media="(max-width: 576px)" href="small.css" />
```

```
<!--Para pantallas entre 576px y 768px de ancho -->  
<link rel="stylesheet" media="(max-width: 768px)" href="medium.css" />
```

```
<!--Para pantallas entre 768px y 992px de ancho -->  
<link rel="stylesheet" media="(max-width: 992px)" href="large.css" />
```

En el diseño responsivo se prioriza comenzar por pantallas pequeñas, es decir de menor a mayor tamaño.

IMÁGENES RESPONSIVAS

Optimización. Imágenes PNG/GIF/JPEG

En estos casos, si no nos importa la optimización es suficiente con usar una imagen de gran resolución y dimensiones y acotarla dentro de un contenedor.

Por ejemplo:

```
<div><img src="" ..... "" /></div>
```

```
div {  
  /* dimensiones deseadas */  
}  
  
img {  
  max-width: XXXXXpx;  
  width: 100%;  
}
```

TABLAS RESPONSIVAS

Scroll controlado

Esta técnica consiste en acotar el scroll horizontal para que si ha de aparecer solo afecte a la tabla y no a la página entera.

Tiene las siguiente ventajas:

- Conseguimos un diseño responsivo.
- No pierdo información.

Aunque también tiene desventajas:

- No priorizo la información.
- Aunque local sigue habien un scroll horizontal.

Para conseguir esto debemos “envolver” la tabla en un contenedor y darle las siguiente propiedades:

```
<div class="localscroll">
  <table>
    .....
  </table>
</div>
div.localscroll {
  overflow-x: auto;
  width: 100%;
}
```

TEXTO RESPONSIVO

Lo ideal, según estudios, es una letra de un tamaño adecuado, para poder leer sin forzar la vista, y que se disponga formando líneas de entre 60-80 caracteres.

Para intentar solucionar esto con garantías lo más recomendable es utilizar para el texto unidades de tamaño relativas al viewport. Tendremos:

- **vw** en relación a la anchura del viewport.
- **vh** en relación a la altura del viewport.

```
header h1 {
  font-size: 8vw;
}
```