

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso  
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148  
[www.sceu.frba.utn.edu.ar/e-learning](http://www.sceu.frba.utn.edu.ar/e-learning)

**Curso:**

# **PROGRAMADOR WEB INICIAL**



UTN.BA  
UNIVERSIDAD TECNOLÓGICA NACIONAL  
FACULTAD REGIONAL BUENOS AIRES

Centro de  
e-Learning  
Secretaría de Cultura y Extensión Universitaria

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso

(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148

[www.sceu.frba.utn.edu.ar/e-learning](http://www.sceu.frba.utn.edu.ar/e-learning)

Módulo 2:

## PROGRAMACIÓN WEB (PARTE 1)

Unidad 2:

## DESARROLLO WEB MULTIPLATAFORMA

# Presentación

En esta unidad aprenderemos cómo crear diseños responsivos. El Diseño Responsive se refiere a la idea de que un sitio web debería mostrarse igual de bien en todo tipo de dispositivo, desde monitores de pantalla panorámica hasta teléfonos móviles.

Es un enfoque para el diseño y desarrollo web que elimina la distinción entre la versión amigable para dispositivos móviles de un sitio web y su contraparte de escritorio. Con un diseño responsive ambos son lo mismo y debe pensarse de manera integral.

# Objetivos

## Que los participantes logren...

- Conocer qué es el Diseño Responsive con Mobile First.
- Aplicar Media Queries para modificar los elementos en función del tipo de dispositivo.

# Bloques temáticos

1. Responsive design
  - a. ¿Qué es el Responsive Design?
  - b. Tipos de Responsive Designs
  - c. Flujo de desarrollo de un sitio responsive
2. Media queries
  - a. Viewport
  - b. @media de CSS3
3. Mobile First
  - a. Concepto
  - b. Ejemplos

# Responsive design

Gracias a la gran utilización de los dispositivos móviles, nos vemos obligados a adaptar nuestra web a ellos, pero sin perder el diseño de nuestra página en equipos de escritorio.

## ¿Qué es el Responsive Design?

A la hora de crear nuestra página web, debemos tener en cuenta que se va a visualizar desde diferentes dispositivos, ya sea computadoras de escritorio, teléfonos celulares, tablets, televisores smart, entre muchos otros.

Vamos a querer que nuestra página se visualice correctamente en todos los lugares donde se ingrese. Por eso, recurriremos a la técnica de diseño web responsive. Crearemos la estructura de nuestra página web de tal manera que se adapte a todas las resoluciones de pantalla utilizadas. Usaremos hojas de estilo CSS o Javascript, pero recordemos que este último puede no estar activado en todos los dispositivos.

**Las técnicas responsive se focalizan en diseñar para:**

- Todos los navegadores y sistemas.
- Todas las resoluciones de pantalla.
- Todas las velocidades de conexión.

## Tipos de Responsive Designs

Tenemos varias maneras de adaptar nuestros sitios. Las exponemos a continuación:

- **Adaptarse al ancho**

Es el tipo de diseño responsive más utilizado; no se modifican en exceso los elementos del sitio, se tratan de adaptar al ancho de la pantalla. Se reduce el

menú, muchas veces cambiando su disposición y se redimensionan las imágenes.

- **Cascada de columnas**

Si tenemos un sitio que posee varias columnas, estas se ubicarán una debajo de la otra, para poder visualizarse correctamente en pantallas estrechas.

- **Reestructuración**

Como lo indica su nombre, aquí realizaremos una reestructuración de nuestro sitio. Colocaremos los elementos de manera diferente según el tamaño de pantalla que tenga el dispositivo. La manera en la que la cambiaremos va a depender del diseño que nosotros queramos ya que no hay una forma definida.

- **Cambio del diseño visual**

En esta manera de diseño no cambia la estructura del contenido, sino que se modifica el diseño según el tamaño de la pantalla, con motivos estéticos. Suele utilizarse en páginas informativas para ver la información de manera rápida y concisa.

## Flujo de desarrollo de un sitio responsive

- Crear un HTML con el contenido que deseamos mostrar, lo más semántico posible.
- Aplicar formato con CSS.
  - **Recursos de diseño estético:** estilos que se agregan para mejorar la estética.
  - **Diseño de layout:** jerarquizar la información y presentarla con una estructura que permita el mejor entendimiento.
    - Imágenes flexibles.
    - Sistema de grilla o cajas fluidas.
    - Media queries.

Siempre hay que tener en cuenta que algunos estilos pueden no ser soportados por todos los navegadores o ser interpretados de forma diferente.

## Media queries

CSS3 nos brinda las media queries que nos ayudarán a conocer algunas de las propiedades comunes de los dispositivos que nos visitan. Podremos utilizar esta información para utilizar nuestros estilos sin la ayuda de JavaScript.

Aunque media queries tiene muchas propiedades diferentes, vamos a identificar las 5 principales:

- **Aspect-ratio:** Evalúa las dimensiones relativas del dispositivo expresadas como una relación de aspecto: 16:9 por ejemplo.
- **Width y Height:** Evalúa las dimensiones del área de visualización. Estas pueden ser expresadas en valores mínimos y máximos.
- **Orientation:** Nos dice la orientación de la pantalla del usuario. Puede ser panorámico u horizontal (el ancho es mayor que el alto) o vertical (el alto es mayor que el ancho). Esto nos permite ajustar los diseños para dispositivos con propiedades de giro de la pantalla como smartphones y tablets.
- **Resolution:** Nos dice cuál es la densidad de los píxeles en el dispositivo de salida. Esto es especialmente útil cuando queremos aprovecharnos de las ventajas de los dispositivos que tienen una resolución mayor a 72 dpi.
- **Color, Color-index y monochrome:** Encuentran el número de color o bits por color. Esto nos permite crear diseños específicos para dispositivos monocromáticos.

## Tipos de dispositivos

- **all:** Todos los dispositivos de tipo media.
- **print:** Impresora.
- **screen:** Pantalla.
- **speech:** Se utiliza para los lectores de pantalla que "leen" en voz alta.



# Viewport

Esta meta-etiqueta ha sido creada inicialmente por Apple para sus dispositivos móviles, pero con el tiempo se ha convertido en un estándar para la mayoría de los dispositivos móviles (smartphone y tablets, por ejemplo).

Su uso es totalmente necesario, ya que sino el navegador establece el ancho con el que prefiere visualizar una página en lugar de usar el ancho del que dispone. Es decir que si la pantalla de nuestro móvil tiene 400px y el navegador detecta que lo óptimo sería visualizarla con 700px, así lo hará si no usamos esta meta-etiqueta.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0, userscalable=no">
```

Se pueden usar los siguientes parámetros, siempre separados por comas:

- **Width:** ancho de la página (se puede establecer en píxeles o como devicewidth y usará el ancho del que dispone).
- **Height:** alto de la página, funciona igual que el width.
- **Initial-scale:** escala o zoom inicial de la página (este y los demás tipos de escala se establecen con valores como 1.0 para no tener zoom o 2.5 para tener un zoom del 2,5 de aumento, por ejemplo).
- **Minimum-scale:** zoom mínimo que podemos hacer en la página.
- **Maximum-scale:** zoom máximo que podemos hacer en la página.
- **User-scalable:** establece si está permitido o no hacer zoom, con los valores: yes/no.

## @media de CSS3

@media nos permitirá definir cuándo modificar ciertas reglas en nuestro CSS, evaluando condiciones según el dispositivo de visualización. CSS3 nos permitirá definir conjuntos de estilo dependiendo de propiedades como el alto, el ancho o la relación de aspecto de la pantalla. Así serán las definiciones dentro del bloque de la regla. Por ejemplo, las que figuren en screen solo serán interpretadas por

dispositivos conectados a monitores de computadoras y por otro lado projection solo se aplicaría a proyectores.

```
@media screen {  
    ...  
}  
@media projection {  
    ...  
}
```

La propiedad **@media** nos permitirá incluir media queries directamente en nuestras hojas de estilo CSS y mejoramos el rendimiento de la página, ya que no tendremos que cargar archivos CSS adicionales para los diferentes dispositivos. Cuando queramos definir un conjunto de reglas específicas para un dispositivo determinado, utilizaremos la sintaxis:

```
@media screen and (parámetros) and (parámetros) { ... }
```

Vamos a poner un ejemplo que se basa en un menú que hay en una cabecera y tiene cinco links que nos envían a distintas partes de la web y actuará de la siguiente forma los elementos <a>. En principio, tendrá el siguiente código que mantendrá a los cinco elementos en línea.

```
a.menu {  
    display:inline-block;  
    padding:0px 12px;  
    margin:0px 8px;  
}
```

## Propiedades avanzadas

\*Cuando fue agregado en Media Queries

Propiedad	Descripción	Agregado*
any-hover	¿Hay algún mecanismo de entrada disponible para que el usuario se pueda desplazar sobre los elementos?	Nivel 4
any-pointer	¿Hay algún mecanismo de entrada disponible que sea un dispositivo señalador? ¿Qué tan preciso es?	Nivel 4
aspect-ratio	La relación entre el ancho y la altura de la ventana gráfica.	
color	El número de bits por componente de color para el dispositivo de salida.	
color-gamut	El rango aproximado de colores que son compatibles con el agente de usuario y el dispositivo de salida.	Nivel 4
color-index	Número de colores que puede mostrar el dispositivo.	
grid	Si el dispositivo es una cuadrícula o mapa de bits.	
height	Altura del viewport.	
hover	¿El mecanismo de entrada principal permite que el usuario se desplace sobre los elementos?	Nivel 4
inverted-colors	¿El browser o sistema operativo está invirtiendo los colores?	Nivel 4
light-level	Nivel de luz ambiental actual.	Nivel 4
max-aspect-ratio min-aspect-ratio	La relación máxima o mínima entre el ancho y la altura del área de visualización.	
max-color min-color	El número máximo o mínimo de bits por componente de color para el dispositivo de salida.	
max-color-index min-color-index	Número máximo o mínimo de colores que puede mostrar el dispositivo.	
max-height min-height	Altura máxima o mínima del área de visualización, como una ventana del navegador.	
max-monochrome min-monochrome	El número máximo o mínimo de bits por "color" en un dispositivo monocromático (escala de grises).	
max-resolution	Resolución máxima o mínima del dispositivo usando dpi o dpcm.	

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso

(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148

[www.sceu.frba.utn.edu.ar/e-learning](http://www.sceu.frba.utn.edu.ar/e-learning)

min-resolution		
max-width min-width	El ancho máximo o mínimo del área de visualización, como una ventana del navegador.	
monochrome	Número de bits por color en un dispositivo monocromático (escala de grises).	
orientation	La orientación de la vista (landscape o portrait mode).	
overflow-block	Cómo controla el dispositivo de salida el contenido que desborda la ventana gráfica a lo largo del eje del bloque.	Nivel 4
overflow-inline	¿Se puede desplazar el contenido que desborda a lo largo del eje inline?	Nivel 4
pointer	¿El mecanismo de entrada es un dispositivo señalador? ¿Qué tan preciso es?	Nivel 4
resolution	La resolución del dispositivo de salida, usando dpi o dpcm.	
scan	El proceso de escaneo del dispositivo de salida.	
scripting	¿Está disponible el scripting? (por ejemplo JavaScript)	Nivel 4
update	¿Qué tan rápido puede el dispositivo de salida modificar la apariencia del contenido?	Nivel 4
width	El ancho de la ventana.	

# Mobile First

En un modo de diseñar que nos propone tener en cuenta, en primer lugar, al dispositivo móvil.

"Mobile First" significa crear el código primero para los dispositivos más pequeños que los usuarios probablemente tengan, como teléfonos o tabletas. Trabajar en el dispositivo más pequeño y luego acumular desde allí todo en el mismo código y el mismo proyecto, en lugar de uno nuevo para cada tamaño de pantalla.

## Se recomienda:

1. Primero trabajar el código para que se reproduzca perfectamente en un teléfono.
2. Luego ajustar para que se ejecute en una tableta.
3. Luego en un dispositivo de escritorio.

## Veamos algunos ejemplos:

Si por ejemplo queremos que en las pantallas extra pequeñas (xs) el color de fondo que aplica la clase `.miestilo` sea rojo y para el resto de tamaños sea verde, podríamos hacer:

```
.miestilo {  
    background-color: green;  
}  
@media (min-width: 768px) {  
    .miestilo {  
        background-color: red;  
    }  
}
```

O si por ejemplo queremos variar la alineación del texto que se aplica en una clase a partir de las pantallas tipo escritorio:

```
.miestilo {  
    text-align: center;  
    font-size: 40px;  
}  
@media (min-width: 992px) {  
    .miestilo {  
        text-align: left;  
        font-size: 10px;  
    }  
}
```

## Operador “and” “Y”

Podemos sumar diferentes indicaciones con las Media Queries, se utilizan con el operador “and”. En este caso el estilo que definimos se reproducirá en pantallas que van de 400px a 700px:

```
@media (min-width: 768px) and (max-width: 1024px) {  
    .miestilo {  
        text-align: left;  
    }  
}
```

En este caso solo se reproducirá el estilo si la ventana tiene un ancho de de 700px o más y la pantalla esta en formato horizontal.

Por ejemplo, si deseamos modificar un menú de posición horizontal en pantallas grandes a vertical en pantallas chicas podríamos hacer lo siguiente:

```
<div class="topnav">
  <a href="#">Home</a>
  <a href="#">Nosotros</a>
  <a href="#">Contacto</a>
</div>
/* Top navigation bar */
.topnav {
  overflow: hidden;
  background-color: #333;}

/* Topnav links */
.topnav a {
  background-image: url('imagen1')
  display: none;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none; }

@media screen and (min-width: 600px) {
  .topnav a {
    display: block;
    width: 100%;
  background-image: url('imagen2')
  }
}
```

## En resumen

En esta unidad pudimos conocer qué es el Diseño Responsive y cómo aplicarlo. Nos detuvimos en la estrategia Mobile First y vimos que para lograr un diseño responsive necesitamos hacer uso de "Media Queries" de CSS. Las Media Queries las pensamos como una forma de aplicar condicionales a las reglas de CSS. Por medio de las cuales le decimos al navegador qué reglas debe ignorar o aplicar dependiendo del dispositivo del usuario.

Con los contenidos vistos hasta el momento, podremos crear diseños adaptables a distintos dispositivos empleando las prácticas más actuales en desarrollo web.

## Bibliografía utilizada y sugerida

- CSS - Glosario. MDN Web Docs - Mozilla. Recuperado de <https://developer.mozilla.org/es/docs/Web/CSS>
- HTML5 - Responsive Web Design. Microsoft DOCS. Recuperado de: <https://docs.microsoft.com/en-us/archive/msdn-magazine/2011/november/html5-responsive-web-design>
- Media queries. MDN Web Docs. Recuperado de: [https://developer.mozilla.org/es/docs/Web/CSS/Media\\_Queries](https://developer.mozilla.org/es/docs/Web/CSS/Media_Queries)
- Media Queries. W3C Recommendation. Recuperado de: <https://www.w3.org/TR/css3-mediaqueries/>