



UNIVERSIDAD  
NACIONAL  
DE TUCUMÁN

**Dpto. Ciencias de la Computación**



# Laboratorio I / Laboratorio de Software I / Laboratorio de Computación I - 2022 -

---

**Carreras: Agrimensura / Ing. en Informática / Lic. en Informática / Programador Universitario**

## **Clase Teórica 05**

**Docente: Myriam Ruiz**

Licenciada en Informática

Profesora en Computación y Matemática

Programadora Universitaria

# DISEÑO RESPONSIVE

---

# RWD

---

- Responsive Web Design
  - Diseño Web Responsivo
  - Adaptativo o Sensitivo

# Qué implica un buen RWD?



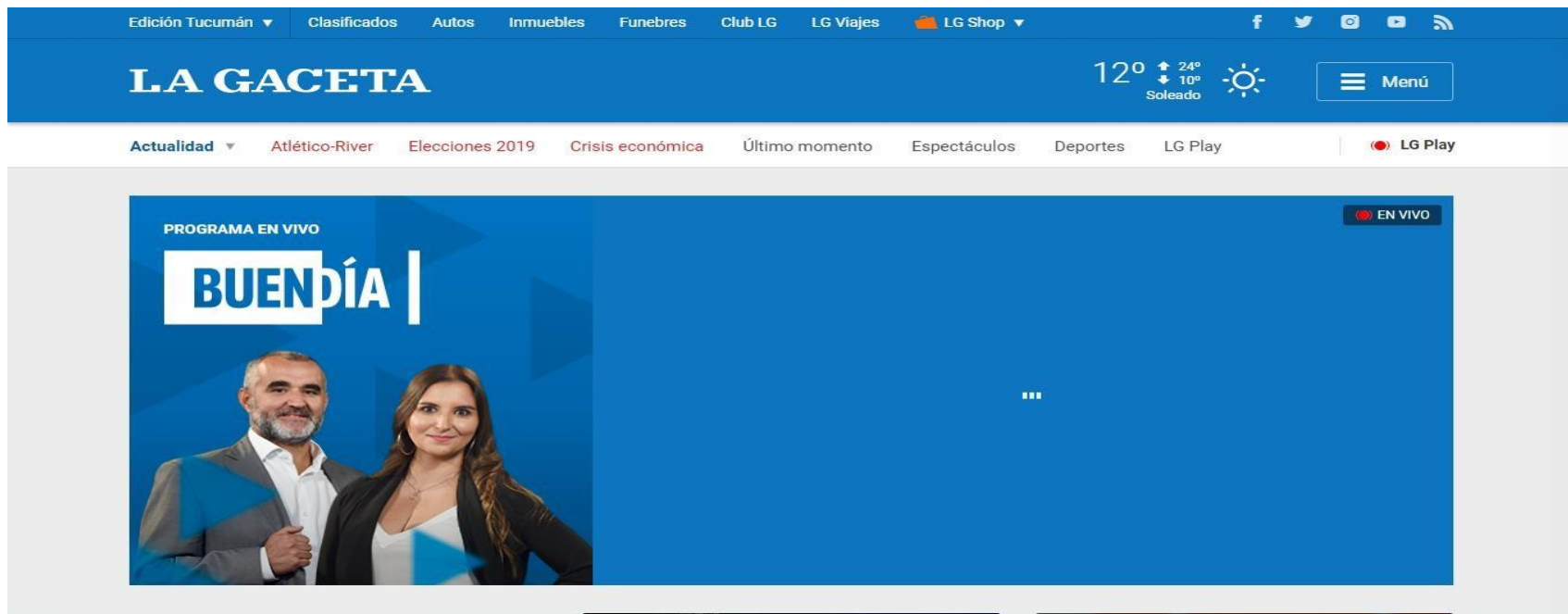
# Qué implica un buen RWD?

---

- El mismo sitio se ve bien en cualquier tamaño de pantalla, sin tener que hacer zoom
- Verse bien no significa que se verá igual, sino que será **cómodo** en el dispositivo que se use

# Ejemplo

## Así luce en PC



# Ejemplo

Así luce en celular Galaxy S5



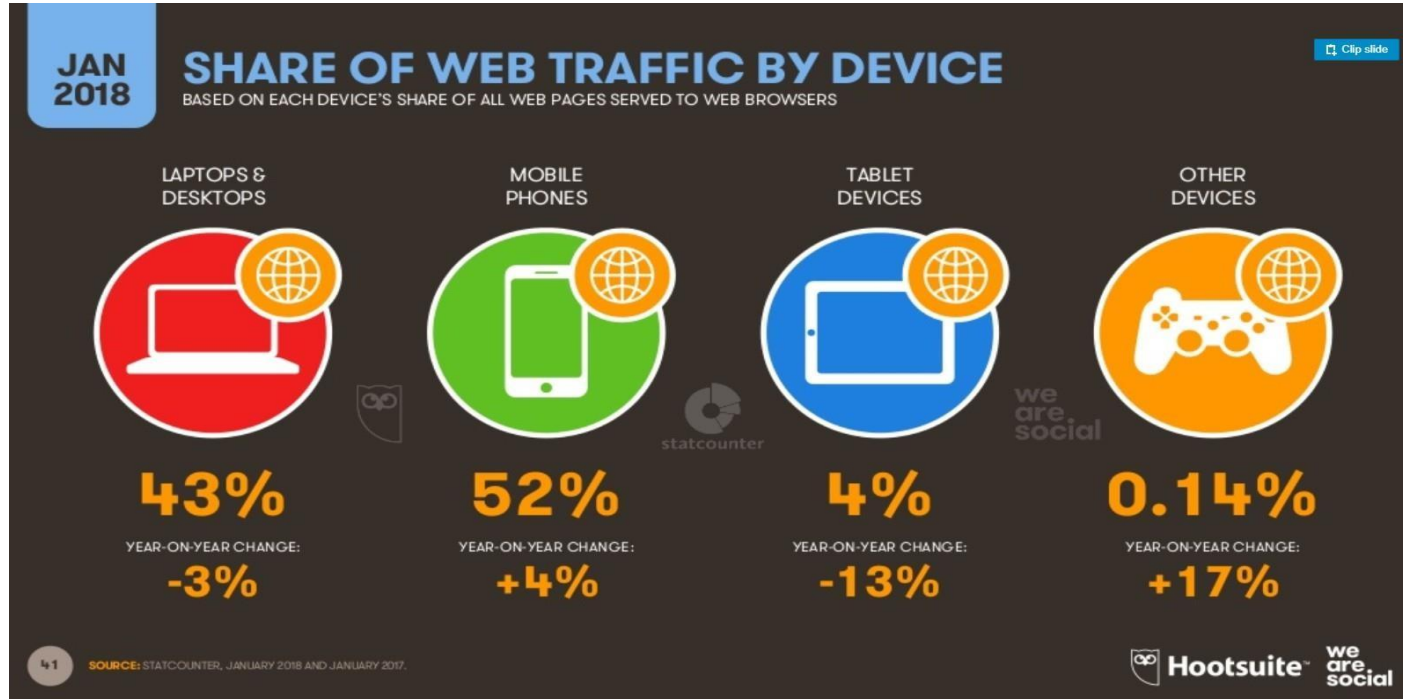
# Por qué pensar más allá del usuario de escritorio?



<https://marketing4ecommerce.net/usuarios-internet-mundo/>



# Por qué pensar más allá del usuario de escritorio?



<https://marketing4ecommerce.net/usuarios-internet-mundo/>

# Análisis de los datos

---

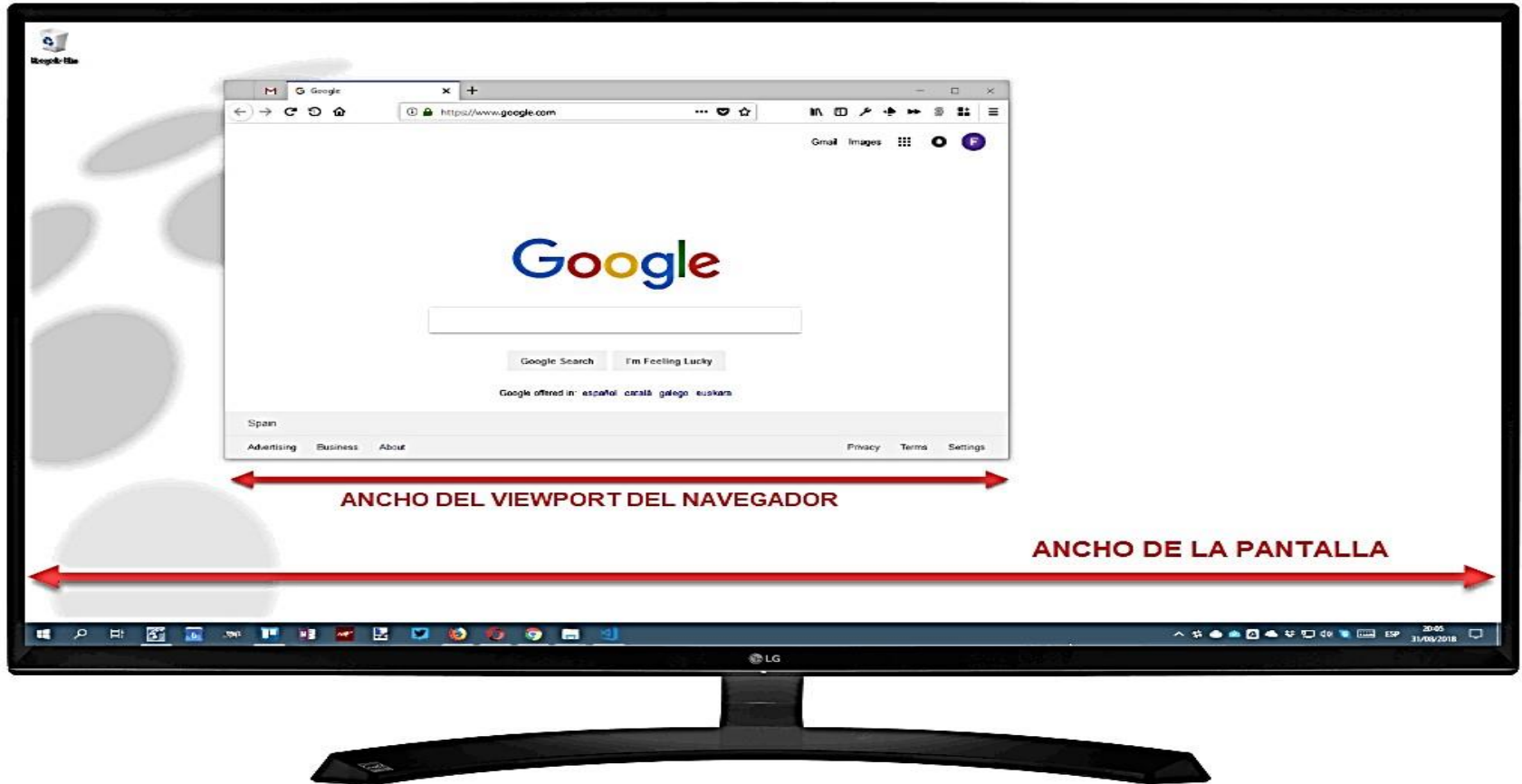
- De la población total del mundo, de más de 7 mil millones, el 67% tiene celular y el 59% tiene Internet.
- Del total de tráfico en Internet, el 52% se realiza desde celulares y 43% desde PC
- Por tanto, el acceso a sitios web está siendo dominado por los celulares

# Resolución vs. Ventana Gráfica

---

- **Resolución** es la cantidad de Píxeles que puede mostrar una pantalla
- **Ventana Gráfica o (Viewport)** es una cantidad de Píxeles simulada por el navegador cuando visualiza una página web. Es el área útil donde se mostrará la página
- Suelen tener cantidad de píxeles diferentes, ya que según cada fabricante se **reescala** el Viewport

# Resolución vs. Ventana Gráfica



# Resolución vs. Ventana Gráfica

---

- **Problemas:**

- Si no configuramos nada y si el tamaño original de la página es superior al tamaño del Viewport, la página se comprime para mostrarse en su totalidad (se ve pequeña)
- Desde el 2016 Google indexa la versión móvil por sobre la de escritorio de un sitio. Debemos tener un buen sitio para móvil.

# Resolución vs. Ventana Gráfica

- Ejemplo: Así luce un sitio con problemas en Galaxy S5



Cómo vemos/  
leemos un sitio  
que no es  
responsive?

Haciendo zoom

# Cómo diseñar un sitio Responsive?

---

Para poder hacer un sitio que se adapte al tamaño de pantalla, debemos:

1. Determinar la dimensión del Viewport en HTML
2. Usar en CSS `@media` para hacer las adaptaciones a distintas pantallas


# Cómo diseñar un sitio Responsive?

---

## 1. Determinar la dimensión del Viewport en HTML

En todas las páginas incorporar en el Head la etiqueta: `<meta name= "viewport"...`

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="author" content="Alumno Laboratorio">
  <meta name="description" content="Prueba de Responsive">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Sitio Responsive</title>
</head>
```





# Cómo diseñar un sitio Responsive?

---

## 1. Determinar la dimensión del Viewport en HTML

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

**width=device-width:** Establece el ancho del viewport tomando el ancho del dispositivo

**initial-scale=1.0:** establece el zoom por defecto para que no tenga zoom

# Cómo diseñar un sitio Responsive?

---

2. Usar en CSS @media para hacer las adaptaciones a distintas pantallas

```
@media destino and (tamaño a comparar) and (orientación)
{
    #menu a {
        display:block;
    }
}
```

# Cómo diseñar un sitio Responsive?

---

@media destino and (tamaño a comparar) and (orientación) {

...

}

destino: tipo de dispositivo en el que se visualizará

- all: Aplicable a todos los dispositivos.
- print: Destinado a material paginado y para documentos visibles en pantalla en modo de vista previa para impresión.
- screen: Destinado a pantallas a color.
- speech: Destinado a sintetizadores de voz.

# Cómo diseñar un sitio Responsive?

---

@media destino and (tamaño a comparar) and (orientación) {

...

}

**tamaño:** se elige pensando en los dispositivos objetivos que esperamos visiten nuestro sitio, se mide en px.

# Cómo diseñar un sitio Responsive?

---

@media destino and (tamaño a comparar) and (orientación) {

...

}

orientación: depende de la posición del celular

- orientation: portrait
- orientation: landscape

# Cómo diseñar un sitio Responsive?

---

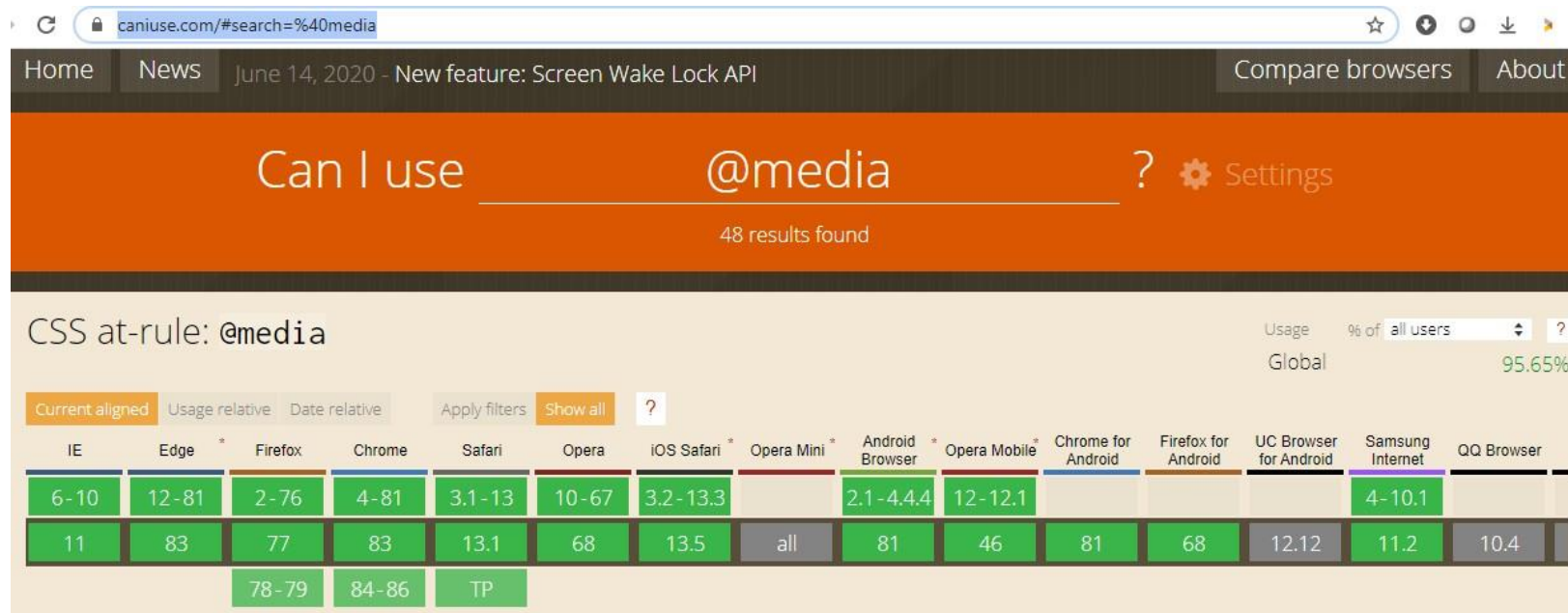
orientation: portrait



orientation: landscape



# Compatibilidad de @media

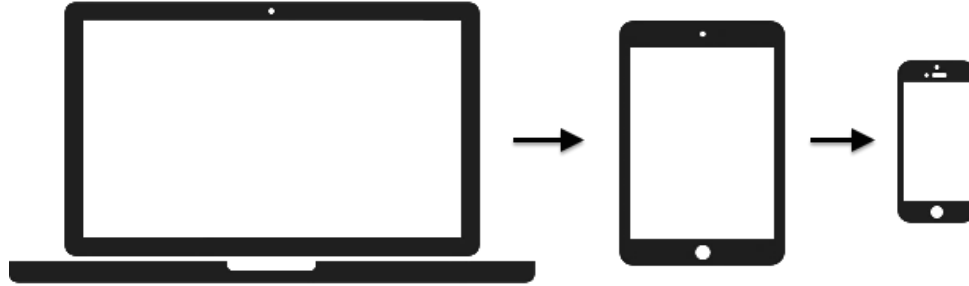


<https://caniuse.com/#search=%40media>

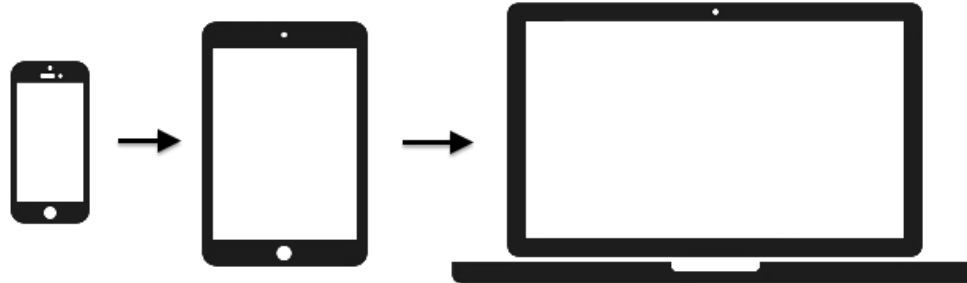
# Dos enfoques distintos

---

Desktop first



Mobile first





# Desktop First

---

La filosofía Desktop First recomienda:

1. Empezar diseñando el sitio de escritorio (siempre que sepamos que la mayor parte de los ingresos es por equipos de escritorio)
2. No se prioriza el contenido relevante al principio, se decide con posterioridad que es lo que se va a quitar o que elementos se adaptarán para pantallas pequeñas.

# Mobile First

---

La filosofía Mobile First recomienda:

1. Empezar diseñando el sitio para móviles
2. Priorizar el contenido relevante y eliminar lo irrelevante de ser necesario, desde el inicio
3. Debe ser cómodo para dispositivos muy pequeños y de carga rápida
4. Se tiene que poder hacer las mismas operaciones en versión móvil y escritorio

# Orden de los CSS

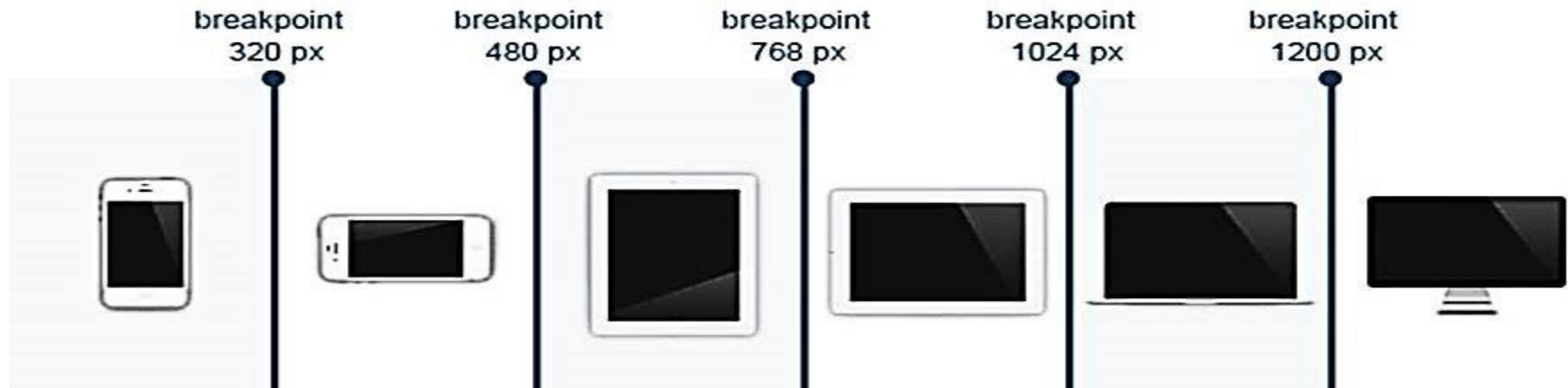
---

- Recomendamos trabajar con Mobile First.
  - Escribir los @media, primero para pantallas de menor tamaño y a continuación las de mayor tamaño

# Breakpoints

---

- Los puntos de quiebre (Breakpoints) son tamaños de pantalla en los cuales el CSS debiera ser distinto. Existen tantos tamaños que sólo nos enfocaremos en los más usados.



# Breakpoints

---

- Otra forma de verlo (siempre se habla de anchos):
- 320px-480px: Celulares
- 481px-768px: iPads, Tablets
- 769px-1024px: Pantallas pequeñas, laptops
- 1025px-1200px: Pantalla grande, PC
- 1201px y más - Pantallas extra grandes, TV

# Ejemplo

---

- Ver ejemplo en un celular:

<https://laboratorio2021.neocities.org/>

# Recomendaciones para Mobile First

---

- Hacer una parte del CSS para todas las pantallas, que no incluya tamaños, padding, border, margin, align, etc. Este irá fuera de los @media.
- Hacer los @media para cada tipo de pantalla, de la más pequeña a la más grande.
- Siempre se utiliza min-width (ancho mínimo) para decidir cuál CSS aplicar.

# Esquema general de los @media

---

```
*** CSS GENERAL ****  
*/  
  
/**** FIN *****/  
  
/*  
*** CSS Móviles ***  
*/  
  
/**** FIN *****/  
  
/*  
*** CSS Tablets ***  
*/  
  
/**** FIN *****/  
  
/*  
*** CSS Monitores ***  
*/  
  
/**** FIN *****/
```



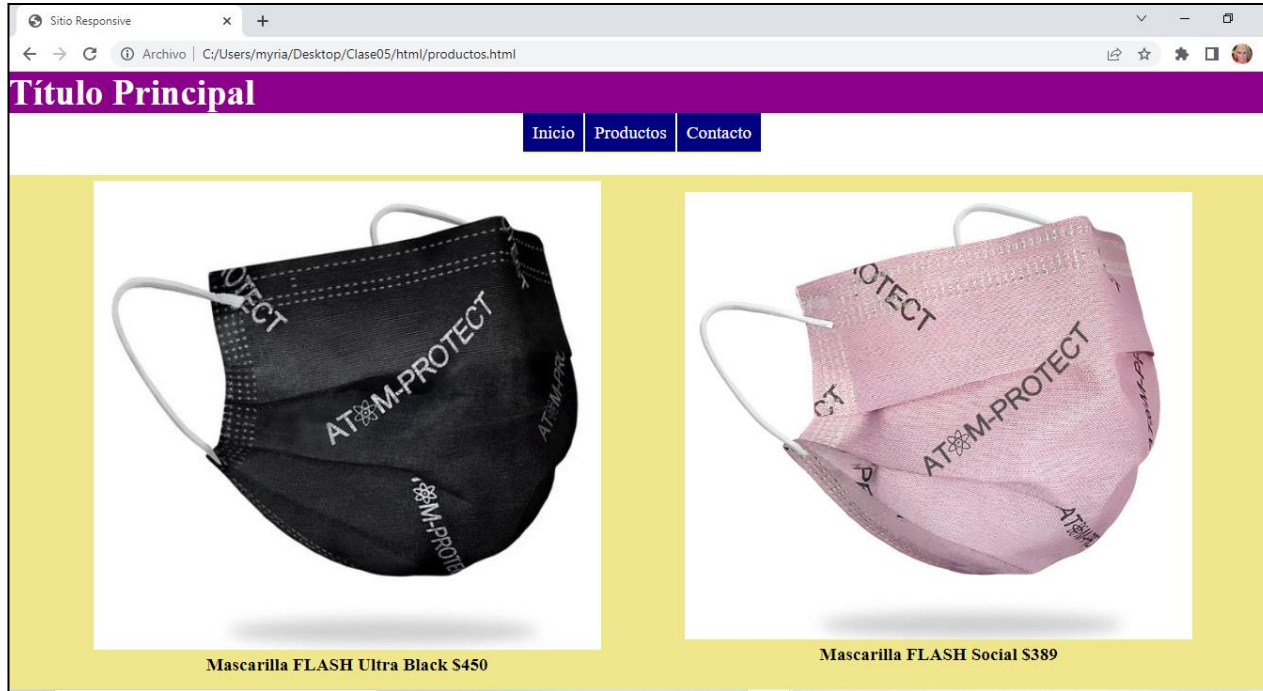
# Recomendaciones para Mobile First

---

- Un buen diseño concentrará la mayoría de las configuraciones en el tamaño más pequeño, modificando en los siguientes tamaños sólo aquello que sea necesario.
- Ej: Si en tamaño pequeño usamos `flex-direction: column`; en tamaño grande quizá deba "pisar" dicha propiedad con `flex-direction: row`

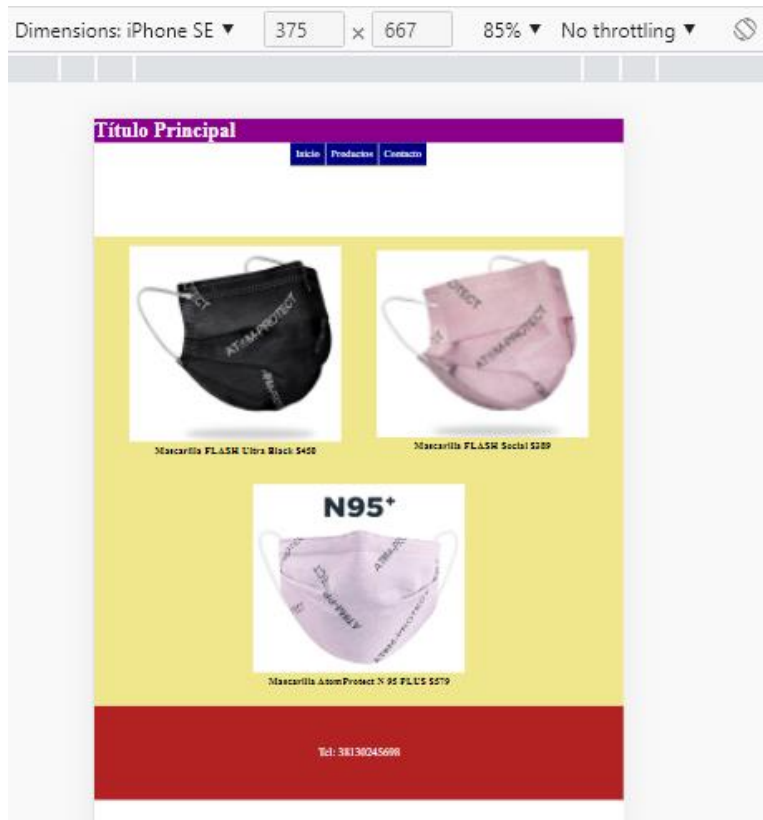
# Ej: Sitio con CSS sin @media ni viewport

Cómo luce en escritorio:



# Ej: Sitio con CSS sin @media ni viewport

Cómo luce en celular:



# Recomendaciones - Ejemplo

---

- Hacer una parte del CSS para todos los sitios, que no incluya tamaños, padding, border, margin, etc.

```
* {  
    margin:0;  
    padding:0;  
    border:0;  
}  
  
/* CSS General */  
#contenedor-menu a {  
    text-decoration: none;  
    padding: 0.5em;  
    color: maroon;  
}
```

```
#contenedor-enlaces a {  
    color: white;  
    margin-right: 0.1em;  
}  
  
#contenedor-enlaces {  
    list-style-type: none;  
}  
  
main {  
    background-color: khaki;  
}
```

# Recomendaciones - Ejemplo Mobile First

- Hacer los @media para cada tipo de pantalla, de la más pequeña a la más grande, usando min-width

```
/* CSS para Celular */
@media all and (min-width:320px) {
  header {
    background-color: ■maroon;
    color: □white;
    font-size: 1.3em;
  }

  #contenedor-menu #contenedor-enlaces {
    height: 0vh;
    transition: 1s height ease;
    overflow: hidden;
  }
}
```

```
/* CSS Pantalla 480px y superior */
@media all and (min-width:480px) {
  header {
    background-color: ■darkmagenta;
    font-size: 1.2em;
  }

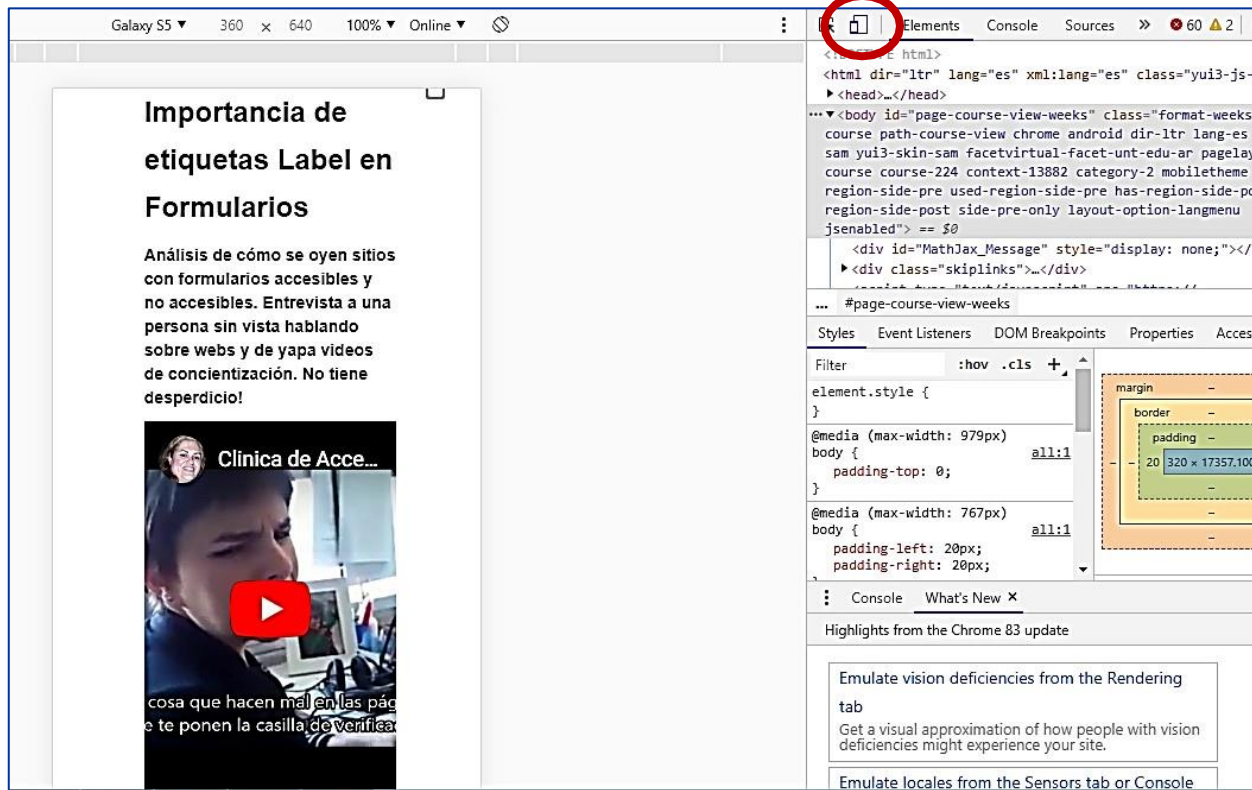
  #contenedor-menu #contenedor-enlaces {
    height: 10vh;
  }
}
```

# Recomendaciones - Ejemplo



# Cómo simular un tamaño de Viewport?

- En Google Chrome: Presionar F12



# Cómo simular un tamaño de Viewport?

- En Google Chrome. Se puede girar el dispositivo





# Cómo simular un tamaño de Viewport?

- En Mozilla Firefox: Presionar F12 o Ctrl + Shift + M



# Decisiones de Diseño

---

- Al momento de desarrollar un sitio deberemos tomar ciertas decisiones en cuánto a elementos que requieren ser adaptados a cada tamaño de pantalla. Estos elementos son:
  - Menú
  - Imágenes (y videos)
  - Formularios
  - Tablas
  - Columnas

# TRANSITION

---

# Transition

---

- Permite definir la transición entre dos estados de un elemento, o sea entre dos valores.
- Existen 3 formas de trabajo:
  - Incorporar la propiedad en el estado normal del elemento (Recomendada)
  - Incorporar la propiedad en el estado normal y en el estado de focus y hover
  - Incorporar en el estado de target (cuando un elemento es el destino de un enlace)
    - Sólo si no se desea que vuelva del estado al que pasó

# Transition

---

**transition-property**: propiedad a modificar

**transition-duration**: duración de la transición (1s, 2s)

**transition-timing-function**: ritmo o aceleración de la transición

**transition-delay**: demora para iniciar la transición

# Transition

---

**transition-timing-function**: ritmo o aceleración de la transición

- **linear**: La animación se realiza de manera uniforme
- **ease**: La animación acelera al inicio se retarda un poco y se acelera al final de nuevo
- **ease-in**: La animación se retarda al inicio, pero acelera al final
- **ease-out**: La animación se acelera al inicio, pero se retarda al final
- **ease-in-out**: La animación se retarda al inicio, se acelera un poco, luego se retarda al final de nuevo

# Transition - Ejemplo

---

Ej: Cambiar el color del texto al hacer hover sobre un enlace, con duración de 1 segundo

```
#menu a {  
    background-color: blue;  
    color: white;  
    transition-property: color;  
    transition-duration: 1s;  
    transition-timing-function: ease;  
    transition-delay: 0s;  
}  
  
#menu a:hover {  
    color: yellow;  
}
```

# Transition

---

En una única instrucción

**transition**: propiedad duración ritmo demora;



# Transition suave con una propiedad

---

En una única instrucción

```
#menu a {  
    background-color: blue;  
    color: white;  
    transition: color 1s ease 0s;  
}  
  
#menu a:hover {  
    color: yellow;  
}
```

# Transition suave con más de una propiedad

---

Se puede cambiar más de una propiedad

```
#menu a {  
    background-color: blue;  
    color: white;  
    transition: color 1s ease 0s, background-color 1s ease 0s;  
}  
#menu a:hover {  
    color: yellow;  
    background-color: green;  
}
```

# Transition suave

---

## Explicación del transition:

- El hecho de poner el transition en el estado original del enlace (en a) hace que el cambio se produzca tanto al producirse el hover como al volver al estado original (recomendable para menú)
- Si pusiéramos el transition en el hover (a:hover), esto haría que cambie de estado suavemente al ir hacia el hover, pero al volver al estado original se vería una vuelta brusca, sin transición.
- Depende el efecto que se quiera lograr, conviene poner el transition en el estado original o en el final.

# Transition

---

- Para versiones viejas de navegadores usar los prefijos de compatibilidad:

```
transition: padding 500ms ease;  
-ms-transition: padding 500ms ease;  
-webkit-transition: padding 500ms ease;  
-moz-transition: padding 500ms ease;  
-o-transition: padding 500ms ease;
```

# MENÚ PARA MÓVILES

---

# Menú para móviles

---

- Se puede optar por un menú que se muestre al principio, desplegado todo el tiempo (menos común, no tan estético)
- Otra opción es diseñar un menú que aparezca oculto y que se despliegue solo cuando lo deseamos (más común)

# Menú para móviles

---

- Se puede optar por un menú que se muestre al principio, desplegado todo el tiempo (menos común)
  - Consiste en mostrar la etiqueta `<a>` como bloque y darle el ancho de toda la pantalla del móvil

# Menú para móviles – Todo el tiempo igual

- Ejemplo:



```
@media all and (min-width: 320px) {  
  #menu li a{  
    display: block;  
    max-width: 100%;  
    margin-bottom: 0.4%;  
    padding: 3%;  
    font-size: 1.3em;  
  }  
}
```



# Menú desplegable (para móviles)



# Menú desplegable (para móviles)

---

La idea consiste en tener un contenedor (section) del menú y utilizar enlaces internos, a fin de poder “detectar” cuando le está llegando (target) o no un enlace al contenedor y según eso ocultar o mostrar el menú (cambiando su altura). La transition le da un efecto de despliegue progresivo.

# Menú para móviles - Desplegable

---

## Código html

```
<section id="contenedor-menu">
  <a href="#contenedor-menu"></a>
  <a href="#"></a>

  <nav>
    <ul id="contenedor-enlaces">
      <li><a href="#">Inicio</a></li>
      <li><a href="html/quienes.html">Quiénes somos</a></li>
      <li><a href="html/novedades.html">Novedades</a></li>
      <li><a href="html/contactenos.html">Contactenos</a></li>
    </ul>
  </nav>
</section>
```

# Menú para móviles - Desplegable

---

```
/* DISEÑO PARA CELULAR */
@media all and (min-width: 320px) {
  /* Menú */
  #contenedor-menu img {
    width: 10vw;
  }

  #contenedor-menu #cerrar {
    display: none; /* Oculta imagen de X */
  }

  #contenedor-menu #contenedor-enlaces {
    height: 0; /* Cambia altura a cero */
    transition: height 1s ease; /* Efecto de transición */
    overflow: hidden; /* oculta lo que sobresalga */
  }
}
```

# Menú para móviles - Desplegable

---

```
#contenedor-menu:target #abrir {  
    display: none; /* Oculta el la imagen de hamburguesa */  
}  
  
#contenedor-menu:target #cerrar {  
    display: block; /* Muestra la imagen de X */  
}  
  
#contenedor-menu:target #contenedor-enlaces {  
    height: 40vh;  
    transition: height 1s ease;  
}
```

**target** sucede  
cuando recibe  
un enlace

# Menú para móviles

---

```
#contenedor-enlaces a {  
    display: block; /* Pone uno debajo del otro */  
    width: 100%;  
    margin-bottom: 0.25em;  
    margin-top: 0.25em;  
    text-decoration: none;  
    background-color: □white;  
    padding: 1em; /* Genera espacio a la vuelta, lo hace más tocable */  
    color: ■black;  
}  
  
#contenedor-menu {  
    background-color: ■brown;  
}
```

# Menú para móviles (Vuelta a menú horizontal)

---

```
@media all and (min-width: 769px) {  
  #abrir, #cerrar {  
    display: none; /* oculta imágenes del menú */  
  }  
  
  #contenedor-menu #contenedor-enlaces {  
    display: flex;  
    flex-direction: row;  
    height: 10vh;  
  }  
  
  #contenedor-menu #contenedor-enlaces a:hover,  
  #contenedor-menu #contenedor-enlaces a:focus {  
    background-color: #ffffaa;  
  }  
}
```

# Imágenes

---

- Las imágenes son fácilmente adaptables utilizando un ancho en porcentaje acorde a cada tamaño de pantalla.
- Problemas:
  - Ampliar con CSS una imagen de baja resolución hace que se vea pixelada, muy cuadriculada, sin nitidez.
  - Como contraparte, poner una imagen de alta resolución y luego reducir su tamaño con CSS hace que se vea bien, aunque con una sobrecarga innecesaria de transferencia de datos



# Imágenes

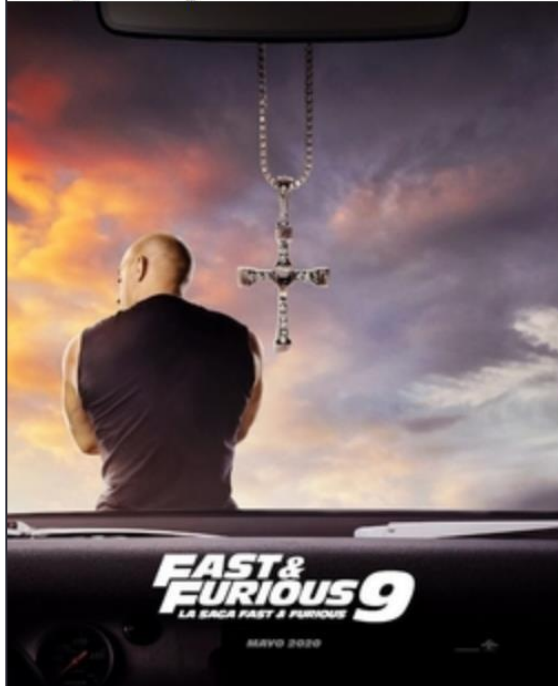
---

- Una posible solución:
  - Para evitar el pixelado de la imagen cuando se adapta a un tamaño que excede tu tamaño original, se puede utilizar la propiedad **max-width** en vez de **width**, a fin de que se adapte siempre que no exceda tu tamaño real.

# Imágenes

- Así luce con **width**: 100% para un contenedor con el doble del ancho original de la imagen

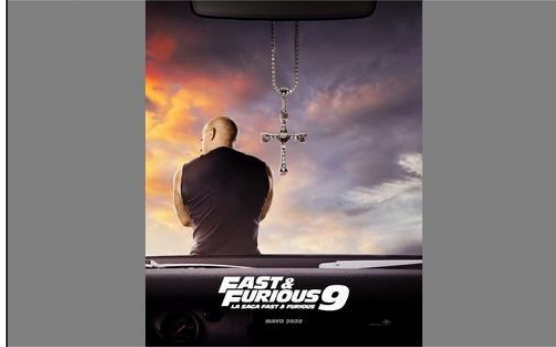
Imagen Responsivo



# Imágenes

- Así luce con **max-width**: 100% para un contenedor (fondo gris) con el mismo ancho del ejemplo anterior

Imagen Responsivo



# Galería Responsive

---

- Para lograr que una galería se adapte al tamaño de pantalla, deberemos utilizar en el contenedor de la galería (article), como mínimo:
  - flex-direction (para mostrarla vertical u horizontal)
  - flex-wrap: wrap (para que las imágenes puedan ubicarse debajo)
- Además, en cada tamaño de pantalla, se deberá cambiar el ancho de figure, a fin de que se muestren más o menos imágenes por línea.

# Galería Responsive

- html:

```
<section>
  <article id="galeria">
    <figure>
      
      <figcaption>Mascarilla FLASH Ultra Black $450</figcaption>
    </figure>
    <figure>
      
      <figcaption>Mascarilla FLASH Social $389</figcaption>
    </figure>
    <figure>
      
      <figcaption>Mascarilla AtomProtect N 95 PLUS $579</figcaption>
    </figure>
  </article>
</section>
```

# Galería Responsive

---

- Tamaño pequeño:

```
/* CSS para Celular */
@media all and (min-width:320px) {
  #galeria {
    display: flex;
    flex-direction: column;
    align-items: center;
  }

  main figure {
    max-width: 95%;
    margin-top: 1vh;
    margin-bottom: 2vh;
    font-size: 1.2em;
    font-weight: bold;
  }
}
```

```
main figcaption {
  text-align: center;
}

main figure img {
  width: 100%;
}
```

# Galería Responsive

---

- Tamaño más grande:

```
/* CSS Pantalla 480px y superior */
@media all and (min-width:480px) {
  /* Galería */
  #galeria {
    display: flex;
    flex-direction: row;
    flex-wrap: wrap;
    justify-content: space-evenly;
  }

  #galeria figure {
    width: 40%;
  }
}
```

# Selector de Hijos

Aún sin trabajar con Flexbox, se considera que las etiquetas dentro de otras son sus hijos.

Existen pseudo-clases CSS que permiten seleccionar de manera particular a estas etiquetas hijo.

También existe un tipo de selector para hijos directos, conocido como mayor que, >



# Selector de Hijos: Pseudo-clases

**Seleccionar Primer o Último hijo:**

```
elemento:first-child {
```

```
}
```

```
elemento:last-child {
```

```
}
```

# Selector de Hijos: Pseudo-clases

## Seleccionar Primer hijo:

```
<section>
  <article>
    <p>Grupo 1: Primer párrafo</p>
    <p>Grupo 1:Segundo párrafo</p>
    <p>Grupo 1:Tercer párrafo</p>
    <p>Grupo 1:Último párrafo</p>
  </article>
  <article>
    <p>Grupo 2: Primer párrafo</p>
    <p>Grupo 2: Segundo párrafo</p>
    <p>Grupo 2: Tercer párrafo</p>
    <p>Grupo 2: Último párrafo</p>
  </article>
  <p>Grupo 3: Primer Párrafo</p>
</section>
```

```
p:first-child {
  background-color: yellow;
}
```

Grupo 1: Primer párrafo

Grupo 1:Segundo párrafo

Grupo 1:Tercer párrafo

Grupo 1:Último párrafo

Grupo 2: Primer párrafo

Grupo 2: Segundo párrafo

Grupo 2: Tercer párrafo

Grupo 2: Último párrafo

Grupo 3: Primer Párrafo

# Selector de Hijos: Pseudo-clases

Observe que **:first-child** detecta los casos que el item es 1er hijo de un padre, pero en el párrafo que dice Grupo 3, no tiene hermanos del mismo tipo que le permitan al CSS determinar que es el 1er hijo

Grupo 1: Primer párrafo

Grupo 1: Segundo párrafo

Grupo 1: Tercer párrafo

Grupo 1: Último párrafo

Grupo 2: Primer párrafo

Grupo 2: Segundo párrafo

Grupo 2: Tercer párrafo

Grupo 2: Último párrafo

Grupo 3: Primer Párrafo



# Selector de Hijos: Pseudo-clases

**Seleccionar Hijos Pares o Impares:**

```
elemento:nth-child(odd) {
```




```
}
```

```
elemento:nth-child(even) {
```

```
}
```

# Selector de Hijos: Pseudo-clases

## Seleccionar Hijos Impares (odd) y Pares (even):

```
p:nth-child(odd) {  
    background-color:  yellow;  
}  
  
p:nth-child(even) {  
    background-color:  brown;  
    color:  white;  
}
```

Grupo 1: Primer párrafo

Grupo 1: Segundo párrafo

Grupo 1: Tercer párrafo

Grupo 1: Último párrafo

Grupo 2: Primer párrafo

Grupo 2: Segundo párrafo

Grupo 2: Tercer párrafo

Grupo 2: Último párrafo

Grupo 3: Primer Párrafo

# Selector de Hijos: Pseudo-clases


Seleccionar un Hijo en particular

```
elemento:nth-child(número) {  
  
}
```

```
elemento:nth-child(fórmula) {  
  
}
```

# Selector de Hijos: Pseudo-clases

**Seleccionar un Hijo en particular: En este caso vamos a elegir el hijo 2 y los que cumplan la fórmula  $2n+1$  (es como en Matemática)**

```
p:nth-child(2) {  
  background-color:  #4fff4f;  
}  
  
p:nth-child(2n+1) {  
  background-color:  brown;  
  color:  white;  
}
```

Grupo 1: Primer párrafo

Grupo 1: Segundo párrafo

Grupo 1: Tercer párrafo

Grupo 1: Último párrafo

Grupo 2: Primer párrafo

Grupo 2: Segundo párrafo

Grupo 2: Tercer párrafo


Grupo 2: Último párrafo

Grupo 3: Primer Párrafo

# Selector de Hijos directos: >

## Seleccionar un Hijo directo

```
<article>
  <ul class="lista-uno">
    <li>Lista 1 - Item 1</li>
    <li>Lista 1 - Item 2</li>
    <ul class="lista-dos">
      <li>Lista 2 - Item 1</li>
      <li>Lista 2 - Item 2</li>
    </ul>
  </ul>
</article>
```



```
.lista-uno > li {
  background-color: #51a8ff;
}
```

- Lista 1 - Item 1
- Lista 1 - Item 2
  - Lista 2 - Item 1
  - Lista 2 - Item 2



# Selector de Hijos directos: >

**Sin selector de Hijo directo (se seleccionan también "nietos", hijos del item hijo ul)**

```
<article>
  <ul class="lista-uno">
    <li>Lista 1 - Item 1</li>
    <li>Lista 1 - Item 2</li>
    <ul class="lista-dos">
      <li>Lista 2 - Item 1</li>
      <li>Lista 2 - Item 2</li>
    </ul>
  </ul>
</article>
```

```
.lista-uno li {
  background-color: #51a8ff;
}
```

- Lista 1 - Item 1
- Lista 1 - Item 2
  - Lista 2 - Item 1
  - Lista 2 - Item 2

# Tabla Responsive

---

- Una forma de hacer una tabla que se adapte al tamaño de los celulares (o tamaños pequeños) es usar Flexbox para cambiar el sentido de las filas (se deben mostrar como una columna).
- Además, se debe ocultar las filas de encabezado originales y hacer "aparecer" (incluso se oirá con NVDA) los títulos para cada fila

# Tabla Responsive

- Ej: Tabla en Desktop Vs. en Celular
- Resultado Final en desktop (Monitores)



The screenshot shows a web browser window with the title 'Tabla Responsive'. The address bar shows the file path 'C:/Users/myria/Desktop/Clase05/html/tabla.html'. The page has a purple header with the text 'Título Principal'. Below the header is a navigation bar with three buttons: 'Inicio', 'Productos', and 'Contacto'. The main content area has a yellow background and contains a table with the caption 'Tabla de ejemplo'. The table has 6 columns and 4 rows. The first row contains the column headers: 'Titulo 1', 'Titulo 2', 'Titulo 3', 'Titulo 4', 'Titulo 5', and 'Titulo 6'. The subsequent three rows contain data: 'Fila 1, Col 1' through 'Fila 1, Col 6', 'Fila 2, Col 1' through 'Fila 2, Col 6', and 'Fila 3, Col 1' through 'Fila 3, Col 6'. At the bottom of the page is a red footer with the text 'Tel: 38130245698'.

Título Principal					
Inicio Productos Contacto					
Tabla de ejemplo					
Titulo 1	Titulo 2	Titulo 3	Titulo 4	Titulo 5	Titulo 6
Fila 1, Col 1	Fila 1, Col 2	Fila 1, Col 3	Fila 1, Col 4	Fila 1, Col 5	Fila 1, Col 6
Fila 2, Col 1	Fila 2, Col 2	Fila 2, Col 3	Fila 2, Col 4	Fila 2, Col 5	Fila 2, Col 6
Fila 3, Col 1	Fila 3, Col 2	Fila 3, Col 3	Fila 3, Col 4	Fila 3, Col 5	Fila 3, Col 6

Tel: 38130245698

# Tabla Responsive

- Ej: Tabla en Desktop Vs. en Celular. Resultado Final en celular:

The screenshot shows a web browser window with the title "Tabla Responsive" and the URL "C:/Users/myria/Desktop/Clase05/html/tabla.html". The browser dimensions are set to 360 x 740 at 75% zoom, simulating a Samsung Galaxy S... device. The page content includes a main title "Título Principal" in a red box, a hamburger menu icon, and a table titled "Tabla de ejemplo". The table has two sections: "Fila 1" and "Fila 2", each with 6 columns. The first section is highlighted in yellow, and the second section is highlighted in green. The Chrome DevTools interface is open on the right, showing the "Network" tab. A red arrow points to the "Always match Chrome's language" button in the DevTools toolbar.

Tabla de ejemplo					
Título 1	Fila 1, Col 1				
Título 2	Fila 1, Col 2				
Título 3	Fila 1, Col 3				
Título 4	Fila 1, Col 4				
Título 5	Fila 1, Col 5				
Título 6	Fila 1, Col 6				
Título 1	Fila 2, Col 1				
Título 2	Fila 2, Col 2				
Título 3	Fila 2, Col 3				
Título 4	Fila 2, Col 4				
Título 5	Fila 2, Col 5				
Título 6	Fila 2, Col 6				

# Tabla Responsive

- Ej: Código HTML (4 filas y 6 columnas)

```
<article>
  <table id="tabla1">
    <caption>Tabla de ejemplo</caption>
    <thead>
      <tr id="tabla-encabezado">
        <th scope="col">Título 1</th>
        <th scope="col">Título 2</th>
        <th scope="col">Título 3</th>
        <th scope="col">Título 4</th>
        <th scope="col">Título 5</th>
        <th scope="col">Título 6</th>
      </tr>
    </thead>
    <tbody>
```

```
<tbody>
  <tr>
    <td>Fila 1, Col 1</td>
    <td>Fila 1, Col 2</td>
    <td>Fila 1, Col 3</td>
    <td>Fila 1, Col 4</td>
    <td>Fila 1, Col 5</td>
    <td>Fila 1, Col 6</td>
  </tr>
  <tr>
    <td>Fila 2, Col 1</td>
    <td>Fila 2, Col 2</td>
    <td>Fila 2, Col 3</td>
    <td>Fila 2, Col 4</td>
    <td>Fila 2, Col 5</td>
    <td>Fila 2, Col 6</td>
```

# Tabla Responsive

---

- Ej: inicio del CSS (no la hace responsive aún)

```
/* CSS para Celular */
@media all and (min-width:320px) {
  /* TABLA */
  #tabla1 {
    border-collapse: collapse;
    width: 95%; /* ancho total de la tabla */
    margin: 0 auto; /* centra la tabla */
  }

  #tabla1 th, #tabla1 td {
    border: 0.1em solid navy;
    padding: 0.3em;
  }

  #tabla1 td {
    background-color: white;
  }
}
```

# Tabla Responsive

---

- Ej: continuación del CSS (modificaciones para que sea vea en celular)

```
#tabla1 #tabla-encabezado {  
  display: none; /* oculta fila de títulos */  
}  
  
#tabla1 tr {  
  display: flex;  
  flex-direction: column; /* muestra una fila como columna */  
  margin-bottom: 0.7em; /* separa filas */  
}
```

# Tabla Responsive

---

- Ej: Hasta acá, en celular lucirá así (están las filas pero no se sabe a qué título pertenece el dato):

Tabla de ejemplo					
Fila 1,Col 1					
Fila 1,Col 2					
Fila 1,Col 3					
Fila 1,Col 4					
Fila 1,Col 5					
Fila 1,Col 6					
Fila 2, Col 1					
Fila 2, Col 2					
Fila 2, Col 3					
Fila 2, Col 4					
Fila 2, Col 5					
Fila 2, Col 6					



# Tabla Responsive

---

- Para hacer aparecer los títulos de las columnas vamos a recurrir a una nueva pseudo-clase:
  - elemento:**nth-of-type**(número)
    - Permite seleccionar un elemento de un tipo dado, según el número de orden que ocupe con respecto a sus hermanos
- Y un pseudo-elemento:
  - elemento:**before**      **before** significa "antes de" y posibilita agregar contenido a la izquierda del elemento

# Tabla Responsive

---

- Vamos a agregar los títulos usando la nueva pseudo-clase y el pseudo-elemento así:

```
td:nth-of-type(1):before {  
    content: "Título 1";  
}
```

En este caso se elije de los td de toda fila, el 1ero (sería Fila x, columna 1), y antes de este se escribirá la frase "Título 1"

# Tabla Responsive

---

```
td:nth-of-type(2):before {  
    content: "Título 2";  
}
```

En este caso se elije de los td de toda fila, el 2do (sería Fila x, columna 2), y antes de este se escribirá la frase "Título 2"

# Tabla Responsive

---

- Ej: continuando con el CSS que teníamos, deber quedar así:

```
#tabla1 td:nth-of-type(1):before {  
|   content:"Título 1";  
|}  
  
#tabla1 td:nth-of-type(2):before {  
|   content:"Título 2";  
|}  
  
#tabla1 td:nth-of-type(3):before {  
|   content:"Título 3";  
|}
```

```
#tabla1 td:nth-of-type(4):before {  
|   content:"Título 4";  
|}  
  
#tabla1 td:nth-of-type(5):before {  
|   content:"Título 5";  
|}  
  
#tabla1 td:nth-of-type(6):before {  
|   content:"Título 6";  
|}
```

# Tabla Responsive

---


- Ej: hasta acá ya casi está, faltan detalles de estética (ya se ve título a la izquierda del dato):

Tabla de ejemplo	
Título 1	Fila 1, Col 1
Título 2	Fila 1, Col 2
Título 3	Fila 1, Col 3
Título 4	Fila 1, Col 4
Título 5	Fila 1, Col 5
Título 6	Fila 1, Col 6
Título 1	Fila 2, Col 1
Título 2	Fila 2, Col 2
Título 3	Fila 2, Col 3
Título 4	Fila 2, Col 4

# Tabla Responsive


---

- Ej: Se logran los detalles usando la pseudo-clase before para los td:

```
#tabla1 td:before {  
    background-color:  gray;  
    padding: 0.3em;  
    margin-right: 0.5em;  
}
```

# Tabla Responsive

- Resultado final para celular

Título Principal	
	
Tabla de ejemplo	
Título 1	Fila 1, Col 1
Título 2	Fila 1, Col 2
Título 3	Fila 1, Col 3
Título 4	Fila 1, Col 4
Título 5	Fila 1, Col 5
Título 6	Fila 1, Col 6
Título 1	Fila 2, Col 1
Título 2	Fila 2, Col 2
Título 3	Fila 2, Col 3
Título 4	Fila 2, Col 4
Título 5	Fila 2, Col 5
Título 6	Fila 2, Col 6
Título 1	Fila 3, Col 1
Título 2	Fila 3, Col 2

# Tabla Responsive

- En el mismo archivo CSS, a continuación, el @media para tamaños más grandes (debe volver a lucir como tabla):

```
@media all and (min-width:480px) {  
    /* TABLA */  
    #tabla1 {  
        width: 60%;  
    }  
  
    #tabla1 #tabla-encabezado {  
        display: table-row;  
    }  
}
```

```
#tabla1 tr {  
    display: table-row;  
}  
  
#tabla1 td:before {  
    display: none;  
}
```



# Formulario Responsivo

---

- Resultado Final:

Formulario para Monitor

Vs.

Para Pantalla Pequeña

Apellido:

Correo:

Recibir novedades:  
Covid: ☐ Vacunación: ☐ Nuevas olas: ☐

Apellido:

Correo:

Recibir novedades:  
Covid: ☐  
Vacunación: ☐  
Nuevas olas: ☐

# Formulario Responsive

---

- Pasos para Formulario Pantallas Pequeñas:
  - Darle al formulario el 100% (o 95%) del ancho de pantalla
  - Mostrar los label e input como bloque, con ancho del 100% del form.
  - Botones uno debajo del otro, ocupando el 100% del form
  - Border reducido acorde a como se verá en la pantalla
  - Altura y Padding necesarios para tener una **mayor superficie de contacto**, que permita presionar o seleccionar elementos con la punta de un dedo.

# Formulario Responsivo

---

- Pasos para Formulario para Monitor:
  - Darle un ancho en porcentual al form
  - Centrar el Formulario
  - Darle un ancho a los label y a los input, y mostrarlos inline-block, de modo que se pida un dato por línea
  - Alinear el texto del label hacia la derecha para que quede más prolijo
  - Botones mostrados en línea
  - Padding necesarios para poder llenar el formulario usando un teclado físico

# Formulario Responsivo

---

- Desarrollo del ejemplo. El html es:

```
<article>
  <form action="mailto:micorreo@gmail.com" method="post">
    <section>
      <label for="ape">Apellido:</label>
      <input type="text" id="ape" name="apellido">
    </section>
    <section>
      <label for="cor">Correo: </label>
      <input type="email" id="cor" name="correo">
    </section>
```

# Formulario Responsivo

---

- Desarrollo del ejemplo. El html es:

```
<fieldset>
  <legend>Recibir novedades:</legend>
  <section class="radio">
    <label for="not1">Covid: </label>
    <input type="radio" id="not1" name="noticia">
  </section>
  <section class="radio">
    <label for="not2">Vacunación: </label>
    <input type="radio" id="not2" name="noticia">
  </section>
  <section class="radio">
    <label for="not3">Nuevas olas: </label>
    <input type="radio" id="not3" name="noticia">
  </section>
</fieldset>
```

# Formulario Responsivo

---

- Desarrollo del ejemplo. El html es:


```
<section class="radio">
  <label for="not3">Nuevas olas: </label>
  <input type="radio" id="not3" name="noticia">
</section>
</fieldset>
<section id="boton-form">
  <input type="submit" value="Guardar">
  <input type="reset" value="Limpiar">
</section>
</form>
</article>
```

# Formulario Responsive – CSS Tamaño Pequeño

---

```
/* Formulario */
form {
    background-color:  darkgrey;
    width: 96%;
    margin: 0 auto;
    padding: 0.5em;
}

form section, fieldset {
    display: flex;
    flex-direction: column;
    font-size: 1.4em;
    font-weight: bold;
}
```

```
form input {
    padding: 0.5em;
    border: 0.1em solid  black;
    font-size: 1.2em;
}

form label {
    width: 100%;
}

form #botones input {
    margin-bottom: 1%;
    font-weight: bold;
    font-size: 1em;
    padding: 0.2em;
}
```

# Formulario Responsive – CSS Tamaño Pequeño

---

```
form input[type=radio] {  
    height: 0.9em;  
    width: 100%;  
}  
  
.radio {  
    display: flex;  
    flex-direction: row;  
    font-size: 1em;  
}  
  
/* Fin Formulario */
```



# Formulario Responsive – CSS Tamaño grande

---

```
form {  
  width: 60%;  
  margin: 0 auto;  
  padding: 0.5em;  
}  
  
form section, fieldset {  
  display: flex;  
  flex-direction: row;  
  font-size: 1.2em;  
}
```

```
form input {  
  padding: 0.3em;  
  border: 0.1em solid black;  
  font-size: 1em;  
}  
  
fieldset {  
  display: flex;  
  flex-direction: row;  
  width: 100%;  
}  
  
form input[type=radio] {  
  height: 1em;  
  width: 30%;  
}
```

# Formulario Responsive – CSS Tamaño grande

---

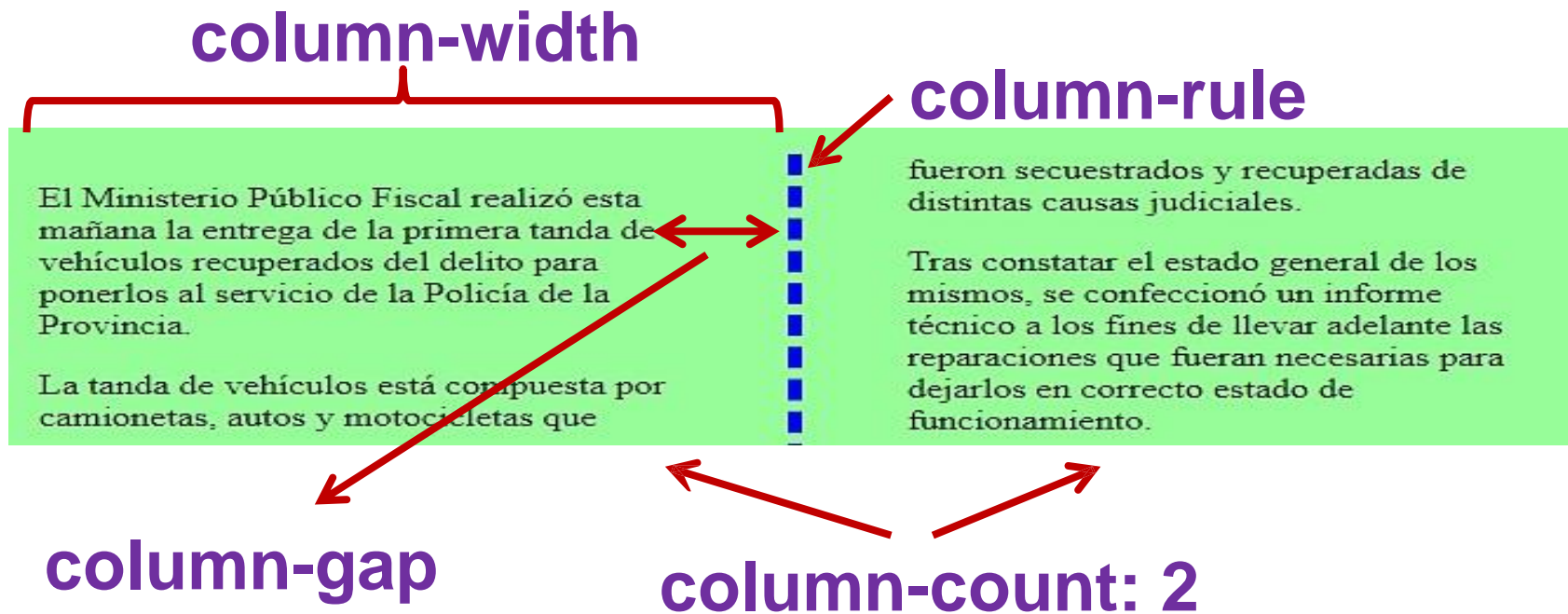
```
form label {  
    width: 30%;  
    margin-right: 2%;  
}  
  
#boton-form {  
    display: flex;  
    flex-direction: row;  
    justify-content: center;  
}
```

```
.radio {  
    display: flex;  
    flex-direction: row;  
    width: 100%;  
}  
  
.radio label {  
    width: 100%;  
}
```

# COLUMNAS

---

# CSS para Texto en Columnas



# Texto en Columnas

---

- **column-count**: Cantidad de columnas
- **column-width**: Ancho de cada columna
- **column-gap**: Separación entre columnas
- **column-rule**: Línea divisoria entre columnas

# Texto en Columnas - HTML

---

```
<article class="columnas"><p>El Ministerio Público Fiscal realizó  
esta mañana la entrega de la primera tanda de vehículos recuperados  
del delito para ponerlos al servicio de la Policía de la Provincia.  
</p>  
  <p>La tanda de vehículos está compuesta por camionetas, autos y  
motocicletas que fueron secuestrados y recuperadas de distintas  
causas judiciales.</p>  
  <p>Tras constatar el estado general de los mismos, se  
confeccionó un informe técnico a los fines de llevar adelante  
las reparaciones que fueran necesarias para dejarlos en  
correcto estado de funcionamiento.</p>  
</article>
```

# Texto en Columnas - CSS

---

```
.columnas{  
    column-count: 2;  
    column-gap: 15%;  
    column-width: 30%;  
    column-rule: 0.4em dashed blue;  
}
```

# Texto en Columnas - Ejemplo

---

El Ministerio Público Fiscal realizó esta mañana la entrega de la primera tanda de vehículos recuperados del delito para ponerlos al servicio de la Policía de la Provincia.

• La tanda de vehículos está  
• compuesta por camionetas, autos y  
• motocicletas que fueron  
• secuestrados y recuperadas de  
• distintas causas judiciales.  
•  
•

• Tras constatar el estado general de  
• los mismos, se confeccionó un  
• informe técnico a los fines de llevar  
• adelante las reparaciones que fueran  
• necesarias para dejarlos en correcto  
• estado de funcionamiento.  
•

**column-count: 3;**  
**column-rule: 0.4em dotted red;**  
**column-gap: 5%;**



# Texto en Columnas - Compatibilidad

- Para versiones viejas de navegadores usar:

-ms-column-count

-webkit-column-count

-moz-column-count

-o-column-count

column-count

# Columnas Responsive

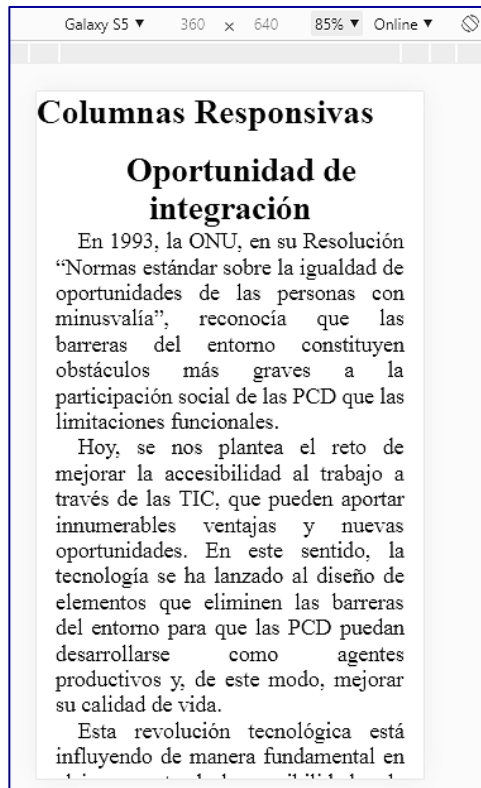
---

- En el caso de las columnas la adaptación consiste en decidir cuántas columnas mostraremos para distintos tamaños de pantalla.
- Posiblemente haya que modificar también cuánto gap utilizar y el formato de rule para que tenga un borde más o menos grueso.

# Columnas Responsive

---

- Para pantalla pequeña de 360px a 767px (una columna)



# Columnas Responsive

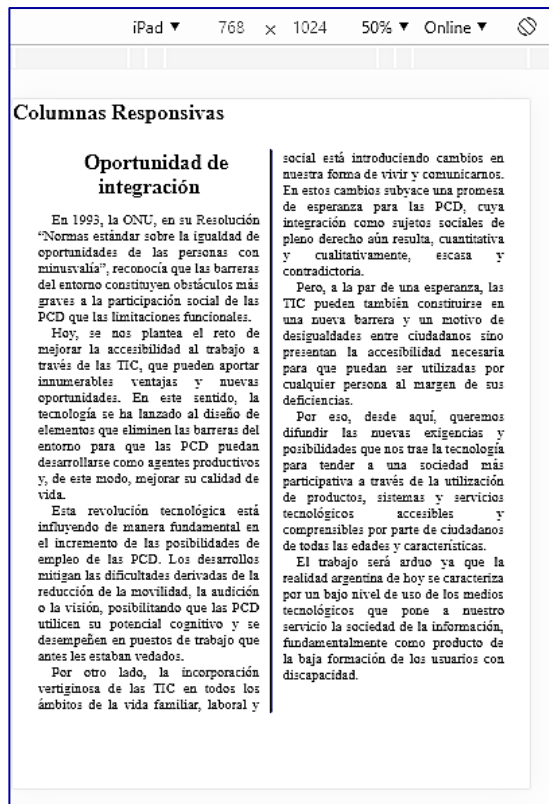
---

- Para pantalla pequeña de 360px a 767px (una columna)

```
*** CSS Móviles ***  
*/  
@media all and (min-width:320px) {  
    section {  
        padding: 1%;  
    }  
  
    .contenedor {  
        column-count: 1;  
        text-align: justify;  
        text-indent: 1em;  
        font-size: 1.3em;  
        padding: 4%;  
    }  
  
    .contenedor h2 {  
        text-align: center;  
    }  
}  
/*** FIN *****/
```

# Columnas Responsive

- Para pantalla mediana de 367px a 1023px (2 columnas)



# Columnas Responsive

---

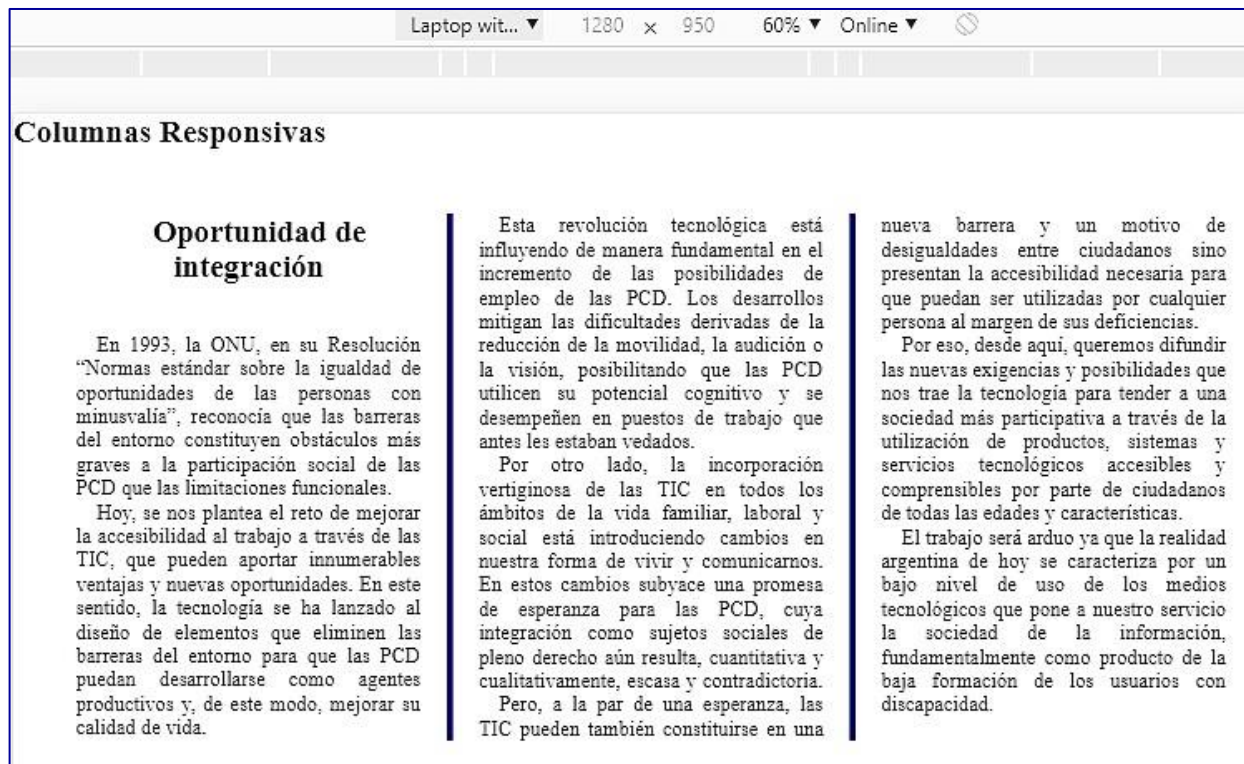
- Para pantalla mediana de 367px a 1023px (2 columnas)

```
*** CSS Tablets ***
*/
@media all and (min-width:768px) {
  section {
    padding: 1%;
  }
  .contenedor {
    column-count: 2;
    column-gap: 5%;
    column-rule: 2px navy solid 0.5vw;
    text-align: justify;
    text-indent: 1em;
    font-size: 1.3em;
  }

  .contenedor h2 {
    margin-bottom: 2vh;
  }
}
```

# Columnas Responsive

- Para pantalla mayor a 1024px (3 columnas)



# Columnas Responsive

---

- Para pantalla mayor a 1024px (3 columnas)

```
*** CSS Monitores ***
*/
@media all and (min-width:1024px) {
  section {
    padding: 1%;
  }
  .contenedor {
    column-count: 3;
    column-gap: 5%;
    column-rule: 1px solid navy;
    text-align: justify;
    text-indent: 1em;
    font-size: 1.3em;
  }
  .contenedor h2 {
    margin-bottom: 5vh;
  }
}
```