README.md

Curso de Responsive Design

Tabla de Contenido

¿Qué es Responsive Design?

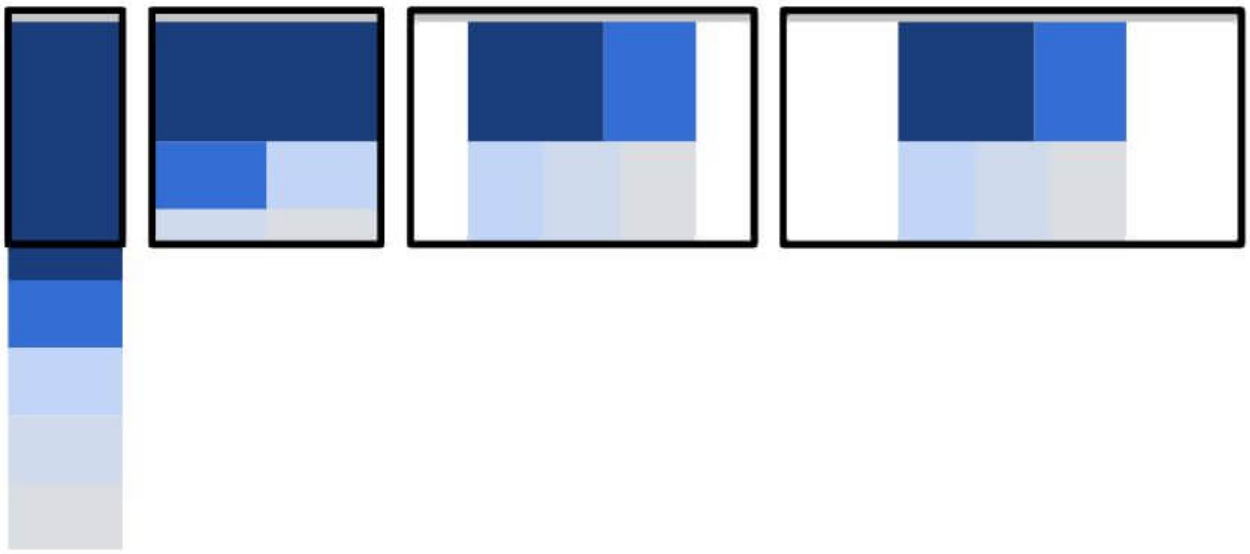
Responsive design son todas esas técnicas que usamos para adaptar nuestras aplicaciones web a la mayor cantidad de pantallas.

↑ volver al inicio

Patrones de Responsive Design

Mostly Fluid

Empieza el contenido en una caja, pero cuando ya no alcanza todo el contenido en el viewport, el sitio se empieza a redimencionar para aprovechar mejor el contenido.

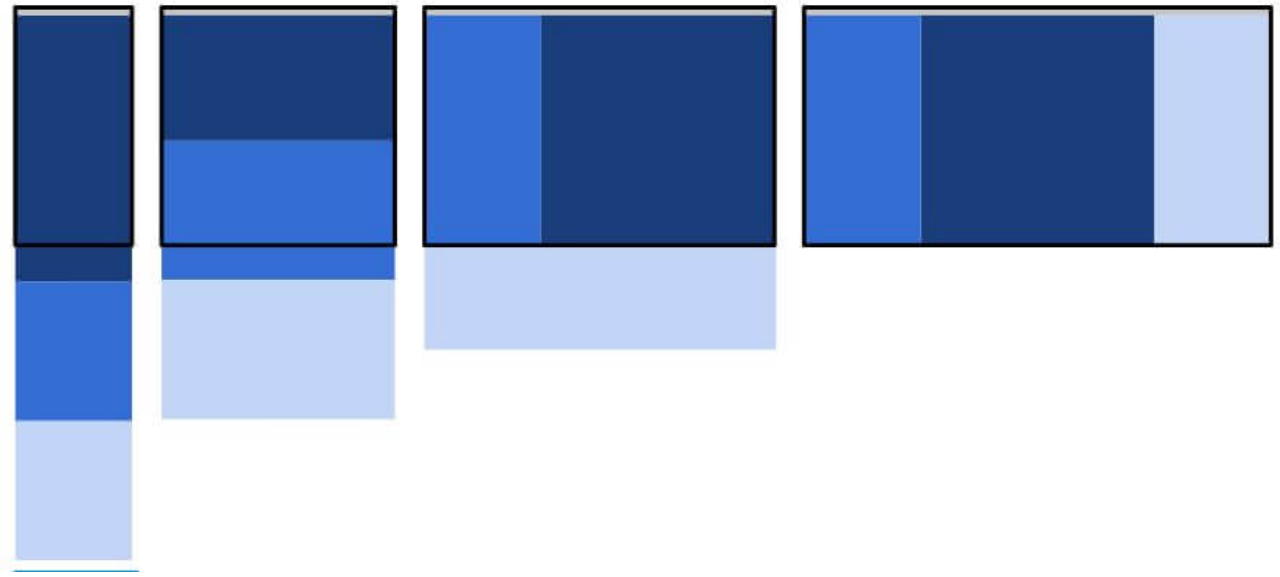


Mostly Fluid

↑ volver al inicio

Colocación de Columnas

Las columnas se van poniendo una debajo de otras dependiendo del tamaño del viewport.



Colocación de Columnas

↑ volver al inicio

Layout Shifter

Haces variaciones del layout dependiendo del tamaño del viewport.

Layout Shifter

↑ volver al inicio

Tiny Tweaks

Hacer cambios pequeños en algunos elementos como:

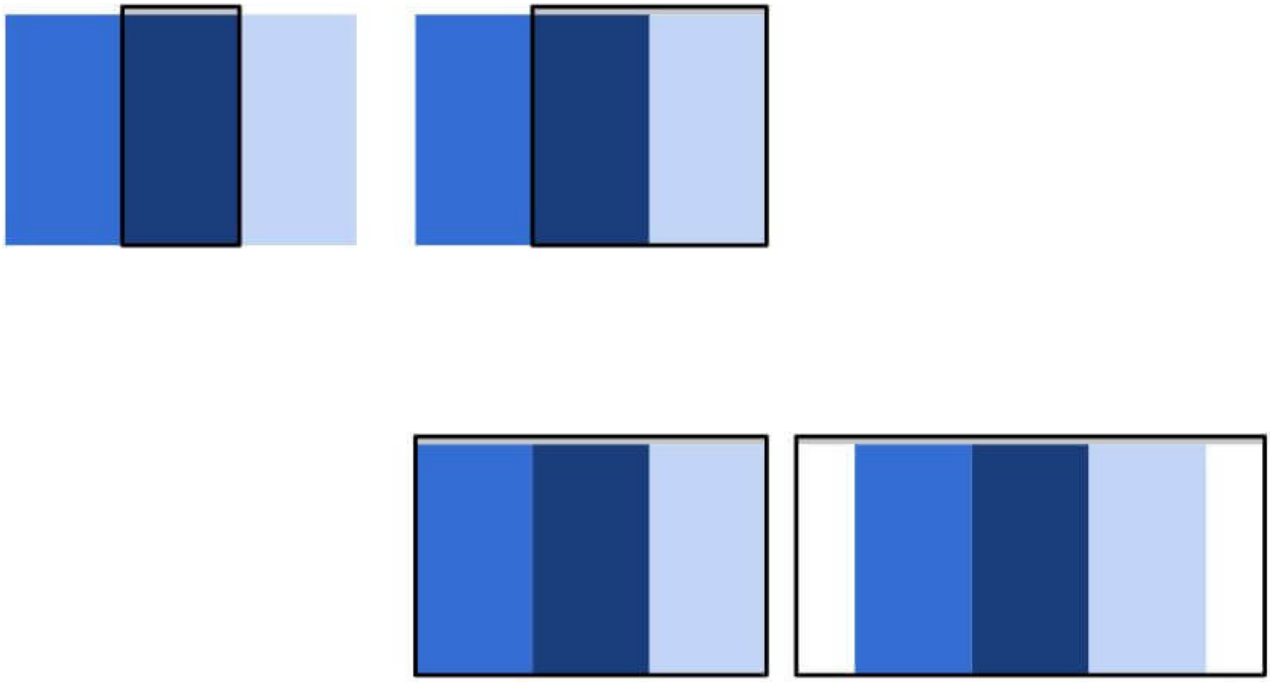
- Tamaño del texto.
- Tamaño de imágenes.

Tiny Tweaks

↑ volver al inicio

Off Canvas

Tienes ciertos elementos fuera del viewport que aparecen de acuerdo a ciertas acciones. Un ejemplo es el hamburger menu.



Off Canvas

↑ volver al inicio

Conceptos Elementales

- **Viewport**: área visible del navegador
- **Portrait**: vertical
- **Landscape**: horizontal
- **Mobile first**: empezar una website desde la menor resolución soportada
- **Desktop first**: empezar una website desde la mayor resolución soportada

¿Cúal es mejor? Técnicamente Mobile First

↑ volver al inicio

Developer tools para Responsive Design

Para activar las herramientas de responsive de Chrome, vamos a hacer lo siguiente:

1. Click derecho
2. Inspeccionar
3. Click en ínico de responsive en la esquina supeior izquierda o Control + Shift + M

↑ volver al inicio

Meta viewport

El viewport es el área visible del navegador.

Para setear el viewport, se va a hacer desde una etiqueta meta.

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

- `width=device-width` para que se adapte según la pantalla del dispositivo
- `initial-scale=1.0` para indicar el escalado según el dispositivo

↑ volver al inicio

Medidas Relativas

- **Porcentaje**: Longitud referente al tamaño de los elementos padre.
- **em**: Unidad relativa al tamaño de fuente especificada más cercano. Incluye al propio elemento.

```
<nav> font-size: 16px;
  <ul> font-size: 2em; 32px;
    <li> font-size: 1em; 32px;
      <a href="">hola!</a> font-size: .5em; 16px; padding: 2em; 32px;
    </li>
  </ul>
</nav>
```

Ejemplo: Medida em

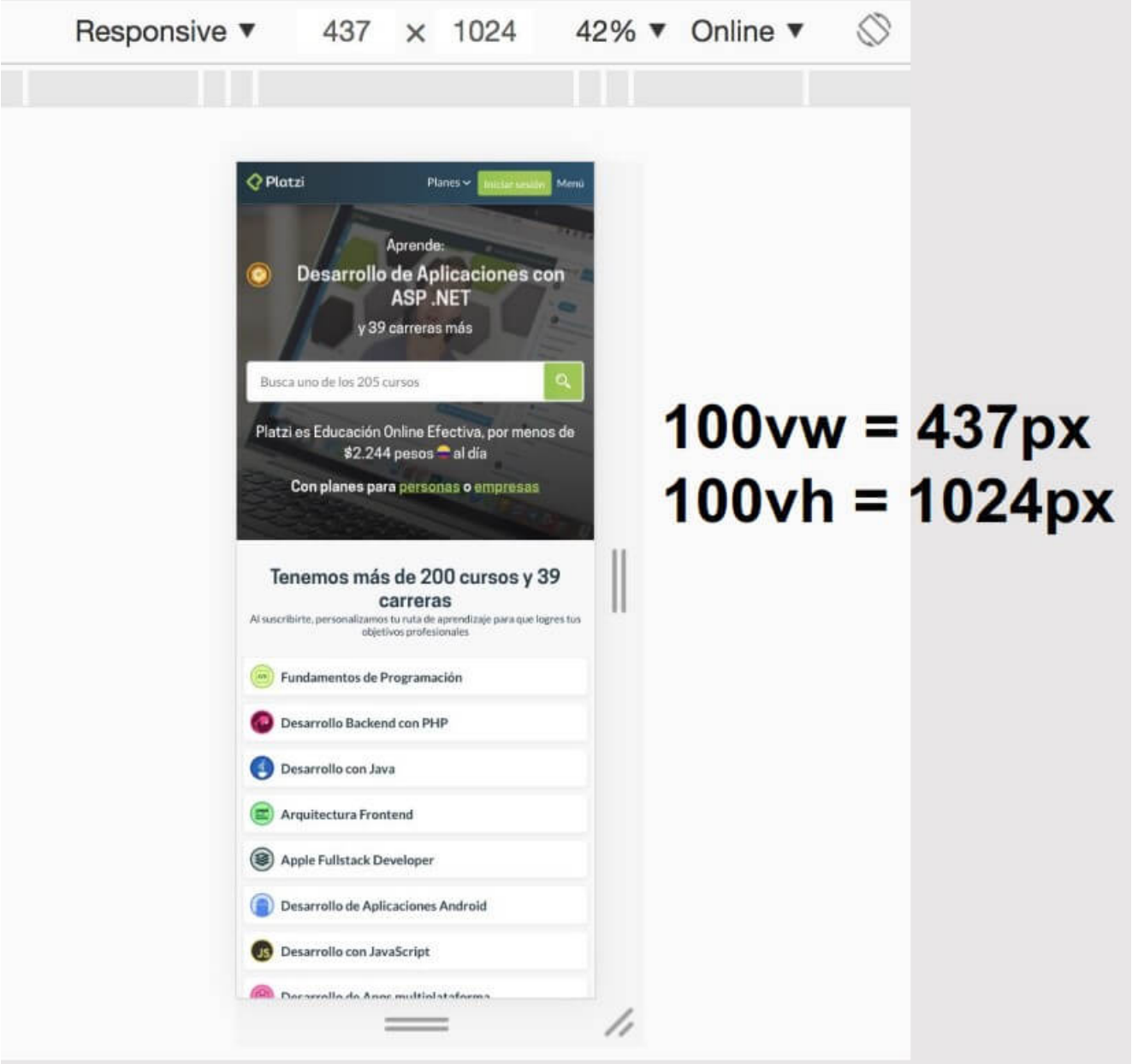
- **rem**: Unidad relativa al tamaño de fuente especificada en el ancestro más lejano (html o body).

```
<!DOCTYPE html> font-size: 32px;
<html lang="en">
<head>
</head>
<body> font-size: 16px;
    <ul> font-size: 2rem;
        <li>font-size: 1rem;
            <a href=""></a> font-size: .5rem; 16px;
        </li>
    </ul>
</body>
</html>
```

Ejemplo: Medida rem

- **vw / vh**: Unidad relativa porcentual con respecto al viewport.

Ejemplo: Medida vw/vh



Ejemplo: Medida vw/vh

↑ volver al inicio

Media Queries

El media queries es un módulo de css que hace posible al responsive design, éste existe desde el 2010 y se encarga de adaptar la representación del contenido a características del dispositivo.

Estructura del Media Querie:

```
@media media type and (condición) { }
```

Ejemplos:

```
/* Todas las pantallas con con un ancho inferior o igual a 768px cumplen esta condició
@media screen and (max-width: 768px) { }

/* Todas las pantallas con un ancho de 480px hasta 768px cumplen esta condición */
@media screen and (max-width: 768px)  and (min-width: 480px) { }
```

Mobile first

Usa min-width
min-width = desde

```
@media screen and (min-width: 320px)  { }
@media screen and (min-width: 480px)  { }
@media screen and (min-width: 768px)  { }
@media screen and (min-width: 1024px)  { }
```

Desktop first

Usa max-width
max-width = hasta

```
@media screen and (max-width: 1024px)  { }
@media screen and (min-width: 768px)  { }
@media screen and (min-width: 480px)  { }
@media screen and (min-width: 320px)  { }
```

↑ volver al inicio

CSS Positions

- **static**: es la propiedad por defecto.

Con las otras opciones, se activan las propiedades de top, bottom, left, right y z-index.

- **relative**: el objeto se mueve en base al lugar donde se encuentra originalmente.
- **absolute**: el objeto se ubica de manera absoluta con el elemento más cercano que tenga posición relativa o con el body.
- **fixed**: El elemento se muestra de manera fija en el viewport.
- **sticky**: El elemento se queda de manera fija una vez que aparece en pantalla.

↑ volver al inicio

Video Responsive

Video HTML

Para que un video se ajuste al tamaño de pantalla, se puede hacer lo siguiente:

```
.html-video  {
  width: 100%;
  height: auto;
}
```

↑ volver al inicio

Video Insertado

```
<div class="flexible-video">
  <iframe class="youtube-video" width="560" height="315" src="https://www.youtube.com/
</div>
```

```
.youtube-video {
  position: absolute;
  top: 0;
  bottom: 0;
  right: 0;
  left: 0;
  width: 100%;
  height: 100%;
}

.flexible-video {
  width: 100%;
  height: 0;
  /* height * 100 / width */
  padding-top: 56.25%;
  position: relative;
}
```

↑ volver al inicio

Burger Menu

```
<!-- HTML -->
<i class="icon-menu burger-button"></i>
```

```
/* CSS */
.burger-button {
  width: 40px;
  height: 40px;
  border-radius: 50%;
  background-color: rgba(0,0,0,.8);
  display: none;
  line-height: 40px;
  text-align: center;
  position: fixed;
  z-index: 4;
  left: 5px;
  top: 5px;
  color: #fff;
}

@media screen and (max-width: 767px) {
  .burger-button {
    display: block;
  }
  .menu {
    position: fixed;
    background: rgba(5, 111, 255, .9);
    z-index: 3;
    top: 0;
    left: -100vw;
    width: 100vw;
    bottom: 0;
    display: flex;
    align-items: center;
    justify-content: center;
    transition: .3s;
  }
  .menu.is-active {
    left: 0;
  }
}
```

```
// Javascript
const menu = document.querySelector('.menu');
const burgerButton = document.querySelector('#burger-button');
burgerButton.addEventListener('click', () => {
```

```
  menu.classList.toggle('is-active');
})
```

↑ volver al inicio

Media Queries con JavaScript

Hay ocaciones en las que solo se van a usar eventos dependiendo del tamaño de pantalla. Por ejemplo, en los burger menu. Para estos casos, se puede usar media queries en javascript

```
//Media query
const ipad = window.matchMedia('screen and (max-width: 767px)');

//Activa la primera vez que se entra
if(ipad.matches)
  burgerButton.addEventListener('click', hideShow);

//Activa o desactiva al hacer resize de la pantalla
ipad.addListener((event) => {
  if(event.matches)
    burgerButton.addEventListener('click', hideShow);
  else
    burgerButton.removeEventListener('click', hideShow);
});
```

↑ volver al inicio

Propiedades CSS Útiles

Flex-Wrap

Pone un elemento debajo de otro si no entran en el viewport.

```
flex-wrap: wrap;
```

↑ volver al inicio

Remote Debugging

Servidor Estático en Node

Para poder hacer debugging se necesita un servidor statático.

1. Instalar static-server

```
npm i -g static-server
```

2. ir a la carpeta del proyecto en la terminal.
3. Ejecutar el comando `static-server` . Se va a generar una dirección con la aplicación.
4. Verificar la ip de la computadora

```
$ ipconfig
```

5. Ingresar desde el dispositivo móvil ingresando **ip:puerto**

↑ volver al inicio

Remote Debugging en iOS

Para esto es necesario tener una Mac.

1. Conectar el celular con la computadora via USB.
2. Abrir Safari.
3. Ir a Preferencias de Safari.
4. Habilitar "Mostrar el menú de Desarrollo en la barra de menús". Aparecerá un nuevo menú de desarrollo.
5. Abrir el menú de desarrollo. Aparecerá el dispositivo en la lista.
6. Abrir la ip del proyecto.

↑ volver al inicio

Remote Debugging en Android

1. Conectar el celular con la computadora via USB.
2. Activar el menú de desarrolldor en Android.
3. Ir a acerca del contenido
4. Hacer tab en **número de compilación** varias veces hasta que se active el menú de programnador.
5. Ir al menú de programación.
6. Activar la opción de **Depuración por USB**.
7. Abrir Chrome en la computadora.

8. Ir a **chrome://inspect**. Aparecerá una lista de los teléfonos con las ventanas que está navegando.
9. Hacer click en inspect.