RWD (Responsive Web Design)

Diseño de Interfaces Web

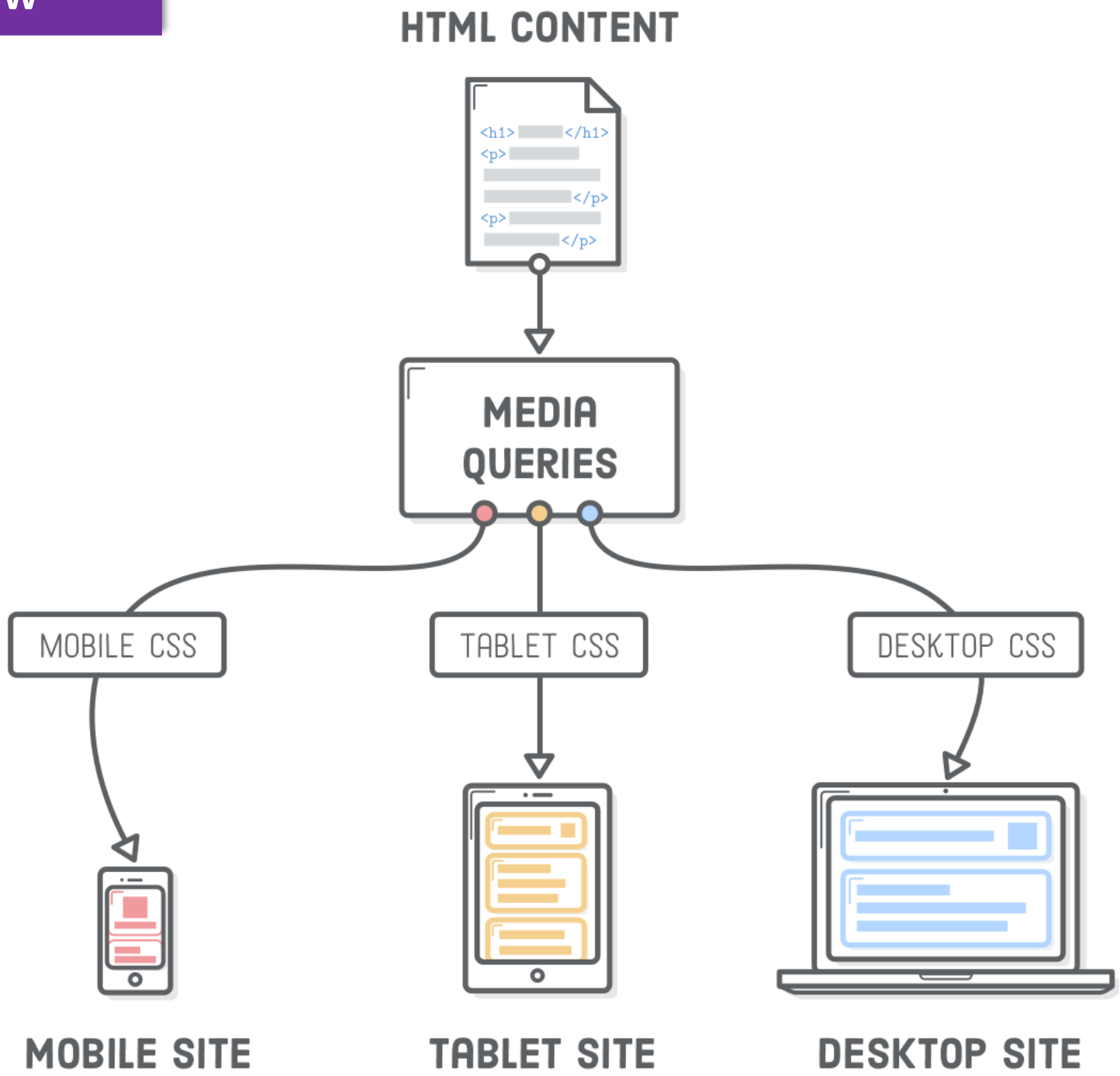
José María Torresano
Diciembre 2018
daw.cierva@gmail.com

1. Media Queries
2. viewport
3. Diseño adaptativo
4. Media Queries en RWD
5. Dispositivo vs contenido

Definición de

Diseño Web Responsive

Es una filosofía de diseño y desarrollo cuyo objetivo es adaptar la apariencia de las páginas web al dispositivo que se esté utilizando para visitarlas mediante “media queries” e incluyendo tipografía y medios adaptados (fotos, videos, mapas, tablas, sliders) y navegación acorde al dispositivo.


HTML CONTENT
<h1> </h1>

MEDIA QUERIES
MOBILE CSS
TABLET CSS
DESKTOP CSS
MOBILE SITE
TABLET SITE
DESKTOP SITE

ejemplos

http://colly.com/

https://www.fork-cms.com/

http://getbootstrap.com/

https://skinnyties.com/

http://foodsense.is/

http://www.morehazards.com/

MEDIA-QUERIES

Desde la especificación de CSS 2.1, ha sido posible modificar el aspecto de los documentos HTML en función del tipo de dispositivo en el que se mostraban. El caso más común es el de crear una hoja de estilos que se aplica al imprimir los documentos:

```
<link rel="stylesheet" type="text/css" href="web.css" media="screen" />
<link rel="stylesheet" type="text/css" href="imprimir.css" media="print" />
```

La especificación CSS ofrece una buena cantidad de tipos de medios (*media type*) para los que podemos aplicar un diseño específico:

```
all, braille, embossed, handheld, print,
projection, screen, speech, tty, y tv
```

W3C introdujo los *Media Query* como parte de la especificación de CSS 3, mejorando de manera notable el objetivo de los *Media Type*. Un Media Query no sólo nos permite seleccionar el tipo de medio, sino consultar otras características sobre el dispositivo que esta mostrando la página.

```
<link rel="stylesheet" type="text/css"
    media="screen and (max-device-width: 480px)" href="mihoja.css" />
```

Esta consulta contiene dos componentes: un tipo de medio (screen) y una consulta concreta sobre la característica del medio (max-device-width) y el valor objetivo (480px).

En otras palabras, estamos preguntando al dispositivo si su resolución horizontal (max-device-width) es igual o menor a 480px. Si se cumple la condición, el dispositivo cargará la hoja de estilos *mihoja.css*. De otra forma, el link será ignorado.

```
@media screen and (min-width: 1024px) {
  body  {
    font-size: 100%;
  }
}
```

Pueden estar definidos en la misma hoja de estilo...

... o utilizando una sentencia @import

```
@import url("mihoja.css") screen and (min-width: 1024px);
```

descriptores de medios

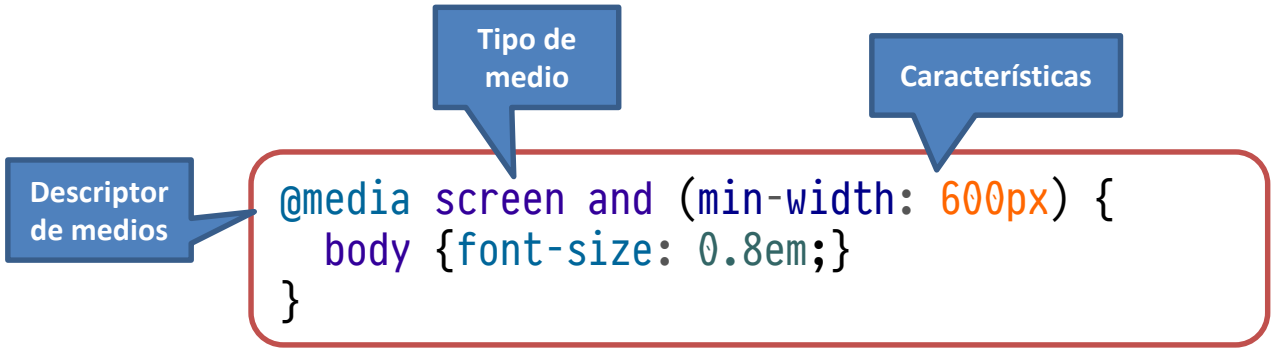
```
<link href="print-color.css" type="text/css"
  media="print and (color)" rel="stylesheet">

@import url(print-color.css) print and (color);


<link href="print-color.css" type="text/css"
media="print and (color),
      screen and (color-depth: 8)" rel="stylesheet">

@import url(print-color.css) print and (color),
                             screen and (color-depth: 8);
```

En cualquier situación donde una de las consultas de medios se evalúe como `true`, se aplica la hoja de estilos asociada.

Por lo tanto, dado el `@import` anterior, se aplicará *print-color.css* si se procesa en una impresora en color o en un entorno de pantalla con suficiente número de colores. Si se imprime en una impresora en blanco y negro, ambas consultas se evaluarán como `false` y *print-color.css* no se aplicará al documento. Lo mismo es `true` en cualquier medio de pantalla, y así sucesivamente.

Cada descriptor de medios se compone de un tipo de medio y una lista de una o más características, con cada descriptor de características de medios entre paréntesis. Si no se proporciona ningún tipo de medio, entonces se supone que es `all`, lo que hace que los dos ejemplos siguientes sean equivalentes:

```
@media all and (min-resolution: 96dpi) {...}
@media (min-resolution: 96dpi) {...}
```



Se pueden vincular múltiples descriptores de características con la palabras `and` y `not`

Sintaxis

Los *Media Queries* pueden contener una o más expresiones, funciones multimedia, que se resuelven a `true` o `false`. El resultado del *query* devuelve `true` si el *media type* especificado en el Media Query coincide con el tipo de dispositivo en que el documento está siendo mostrado y todas las expresiones en el Media Query devuelven también `true`.

Cuando un *Media Query* devuelve `true`, se aplica la hoja de estilo/reglas correspondientes, siguiendo las reglas habituales de CSS.

> Las hojas de estilo con *Media Queries* con <link> se descargarán, incluso si sus Media Queries resultan `false`; sin embargo, no se aplicarán.

Se pueden crear *Media Queries* complejos utilizando operadores lógicos: **`not`**, **`and`** y **`only`**. El operador **`and`** se utiliza para combinar múltiples *media features* en un sólo Media Query; cada función debe se **`true`** para que el Query también lo sea. El operador **`not`** se utiliza para negar un Media Query completo y el operador **`only`** se usa para aplicar un estilo sólo si el Query completo es correcto.

Además, se pueden combinar múltiples *Media Queries* separados por **`comas`** en una lista; si alguno de los Queries devuelve **`true`**, todo la instrucción devolverá **`true`**. Esto es equivalente a un operador **`or`**.

and

Vincula dos o más funciones de medios de tal manera que todas deben ser verdaderas para que la consulta sea verdadera.

Por ejemplo, `(color) and (orientation: landscape) and (min-device-width: 800px)` significa que se deben cumplir las tres condiciones: si el medio tiene color, está en orientación horizontal y la pantalla del dispositivo es de al menos 800 píxeles de ancho, se usa la hoja de estilo.

not

Niega toda la consulta de manera que si todas las condiciones son verdaderas, entonces la hoja de estilo no se aplica.

Por ejemplo, `not (color) and (orientation: landscape) and (min-device-width: 800px)` significa que si se cumplen las tres condiciones, se niegan. Así que si el medio tiene color, está en orientación horizontal y la pantalla del dispositivo es de al menos 800 píxeles de ancho, NO se usa la hoja de estilo. En todos los demás casos, sí.

Ten en cuenta que `not` solo se puede usar solo al comienzo de una consulta de medios. `(color) and not (orientation: landscape)` es ilegal y se ignorará

,

Hace la función lógica de `or` (que no existe como palabra clave).

Por ejemplo, `(color) and (orientation: landscape), (min-device-width: 800px)` significa que se deben cumplir las 2 primeras condiciones o la última para usarse la hoja de estilo.

only

Se usa para ocultar una hoja de estilo de navegadores demasiado antiguos para comprender las consultas de los medios.

Por ejemplo, `@import url(css.css) only all` significa que en navegadores que comprendan las *media queries* ignorarán la palabra `only` y se aplicará la hoja de estilos. En navegadores que NO comprendan las *media queries* 'ven' `only all` que no es válido así que no aplicaran la hoja de estilos.

Ten en cuenta que `only` solo se puede usar solo al comienzo de una consulta de medios.

Features (2018)

- width
- min-width
- max-width
- device-width
- min-device-width
- max-device-width
- height
- min-height
- max-height
- device-height
- min-device-height
- max-device-height
- aspect-ratio
- min-aspect-ratio
- max-aspect-ratio
- device-aspectratio
- min-device-aspectratio
- max-device-aspectratio
- color
- min-color
- max-color
- color-index
- min-color-index
- max-color-index
- monochrome
- min-monochrome
- max-monochrome
- resolution
- min-resolution
- max-resolution
- orientation
- scan
- grid

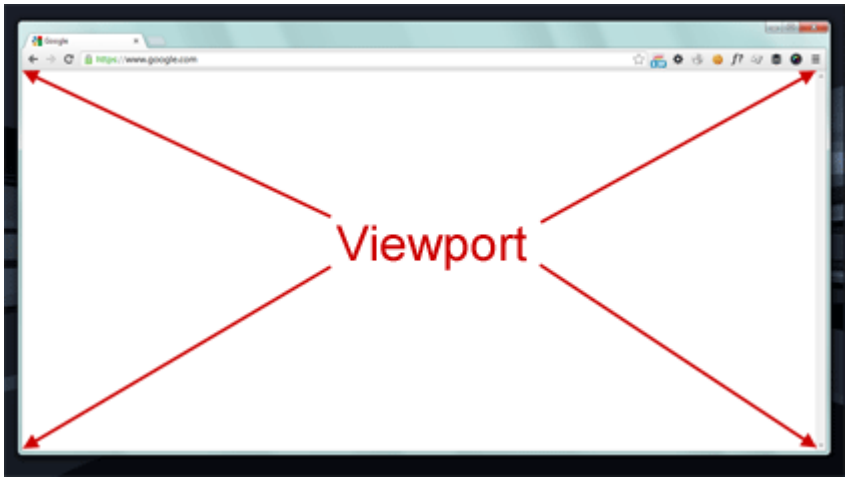
<ratio>
<resolution>

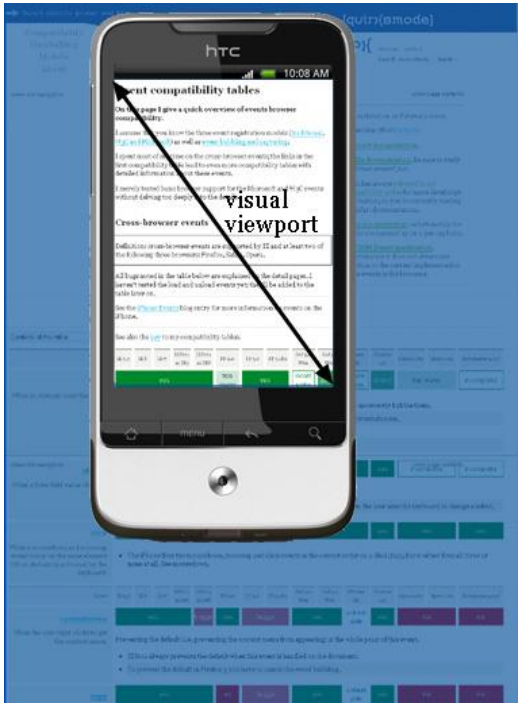
VIEWPORT

viewport (ventana gráfica)

¿Qué es un viewport?

El **viewport** es la parte visible en la pantalla del usuario que se puede usar para mostrar contenido en línea en el navegador. También existe la etiqueta HTML "viewport" que permite a los desarrolladores web indicarle al navegador cómo tratar las dimensiones y la escala de la página dentro de la ventana gráfica.

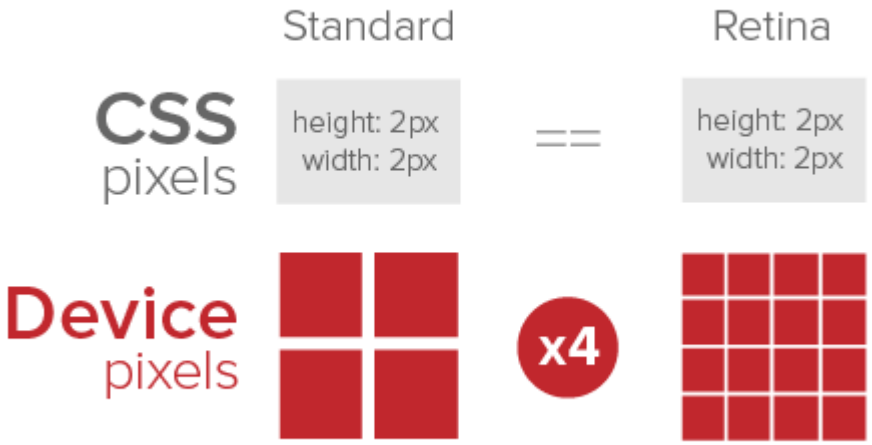




{diw} daw.cierva@gmail.com

viewport (ventana gráfica)

Es importante saber que el tamaño del viewport se proporciona en píxeles virtuales o CSS que difieren de los que los fabricantes de dispositivos enumeran en las especificaciones de los teléfonos inteligentes. Para teléfonos de alta resolución, como Samsung Galaxy S8 y iPhone X, lo que el navegador cuenta como 1 píxel contiene más de 1 píxel físico de la pantalla. Un ejemplo es la pantalla Retina de iPhone que tiene 4 píxeles físicos dentro de cada píxel virtual. Solo para algunos teléfonos más antiguos con pantallas de baja resolución, 1 píxel virtual puede ser igual a 1 píxel físico.



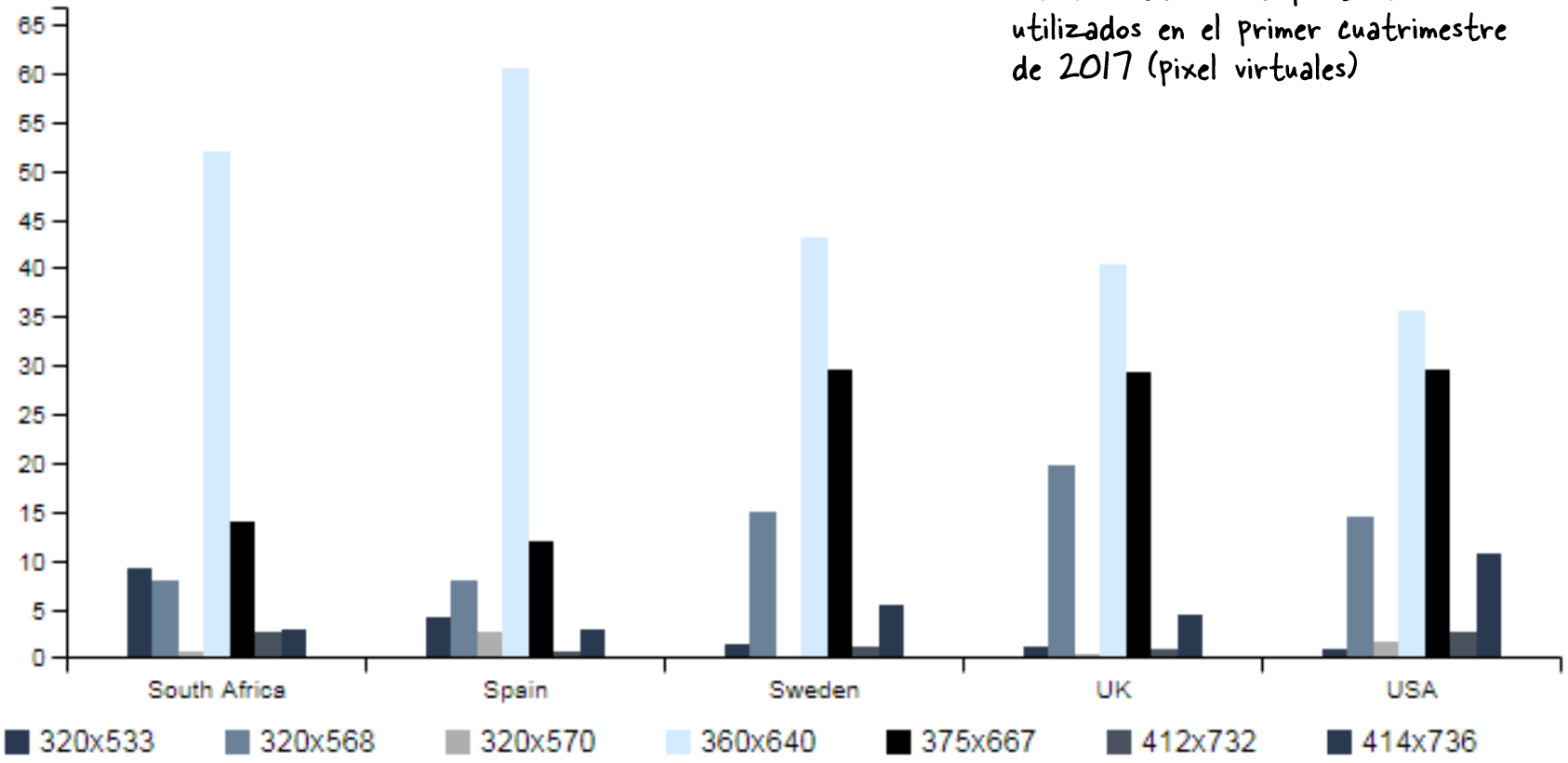
dpi — dots per inch
dpcm — dots per centimeter
dppx — dots per px unit

1dppx → 96px CSS

{diw} daw.cierva@gmail.com

viewport (ventana gráfica)

Tamaños de los viewport más utilizados en el primer cuatrimestre de 2017 (pixel virtuales)



{diw} daw.cierva@gmail.com

meta viewport

En los dispositivos móviles los navegadores muestran la página en el *viewport* y luego lo reducen para que se adapte al tamaño de la pantalla del dispositivo. La etiqueta *viewport* nos permite definir el ancho, alto y escala del área usada por el navegador para mostrar contenido

```
<meta name="viewport" content="width=device-width, initial-scale=1, maximun-scale=1">
```

width : Define el ancho del área visible. El valor será el número de píxeles, o la constante que tomará el ancho del dispositivo: ***device-width***.

height : Define el altura del área visible. El valor será el número de píxeles, o la constante que tomará el ancho del dispositivo ***device-height***.

initial-scale : Define la escala inicial del área visible. El valor será un número real que irá de 0.1 a 1.

minimum-scale : Define la escala mínima del área visible. El valor será un número real que irá de 0.1 a 1.

maximum-scale : Define la escala máxima del área visible. El valor será un número real que irá de 0.1 a 1. El valor 1 impide al usuario el hacer zoom.

user-scalable : Define los permisos de si se puede o no escalar el área visible. El valor será: *yes*/*no*.

meta viewport

{diw} daw.cierva@gmail.com

viewport (medidas)

vh y vw

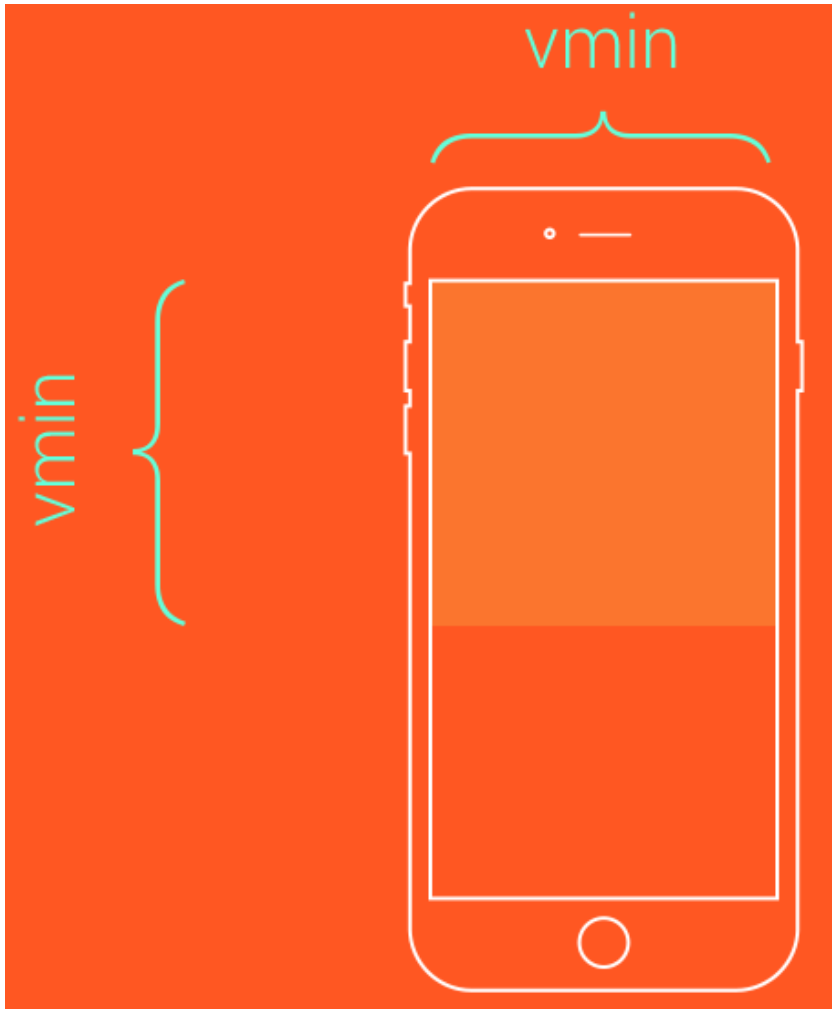
Las técnicas de diseño web responsivo dependen fuertemente en reglas de porcentaje. Sin embargo, el porcentaje CSS no siempre es la mejor solución para cada problema. El ancho CSS es relativo al elemento padre contenedor más cercano. ¿Y si quisieras usar el ancho o alto del viewport en lugar del ancho del elemento padre? Esto es exactamente lo que proporcionan las unidades `vh` y `vw`.

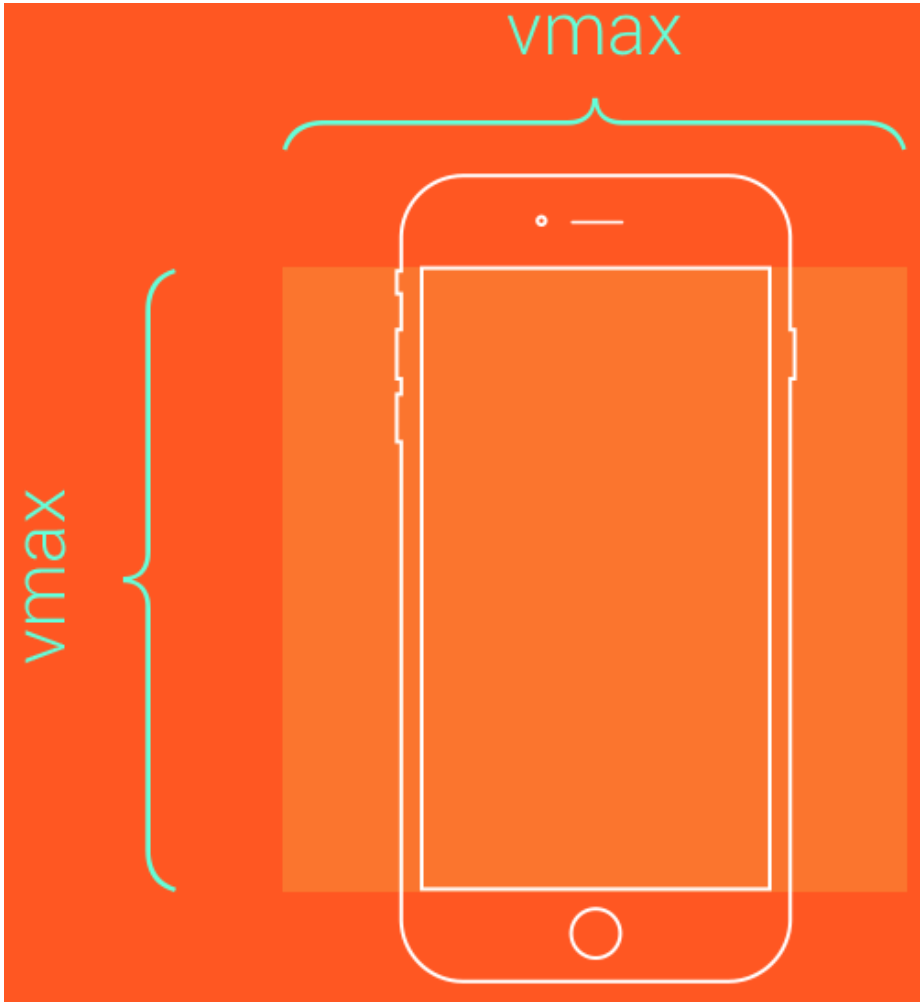
El elemento `vh` es igual a `1/100` de la altura del viewport. Por ejemplo, si la altura del navegador es `900px`, `1vh` evaluaría a `9px`.
De manera similar, `vw` es igual a `1/100` del ancho del viewport. Si el ancho del viewport es `750px`, `1vw` evaluaría a `7.5px`.

vmax y vmin

Mientras que `vh` y `vm` siempre están relacionadas a la altura y el ancho del viewport respectivamente, `vmin` y `vmax` están relacionadas al máximo y mínimo de aquellos anchos y altos, dependiendo de cuál es más pequeño y más grande. Por ejemplo, si el navegador está establecido a `1100px` de ancho y `700px` de alto, `1vmin` sería `7px` y `1vmax` sería `11px`. Sin embargo, si el ancho fue establecido a `800px` y el alto establecido a `1080px`, `vmin` sería igual a `8px` mientras que `vmax` sería establecido a `10.8px`.

viewport (medidas)



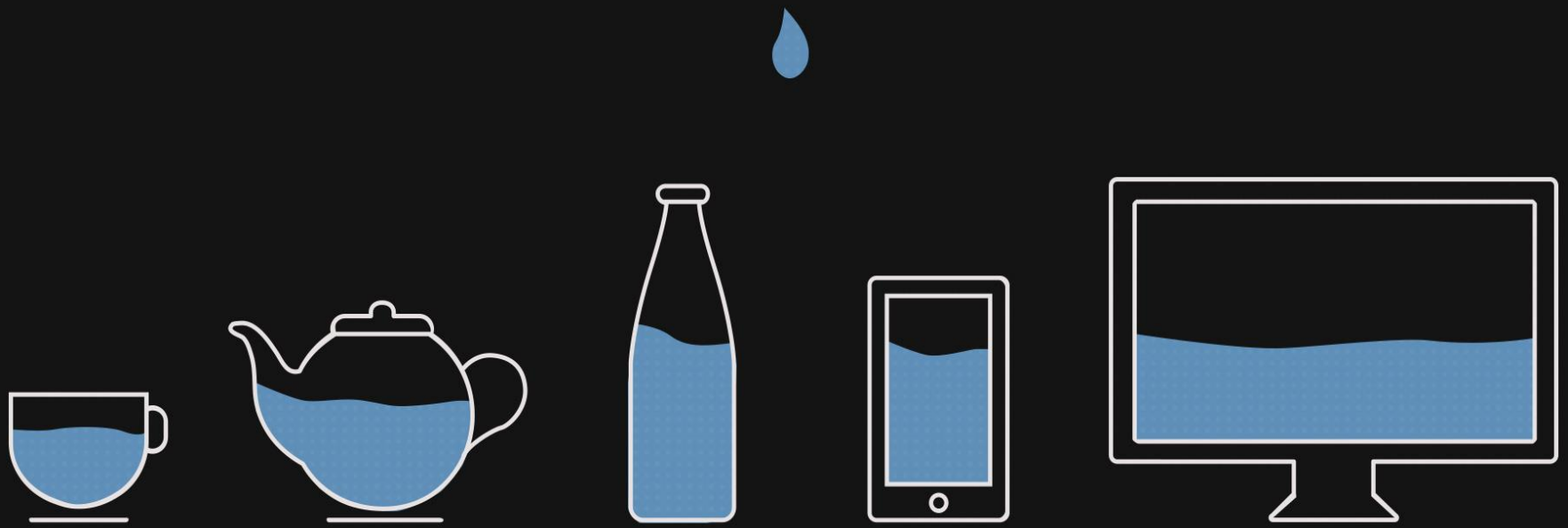


{diw} daw.cierva@gmail.com

DISEÑO ADAPTATIVO

{diw} daw.cierva@gmail.com

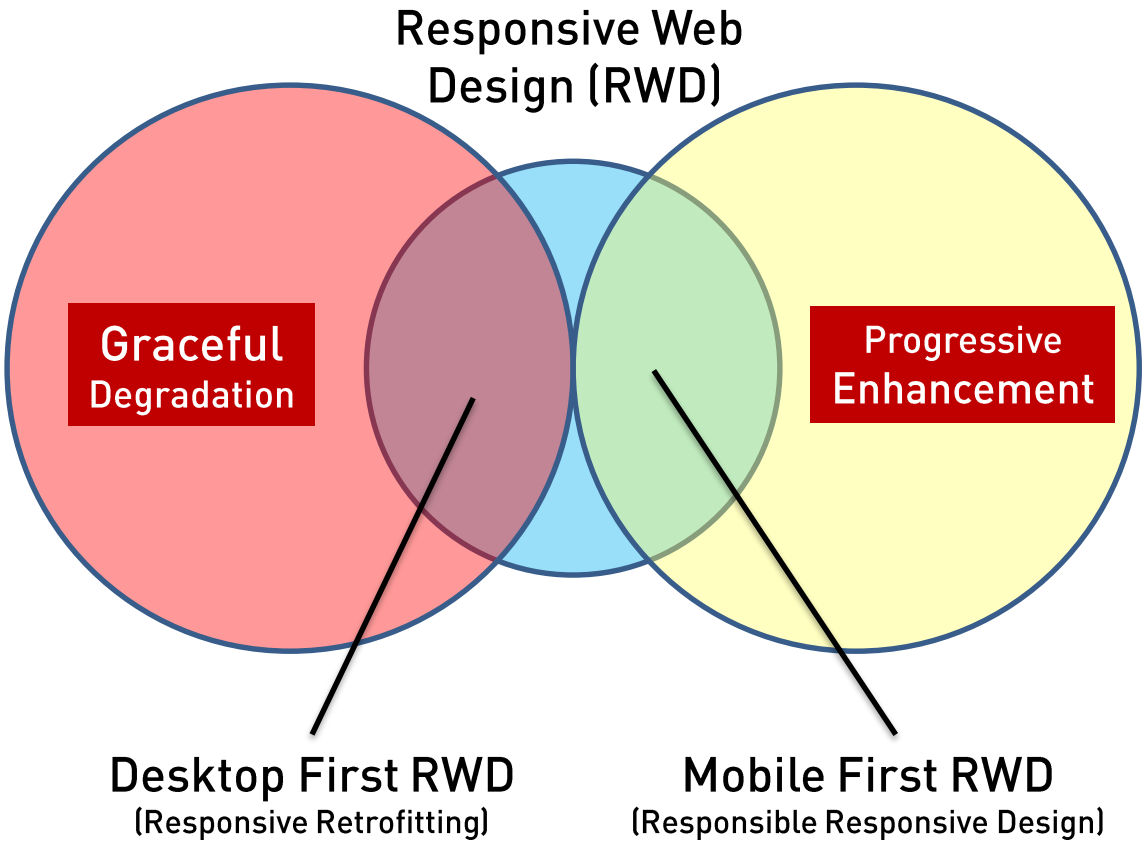
CONTENT IS LIKE WATER



"You put water into a cup it becomes the cup.
You put water into a bottle it becomes the bottle.
You put it in a teapot, it becomes the teapot."

Josh Clark *(originally Bruce Lee)* - Seven deadly mobile myths

Illustration by Stéphanie Walter


Responsive Web
Design (RWD)
Graceful
Degradation
Progressive
Enhancement
Desktop First RWD
(Responsive Retrofitting)
Mobile First RWD
(Responsible Responsive Design)

Graceful Degradation

Se desarrolla para los últimos navegadores con posibilidad de que funcionen en navegadores antiguos

desktop vs mobile

Primero el desktop y, al final, el móvil.

Progressive Enhacement

Se desarrolla para la versión básica, completamente operativa, con la posibilidad de ir añadiendo mejoras para los últimos navegadores.

desktop vs mobile

Primero el mobile y, al final, el desktop.

{diw} daw.cierva@gmail.com

Beneficios de ambos métodos

Reducción de costes	**Eficiencia en la actualización**	**Mejora la usabilidad**
Solo hay que tener una versión de la página	La actualización se refleja en todos los dispositivos	Experiencias parecidas con independencia del dispositivo

Mejora del SEO	**Impacto en el visitante**
Una web que se vea en un móvil mejora el ranking de la página	Asociación a la marca de creatividad e innovación

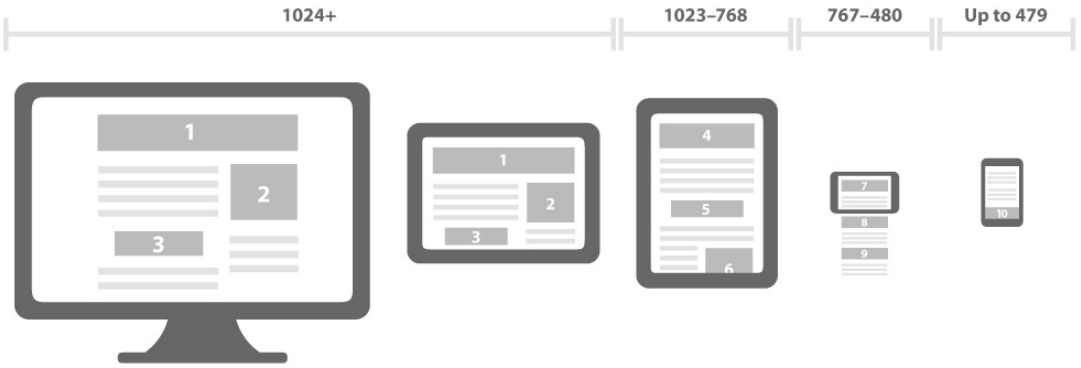
MEDIA QUERIES EN RWD

1. Configurar el viewport

```
<meta name="viewport" content="user-scalable=no, width=device-width, initial-scale=1">
```

2. Decidir los puntos de ruptura y definir sus rangos

Los puntos de ruptura no se hacen orientados a dispositivos sino a tu diseño web



3. Colocar los media query en orden en función del diseño elegido

a. Colocar todos los estilos que se aplicarán independientemente del tamaño y aquellos relativos al diseño básico (para MF, tamaño más pequeño; para DF, tamaño más grande). No hay MQ
b. Dentro de MQ correspondientes a los puntos de ruptura aplicar los estilos necesarios para las siguientes anchuras

Mobile first → medias ascendentes
Desktop first → medias descendentes

CENTRADO EN EL DISPOSITIVO VS CENTRADO EN EL CONTENIDO

Centrado en dispositivos (insostenible)

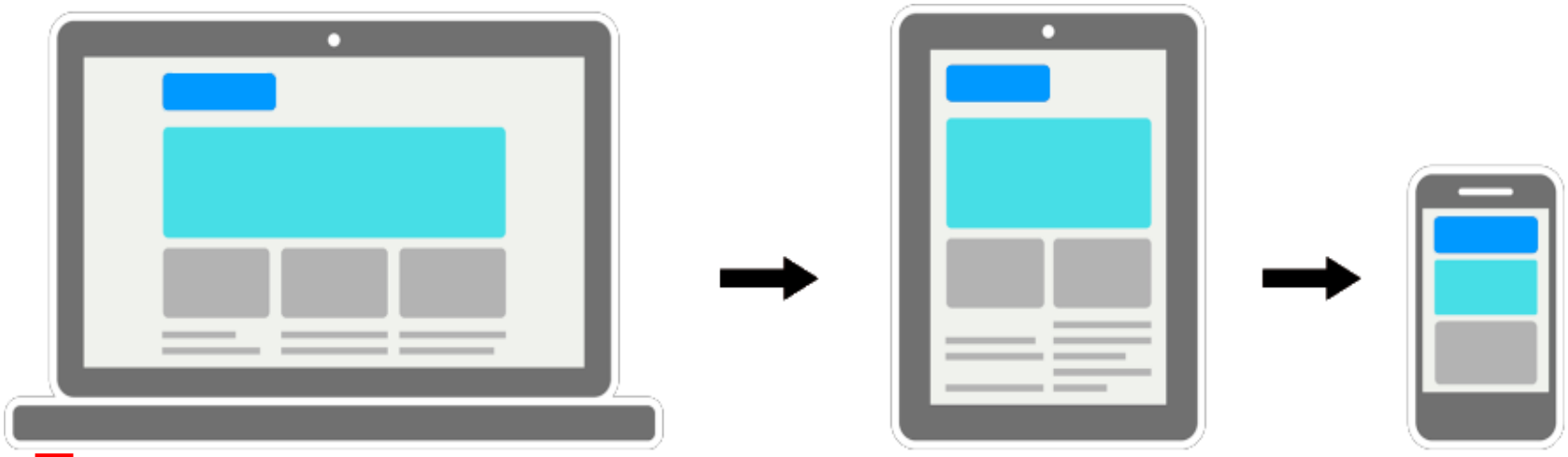
1. Se hace una lista de 5 o 6 dispositivos más vendidos.
2. Se anota el ancho en pixeles de cada dispositivo
3. Se definen los breakpoints con una media querie en pixeles para cada resolución.

NO CONSIDERA LOS CONTENIDOS

Centrado en contenidos (content-out)

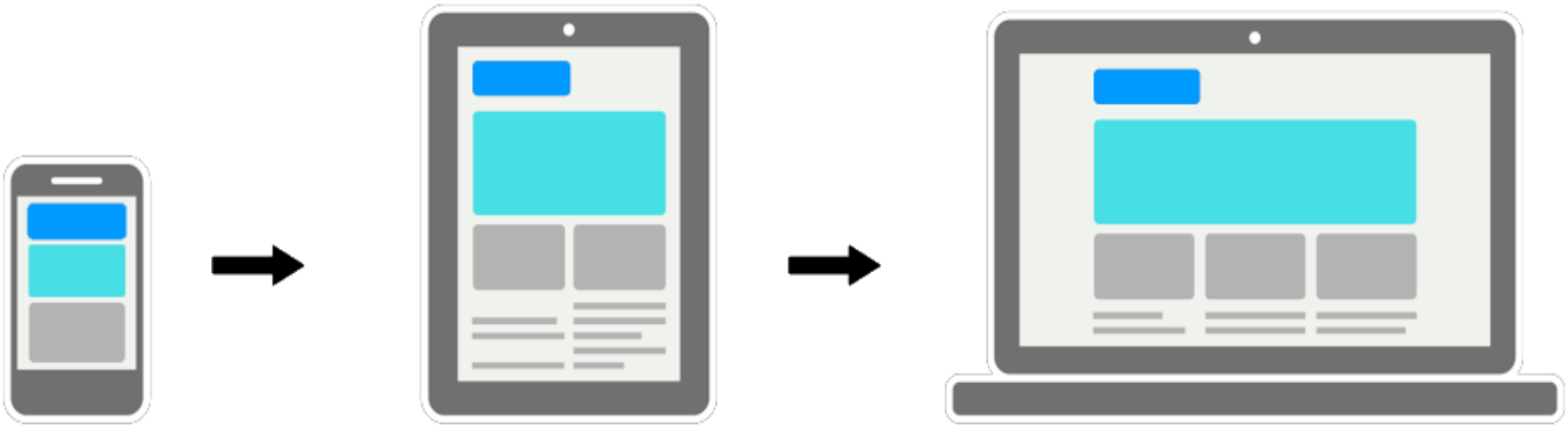
1. Se evalúan los anchos de renglón del contenido.
2. Se hace una lista de anchos de ventana del navegador a los que ESE contenido ya NO ES LEGIBLE.
3. Se definen los breakpoints en función de esos anchos (convertidos a em) .

DEPENDE DE LOS CONTENIDOS

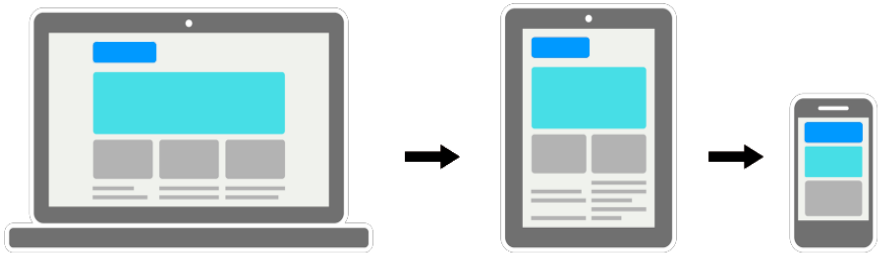


Boceto

Código



{diw} daw.cierva@gmail.com



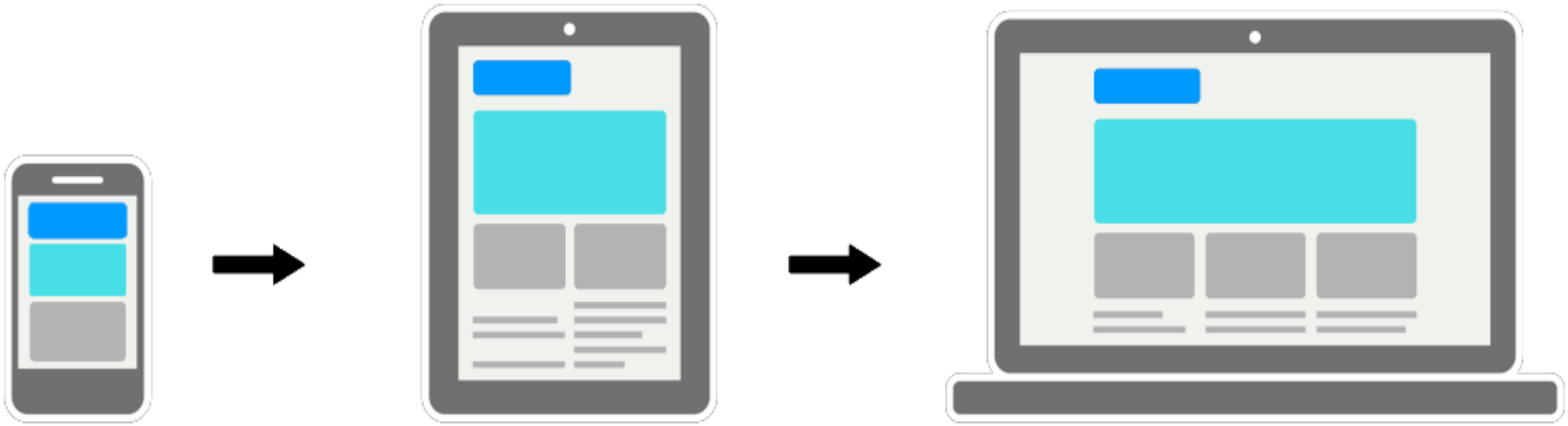
Boceto de mayor a menor

Considerar qué deja de flotar en cada breakpoint

Soluciones posibles

1. **Mantener** igual (logo azul)
2. Dejar que se **estreche** el ancho en porcentaje (slider celeste)
3. **Dejar de flotar** y **aumentar el ancho** en porcentaje (columnas grises)
4. **Ocultar**
5. **Mostrar**

Código de menor a mayor



HTML y CSS para estilos

Flotar y estirar (CSS)

Flotar y estirar (CSS)

RWD (Responsive Web Design)

Diseño de Interfaces Web

José María Torresano
Diciembre 2018
daw.cierva@gmail.com