



# UT5: Diseño Responsivo

DIW

## ¿Qué es?

Es 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.

mobile-seo

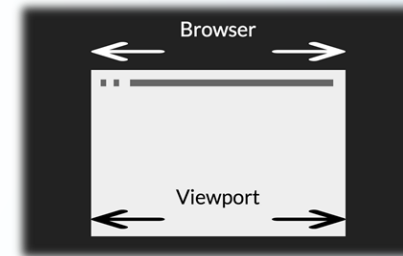


## Conceptos

### ➤ Viewport

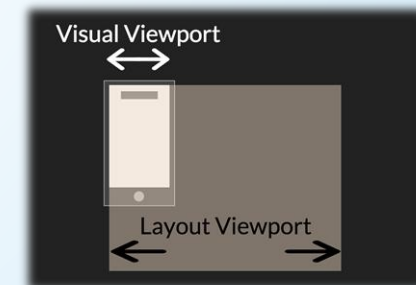
Área de la pantalla en la que el navegador puede renderizar contenido, es decir, el espacio disponible para mostrar mi página web.

- Desktop: el viewport es la pantalla de nuestro navegador.



- Tablets/móviles:

- Layout-viewport: donde se aplican las reglas CSS.
- Visual-viewport: lo que realmente ve el usuario.

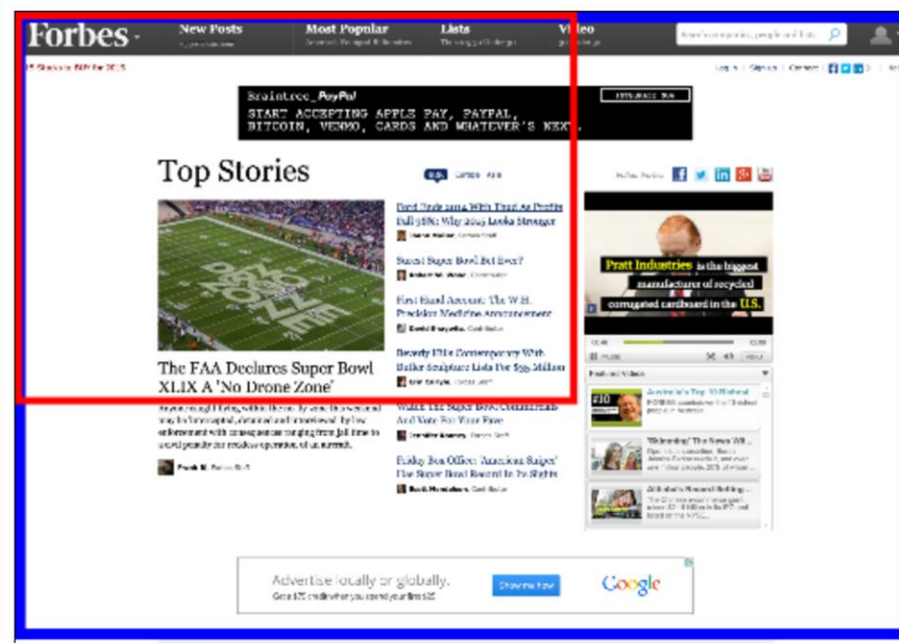


# Conceptos

## ➔ Viewport

Cuando hacemos zoom, **no cambia** el layout-viewport, mientras que el visual-viewport **si cambia**. [Demo](#)

### What the browser sees



Page Scale



Browser Implementation

- ☒ Chrome
- ☐ Edge
- ☐ Firefox
- ☐ Safari

Visual Viewport

Layout Viewport

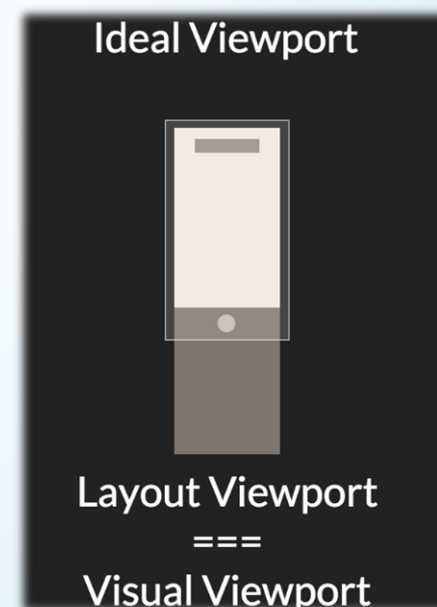
## Conceptos

### ► Metaetiqueta viewport

La declaración viewport nos permite definir los parámetros de visualización de un página web en los diferentes dispositivos.

Consiste en una metaetiqueta mediante la que se establece si se puede hacer zoom en una página, el zoom inicial o la anchura de la pantalla del dispositivo .

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



## Conceptos

### ► Metaetiqueta viewport

Atributo	Valor	Descripción
width	Valor en pixels o constante device-width	Ancho del viewport
height	Valor en pixels o constante device-height	Alto del viewport
initial-scale	Número real entre 0,1 en adelante. 1 = NO ESCALA	Escala inicial del viewport
user-scale	"yes" / "no"	Permisos para que el usuario escale el vwp.
minimum-scale	Número real entre 0,1 en adelante. 1 = NO ESCALA	Escala mínima del viewport
maximum-scale	Número real entre 0,1 en adelante. 1 = NO ESCALA	Escala máxima del viewport



# Conceptos

## ➔ Tipos de layout

No siempre un píxel es un píxel. En la resolución de las pantallas podemos encontrarnos:

- ➔ **Fixed:** Anchura de página fijo expresado en pixels. Predecible y consistente.
- ➔ **Elastic:** Anchura de página fijo expresado en ems (múltiplos del tamaño de letra). Usar rem que es relativo al tamaño de elemento raíz.
- ➔ **Fluid/Liquid/Relative:** Anchura de página depende del tamaño del viewport del usuario y se expresa en porcentaje (%)

## Conceptos

### ➔ Breakpoint

Es la **anchura** en la que se produce un **cambio en el layout** en respuesta a ciertas condiciones del dispositivo (media queries) . [Bootstrap](#)

Breakpoint	Class infix	Dimensions
Extra small	<i>None</i>	<576px
Small	<i>sm</i>	≥576px
Medium	<i>md</i>	≥768px
Large	<i>lg</i>	≥992px
Extra large	<i>xl</i>	≥1200px
Extra extra large	<i>xxl</i>	≥1400px



# Conceptos

## ► Media queries

Módulo de CSS3 que permite adaptar la representación del contenido a las características del dispositivo.

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 el color.

```
@media not|only mediatype and|not|only  
(media feature) {  
  .my-code { ... }  
}
```

## Conceptos

### ► Media queries: *mediatype*

Este *mediatype* es el encargado de seleccionar **qué tipo de formato** será objeto de estas reglas.

Con los operadores *not* y *only* podemos agrupar o excluir un tipo de *mediatype* concreto.

```
@media not|only mediatype  
all  
print  
screen  
speech  
braile / projection / tv (Deprecated)
```

## Conceptos

### ► Media queries: media feature

Indicamos al navegador **qué condición** debe cumplir el dispositivo de salida especificado anteriormente para que la condición sea verdadera y se aplique este código contenido en nuestra media query.

Las más comunes hacen referencia a las dimensiones de los dispositivos, pero hay muchas.

```
@media and|not|only (media feature)
height
width
max-height / min-height
max-width / min-width
```

# Conceptos

## ➔ Media queries: ejemplos

```
@media (min-width: 30em) and (orientation: landscape) {  
    /* ... */  
}  
  
@media (min-width: 600px) and (max-width: 768px) {  
    body {  
        background-color: #de3163;  
    }  
}  
  
@media screen and (min-width: 30em) and (orientation: landscape)  
    /* ... */  
}  
  
@media (min-height: 680px), screen and (orientation: portrait) {  
    /* ... */  
}
```

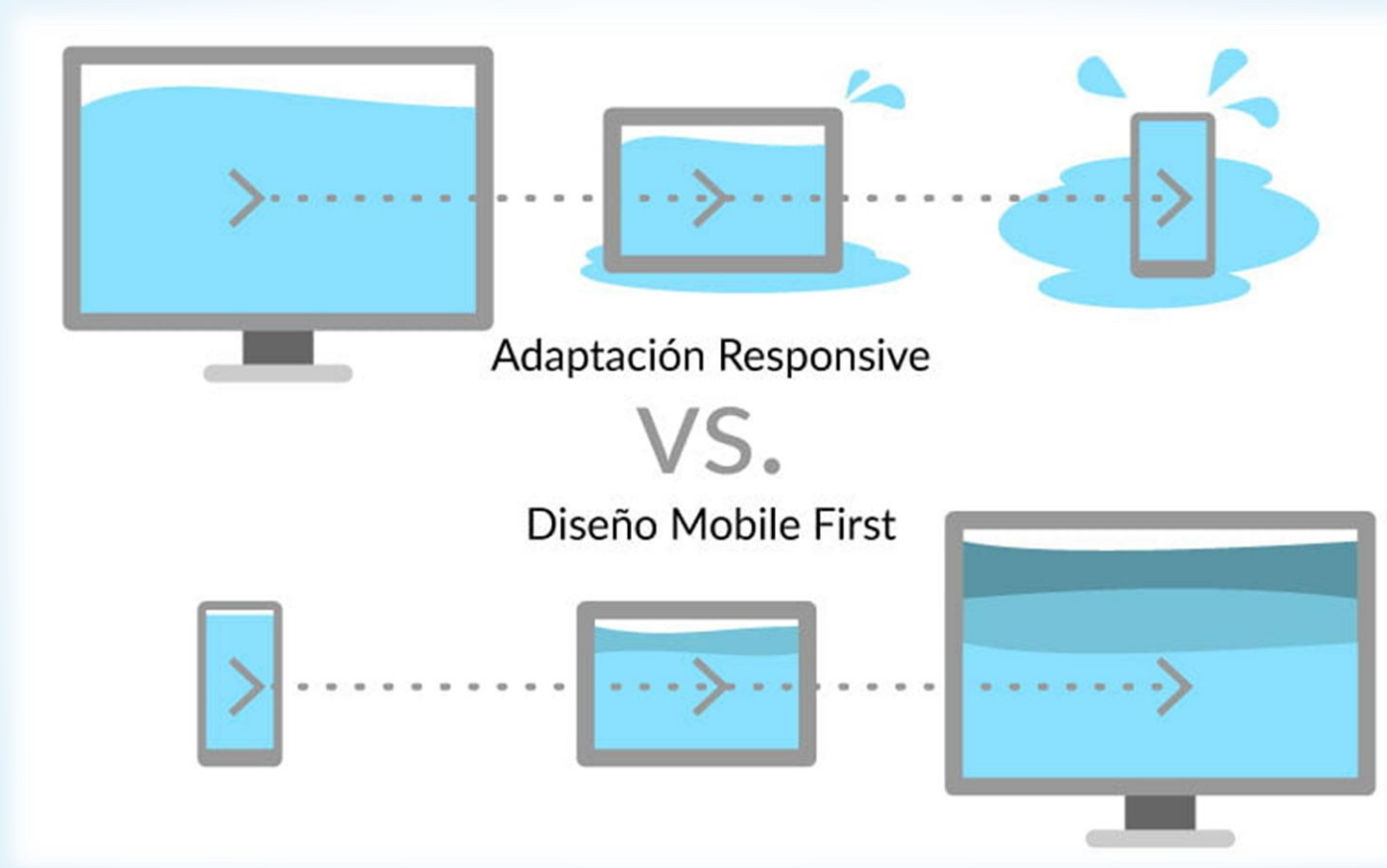
## Conceptos

### ► Media queries: hojas de estilo diferentes

```
<link rel="stylesheet" media="(max-width: 576px)" href="small.css" />
```

```
<link rel="stylesheet" media="(max-width: 768px)" href="medium.css" />
```

```
<link rel="stylesheet" media="(max-width: 992px)" href="large.css" />
```



## Proceso de diseño

### ► Implementación Mobile First

Es necesario colocar las media queries de **menor a mayor** tamaño, usando min-width, por ejemplo: *@media screen and (min-width: 768px)*, lo que significa que esa media query se aplicará desde 768px en adelante, dejando los estilos para mobile fuera de las media queries, de esta manera, los estilos por defecto serían los que se aplican para los dispositivos móviles



sin media querye



min-width: 768px



min-width: 992px



## Proceso de diseño

### ➔ Patrones de diseño

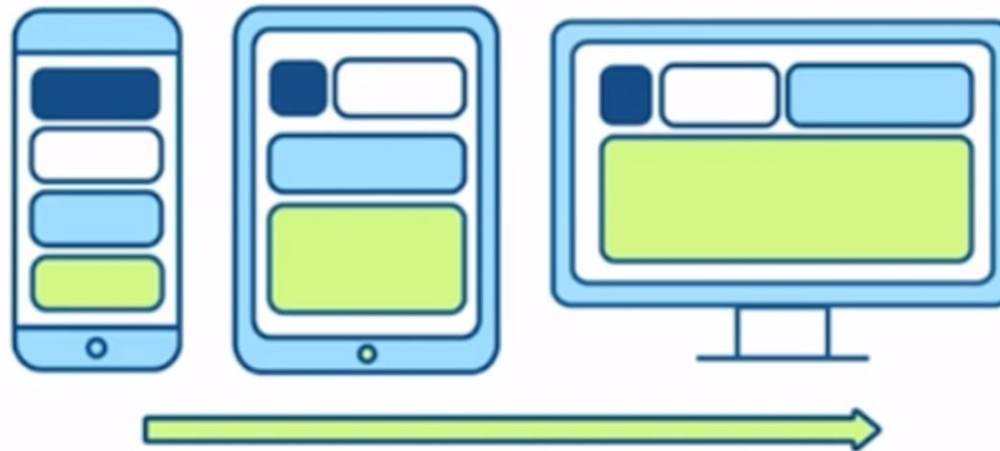
Los patrones son **soluciones** que se han dado por buenas para cierto tipo de problemas. En nuestro caso, el diseño de páginas web responsivas.

- ➔ COLUMN DROP
- ➔ MOSTLY FLUID
- ➔ LAYOUTSHIFTER
- ➔ OFF CANVAS
- ➔ MEZCLA DE VARIOS...
- ➔ PEQUEÑOS AJUSTES (TINY TWEAKS..)

## Patrones de diseño

### ➔ Column Drop

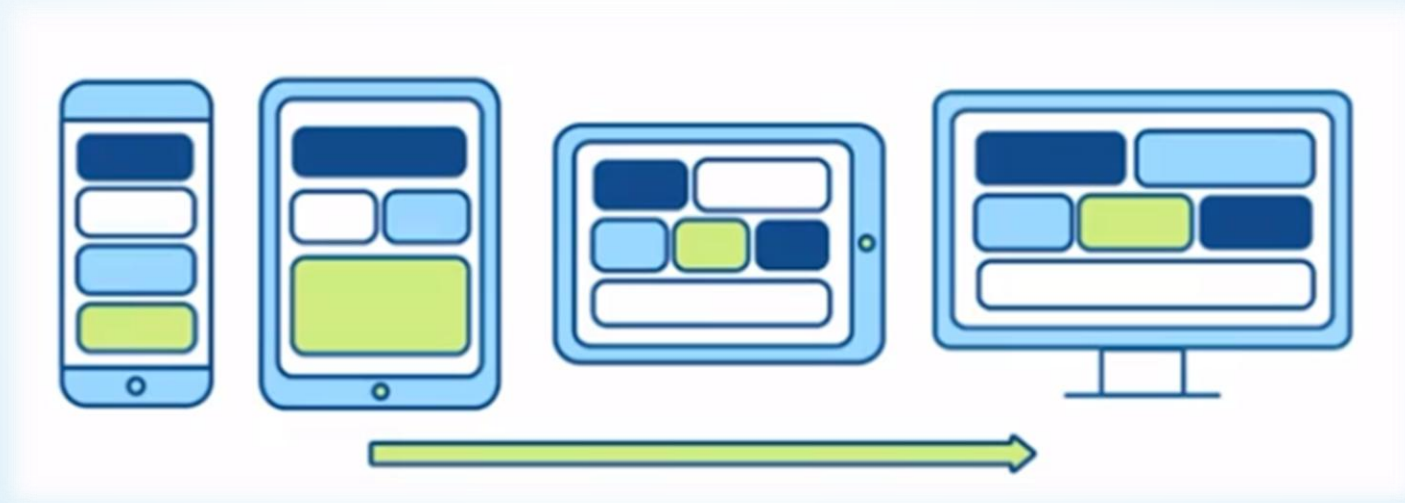
Es el patrón más básico y básicamente consiste en que en cada breakpoint se va apilando un elemento, pudiendo cambiar el orden si es necesario.



## Patrones de diseño

### ➔ Mostly Fluid

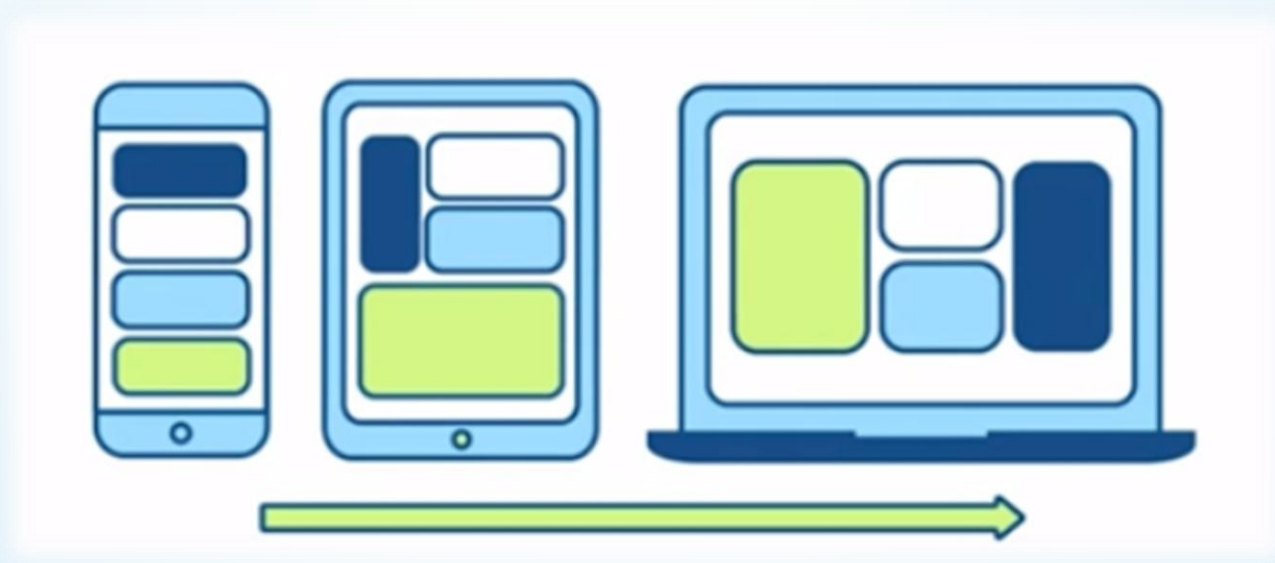
En pantallas grandes o medianas se mantiene el tamaño de la página, ajustando los márgenes de las pantallas más anchas. Podemos determinar un ancho máximo y hacer que, tras ese ancho, simplemente haya más margen a lado y lado, dejando centrada la página.



## Patrones de diseño

### ► Layout Shifter

Es el patrón más “responsivo”, se produce un cambio del diseño en cada Breakpoint, en vez de únicamente organizar el flujo y la anchura de los elementos.

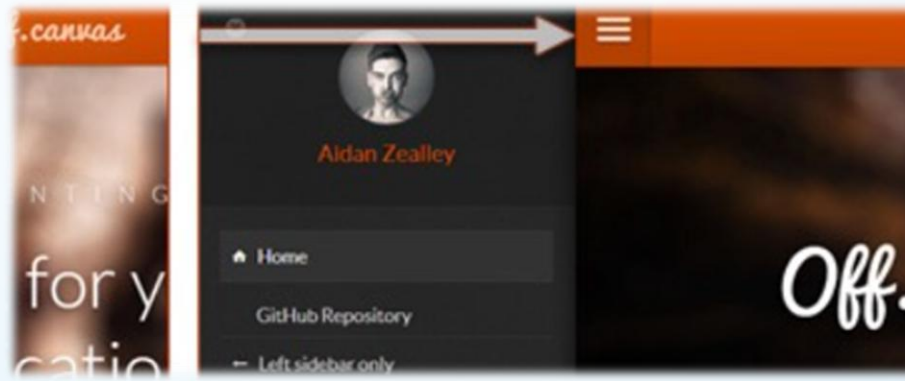


## Patrones de diseño

### ➔ Off Canvas

En vez de apilar contenidos éstos se colocan fuera de la pantalla cuando el tamaño de pantalla no es lo suficientemente grande.

Es importante darle al usuario una forma de revelar estos elementos mediante algún tipo de interacción sencilla e intuitiva, como es pulsando un icono.



## Imágenes responsivas

Las imágenes son un elemento fundamental de todas las páginas y representan una gran parte del “peso” de la misma. Esta situación plantea ciertos retos a la hora de hacer diseño responsivo.

- A nivel de **optimización**: tráfico que genera mi web.
- A nivel de **diseño**: dependiendo del tipo de pantalla puedo querer mostrar una u otra imagen.



## Imágenes responsivas

### ➔ Optimización de imágenes

En diseño responsivo optimizar el uso de imágenes consiste en:

- ➔ Consumir el menor ancho de banda.
- ➔ Elegir la versión de una misma imagen más adecuada para la resolución.

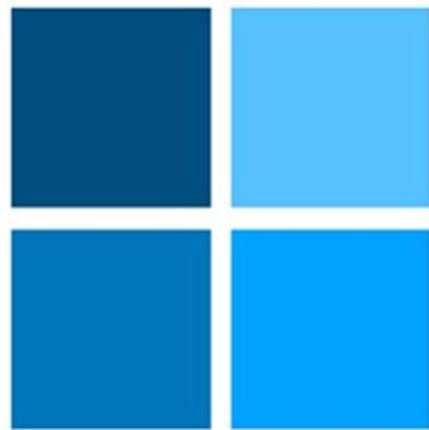
Algunos de los aspectos a tener en cuenta son el **ancho del dispositivo**, las **dimensiones de la imagen** o la **resolución** de la misma (cuidado con dispositivos con resolución Retina Display)



## Imágenes responsivas

### ► Retina display

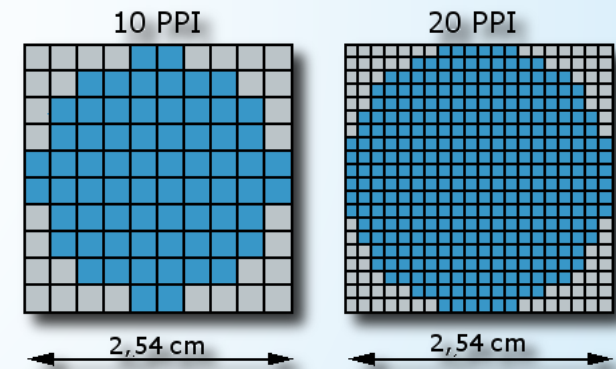
La pantalla Retina incluye cuatro veces más píxeles en el mismo espacio, lo que da como resultado una densidad de píxeles mucho mayor.



Standard display



Retina display



## Imágenes responsivas

### ➔ Optimización de imágenes



## Imágenes responsivas

### ➔ Optimización de imágenes

La solución más fácil es usar **imágenes SVG**, que son gráficos vectoriales que escalan y encogen sin perder resolución. No siempre es posible.

Si tengo imágenes de mapa de bits (png, gif, jpeg):

- ➔ Sin optimización: uso una imagen con gran resolución y luego le doy el tamaño atendiendo al % de la zona que va ocupar.
- ➔ Optimizando: utilizaremos los atributos **srcset** y/o **sizes** de la imagen que queremos mostrar.

## Imágenes responsivas

### HTML

```
<div>
  
</div>
```

### CSS

```
div {
  /* dimensiones deseadas */
}

img {
  max-width: XXXXXpx;
  width: 100%;
}
```

### Considerando resolución:

```
<div class='container'>
  <img src='img/small.jpg'
    srcset='img/big.jpg 2x,img/small.jpg 1x' />
</div>
```

### Considerando dimensiones:

```
<div class='container'>
  <img src='img/small.jpg'
    srcset='img/big.jpg 2000w, /* W → Tamaño real
    img/small.jpg 1000w'
    sizes='(min-width: 960px) 960px,100vw' />
</div>
```

## Imágenes responsivas

```

```

```

```



## Imágenes responsivas

### ► Diseño “Art-Director”

La técnica de diseño responsivo Art Director consiste en elegir una u otra imagen utilizando la **etiqueta source** dentro la **etiqueta picture** y sus **atributos srcset** para indicar la imagen y **media** que funciona de manera similar a una media query.

```
<div class="container">
  <picture>
    <source media="(min-width: 993px)" srcset="img/big-art.jpg" />
    <source media="(min-width: 576px)" srcset="img/medium-art.jpg" />
    <source media="(max-width: 575px)" srcset="img/small-art.jpg" />
     En caso de no soportar picture
  </picture>
```



KEEP  
CALM  
AND  
CARRY  
ON