

# DISEÑO DE INTERFACES WEB

UT4. Aplicar estilos CSS (II):  
Estilos avanzados.  
Diseño adaptativo.



CFGs Desarrollo de Aplicaciones Web  
CIFP Juan de Colonia  
Curso 2020-2021

## Tabla de contenido

1.	Hojas de estilo alternativas .....	2
2.	Hojas de estilo dependientes del medio .....	4
2.1.	Tipos de medios.....	4
2.2.	Propiedades CSS definidas para cada grupo de medios.....	4
2.3.	Cómo definir estilos para cada tipo de medio .....	5
2.4.	Hojas de estilo para impresión .....	6
2.5.	Hojas de estilo auditivas.....	7
3.	Introducción al diseño adaptativo .....	8
3.1.	La filosofía del diseño web adaptativo.....	8
3.2.	Técnicas utilizadas en Diseño Web Adaptativo .....	10
3.3.	Otros aspectos en el diseño adaptativo y del diseño web .....	15
4.	Anexos .....	17
4.1.	Tipos de medios y grupos de medios .....	17
4.2.	Características que se pueden tener en cuenta en las media queries	18
5.	Referencias y bibliografía .....	19

## 1. Hojas de estilo alternativas<sup>1</sup>

Una página web puede tener enlazadas varias hojas de estilo distintas para que el usuario elija cuál se aplica. Esto se logra enlazando las hojas de estilo con la etiqueta <link>.

**Hojas de estilo persistentes: se aplican siempre.**

- El usuario no puede elegir no aplicarlas.
- Son las enlazadas por un <link> que no tiene el atributo **title**.

**Hojas de estilo alternativas y hojas de estilo alternativas preferidas.**

- Son las enlazadas por un <link> que tiene el atributo **title**.
- Puede haber un grupo de hojas que tenga el atributo **title** con el mismo valor. En este caso, el usuario puede elegir qué hoja (o grupo de hojas) aplicar, seleccionándolas por el valor de title.
- Por defecto se aplican las hojas que aparezcan primero. Sin embargo, si se enlazan hojas de estilo con el atributo **rel="stylesheet"** y otras con **rel="alternate stylesheet"**, se entenderá que las primeras son las hojas de estilo alternativas preferidas.

*Ejemplo:*

```
<!-- Hoja de estilos persistente -->
<link rel="stylesheet" href="css/todos.css">

<!-- Hoja de estilos alternativa preferida -->
<link rel="stylesheet" href="css212/212.css" title="CSS Zen Garden
212">

<!-- Otras hojas de estilos alternativas -->
<link rel="alternate stylesheet" href="css214/214.css" title="CSS Zen
Garden 214">

<link rel="alternate stylesheet" href="css215/css/215.css" title="CSS
Zen Garden 215">
```

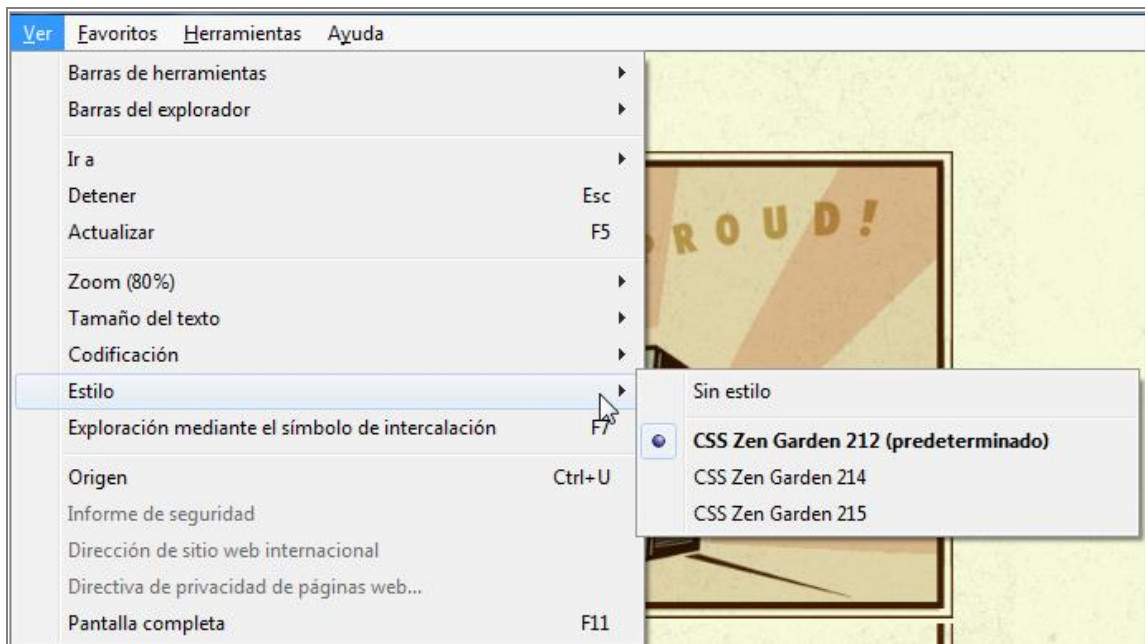
---

<sup>1</sup> [http://www.mcilibre.org/consultar/htmlcss/css/css\\_cssalternativas.html](http://www.mcilibre.org/consultar/htmlcss/css/css_cssalternativas.html)

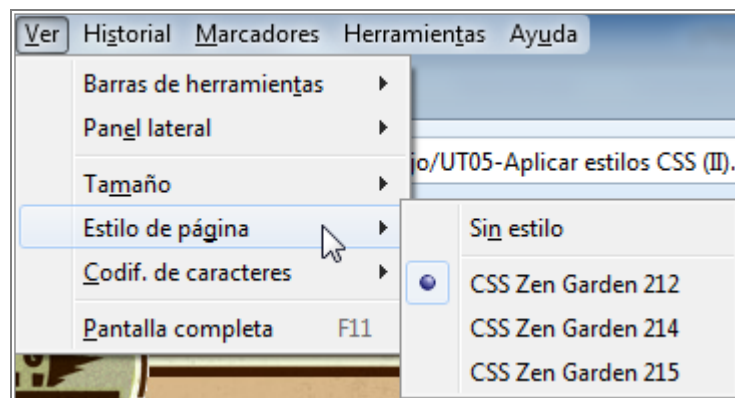
[http://librosweb.es/libro/css\\_avanzado/capitulo\\_6/estilos\\_alternativos.html](http://librosweb.es/libro/css_avanzado/capitulo_6/estilos_alternativos.html)

### Seleccionar una hoja de estilos en el navegador:

- **Internet Explorer:** Mostramos el menú superior (con la tecla **Alt**). Seleccionar **Ver > Estilo**.



- **Firefox:** Mostramos el menú superior (con la tecla **Alt**). Seleccionar **Ver > Estilo página**.



- **Chrome:** No permite seleccionar hojas alternativas, aunque se pueden instalar extensiones que permiten hacerlo.
- **Microsoft Edge:** No permite seleccionar hojas de estilo alternativas.

## 2. Hojas de estilo dependientes del medio

- En las páginas web se puede definir qué estilos se aplican, dependiendo del tipo de dispositivo (media) a donde "se envía" la página Web.
- Habitualmente las páginas web "se envían" a dispositivos con pantalla, como ordenadores, tabletas, móviles, etc.
- Pero también se pueden enviar a impresoras o a dispositivos que las "leen" para personas con dificultades de visión. En estos casos, puede interesar definir reglas CSS específicas.

### 2.1. Tipos de medios

- CSS define varios tipos de medios, entre los que están: **all** (todos los tipos de medios), **screen**, **print**, etc.
- Los nombres de los "tipos de medios" son palabras reservadas, por lo que se pueden escribir en mayúsculas o minúsculas indistintamente (son *case-insensitive*) y no irán entre comillas.

### Relación entre los agentes de usuario y los medios

- Un agente de usuario (navegador, lector de pantalla, etc.) sólo puede soportar un tipo de medio en el momento de interpretar un documento.
- Sin embargo, los agentes de usuario pueden tener diversos modos que soporten diferentes medios. Por ejemplo, un navegador puede soportar los modos "screen" y "print" (vista preliminar).

### 2.2. Propiedades CSS definidas para cada grupo de medios

#### Grupos de medios

- Los medios se agrupan en **grupos de medios** según sus propiedades.
- Por ejemplo, el medio "screen" pertenece a los grupos: continuo, visual, audio, bitmap, interactivo, estático.

#### Propiedades CSS y grupos de medios

- En la definición de cada propiedad CSS se especifica a qué grupo o grupos se aplica esa propiedad.
  - Por ejemplo, la propiedad page-break-before se define para medios paginados (no se puede aplicar a la pantalla).

#### 'page-break-before'

Valor:	auto   always   avoid   left   right   <u>inherit</u>
Inicial:	auto
Se aplica a:	Elementos a nivel de bloque (block-level elements)
Se hereda:	no
Porcentajes:	N/A
Medio:	<u>visual</u> , <u>paginados</u>
Valor computado:	Como el especificado

## 2.3. Cómo definir estilos para cada tipo de medio

- Con elemento `<link>`: atributo "media":

```
<link rel="stylesheet" type="text/css" media="print, handheld" href="foo.css">
```

- Con la regla `@media`, en el elemento `<style>` o en un archivo `.css`.

Con esta regla se pueden definir reglas para distintos medios en la misma hoja de estilo:

```
<style>
  @media print {
    body { font-size: 10pt }
  }
  @media screen {
    body { font-size: 13px }
  }
  @media screen, print {
    body { line-height: 1.2 }
  }
</style>
```

- Con la regla `@import`, en el elemento `<style>` o en un archivo `.css`.

```
<style>
  @import url("fancyfonts.css") screen;
</style>
```

### Qué ocurre si se especifica un medio incorrecto

- Se tratan como si ese tipo de medio no estuviera presente, se ignoran.

Por ejemplo, la siguiente regla se aplicará en el modo "screen" para los elementos `p`, aunque el tipo de medio `3D` no existe:

```
@media screen, 3D {
  p { color: green; }
}
```

### Medio por defecto

El medio por defecto es **all**.

## 2.4. Hojas de estilo para impresión<sup>2</sup>

En algunos sitios web puede ser aconsejable preparar una hoja de estilos para la versión impresa de dicha página. Estas hojas de estilos eliminan contenidos superfluos, eliminan o modifican las imágenes y contenidos de fondo, etc.

### Cómo definir estilos diferentes para imprimir y para visualizar una página

Definiendo reglas CSS específicas para cada tipo de medio.

### Prácticas habituales para crear una hoja de estilos para impresión

Una práctica habitual (no la única) es:

- Definir primero los estilos sin especificar el tipo de medio. De este modo se definen las páginas como se verán en pantalla, pero también son estilos que se aplicarán en el momento de imprimir.
- Después, definir estilos específicos para la impresión, es decir, indicando el tipo de medio. Estas reglas se pueden definir con **!important**; para que sobrescriban la definición por defecto.

Algunos estilos que se pueden cambiar (y ejemplos):

1. **Ocultar los elementos** que no se van a imprimir, utilizando el atributo **display: none**

```
#cabecera, #menu, #lateral, #comentarios {  
    display: none !important;  
}
```

2. **Corregir la estructura de la página**, quitando los márgenes, los rellenos, haciendo que todas las capas ocupen todo el ancho, y que no haya capas flotantes.

```
body, #contenido, #principal, #pie {  
    float: none !important;  
    width: auto !important;  
    margin: 0 !important;  
    padding: 0 !important;  
}
```

3. **Modificar los colores y tipos de letra**

- El color de texto puede ser negro o gris, para aumentar el contraste con el fondo de la hoja blanca.
- El tipo de letra puede ser alguno que facilite la lectura del texto.

```
body {  
    color: #000; font: 100%/150% Georgia, "Times New Roman",  
    Times, serif;  
}
```

---

<sup>2</sup> [http://librosweb.es/libro/css/capitulo\\_13/version\\_para\\_imprimir.html](http://librosweb.es/libro/css/capitulo_13/version_para_imprimir.html)

#### 4. Imprimir los enlaces con las pseudoclases **after** y **before**

Nos puede interesar imprimir las rutas asociadas a los hiperenlaces.

- Si queremos que se imprima después del enlace:

```
a:after {  
    content: " (" attr(href) ") ";  
}
```

- Si queremos que se imprima antes del enlace:

```
a:before {  
    content: " (" attr(href) ") ";  
}
```

### 2.5. Hojas de estilo auditivas<sup>3</sup>

Normalmente, para facilitar la accesibilidad a una página web por parte de una persona ciega o con deficiencias visuales, el documento es leído por un sintetizador de voz, como Jaws o NVDA (para Windows)

Las hojas de estilo auditivas pretenden proporcionar una alternativa sonora a la presentación visual, es decir configurar cómo tienen que leer. CSS 2.1. reserva el tipo de medio "**speech**" pero no define qué propiedades se le deben aplicar. CSS 2 definió el tipo de medio "**aural**" (auditivo), pero CSS 2.1. lo declaró obsoleto. Sin embargo, está definido.

Esto significa que una hoja de estilos como la siguiente, es válida, pero no está definida por CSS 2.1.

```
@media speech {  
    body { voice-family: Paul }  
}
```

Sin embargo, lo siguiente, está obsoleto, pero sí está definido en CSS:

```
@media aural {  
    body { voice-family: Paul }  
}
```

Existen propiedades CSS específicas para las hojas de estilo auditivas, como las que se muestran en el siguiente ejemplo, tomado del W3C.

#### ***Ejemplo de hoja de estilos auditiva***

Los títulos se leerán con una voz llamada "paul", en un tono plano, pero en voz muy alta. Antes de decir los títulos, se escuchará un sonido.

Los párrafos con clase "heidi" parecerán venir del frente-izquierdo (si el sistema de sonidos es soporta el audio espacial), y los párrafos con clase "peter" vendrán de la derecha.

Los párrafos con clase "goat" serán muy suaves.

---

<sup>3</sup> El soporte por parte de los navegadores es desigual. Para NVDA se aconseja utilizar Firefox, pero no se ha podido comprobar que interprete la hoja de estilos.



```
h1, h2, h3, h4, h5, h6 {  
    voice-family: paul;  
    stress: 20;  
    richness: 90;  
    cue-before: url("ping.au")  
}  
  
p.heidi { azimuth: center-left }  
p.peter { azimuth: right }  
p.goat { volume: x-soft }
```

### 3. Introducción al diseño adaptativo

#### 3.1. La filosofía del diseño web adaptativo

En mayo de 2010, Ethan Marcotte publica un artículo y habla por primera vez de *Responsive Web Design*<sup>4</sup>. El artículo va acompañado de un ejemplo.

En castellano, la expresión *Responsive Web Design* (RWD) se ha traducido como Diseño Flexible, Diseño Adaptable o incluso Diseño Responsivo. Sin embargo, la expresión más utilizada es *Diseño Web Adaptativo*.

#### El problema de la variedad de dispositivos de acceso a la web

En ese artículo, el autor se acerca al concepto de diseño adaptativo desde la arquitectura. Se plantea que cuando se hace un edificio, el diseño es algo rígido, porque va a durar muchos años. Sin embargo, en el diseño Web, los sitios se rediseñan cada pocos años.

Existe un problema que crece de año en año: la variedad de dispositivos con los que se accede a una página Web<sup>5</sup>. Esto nos plantea un problema: no sabemos con qué dispositivo se va a acceder al sitio Web, es imposible diseñar para todos los dispositivos posibles.

#### Modos de enfocar el problema

- **Crear una aplicación móvil.** Esta es una buena solución, de hecho, muchos sitios web ofrecen además, una app para acceder a los contenidos.
- **Desarrollar una web para escritorio y otra para móvil.** Normalmente tendrán dos URL diferentes. El inconveniente de esta solución es que tendremos dos sitios web que mantener.
- **Detectar el dispositivo.** Consiste en detectar el dispositivo en el lado del servidor y, dependiendo de este dato, enviar al usuario un HTML o un CSS diferente. En este caso, se trata de la misma URL, pero muestra contenidos diferentes dependiendo del dispositivo desde el que se accede. En la práctica, tenemos también dos sitios web a mantener.

---

<sup>4</sup> Artículo original: <http://alistapart.com/article/responsive-web-design>; traducción al castellano <http://xn--diseowebresponsivo-q0b.com.ar/>

<sup>5</sup> <http://gs.statcounter.com/>

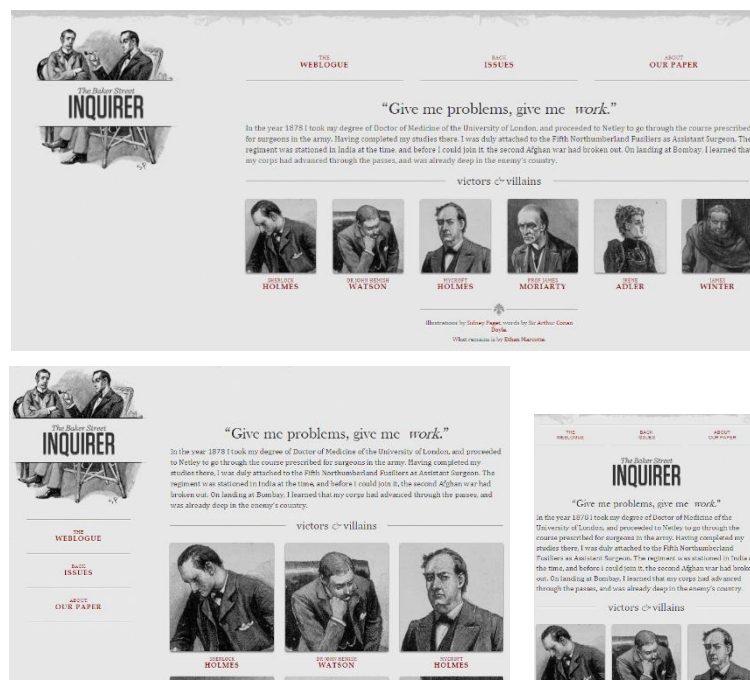
- **Diseño Web Adaptativo.** En este caso, hay una sola Web (un sólo HTML) que se adaptará al dispositivo de salida utilizando CSS.

## El fundamento del diseño adaptativo

El fundamento del diseño adaptativo es que la página sea capaz de adaptarse al dispositivo, utilizando principalmente CSS. En <http://mediaqueri.es/> se recogen muchos ejemplos de sitios adaptativos.

La ventaja de este enfoque es que tenemos un sólo sitio web que mantener.

¿Cómo es un sitio web adaptativo? Utilizaremos el ejemplo de Ethan Marcotte: si vamos estrechando la pantalla, vemos tres distribuciones de los elementos en la página:



Vemos que el contenido es siempre el mismo; lo que cambia es la presentación del mismo.

Si visualizamos la página y cambiamos el ancho de la ventana del navegador, podemos observar que:

1. El diseño es fluido (flexible). Vemos que el ancho de las columnas se adapta.
2. Además, el tamaño de las imágenes también se adapta al espacio disponible.
3. Hay puntos en los que el diseño de la página, la estructura o layout, cambia. Esto ocurre para determinados anchos de página (los que el diseñador de la página haya decidido).

Estos tres aspectos se verán a continuación:

1. Diseño fluido.
2. Imágenes flexibles.
3. Media queries CSS.

### 3.2. Técnicas utilizadas en Diseño Web Adaptativo

#### Diseño fluido (*fluid grid*, cuadrícula fluida)

Lo hemos visto en la unidad anterior: consiste en utilizar porcentajes para definir el ancho de los elementos.

#### Imágenes flexibles (y los elementos multimedia)

Se consigue aplicando la siguiente regla a las imágenes:

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

De este modo, la imagen tomará, como máximo, el ancho de su contenedor, independientemente de su tamaño original. El navegador redimensionará la imagen manteniendo las proporciones, cuando el contenedor cambie de tamaño.

Esta regla se puede aplicar al resto de elementos multimedia:

```
img,  
embed,  
object,  
video {  
    max-width: 100%;  
}
```

Compatibilidad:

- IE6 no soporta la propiedad max-width.

Nota

- Si se especifica también la altura (height) de la imagen en el HTML, o en el CSS; la imagen se puede deformar.

#### Imágenes de fondo flexibles

La solución al problema de las imágenes de fondo flexible viene dada por la propiedad **background-size** de CSS3.

#### No basta con redimensionar las imágenes

La gestión de imágenes en el diseño adaptativo no es tan simple. Utilizando esta solución probablemente estaremos enviando una imagen de un tamaño superior al que probablemente se necesita. Hay varias soluciones. Una de ellas, es servir imágenes diferentes en función del tamaño del dispositivo.

El elemento HTML **<picture>** es un contenedor usado para especificar múltiples elementos **<source>** y un elemento **<img>** contenido en él para proveer versiones de una imagen para diferentes escenarios de dispositivos. Si no hay coincidencias con los elementos **<source>**, se utiliza el archivo especificado en el atributo **src** del elemento **<img>**. La imagen seleccionada es entonces presentada en el espacio ocupado por el elemento **<img>**.

### Ejemplo:

```
<picture>
  <source media="(min-width: 650px)" srcset="flores/flor_rosa.jpg">
  <source media="(min-width: 465px)" srcset="flores/flor_amarilla.jpg">
  
</picture>
```

### La etiqueta <meta name="viewport" ...>

Para utilizar las media queries, hay que **definir el viewport de la página**.

#### Ancho de página en los navegadores de móviles

Los navegadores de los móviles ajustan las páginas web a un "lienzo" (viewport) con un ancho predeterminado (ej: 980px en iPhone, 850px en Opera, 800px en Android) y después ajustan esa página a la pantalla del móvil, cambiando el tamaño del contenido para que quepa en la pantalla.

Si accedemos a una página que no esté preparada para móviles<sup>6</sup> desde un emulador de navegador de móvil, veremos la pantalla como si estuviera en un navegador de escritorio. Como el ancho del móvil es menor, todo el contenido es muy pequeño y difícil de ver.

#### Etiqueta <meta name="viewport" content="...">

Para solucionar el problema para el iPhone, Apple utilizó la etiqueta <meta>, para definir el ancho del viewport. Esta etiqueta ha sido adoptada por la mayoría de los navegadores móviles, convirtiéndose en un estándar de facto.

En el "contenido" se definen pares propiedad=valor separadas por comas.

```
<meta name="viewport" content="prop1=valor1 {, prop=valor}">
```

Esta etiqueta permite definir el ancho, alto y escala del área usada por el navegador para mostrar el contenido (aunque el alto, height, no se suele definir).

Las propiedades más utilizadas son:

- **width**: Define el ancho que tendrá la página. Puede ser un valor numérico en píxeles (ej. 320px, o la constante **device-width**, que indica el ancho total del dispositivo).
- **initial-scale**: Define el nivel del zoom inicial al cargarse la página. Su valor puede ser un número real desde 0.1 en adelante. Un valor de 1 indica que la página se muestra al 100% de su tamaño.
- **minimum-scale** y **maximum-scale**: Definen los valores mínimo y máximo de zoom permitido. El valor será un número real de 0.1 en adelante.
- **user-scalable**: Define si el usuario puede hacer zoom para escalar el contenido. Puede tomar dos valores: yes (valor por defecto) o no.

Es recomendable que los usuarios puedan hacer zoom por accesibilidad y usabilidad.

<sup>6</sup> Ej: <http://www.mecd.gob.es/portada-mecd/>

La declaración del viewport es necesaria para controlar el modo en que la página se verá en los dispositivos móviles. Es importante definir la anchura y la escala inicial. No se suele definir la altura, porque se calcula automáticamente en función de la anchura.

Una declaración habitual es:

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

*Ejemplo.* Vamos a ver el efecto de la etiqueta `<meta name="viewport"...>` en el ejemplo elaborado por Ethan Marcote.

1. Modificamos el ejemplo desarrollado por Ethan Marcote, comentando la línea siguiente:

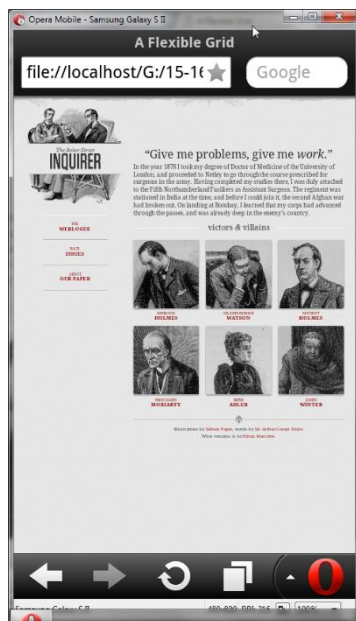
```
<!--meta name="viewport" content="width=device-width, initial-scale=1.0" /-->
```

2. Ahora accedemos con un navegador, con la vista de emulador de móvil, o desde un emulador como el Opera Mobile Emulator. Observamos que la página se ha redimensionado para que quepa en la ventana del dispositivo.

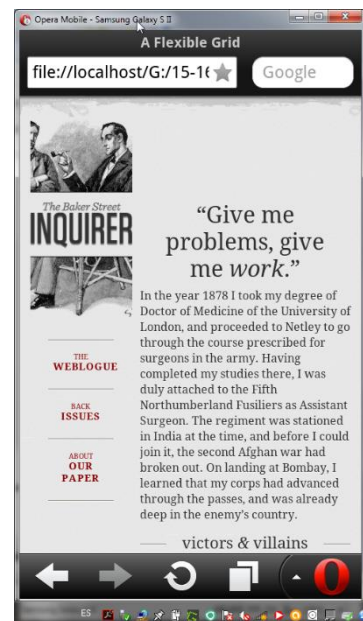
3. Descomentamos la línea del viewport:

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

4. Volvemos a visualizar la página. Vemos que ahora la página se adapta al tamaño del dispositivo.



Página sin definir el viewport.



Página definiendo el viewport.

### La regla @viewport

La tendencia sería a que la definición del viewport se haga desde la hoja de estilos con una regla-arroba @viewport estandarizada. Se definiría así:

```
@viewport { width: device-width; zoom: 1.0; }
```

Consultar la compatibilidad, porque aún no está correctamente implementado en todos los navegadores.

## Media queries

Por último, en el diseño adaptativo hay que aplicar reglas CSS diferentes según sea el ancho del navegador. Esto se consigue con las media queries.

### Sintaxis de las media queries

Una *media query* es una extensión de la definición de reglas para distintos medios. Está formada por un tipo de medio (opcional) y cero o más expresiones con las que especificar condiciones sobre las características del medio. Las expresiones se escriben entre paréntesis.

Ejemplo:

```
@media screen and (min-width: 400px)
```

### Varias condiciones

```
@media screen and (min-width: 400px) and (max-width: 700px)
```

### Dónde definir las media queries

Las *media queries* se pueden definir en el elemento `<link>`, en la regla `@import` y en `@media`.

*Ejemplo:* Queremos aplicar una hoja de estilos (`example.css`) si a los dispositivos de un tipo de medio ("screen") con tamaño de pantalla mínimo de 400px.

Se puede definir en el elemento `<link>`

```
<link rel="stylesheet" media="screen and (min-width: 400px)"  
href="example.css" />
```

En la regla `@import`:

```
@import url(example.css) screen and (min-width: 400px);
```

Y en la regla `@media`:

```
@media screen and (min-width: 400px) { ... }
```

### Utilizar link, @import o @media

Si definimos los estilos en archivos separados, hay que hacer una petición HTTP al servidor para acceder a él. Esto ocurre si definimos los estilos con una etiqueta `link` o con `@import`.

A cambio, una vez que descargamos el archivo, lo pueden utilizar todas las páginas a las que accedamos. Normalmente los desarrolladores utilizan archivos para los diseños más importantes y `@media` para los ajustes.

### No es necesario especificar el medio

Si no se indica el tipo de medio, se tomará el valor por defecto (**all**).

*Ejemplo:* las dos reglas siguientes son idénticas:

```
@media all and (min-width:400px) { ... }  
@media (min-width:400px) { ... }
```

## Combinar varias media queries

Se pueden combinar varias *media queries* en una **media query list**. Se utiliza una coma para separar las *media queries*.

*Ejemplo:*

```
@media screen and (min-width:400px), projection and (min-width:400px) { ... }
```

## Características que se pueden evaluar con *media queries*.

Además del ancho de la pantalla (width), se pueden evaluar otras características (ver anexo 2).

## Ejemplo de media queries

Pantalla, tamaño máximo 700px y orientación landscape:

```
@media screen and (max-width: 700px) and (orientation: landscape) {...}
```

## Cuántas media queries utilizar

Al hacer el diseño adaptativo, hay que tomar una decisión: ¿cuántas media queries vamos a definir y para qué anchuras? ¿A qué dimensiones queremos que se produzcan los cambios en el aspecto? A estas dimensiones se les llama *breakpoints* o puntos de ruptura.

¿Cuántos puntos de ruptura definir? En general, se pueden definir puntos de ruptura para tres diseños: uno para móvil, otro para tablet y un tercero para pantallas grandes de escritorio. Sin embargo, no siempre será así.

*Ejemplo.* Modificar el diseño original de Ethan Marcote, según su artículo, añadiendo CSS al final del elemento <style>.

Para anchos inferiores a 600px, no habrá columnas.

```
@media screen and (max-width: 600px) {  
  .mast,  
  .intro,  
  .main,  
  .footer {  
    float: none;  
    width: auto;  
  }  
}
```

Se cambia el tamaño de las imágenes para tamaños más pequeños, para que haya dos imágenes por fila.

```
@media screen and (max-width: 400px) {  
  .figure, li#f-mycroft {  
    margin-right: 3.317535545023696682%; /* 21px / 633px */  
    width: 48.341232227488151658%; /* 306px / 633px */  
  }  
  
  li#f-watson, li#f-moriarty {  
    margin-right: 0;  
  }  
}
```

Por último, para pantallas muy anchas, se muestran todas las imágenes en la misma fila

```
@media screen and (min-width: 1300px) {  
  .figure, li#f-mycroft {  
    margin-right: 3.317535545023696682%; /* 21px / 633px */  
    width: 13.902053712480252764%; /* 88px / 633px */  
  }  
}
```

### 3.3. Otros aspectos en el diseño adaptativo y del diseño web

#### Estrategia "Mobile first"

Se trata de una tendencia en el desarrollo de sitios adaptativos. Consiste en empezar desarrollando el diseño más sencillo, más pequeño; y usar media queries para definir los estilos de los diseños para dispositivos con ancho de pantalla mayor, de forma que estos últimos estilos sobrescriban los anteriores.

Este modo de diseñar fue propuesto por Luke Wroblewski. Cuando lo expuso, dio tres razones:

- El uso de móviles está en continuo aumento.
- Las dimensiones de la pantalla nos obligan a establecer prioridades, a centrarnos en lo esencial, ya que no hay sitio para lo superfluo.
- La versión móvil no tiene por qué ser una versión reducida de la página web. Al contrario, los dispositivos móviles proporcionan nuevas características, como la geolocalización.



En esta estrategia, las media queries suelen ser de este tipo:

```
/* Estilos de la version móvil */  
  
...  
  
/* Estilos para pantallas pequeñas */  
@media screen and (min-width: 600px) { ... }  
  
/* Estilos para pantallas grandes */  
@media screen and (min-width: 800px) { ... }
```

### Los menús de navegación y las pantallas pequeñas<sup>7</sup>

Algunos elementos de las páginas web requieren un tratamiento particular al adaptarse a pantallas pequeñas. Uno de los más importantes son los menús de navegación.

Algunas soluciones utilizadas son:

- Utilizar un selector de formulario.
- Reorganizar el tamaño del menú. Si el número de opciones del menú no es muy elevado, puede bastar con reducir su tamaño o modificar su disposición.
- Ocultar el menú. Es una de las soluciones más aplicadas últimamente. Consiste en ocultar el menú reduciéndolo a un icono. Se puede construir con CSS o con JavaScript.

### Compatibilidad con IE8 y anteriores

IE8 no soporta los *media queries*. Una solución: utilizar los comentarios condicionales que proporciona Internet Explorer.

Supongamos que estamos utilizando la estrategia "**mobile first**" y se han creado varias hojas de estilo, para aplicar dependiendo del ancho del navegador.

Lo que haremos será crear una nueva hoja de estilos con los estilos que se aplicarán a IE8. Se puede hacer simplemente copiando y pegando, por orden, todos los estilos definidos en las hojas de estilo ya creadas. Supongamos que se llama `ie8_styles.css`.

```
<link rel="stylesheet" href="estilos.css">  
<!--[if(lt IE 9)&(!IEMobile)]>  
    <link rel="stylesheet" href="ie8_styles.css">  
<![endif]-->
```

---

<sup>7</sup> Menú de navegación adaptable a dispositivos móviles:

<http://www.falconmasters.com/web-design/menu-de-navegacion-responsive/>

<http://www.falconmasters.com/web-design/menu-desplegable-adaptable-dispositivos-moviles/>

## La mejora progresiva

La mejora progresiva significa que los sitios web se deben pensar para navegadores más o menos avanzados. Este concepto supera la idea de que todos los navegadores deben mostrar la página exactamente del mismo modo. Este concepto no se aplica sólo al diseño adaptativo, sino a todo el proceso de diseño.

Por ejemplo, no todos los navegadores reconocen las funciones CSS3 como `rgba()`. En este caso, la mejora progresiva significa que debemos preparar nuestra página web para que se vea también en esos navegadores; aunque en un navegador que admita transparencias, aprovechemos esa característica.

## 4. Anexos

### 4.1. Tipos de medios y grupos de medios

#### Tipos de medios

Medio	A qué dispositivos se aplica
<b>all</b>	Todos los dispositivos
<b>braille</b>	Dispositivos táctiles de braille
<b>embossed</b>	Impresoras braille
<b>handheld</b>	Dispositivos de mano (pantalla pequeña, ancho de banda reducido)
<b>print</b>	Dispositivos paginados y mostrados en vista previa de impresión
<b>projection</b>	Dispositivos de proyección de presentaciones
<b>screen</b>	Pantallas a color de equipos informáticos
<b>speech</b>	Para sintetizadores de voz (similar a "aural" en CSS2)
<b>tty</b>	Dispositivos de visualización con rejilla de caracteres de espaciado fijo, como terminales. No se deben utilizar unidades px
<b>tv</b>	Dispositivos de tipo televisión: baja resolución, color, desplazamiento limitado, sonido disponible

## Grupos de medios

*Relación entre los grupos de medios y los tipos de medios*

Tipos De Medios	Grupos De Medios			
	continuos/ paginados	visual/audio/ voz/táctiles	rejillas/bitmap	interactivos/ estáticos
<b>braille</b>	continuo	táctil	rejilla	ambos
<b>embossed</b>	paginado	táctil	rejilla	estático
<b>handheld</b>	ambos	visual, audio, voz	ambos	ambos
<b>print</b>	paginado	visual	BITMAP	estático
<b>projection</b>	paginado	visual	BITMAP	interactivo
<b>Screen</b>	continuo	visual, audio	BITMAP	Ambos
<b>Speech</b>	continuo	Voz	N/A	Ambos
<b>Tty</b>	continuo	visual	rejilla	Ambos
<b>TV</b>	ambos	visual, audio	BITMAP	Ambos

### 4.2. Características que se pueden tener en cuenta en las media queries

Característica	Descripción
width y height	Dimensiones del área de visualización (viewport). Admite medidas fijas y relativas.
device-width y device-height	Igual que el anterior, pero referido a la resolución de la pantalla
orientation	Orientación del dispositivo. Puede ser vertical (portrait) y horizontal (landscape).
aspect-ratio	Relación ancho-alto del área de visualización.
device-aspect-ratio	Igual que el anterior, pero referido a la resolución de la pantalla.
color	Profundidad del color (número de bits por color). Si el dispositivo no tiene color, el valor es cero.
color-index	Número de colores en la tabla de colores
monochrome	Número de bits por píxel en dispositivos monocromo
resolution	Resolución del dispositivo expresada en puntos por pulgada (dpi) ( <i>dots per CSS inch</i> ) o en puntos por centímetro, dpcm ( <i>dots per CSS centimeter</i> ).
device-pixel-ratio	Densidad del píxel (puntos por píxel) del dispositivo. Los valores habituales son 1.5 y 2 para las pantallas retina. Se utilizan los prefijos webkit y moz en referencia al motor de renderizado del navegador.

Todas las propiedades admiten los prefijos min- y max- (excepto la orientación).

## 5. Referencias y bibliografía

### REFERENCIAS WEB

#### Responsive design

- Responsive web design, de Ethan Marcotte: <http://xn--diseowebresponsivo-q0b.com.ar/> (Original: <http://alistapart.com/article/responsive-web-design/>).
- Etiqueta viewport. Explicación de las propiedades de esta etiqueta, con un ejemplo detallado: <http://emiliocobos.net/meta-viewport-html/>.
- Referencia viewport: CSS Device Adaptation. Editor's Draft 9 August 2019 (aún no es W3C Recommendation): <http://dev.w3.org/csswg/css-device-adapt/>.
- Media Queries. W3C Recommendation 19 June 2012: <http://www.w3.org/TR/2012/REC-css3-mediaqueries-20120619/>
- Media Queries Level 4 W3C Candidate Recommendation, 5 September 2017 <https://www.w3.org/TR/mediaqueries-4/>
- Flujo de trabajo para sitios responsivos. <http://msalazar.aiux.cl/flujo-de-trabajo-para-sitios-responsivos/>. (Original: <http://viljamis.com/blog/2012/responsive-workflow/>.)
- Sitios Web Optimizados para móviles. <https://developers.google.com/webmasters/mobile-sites/>

### BIBLIOGRAFÍA

- BONILLA, P.C., Diseño Web Adaptativo (Responsive Web Design). Madrid. Anaya Multimedia 2014.
- NIEDERST ROBBINS, J.; Learning Web Design. O'Reilly, 2012.
- CÓRCOLES TENDERO, J.E.; MONTERO SIMARRO, F. Diseño de Interfaces. Madrid: Ra-Ma, 2012. p. 43-93.