

LMSGI > UT02 > Apuntes > Anexo III

Tema 2. Anexo 3. Maquetación Web

Actualizado al 11/10/2022

Este anexo complementa la información aportada por los contenidos, ya que ésta no es suficiente para abordar con garantías la tarea de plantear una web desde cero.

Además, hay un par de etiquetas que no han quedado suficientemente explicadas. Son las siguientes:

- **<div>** Establece una capa, generalmente interpretada como un elemento contenedor.
- **** Delimita una región a la cual se le desea aplicar un estilo.

De las dos, es **<div>** el elemento que más nos va a ayudar en nuestra labor de maquetación, ya que será a base de “capas” como demos estructura visual a nuestra web.

****, por su parte, nos ayudará a hacer excepciones al formato general que se haya aplicado a una parte de la web. Su uso es más ocasional y no suele implicar cambios en la estructura.

Pero vayamos a la cuestión. Se trata de plantear una web desde cero. Bien, para ello vamos a usar la web que podemos visitar en <http://www.km77.com/>. Su aspecto es el siguiente:



Gama Civic por 140€ al mes.
 Entrada: 5.089,30€. 36 cuotas. Cuota final: 6.469€. TAE: 8,38%
Descúbrelo




km77.com

comparador de seguros buscador de imágenes buscador de precios coches de segunda mano

precios y datos informaciones y pruebas imágenes blogs seguros otras secciones







Volkswagen Beetle Dune 2016
 El Volkswagen Beetle Dune es una variante del Beetle con 10 milímetros más de altura libre al suelo y algunos elementos decorativos específicos. Estará a la venta a principios de 2016 con carrocería cerrada y descapotable.
 información - imágenes exteriores - imágenes interiores



Porsche 911 Carrera
 Tiene motores boxer turboalimentados de 370 y 420 caballos que dan mucho empuje a medio régimen y gastan menos carburante que los del anterior Porsche 911. También hay cambios en el chasis (como un sistema de dirección en las cuatro ruedas), en el equipamiento y en el aspecto de la carrocería. El 911 Carrera más barato cuesta 107 653 euros, más que sus principales rivales.
 primeras impresiones - exteriores - interiores - precio



HONDA
 The Power of Dreams
Gama Civic por 140€ al mes.
 Entrada: 5.089,30€. 36 cuotas. Cuota final: 6.469€. TAE: 8,38%
Descúbrelo >>

Redacción (34) 91 724 05 70 - Publicidad (34) 91 005 21 04 - © copyright 1999-2015 Ruedas de Prensa S.L. - [Notificar errores](#) - [Condiciones legales](#) - [Mapa web](#) - [Seguros](#)

USO DE COOKIES. Utilizamos cookies propias y de terceros para facilitar la navegación por nuestra web, así como para mejorar nuestros servicios y mostrarte la publicidad relacionada con tus preferencias mediante el análisis de los hábitos de navegación. Si continúas navegando, consideramos que aceptas su uso. Puedes obtener más información, o bien conocer cómo cambiar la configuración, en nuestra [Política de cookies](#)

La página incorpora imágenes y textos. Muchos de los elementos se comportan como enlaces.

El primer paso para llevar a cabo la maquetación es identificar la estructura de contenedores que posee la página. Esta estructura es jerárquica, de forma que comenzamos por un contenedor raíz (que es la propia página) y vamos identificando “ramas” de esa raíz, que a su vez se comportan como contenedores “padre” para otros sub-contenedores.

En la ilustración de la izquierda vemos el contenido de toda la página. En la ilustración de la derecha vemos el primer nivel de descomposición. Ahora lo explicaremos:



La franja roja se corresponde con el primer contenedor analizado. Es lo que conocemos como “cabecera”. Muestra una parte “fija” que siempre aparece en la zona superior de la web.

¿Por qué hemos puesto ahí la división? Porque, como decimos, el contenido de la cabecera siempre es el mismo (siempre hay un anuncio, siempre los mismos menús, logotipos, enlaces a redes sociales, etc.).

En este caso, la cabecera es bastante compleja. No hay problema en ello, pues después la volveremos a subdividir. Recordemos: en informática “divide y vencerás” es una buena praxis.

La franja azul es la que se conoce como el “pie”. También se corresponde con un elemento fijo que, al contrario que la cabecera, aparece en la parte inferior de la web.

Lo que queda en medio es una parte, que hemos coloreado en verde, que se corresponde con el contenido más dinámico de la web. Es donde van apareciendo los artículos y de donde éstos van desapareciendo, tras un tiempo de, digamos, “caducidad”. Por simplicidad, hemos puesto sólo dos artículos (uno referido a cada coche), aunque en realidad la web suele mostrar muchos más. Podemos llamar a esta sección “contenido”.

En la parte derecha del contenido hay un anuncio y, en la web real, pueden verse otras secciones. En realidad, podríamos haber planteado una división de primer nivel con la forma que hay en la ilustración de la derecha, pero



por simplicidad es mejor realizar las divisiones sólo en una orientación y, para este primer ejemplo, hemos elegido la orientación vertical. Después, si es necesario, dividiremos horizontalmente el “contenido” (franja verde).

En general, se recomienda al alumno realizar las divisiones siempre en el mismo sentido (dentro de un mismo contenedor). Esto facilitará aplicar “Flex-Positioning”, como vamos a ver.

Vamos ahora a crear la primera versión de la web. Para ello crearemos dos archivos: webcoches.html y webcoches.css, cada uno con lo que debe contener. Comenzamos por el primero:

```
<!DOCTYPE html>
<html lang="es-ES">
  <head>
    <meta charset="UTF-8" />
    <title>Web Coches</title>
    <link rel="stylesheet" type="text/css" href="webcoches.css" />
  </head>
  <body>
  </body>
</html>
```

Se entiende que “webcoches.css” está vacío o bien no existe (ahora mismo). En todo caso, ya queda enlazado para cuando tenga algo que decir.

Bien, comienzan a entrar en juego los <div>s. Lo primero es entender que <body> no tiene por qué representar a la página completa. Esto es, se recomienda usar un contenedor <div> que se comporte como elemento raíz de la página. Para identificar correctamente los contenedores, vamos a usar tres trucos que luego puede que quitemos (o no), son:

- Identificar el contenedor con un atributo “id” cuyo valor sea único.
- Poner como contenido un texto a modo de título, para verlo luego en el navegador.
- Darle un color al fondo, a través de una regla en el CSS, para ver qué área abarca.

Para los dos primeros nos basta el código HTML, para lo tercero usamos CSS. El HTML queda:

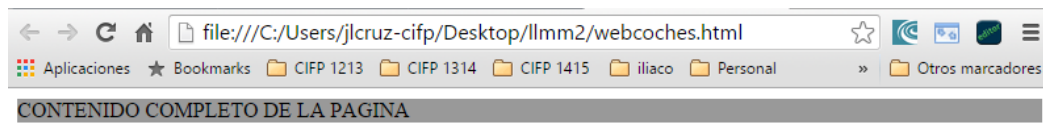
```
<!DOCTYPE html>
<html lang="es-ES">
  <head>
    <meta charset="UTF-8" />
    <title>Web Coches</title>
    <link rel="stylesheet" type="text/css" href="webcoches.css" />
  </head>
  <body>
    <div id="pagina">
      CONTENIDO COMPLETO DE LA PAGINA
    </div>
  </body>
</html>
```

Necesitamos escribir lo de “CONTENIDO COMPLETO DE LA PAGINA” porque, de lo contrario, la capa no ocuparía espacio alguno y, por tanto, no podríamos prever dónde aparece. No obstante, **todos estos rótulos que iremos poniendo (en mayúscula) son provisionales.**

El último punto se concreta en que en el CSS aplicamos una regla a nivel de identificador (con el selector #) para cambiar el color de fondo. En nuestro ejemplo lo vamos a hacer gris:

```
#pagina
{
    background-color: #999;
}
```

El resultado, en Chrome, se vería así:



Es decir, la página sólo ocupa una línea, con el texto indicado entre <div> y </div>, que es la única capa que hay. Hemos aplicado a la capa una regla de estilo que establece el fondo en gris. El resto del formato viene heredado del comportamiento predeterminado del navegador.

Vamos a introducir las 3 capas que hemos identificado en nuestro primer nivel de división jerárquico. Siguiendo el mismo proceso, pensaríamos en 3 identificadores y usaríamos los mismos 3 colores que he puesto en las capturas (para aclararnos).

No obstante, aquí hay que hacer un inciso: HTML 5 tiene algunas etiquetas semánticas para nombrar a capas que tienen un especial significado. Particularmente, sugiere llamar <header> (y no simplemente <div>) a los encabezados, <footer> a la capa que haga de “pie”, etc. Así que **vamos a dejar como <div> la central, pero vamos a usar <header> y <footer> para la cabecera y el pie.** El HTML quedaría:

```
<!DOCTYPE html>
<html lang="es-ES">
  <head>
    <meta charset="UTF-8" />
    <title>Web Coches</title>
    <link rel="stylesheet" type="te
  </head>
  <body>
    <div id="pagina">
      CONTENIDO COMPLETO DE LA PAGINA
      <header>
        CABECERA
      </header>
      <div id="cuerpo">
        CUERPO
      </div>
      <footer>
        PIE
      </footer>
    </div>
  </body>
</html>
```

```
#pagina
{
    background-color: #999;
}

header
{
    background-color: #F99;
}

#cuerpo
{
    background-color: #9F9;
}

footer
{
    background-color: #99F;
}
```

Vemos que los selectores de `<header>` y `<footer>` son diferentes, ya que no se seleccionan por “id”, sino por la propia etiqueta en sí (por tanto, no requieren “#” ni “.”). El resultado visible es:



Bien, ya nos vamos acercando. El comportamiento de `<div>` o cualquier otro contenedor (`<header>`, `<footer>`...) es a nivel de bloque. Por eso aparece uno encima del otro. En el caso de una división vertical (como esta primera que hemos hecho), esto es justamente lo que queremos, así que no hace falta indicar nada más.

Vamos ahora a resolver algo más sobre la cabecera. Para ello hay que subdividirla. Si nos fijamos, podemos encontrar varias secciones dispuestas también en vertical. Serían estas:



Hemos identificado 4 “bandas”, a saber: anuncio (amarillo), logo + menú a webs externas (magenta), menú interno (azul) e iconos de redes sociales (marrón). Ya sabemos cómo hacer esto en HTML, sólo hay que puntualizar algunas cosas:

- Debemos hacerlo dentro del elemento `<header>`, para conservar la estructura que estamos identificando.
- Al menos el menú interno es procedente etiquetarlo como `<nav>`, según HTML5.

Manos a la obra, desarrollaremos estos cambios en el HTML y en el CSS:

```
<header>
CABECERA
  <div id="anuncio_top">ANUNCIO SUPERIOR</div>
  <div id="links_externos">VINCULOS EXTERNOS</div>
  <nav>MENU DE LA WEB</nav>
  <div id="links_social">VINCULOS A REDES</div>
</header>

#anuncio_top
{
  background-color: #FF9;
}

#links_externos
{
  background-color: #F9F;
}

nav
{
  background-color: #9FF;
}

#links_social
{
  background-color: #BA8;
}
```

Y el resultado final, aunque dista mucho de asemejarse a lo que queremos, creedme que posee la estructura correcta para continuar. Vamos por aquí:

CONTENIDO COMPLETO DE LA PAGINA
CABECERA
ANUNCIO SUPERIOR
VINCULOS EXTERNOS
MENU DE LA WEB
VINCULOS A REDES
CUERPO
PIE

Ya tenemos un contenedor casi listo. Es el del anuncio superior, pues basta con cambiar "ANUNCIO SUPERIOR" por el vínculo a la imagen. Las líneas del HTML pueden quedar así:

```
<div id="anuncio_top">
  </img>
</div>
```

Grabando y actualizando la página, la imagen aparecerá en lugar de "ANUNCIO SUPERIOR".

OJO: el fichero "civic_980x180.gif" debe estar grabado en la misma carpeta que la página HTML, de lo contrario habría que usar una ruta relativa o absoluta hasta su ubicación.

Otra cosa que podemos ir resolviendo es la franja con el menú interno. La de color azul. Vemos que se compone de vínculos. Podemos crear estos vínculos con la etiqueta <a>. De momento, vamos a inventarnos los enlaces. El caso es que podrían quedar así en el código HTML:

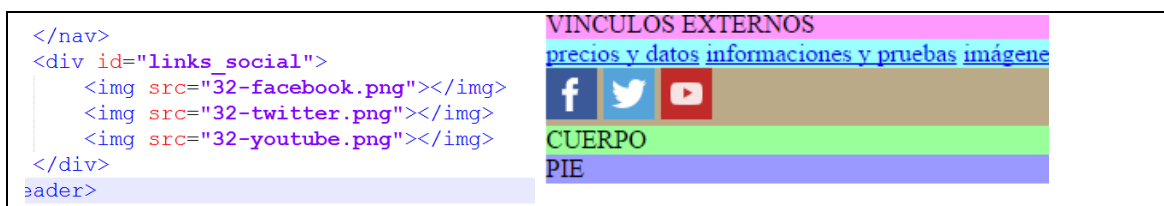
```
<div id="links_externos">VINCULOS EXTERNOS</div>
<nav>
  <a href="webcoches.com/opc1">precios y datos</a>
  <a href="webcoches.com/opc2">informaciones y pruebas</a>
  <a href="webcoches.com/opc3">imágenes</a>
  <a href="webcoches.com/opc4">blogs</a>
  <a href="webcoches.com/opc5">seguros</a>
  <a href="webcoches.com/opc6">otras secciones</a>
</nav>
<div id="links_social">VINCULOS A REDES</div>
```

Y el resultado final, en el que se ve también el anuncio superior ya enlazado:



Vemos que los enlaces son elementos “en línea”, por lo que por defecto ya aparecen uno a continuación de otro, en orden de lectura. Así es como los queremos, luego, perfecto.

Del mismo modo, podemos atajar la banda dedicada a los vínculos a redes. Con el código:



Por supuesto, hemos omitido que los iconos, en realidad, son vínculos (deberían estar encerrados, a su vez, por etiquetas <img.../img>, indicando a donde vincularía un clic sobre ellos. Pero el cometido de este tutorial es hacer un ejemplo de maquetación, principalmente, así que por simplicidad omitimos este aspecto.

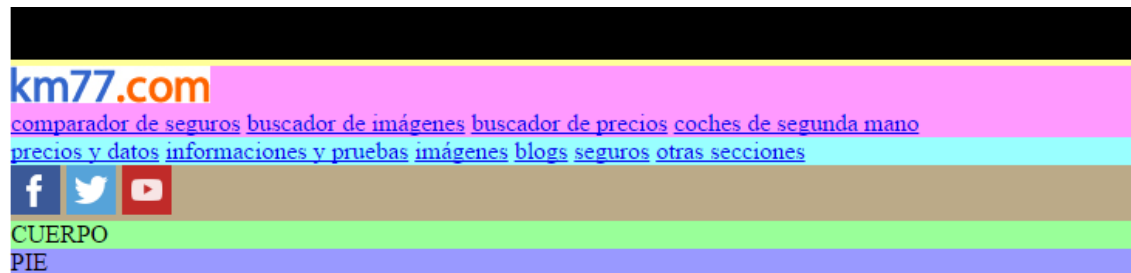
De la cabecera, sólo nos queda la capa resaltada en magenta. En esta banda hay dos cosas: los vínculos externos (que quedan alineados a la derecha) y el logotipo (que queda alineado a la izquierda). Al ser uno solo, el logotipo podemos agregarlo tal cual pero, si queremos hacer un

bloque con los vínculos externos, deberíamos habilitar un contenedor para ellos, esto es, un <div> propio que represente a todo el conjunto de opciones de este menú de enlaces.

Ya no vamos a usar más colores, estamos muy cerca de la solución final. Además, estamos repitiendo el mismo planteamiento de antes. Veamos cómo plantear lo dicho en el HTML:

```
</div>
<div id="links_externos">
  </img>
  <div id="menu_externo">
    <a href="webexterna1.com">comparador de seguros</a>
    <a href="webexterna2.com">buscador de imágenes</a>
    <a href="webexterna3.com">buscador de precios</a>
    <a href="webexterna4.com">coches de segunda mano</a>
  </div>
</div>
<nav>
```

Es decir, dentro de “links_externos” hemos indicado dos elementos: la imagen y el menú. A su vez, dentro del menú (que hemos identificado como “menu_externo”), hemos creado las cuatro opciones mediante cuatro vínculos. El resultado final es:



La cabecera tiene muchas carencias aún. Como nos estamos centrando en la estructura, vamos a resaltar sólo las que tienen que ver con la maquetación, pero no hablamos aún del formato:

1. El logo de km77.com y el “menú externo” han de salir al mismo nivel, alineándose el segundo a la derecha.
2. El menú de redes sociales ha de alinearse a la derecha, igual que el menú externo.
3. El menú de vínculos internos debe ocupar todo el ancho.

Vamos a poner solución a esto por partes:

Para la primera carencia, debemos conseguir que la del logotipo y el <div> del menú externo se comporten como elementos “en línea” y que, además, se alineen a ambos lados rellenando con espacios lo que haga falta. Para ello, entra en juego ya, por fin: Flex.

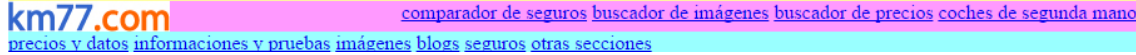
Mediante CSS, vamos a indicar al contenedor padre que se trata de un “Flex-container”. Eso lo hacemos con esta línea:

```
#links_externos
{
    background-color: #F9F;
    display: flex;
}
```



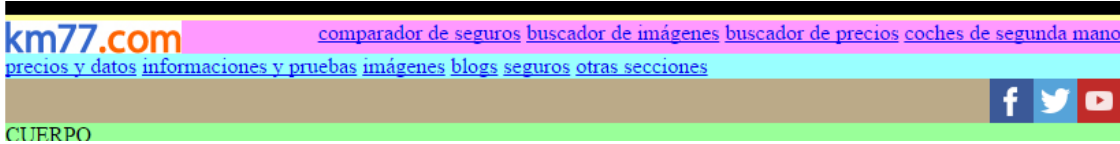
El resultado es instantáneo. “links_externos” pasa a ser un contenedor con visualización (display) de tipo “Flex”, lo que implica que, por defecto, estará compuesto de “Flex-items” que se dispondrán en orden de lectura. Por eso el logo y el menú ya se alinean uno junto a otro, de izquierda a derecha. Ahora hay que buscar un atributo de “Flex-Positioning” que permita distribuir los dos “Flex-items” de forma que ocupen todo el ancho, pero rellenando con espacios. Eso es justamente lo que hace esta pareja de atributo y valor:

```
#links_externos
{
    background-color: #F9F;
    display: flex;
    justify-content: space-between;
}
```



Ya lo tenemos. Vamos a aplicar algo parecido para los vínculos de las redes sociales:

```
#links_social
{
    background-color: #BA8;
    display: flex;
    justify-content: flex-end;
}
```



Como puede verse, el atributo “justify-content” indica cómo se justifican los flex-items dentro del contenedor para quien se indica dicho atributo. Los valores están explicados en el tutorial cuyo enlace se facilitó en el anexo anterior (anexo 2).

Nos queda resolver el menú interno, que debe ocupar todo el ancho. Siguiendo con “Flex-Positioning”, cabe la posibilidad de decirle al contenedor padre que su visualización es “Flex” y, después, decirle a los “ítems” que pueden crecer equitativamente hasta ocupar todo el ancho disponible, lo cual se consigue con el atributo “flex-grow”. Lo primero es fácil:

```
nav
{
    background-color: #9FF;
    display: flex;
}
```

Lo segundo implica cambiar, en principio uno a uno, todos los vínculos internos. En estos casos, lo procedente es crear una “clase” que los agrupe y, usando el selector que comienza por punto, crear una única regla para toda la clase. Cambiamos el HTML de esta forma:

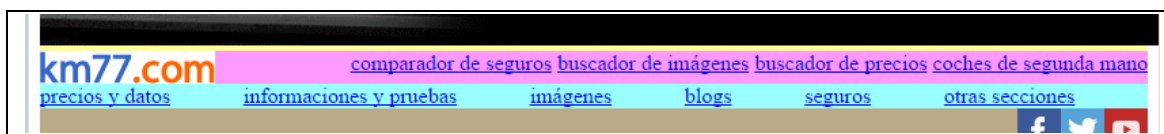
```
<nav>
  <a class="link_interno" href="webcoches.com/opc1">precios y datos</a>
  <a class="link_interno" href="webcoches.com/opc2">informaciones y pruebas</a>
  <a class="link_interno" href="webcoches.com/opc3">imágenes</a>
  <a class="link_interno" href="webcoches.com/opc4">blogs</a>
  <a class="link_interno" href="webcoches.com/opc5">seguros</a>
  <a class="link_interno" href="webcoches.com/opc6">otras secciones</a>
</nav>
```

Con los ítems ya “class”ificados como “link_interno”, creamos una única regla para esa clase:

```
.link_interno
{
    flex-grow: 1;
}
```

Nótese que la regla del CSS lleva un punto (.) como prefijo, indicando así que “link_interno” se refiere a una clase.

El resultado final muestra el menú interno repartíéndose todo el ancho disponible:



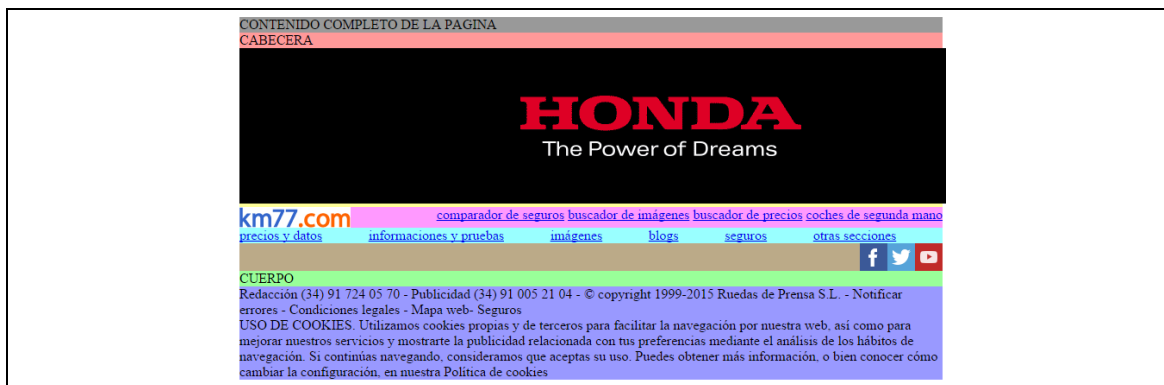
La estructura de la cabecera estaría ya terminada, quedando únicamente mejorar su estilo. Esto lo dejaremos para el final. De momento, proseguimos “estructurando” el pie.

De los tres contenedores del primer nivel de división, el pie de la página es el más sencillo. Únicamente se compone de dos partes claramente diferenciadas: la línea de “Redacción...” que termina con unos vínculos (pero que se ve fácilmente que están en línea) y el aviso legal del “USO DE COOKIES”, que tiene formato diferente y está claramente encerrado en una “caja” separada.

Ya sabemos cómo acometer esta tarea: dos <div>s que, por estar dispuestos verticalmente, no van a necesitar indicación adicional. Únicamente decir que los identificaremos para poder darles formato. Ya de paso, vamos a copiar y pegar lo que deben mostrar (omitiendo los vínculos, de momento). El código HTML quedaría así:

```
</div>
<footer>
  <div id="texto_redaccion">
    Redacción (34) 91 724 05 70 - Publicidad (34) 91 005 21 04 -
  </div>
  <div id="aviso_cookies">
    USO DE COOKIES. Utilizamos cookies propias y de terceros pa
  </div>
</footer>
</div>
```

De momento, mientras no vayamos aplicando formato, es difícil distinguir las capas a las que no les hemos dado un color de fondo exclusivo (como estas dos últimas). Lo que llevamos hasta ahora se muestra así:



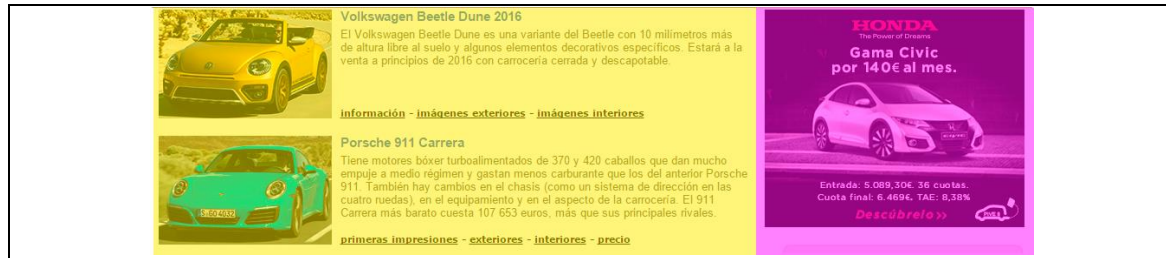
Pero veremos que, gracias a la estructuración que hemos hecho y haciendo uso de CSS, podemos cambiar rápidamente su apariencia. Queda poco, aún hace falta un pequeño esfuerzo de fe.

Antes vamos a terminar de dar estructura a la parte central (verde) que es la que, a priori, podría presentar mayor complejidad. De momento, sólo tenemos un <div id="cuerpo"...>.

Si nos fijamos en la web, ya vimos que este cuerpo central se compone de dos columnas básicas:

- La de la izquierda, la principal, con una secuencia de artículos (de arriba abajo), cada uno referido a un modelo de coche (en nuestro ejemplo aparecen dos artículos).
- La de la derecha, a modo de anexo lateral, en la que aparecerían anuncios y otras cuestiones "anexas" (también de arriba abajo).

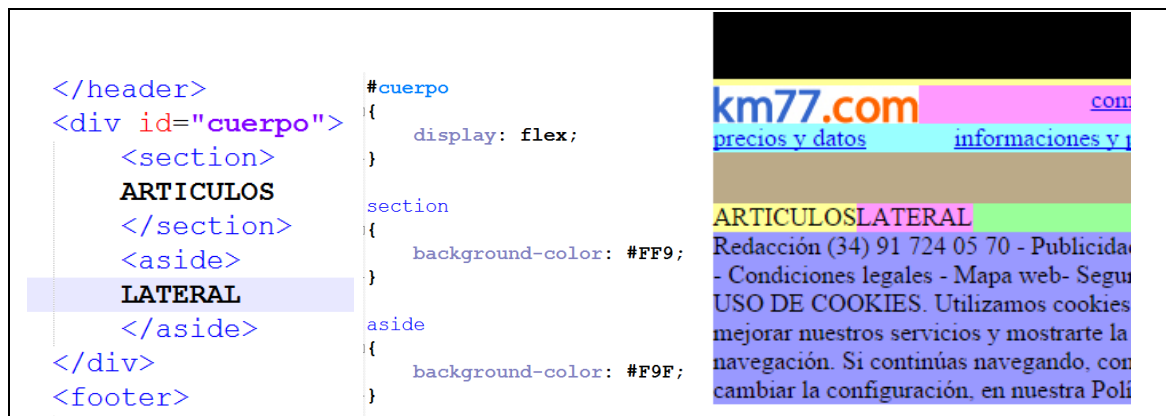
Por tanto, nuestra “descomposición” debe continuar por identificar estas dos partes y luego, dentro de cada una, las que sean necesarias. Volvemos a usar el amarillo y el magenta para identificarlas en esta captura:



Siempre que pensemos en una descomposición horizontal ayuda mucho echar mano de “Flex”, como ya hicimos para la cabecera, así que vamos a plantear una división de “cuerpo” en dos sub-capas: una para los artículos sobre coches (la amarilla) y otra para los anuncios y demás temas misceláneos del lateral derecho (la magenta).

HTML 5 nos recomienda usar <aside> para capas anexas laterales (como la capa derecha) y, por otra parte, <section> para agrupar artículos (y <article> para encerrar artículos, que es la forma en la que podríamos aislar cada uno de los artículos sobre coches que van apareciendo).

Con todo lo dicho (y espero que correctamente justificado), podríamos plantear este código HTML y estas reglas CSS, obteniendo el resultado que puede verse a la derecha:



Gracias a “display: flex”, las dos nuevas capas aparecen en línea. En principio, podríamos fijar una anchura para “LATERAL” y hacer crecer “ARTICULOS” hasta completar la anchura (consumir todo el área verde restante, empujando “LATERAL” hacia la derecha). Eso es muy fácil, completando de la siguiente forma estas dos reglas CSS:

```

section
{
    background-color: #FF9;
    flex-grow: 1;
}

aside
{
    background-color: #F9F;
    width : 300px;
}

```



Ahora vemos que “LATERAL” ocupa 300 píxeles y que “ARTICULOS” ha crecido todo lo posible.

Hablando de “LATERAL”, para nuestro ejemplo, está casi terminado. Sólo hay que incrustar en él una versión “más estrecha” del anuncio. Lo conseguimos con el siguiente código HTML:

```

<div id="cuerpo">
    <section>
        ARTICULOS
    </section>
    <aside>
        
    </aside>
</div>

```



Nótese que una de las ventajas del “truco” de poner un color de fondo a cada capa es que, visualmente, hemos podido comprobar cómo al aumentar de altura la capa “LATERAL” lo ha hecho también la capa “ARTICULOS” (se ve por el fondo amarillo). Este es el comportamiento por defecto del módulo “Flex” para capas hermanas (flex-items) del mismo flex-container. El tutorial del anexo 2 explica cómo cambiar este comportamiento pero, para nuestro ejemplo (y para la mayoría de situaciones), es justamente lo que necesitamos.

Nos queda que modelar cada artículo referido a cada coche. En realidad, haremos un único diseño y lo repetiremos, entendiendo que “ARTICULOS” es un contenedor al que puedo agregar otras capas con visualización por bloques (la por defecto, es decir, que no tengo que hacer nada especial sobre “ARTICULOS”).



Volkswagen Beetle Dune 2016

El Volkswagen Beetle Dune es una variante del Beetle con 10 milímetros más de altura libre al suelo y algunos elementos decorativos específicos. Estará a la venta a principios de 2016 con carrocería cerrada y descapotable.

[información](#) - [imágenes exteriores](#) - [imágenes interiores](#)

Viendo la captura anterior, y teniendo en cuenta lo que hemos dicho de intentar realizar las divisiones sólo en sentido horizontal o vertical (pero no mixto), el diseño de un artículo se compone de dos partes: una imagen (azul celeste) seguida de una descripción (marrón). Eso puede representarse en el código HTML y CSS así (ahora más rápidos, que es igual que antes):

```


<section>
  <article>
    
    <div>
      DESCRIPCION PRIMER COCHE
    </div>
  </article>
</section>

```

```

article
{
  display: flex;
}

```



Una vez hecha esa descomposición horizontal, vemos que la descripción del vehículo se compone, a su vez, de tres partes: Un titular, un texto descriptivo y unos enlaces. Como están en vertical, no requieren visualización especial, nos basta con completar el HTML así:

```

<article>
  
  <div>
    <h3>
      Volkswagen Beetle Dune 2016
    </h3>
    <div>
      El Volkswagen Beetle Dune es una variante del Beetle con 10 milí
    </div>
    <div>
      <a href="webcoches.com/info_coche1">información</a>
      <a href="webcoches.com/info_coche1">imágenes exteriores</a>
      <a href="webcoches.com/info_coche1">imágenes interiores</a>
    </div>
  </div>
</article>

```

Lo cual nos va dejando ya un resultado bastante completo:

CONTENIDO COMPLETO DE LA PAGINA

CABECERA



km77.com

comparador de seguros buscador de imágenes buscador de precios coches de segunda mano

precios y datos informaciones y pruebas imágenes blogs seguros otras secciones

Volkswagen Beetle Dune 2016

El Volkswagen Beetle Dune es una variante del Beetle con 10 milímetros más de altura libre al suelo y algunos elementos decorativos específicos. Estará a la venta a principios de 2016 con carrocería cerrada y descapotable.

[información](#) [imágenes exteriores](#) [imágenes interiores](#)

HONDA

Lo quieres, lo quieres... No lo pienses más.

Redacción (34) 91 724 05 70 - Publicidad (34) 91 005 21 04 - © copyright 1999-2015 Ruedas de Prensa S.L. - Notificar errores - Condiciones legales - Mapa web- Seguros

USO DE COOKIES. Utilizamos cookies propias y de terceros para facilitar la navegación por nuestra web, así como para mejorar nuestros servicios y mostrarte la publicidad relacionada con tus preferencias mediante el análisis de los hábitos de navegación. Si continúas navegando, consideramos que aceptas su uso. Puedes obtener más información, o bien conocer cómo cambiar la configuración, en nuestra Política de cookies

Si ahora repetimos el código HTML y lo adaptamos para el segundo artículo, la cosa quedaría:

CONTENIDO COMPLETO DE LA PAGINA
CABECERA



Gama Civic por 140€ al mes.
Entrada: 5.089,30€. 36 cuotas. Cuota final: 6.469€. TAE: 8,38%

HONDA
The Power of Dreams

[km77.com](#) [comparador de seguros](#) [buscador de imágenes](#) [buscador de precios](#) [coches de segunda mano](#)
[precios y datos](#) [informaciones y pruebas](#) [imágenes](#) [blogs](#) [seguros](#) [otras secciones](#)

[f](#) [t](#) [y](#)



Volkswagen Beetle Dune 2016
El Volkswagen Beetle Dune es una variante del Beetle con 10 milímetros más de altura libre al suelo y algunos elementos decorativos específicos. Estará a la venta a principios de 2016 con carrocería cerrada y descapotable.
[información](#) [imágenes exteriores](#) [imágenes interiores](#)



Porsche 911 Carrera
Tiene motores boxer turboalimentados de 370 y 420 caballos que dan mucho empuje a medio régimen y gastan menos carburante que los del anterior Porsche 911. También hay cambios en el chasis (como un sistema de dirección en las cuatro ruedas), en el equipamiento y en el aspecto de la carrocería. El 911 Carrera más barato cuesta 107 653 euros, más que sus principales rivales.
[información](#) [imágenes exteriores](#) [imágenes interiores](#)



HONDA
The Power of Dreams
Gama Civic por 140€ al mes.
Entrada: 5.089,30€. 36 cuotas. Cuota final: 6.469€. TAE: 8,38%

Redacción (34) 91 724 05 70 - Publicidad (34) 91 005 21 04 - © copyright 1999-2015 Ruedas de Prensa S.L. - Notificar errores - Condiciones legales - Mapa web- Seguros
USO DE COOKIES. Utilizamos cookies propias y de terceros para facilitar la navegación por nuestra web, así como para mejorar nuestros servicios y mostrarte la publicidad relacionada con tus preferencias mediante el análisis de los hábitos de navegación. Si continúas navegando, consideramos que aceptas su uso. Puedes obtener más información, o bien conocer cómo cambiar la configuración, en nuestra Política de cookies

Y hasta aquí lo necesario para crear la maquetación de la web. A partir de este momento, prácticamente con usar CSS (y poco más) podemos terminar de darle su aspecto definitivo.

Lo primero que haríamos sería liberar la página de los colores de fondo y rótulos artificialmente añadidos. Tras hacerlo, el código HTML y CSS quedaría así:



HONDA
The Power of Dreams

[km77.com](#) [comparador de seguros](#) [buscador de imágenes](#) [buscador de precios](#) [coches de segunda mano](#)
[precios y datos](#) [informaciones y pruebas](#) [imágenes](#) [blogs](#) [seguros](#) [otras secciones](#)

[f](#) [t](#) [y](#)



Volkswagen Beetle Dune 2016
El Volkswagen Beetle Dune es una variante del Beetle con 10 milímetros más de altura libre al suelo y algunos elementos decorativos específicos. Estará a la venta a principios de 2016 con carrocería cerrada y descapotable.
[información](#) [imágenes exteriores](#) [imágenes interiores](#)



Porsche 911 Carrera
Tiene motores boxer turboalimentados de 370 y 420 caballos que dan mucho empuje a medio régimen y gastan menos carburante que los del anterior Porsche 911. También hay cambios en el chasis (como un sistema de dirección en las cuatro ruedas), en el equipamiento y en el aspecto de la carrocería. El 911 Carrera más barato cuesta 107 653 euros, más que sus principales rivales.
[información](#) [imágenes exteriores](#) [imágenes interiores](#)



HONDA
The Power of Dreams

Redacción (34) 91 724 05 70 - Publicidad (34) 91 005 21 04 - © copyright 1999-2015 Ruedas de Prensa S.L. - Notificar errores - Condiciones legales - Mapa web- Seguros
USO DE COOKIES. Utilizamos cookies propias y de terceros para facilitar la navegación por nuestra web, así como para mejorar nuestros servicios y mostrarte la publicidad relacionada con tus preferencias mediante el análisis de los hábitos de navegación. Si continúas navegando, consideramos que aceptas su uso. Puedes obtener más información, o bien conocer cómo cambiar la configuración, en nuestra Política de cookies

Ciertamente mejorable. Bien, lo primero sería entender que la página tiene una anchura fija (en este caso), que debe corresponderse con la del banner (anuncio) superior. Eso lo conseguiremos mediante CSS, con el atributo *width* del contenedor que se refiere a toda la composición, esto es:

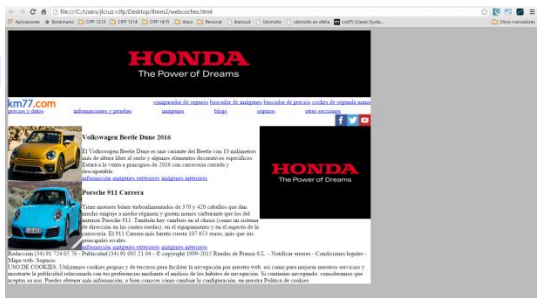
```
#pagina
{
    width: 980px;
}
```



Para ver el “corte” entre la página y el fondo, podríamos usar diferentes colores de fondo para cada cosa. Por ejemplo, poner el fondo en un gris claro y la página, forzarla a que sea blanca:

```
body
{
    background-color: #BBB;
}

#pagina
{
    width: 980px;
    background-color: white;
}
```



Si deseamos centrarla, un recurso fácil es establecer los márgenes de “#pagina” como automáticos, tanto por la izquierda como por la derecha:

```
#pagina
{
    width: 980px;
    background-color: white;
    margin-left: auto;
    margin-right: auto;
}
```



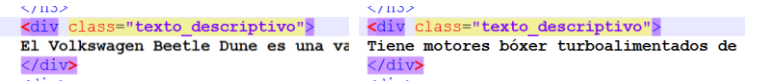
Otro aspecto general es la tipografía. La página original usa la fuente “Verdana” para títulos y “Arial” para descripciones. Como la primera parece la más extendida, vamos a proponerla como de “uso por defecto” en toda la página, esto es, como atributo de la misma regla CSS:

```
#pagina
{
    width: 980px;
    background-color: white;
    margin-left: auto;
    margin-right: auto;
    font-family: Verdana;
}
```



Veis que el aspecto va cobrando semejanza con la original. Si queremos dejar como “Arial” las descripciones de cada coche, lo precedente sería crear una clase (class), ya que puede haber varias apariciones de las mismas. Nos viene bien hacerlo así porque, además, el tipo de letra parece más pequeño del estándar, cosa que ajustaremos en la misma regla para la nueva clase, a la que llamaremos “texto_descriptivo”. Los cambios en el CSS y en el HTML son:

```
.texto_descriptivo
{
    font-family: Arial;
    font-size: small;
}
```



Abandonamos un poco la tipografía para ir a otras cosas que “abultan” más:

Yendo de arriba abajo, vemos que el logo de “Km77.com” y el menú externo, están demasiado pegados al banner superior. Podemos decirle al banner que guarde un margen por la parte inferior con el resto de los elementos. Generalmente una unidad de “em” (es decir, el tamaño de un carácter medio) suele funcionar bien. Sólo hay que tocar el CSS para indicar la regla:

```
#anuncio_top
{
    margin-bottom: 1em;
}
```



El menú de enlaces externos se compone de botones con un degradado gris vertical y letras azules, que no están subrayadas. Una vez más, para poder aplicar el mismo estilo a todos a la vez hay que agruparlos en una clase (por ejemplo class=“link_externo”).

```
<div id="menu_externo">
  <a class="link_externo" href="web">
  <a class="link_externo" href="web">
  <a class="link_externo" href="web">
  <a class="link_externo" href="web">
</div>
```

Los dos efectos visuales se pueden conseguir así:

```
.link_externo
{
    background-image: linear-gradient(to bottom, #DDDDDD 0%, #BBBBBB 100%);
    text-decoration: none;
}
```

El fondo degradado, como puede verse, se basa en un “fondo con imagen” pero para el que se ha usado una función de CSS3 especial “*linear-gradient()*”, que debe ser soportada por el navegador y cuya sintaxis no ha sido explicada en el curso. En todo caso, existen herramientas online que nos ayudan a conseguir el código CSS necesario para generar un degradado. Por ejemplo, podéis visitar la url: <http://www.css3factory.com/linear-gradients/> y jugar con los controles que se os ofrece para obtener el código CSS más abajo.

En cualquier caso, aún no conseguimos el efecto final, pues los botones salen con el rótulo muy pegado a sus bordes, el tamaño de fuente muy grande y la tipografía demasiado delgada. La solución a esto es:

- Abrir hueco entre el rótulo y los bordes: jugando con el “padding”, que para eso está.
- Reducir bastante el tamaño de fuente: usando un tamaño absoluto, como “x-small”.
- Engrosar la tipografía: estableciendo un valor alto para el atributo “text-weight”.

El resultado en el CSS es éste (y al lado la vista previa de cómo queda):

```
.link_externo
{
    background-image: linear-gradient(to bottom, #DDDDDD 0%, #BBBBBB 100%);
    text-decoration: none;
    padding: 0.5em 2em 0.5em 2em;
    font-size: x-small;
    font-weight: bold;
}
```



Con el menú de vínculos internos pasa algo parecido, hay que tocar varios ajustes:

```
.link_interno
{
    flex-grow: 1;
    background-image: linear-gradient(to bottom, #00CCFF 0%, #0044FF 100%);
    padding: 0.5em 2em 0.5em 2em;
    margin: 1px 1px 1px 1px;
    text-decoration: none;
    color: white;
    font-weight: bold;
    font-size: small;
    text-align: center;
}
```

En adición a lo que hicimos con el menú externo, ahora ha hecho falta:

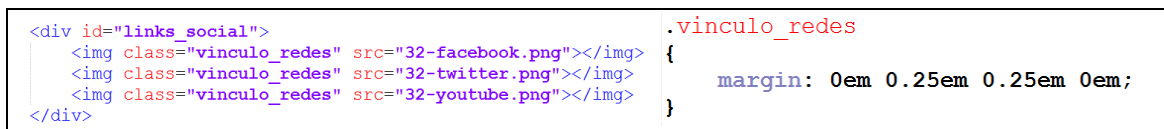
- Establecer un fino margen (margin) de un píxel, para separar las opciones.
- Alinear los rótulos (text-align) al centro, pues su “caja” crecía dejándolos a la izquierda.

El resto de la configuración ya se ha explicado, pues es semejante a la del menú externo.

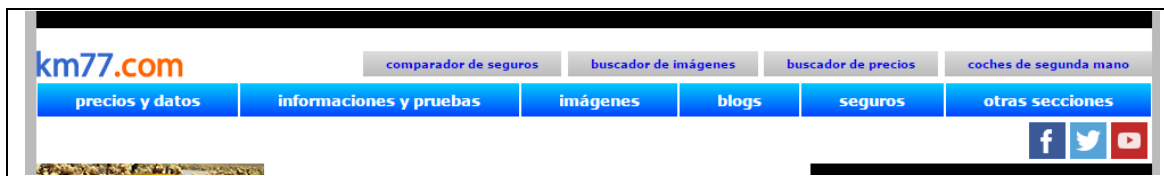
Si en este menú establecemos un margen superior e inferior de un cuarto de carácter (0.25em) el resultado final es bastante cercano al de la web original:



También deberíamos aplicar ese mismo margen sobre los iconos de redes sociales:



Si bien en esta ocasión sólo lo damos por la derecha y por abajo, para evitar que se sumen los de dos iconos adyacentes (y que se sumen con el del menú que hay justo arriba). El resultado:



Tan cercano ya al de la web original que damos la cabecera por zanjada.

En lo referente al pie, hablábamos de dos partes diferenciadas. Cualquiera de las dos tiene un tipo de letra muy pequeño pero, además, la segunda queda “encuadrada” en un fondo gris. Veamos como hacerlo. Por lo pronto, aplicaremos estas dos cualidades:



Vemos algunas carencias:

- Falta un margen que separe el pie de los artículos.
- En la web original hay una franja azul clara que separa los artículos del pie.
- Falta un “padding” que evite que los textos se acerquen demasiado a su borde.

Las instrucciones CSS que corrigen estas carencias son:

```
#texto_redaccion
{
    font-size: x-small;
    border-top-color: #77AAFF;
    border-top-width: 0.5em;
    border-top-style: solid;
    margin-top: 1em;
    padding: 1em 1em 1em 1em;
}

#aviso_cookies
{
    font-size: x-small;
    background-color: #CCCCCC;
    padding: 1em 1em 1em 1em;
}
```

Como puede verse, aplicar un “padding” de “1em” funciona bastante bien para evitar que un texto quede “pegado” a su borde. Por ello, vamos a aplicárselo también al interior de artículos. El problema es que, en realidad, no queremos aplicarlo a <article>s, sino a la capa que hay dentro de cada uno, la de la derecha, que es la que describe el artículo. Si lo aplicáramos al <article>, incurriríamos en dos defectos:

- Le daríamos un espacio innecesario a las imágenes de los coches, por su parte superior, izquierda e inferior.
- No evitaríamos la separación entre dichas imágenes y el texto (y hay que evitarla).

Las capas que almacenan la descripción de cada coche no están clasificadas ni identificadas. Vamos a clasificarlas, por ejemplo, mediante class=“describe_articulo”:

```
<article>
  
  <div class="describe_articulo">
    <h3>
```

```
<article>
  
  <div class="describe_articulo">
    <h3>
```

Para poder así aplicar la regla para ambas capas:

```
.describe_articulo
{
    padding-left: 1em;
    padding-right: 1em;
}
```


El texto ya no se “pega” a sus paredes laterales, pero persisten (o se agravan) otros problemas:

- La imagen se “estira” verticalmente porque el texto ocupa demasiado y esto, comparando nuestro diseño con la web original, es porque hay demasiado espaciado por encima y por debajo del título del artículo (p.ej. “Volkswagen Beetle...”).
- No hay separación entre dos artículos consecutivos.

El segundo problema es el más sencillo. Su solución es darle un margen inferior a los `<article>`s. Si lo hacemos de “1em”, cumpliremos la proporción que estamos siguiendo:

```

article
{
    display: flex;
    margin-bottom: 1em;
}

```



más de altura libre al suelo y algunos elementos decorativos específicos. Estará a la venta a principios de 2016 con carrocería cerrada y descapotable.
[información](#) [imágenes exteriores](#)



Porsche 911 Carrera
Tiene motores boxer turboalimentados de 370 cv. empuje a medio régimen y gastan menos carb.

El primer problema se basa en que decidimos usar `<h3>` para ese titular de artículo. No es mala opción, pues `<h3>` representa un título de nivel intermedio, como cabría esperar de estos titulares. Sin embargo, no hay que perder de vista que estos encabezados poseen un formato por defecto y, por lo que se ve, dicho formato incorpora un margen superior e inferior nada despreciable. Nosotros podemos sobre-escribir dicho comportamiento en nuestro CSS. No hay problema en ello, de hecho, vamos a retocar también el color de fuente y tamaño de un `<h3>`:

```

h3
{
    margin-top: 0em;
    margin-bottom: 0.5em;
    color: blue;
    font-size: small;
}

```

precios y datos | informaciones y pruebas | imágenes | blogs



Volkswagen Beetle Dune 2016
El Volkswagen Beetle Dune es una variante del Beetle con 10 milímetros más de altura libre al suelo y algunos elementos decorativos específicos. Estará a la venta a principios de 2016 con carrocería cerrada y descapotable.
[información](#) [imágenes exteriores](#) [imágenes interiores](#)

Va quedando muy parecido, pero se siguen apreciando dos defectos:

- Los vínculos inferiores son muy grandes. Además deberían ser negros y en negrita.
- Dichos vínculos quedan pegados a la descripción y, en la web original, se quedan alineados a la parte inferior del artículo.

La primera solución es fácil, basta con clasificar dichos vínculos (por ejemplo, como `class="link_articulo"`) y luego cambiar el formato de fuente en el CSS para esa clase:

```

<div class="texto_descriptivo">
El Volkswagen Beetle Dune es una variante
</div>
<div>
<a class="link_articulo" href="webcoo">
<a class="link_articulo" href="webcoo">
<a class="link_articulo" href="webcoo">
</div>
iv>

```

km77.com

comparador de seguros | buscador de imágenes

precios y datos | informaciones y pruebas | imágenes | blogs



Volkswagen Beetle Dune 2016
El Volkswagen Beetle Dune es una variante del Beetle con 10 milímetros más de altura libre al suelo y algunos elementos decorativos específicos. Estará a la venta a principios de 2016 con carrocería cerrada y descapotable.
[información](#) **[imágenes exteriores](#)** **[imágenes interiores](#)**

Ya está. La segunda no es tan fácil, puesto que hay que empujar al contenedor de esos vínculos hasta abajo.

Antes decíamos que, para una disposición vertical de contenedores dentro de un padre común, no necesitábamos “Flex”. Bien, pues ahora sí, porque además de esa disposición necesitamos hacer crecer al elemento central (el de la descripción) todo lo que pueda. Lo de que crezca no es ningún misterio, se consigue con “flex-grow” igual que ya hicimos para los “flex-items” del menú interno. Lo novedoso es establecer la disposición de los “flex-items” no de izquierda a derecha, sino de arriba abajo. Vamos por partes.

Lo primero, convertir a la capa contenedora del título, descripción y vínculos como “flex”. Estas capas se clasifican como `class="describe_articulo"`. Al establecer el atributo de la captura izquierda, se produce el fenómeno de la captura derecha:

```

.describe_articulo
{
    padding-left: 1em;
    padding-right: 1em;
    display: flex;
}

```



Volkswagen Beetle Dune 2016
El Volkswagen Beetle Dune es una variante del Beetle con 10 milímetros más de altura libre al suelo y algunos elementos decorativos específicos. Estará a la venta a principios de 2016 con carrocería cerrada y descapotable.

[información](#)
[imágenes](#)
[exteriores](#)
[imágenes](#)
[interiores](#)

Lo habíamos previsto. Flex, por defecto, organiza sus “flex-items” de izquierda a derecha. Ese comportamiento lo cambiamos así, obteniendo el resultado:

```

.describe_articulo
{
    padding-left: 1em;
    padding-right: 1em;
    display: flex;
    flex-direction: column;
}

```



Volkswagen Beetle Dune 2016
El Volkswagen Beetle Dune es una variante del Beetle con 10 milímetros más de altura libre al suelo y algunos elementos decorativos específicos. Estará a la venta a principios de 2016 con carrocería cerrada y descapotable.

[información](#) [imágenes exteriores](#) [imágenes interiores](#)

Bien, arreglado. Volemos a lo que teníamos. Pero hay una diferencia, ahora podemos usar los atributos de “Flex-Positioning” en los “flex-items” de este contenedor. Particularmente, queremos que la descripción crezca todo lo que pueda, empujando hacia abajo a los vínculos:

```

.texto_descriptivo
{
    font-family: Arial;
    font-size: small;
    flex-grow: 1;
}

```



Volkswagen Beetle Dune 2016
El Volkswagen Beetle Dune es una variante del Beetle con 10 milímetros más de altura libre al suelo y algunos elementos decorativos específicos. Estará a la venta a principios de 2016 con carrocería cerrada y descapotable.

[información](#) [imágenes exteriores](#) [imágenes interiores](#)

“texto_descriptivo” es la clase en la que se agrupan las descripciones de los titulares de cada artículo, por eso ha sido ahí donde hemos indicado que ese “flex-item” puede crecer.

El resultado es visible y correcto, los vínculos están ahora alineados a la parte inferior del área total que puede utilizar la descripción total del artículo (capa de la clase “describe_articulo”).

Con esto damos por concluido el maquetado y formateado básico de la web, cuyo aspecto actual es el siguiente:



Como perlas adicionales y, sin que sean objeto de estudio ni susceptibles de entrar para examen, creo oportuno comentar un par de carencias más que se puede corregir fácilmente.

La primera tiene que ver con mejorar el perímetro, pues hay cosas que quedan muy “a sangre” del borde actual de la página. Para ello, podríamos hacer dos cosas:

- Incrementar en, por ejemplo 1em, el padding de toda la página (elemento con id="pagina"), de forma que quede un espacio alrededor con el mismo color de fondo que la página.
- Crear un efecto de sombra que resalte aún más la página del fondo. Eso puede hacerse con un atributo llamado “box-shadow” y que puede aplicarse a cualquier “caja” (capa). Al igual que los degradados, también hay webs online que nos facilitan calcular sus valores, como por ejemplo: <http://www.css3generator.in/default.html#box-shadow-generator>

Veamos la aplicación CSS de las dos ideas anteriores, junto con el resultado final:

```
#pagina
{
    width: 980px;
    padding: 1em 1em 1em 1em;
    box-shadow: 0px 0px 40px rgba(0,0,0,0.5);
    background-color: white;
    margin-left: auto;
    margin-right: auto;
    font-family: Verdana;
}
```

El “padding” no requiere más explicación. Del “box-shadow” decir que los parámetros son:

- Distancia de separación de la sombra con respecto al eje x (offset x).
- Distancia de separación de la sombra con respecto al eje y (offset y).
- Tamaño de la sombra (40 pixels en nuestro ejemplo).
- Color de la sombra (en nuestro ejemplo, negro semitransparente al 50%).

La segunda se refiere al efecto que provoca el ratón al pasar sobre algunos elementos. Particularmente las opciones de los menús, que cambian de color. Este efecto parece que requiere programación dinámica del lado del cliente (javascript) pero en realidad es un comportamiento que puede ser definido en el propio CSS.

No entraré en profundidad, pero al menos comentar que existen unas denominadas “pseudo-clases” que se refieren a estados en los que pueden estar los selectores que ya conocemos de CSS. Concretamente, la pseudo-clase llamada “hover” se refiere al estado que presenta algo cuando el ratón lo sobrevuela. Así, si yo quiero indicar el formato que ha de tener la clase “link_interno” cuando es sobrevolada por el ratón, lo haría mediante el selector “.link_interno:hover” (los dos puntos es la forma de separar el selector y la pseudo-clase).

En la web original vemos que el degradado se torna anaranjado, en lugar de azul. El código CSS para conseguir este efecto en nuestro diseño es el siguiente:

```
.link_interno:hover, .link_externo:hover
{
    background-image: linear-gradient(to bottom, #FFCC00 0%, #FF4400 100%);
    color: white;
}
```

En la misma regla hemos hecho referencia a los dos menús. Hemos indicado el color anaranjado para el degradado y hemos forzado la fuente a que sea blanca, ya que el menú de vínculos externos tenía por defecto la fuente azul y, con el naranja, no combinaba bien.