

Ministerio de Educación, Cultura y Deporte.

HTML5 en la educación

Módulo 10: Diseño web con estilos.

Diseño web con estilos

El formato con estilo



Las posibilidades que nos brinda el uso de hojas de estilo en cascada para diseñar nuestras páginas web son muy amplias. Veremos los aspectos principales relacionados con el formato de texto, colores, fondos y otros detalles gráficos.

Es difícil abarcar todas las opciones posibles e incluso algunas propiedades se nos quedarán en el tintero. De hecho, el propio estándar CSS3 está en desarrollo y con cada nueva versión de los navegadores web nos encontramos nuevas implementaciones de esas propiedades. No obstante sí que veremos todos los elementos imprescindibles para conseguir un diseño web atractivo para nuestros usuarios.



Programación

Objetivos específicos

- Aplicar estilos relacionados con párrafos, títulos y otros elementos similares.
- Aplicar color y fondos a los elementos de una página web.
- Aplicar diferentes propiedades vinculadas a la apariencia de la página web.

Contenidos

- Estilos de texto.
- Estilos de color y fondos.
- Otros estilos gráficos.

Criterios de evaluación

- Aplicar estilos al texto y otros elementos.
- Modificar los estilos aplicados.



Requisitos mínimos

- Conocimientos básicos sobre HTML.
- Conocimientos básicos sobre navegadores web.
- Conocimientos básicos de procedimientos en el ordenador: seleccionar, cortar y pegar.

Recurso TIC: Diseño web con estilos

En este apartado nos sumergiremos definitivamente en el conocimiento de las diferentes propiedades de las hojas de estilo que nos permiten controlar la apariencia de una página web. Con lo que conocemos hasta ahora podemos decidir con facilidad en qué lugar queremos que se aplique uno u otro estilo, por lo que sólo necesitaremos tener presentes las posibilidades disponibles.

Como punto de partida necesitaremos un documento base con abundante texto y diferentes etiquetas, donde iremos realizando las modificaciones. Probablemente nos resulte cómodo emplear un editor web, como el que hemos venido empleando hasta ahora.

No obstante todas las reglas que vamos a analizar pueden ser aplicadas directamente mediante una hoja de estilos externa, nuestro método favorito de trabajo con estilos.

Propiedades de texto

Sin más preámbulos, analizaremos algunas de las reglas que podemos definir respecto al texto.

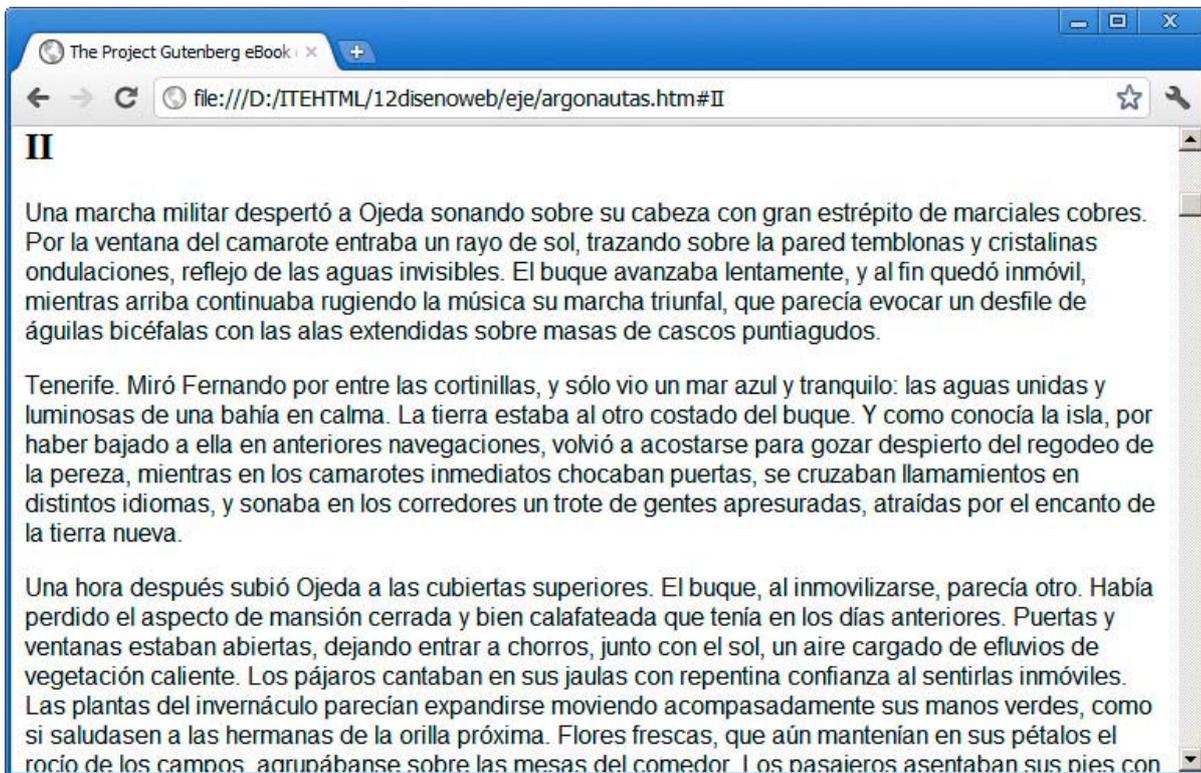
Tipo de letra

La propiedad **font-family**, seguida de una tipografía concreta o de una serie de tipos de letra separadas por comas, define la apariencia del elemento HTML en cuestión.

Hay muchos tipos de letra, pero un pequeño conjunto de ellos se usan con asiduidad, como el tipo *Times New Roman*, *Helvética*, *Georgia*, *Arial*, *Verdana*, etc. Cada uno tiene sus propias características. Normalmente los tipos sin serifa (como *arial* o *verdana*) suelen proporcionar una estética agradable y moderna, mientras que los que sí la tienen (como *times*) se suelen usar para textos largos, como libros o documentos más densos. La práctica nos irá marcando la pauta de qué tipo será el más adecuado en cada caso.

```
h1 {font-family: Georgia, "Times New Roman", Times, Serif;}  
body {font-family: Helvetica, Arial, Sans-serif; }
```

Con este ejemplo estamos definiendo un formato concreto para toda la página y otro para los títulos. El resultado del ejemplo anterior sería el mostrado en la figura.



Al definir un valor para el cuerpo de la página, estamos estableciendo ese tipo de letra para todos los elementos excepto para aquellos en los que se indique otra cosa. Es una práctica muy habitual.

Normalmente siempre indicaremos un grupo de fuentes, ya que, si el navegador no tiene instalada la primera, recurrirá a las alternativas, hasta llegar a la última que hemos indicado, que define una fuente genérica. Las fuentes genéricas son **serif**, **sans-serif**, **cursive**, **fantasy** y **monospace**.



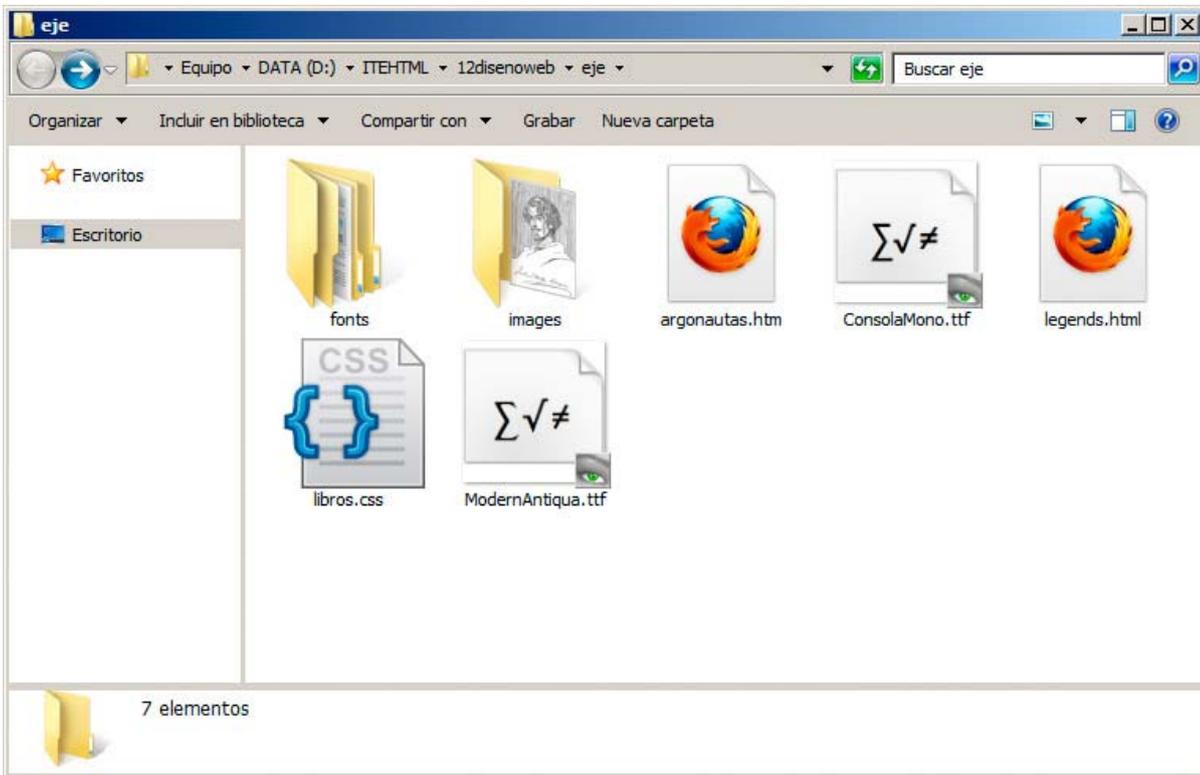
Nota

El que el nombre se ponga en mayúsculas o minúsculas no tiene relevancia; pero, si tiene varias palabras en su nombre, deberíamos ponerlo entre comillas simples o dobles.

Tipos de letra no instalados

Hasta la versión 3 del estándar CSS el uso de tipografías estaba bastante acotado, ya que sólo podíamos emplear aquellos tipos de letra que intuíamos que el usuario podría tener instalados. Ahora eso no es así y podemos indicar al navegador que emplee un tipo de letra que nosotros distribuimos junto a la página web.

Si observamos la figura, veremos que junto a la página web y la hoja de estilos hemos incorporado también un par de tipos de letra, archivos con extensión `.ttf`. Es decir, tipografías en formato *TrueType*.

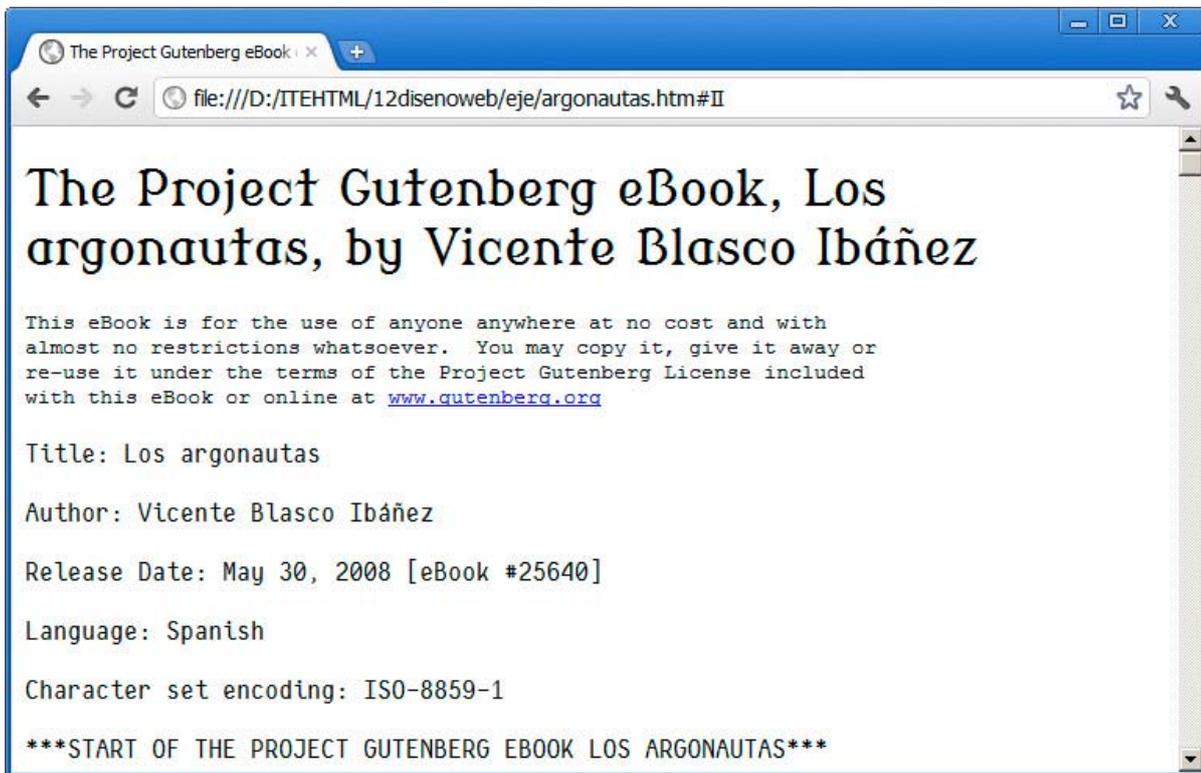


Para incorporar esas tipografías, modificaremos las líneas anteriores estableciéndolas de la siguiente manera:

```
@font-face {  
    font-family: TipoTitulos;  
    src: url('ModernAntiqua.ttf');  
}  
  
@font-face {  
    font-family: TipoGeneral;  
    src: url('ConsolaMono.ttf');  
}  
  
h1 {  
    font-family: TipoTitulos, Serif;  
}  
  
body {  
    font-family: TipoGeneral, Sans-serif;  
}
```

Es decir, mediante la directiva **@font-face** le indicamos al navegador que tenemos un nuevo tipo de letra basado en un archivo que nosotros mismos le suministramos. A su vez, en **font-family** emplearemos el nuevo tipo de letra. Incluso podemos poner una fuente genérica, por si hubiese algún problema de carga de los archivos.

El resultado se muestra en la figura:



Estamos empleando fuentes sin preocuparnos de si el usuario las tiene instaladas o no.



Nota

Como siempre, debemos intentar coger tipografías libres. Las del ejemplo han sido tomadas de la web de *Open Library Font* (<http://openfontlibrary.org/>).



Actividad 1

Localizaremos una tipografía libre en Internet y se la aplicaremos a varias etiquetas del texto, empleando el método visto anteriormente. Comprobaremos su funcionamiento en al menos dos navegadores.

Se pueden encontrar tipografías libres mediante cualquier buscador. También podemos recurrir a algún sitio web concreto dedicado a ello, como <http://openfontlibrary.org/> o <http://www.theleagueofmoveabletype.com>.

Tamaño de letra

Para el tamaño de letra se emplea la propiedad **font-size**.

```
body {font-size: 2em;}
```

En el valor, además de un número, indicaremos la medida que empleamos entre una de éstas:

- **%**: un porcentaje para el texto hará que varíen sus tamaños basados en el valor original. Por ejemplo, **body {font-size: 80%; }** hace que el texto se vea un 20% más pequeño de lo normal, mientras que un 150% lo incrementaría en un 50%.
- **px**, **cm**, **pt** o **in**: para indicar medidas en píxeles, centímetros, puntos o pulgadas. No suele ser muy apropiado, porque cada navegador puede mostrar diferencias en este sentido, aunque para definir estilos de impresión sí que

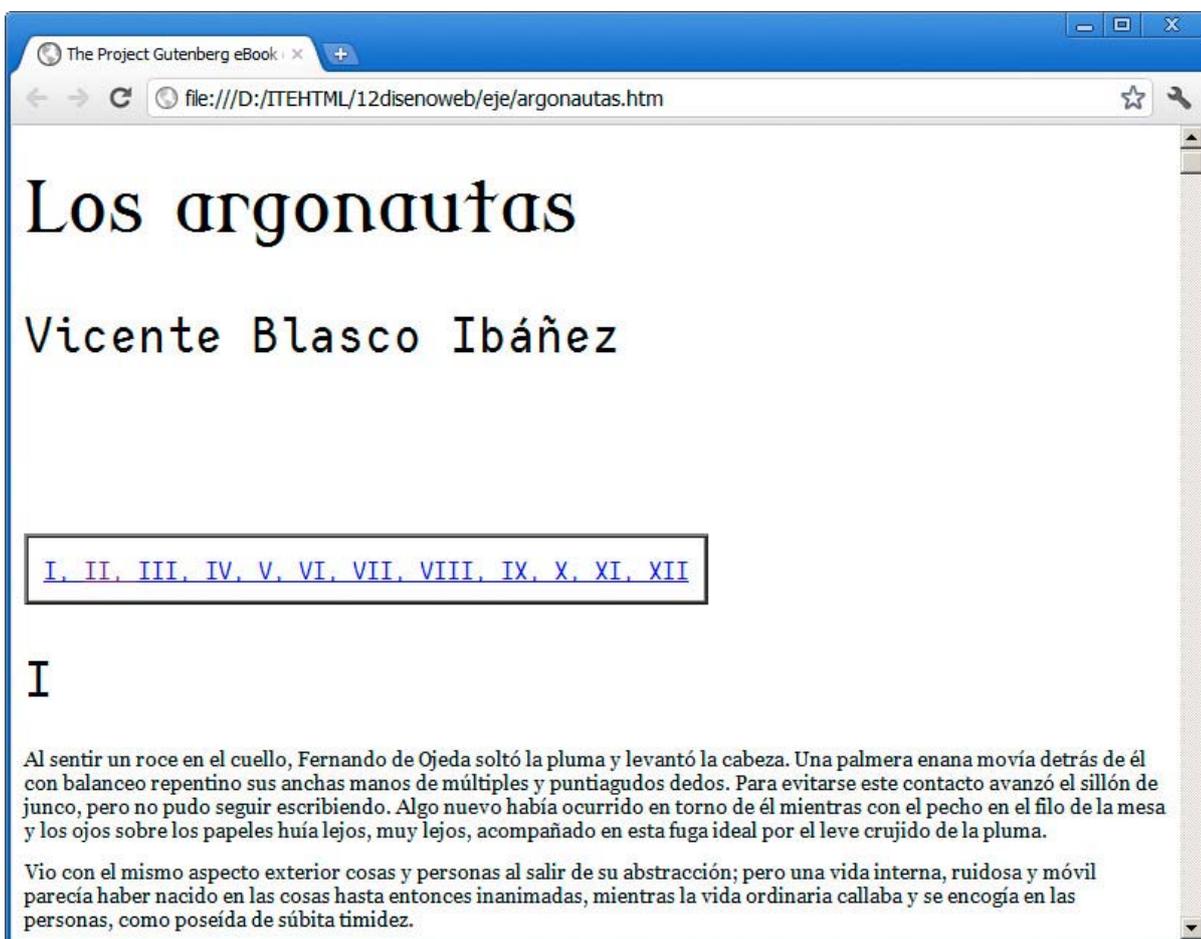
nos dará unos valores más precisos.

- **em**: es el valor recomendado por el W3C, si queremos emplear valores fijos. 1em es su valor base y equivale a 16 píxeles. 2em sería el doble de tamaño .5em la mitad (es lo mismo que 0.5em), etc.
- **Valores fijos**: hay ciertos modificadores fijos que podemos emplear, que van de menor a mayor (**xx-small**, **x-small**, **small**, **medium**, **large**, **x-large** y **xx-large**). Éstos no necesitan un número y asignan siempre el mismo tamaño.
- **Valores relativos**: usando **smaller** (más pequeño) y **larger** (más grande) conseguimos que la tipografía aumente o disminuya su tamaño respecto a su valor original.

Los diferentes valores se pueden mezclar, pero finalmente lo más recomendable sería emplear una escala basada en porcentajes o en **em**. A nuestra plantilla de ejemplo incorporaremos el siguiente bloque:

```
body {font-size:100%;}
h1 {font-size:3em;}
h2 {font-size:2em;}
p {font-size:0.8em;}
```

Conseguiremos unos títulos grandes, con un texto general al 100% de su tamaño, es decir, a 1em y unos párrafos un poco más pequeños. La figura presenta el resultado:



En la hoja de estilos completa hemos combinado las propiedades dentro de sus selectores:

```
@font-face {
    font-family: TipoTítulos;
    src: url('ModernAntiqua.ttf');
}

@font-face {
    font-family: TipoGeneral;
    src: url('ConsolaMono.ttf');
}
```

```
body {  
    font-family: TipoGeneral, Sans-serif;  
    font-size:100%;  
}  
  
h1 {  
    font-family: TipoTitulos, Serif;  
    font-size:3em;  
}  
  
h2 {  
    font-size:2em;  
}  
  
p {  
    font-family: Georgia,"Times New Roman",Serif;  
    font-size:0.8em;  
}
```



Actividad 2

Utilizando valores **em** definiremos diferentes parámetros para nuestra web de ejemplo. Comprueba lo que sucede, si a un elemento le asignamos 0em como tamaño o 1em.

Estilo de fuente

La propiedad **font-style** puede tomar el valor **normal**, que es el predefinido, **italic** (equivalente a cursiva) o **oblique** (equivalente a oblícua), que inclina el texto a la derecha. Normalmente coinciden estos dos últimos resultados pero, según la tipografía, podría haber cierta variación. La figura muestra su apariencia:



Nota

¿Para qué querríamos el valor **normal** si ese es ya el valor predeterminado? Su sentido está en que, si hemos definido una etiqueta superior con ese valor en cursiva, el valor normal nos permitiría establecer que una parte de un contenido no se mostrase en cursiva. Así, si toda la página va a estar en formato **italic**, podríamos hacer que un párrafo o una clase concreta se mostrase normal, indicando **font-style:normal**. Por eso casi la mayoría de las propiedades de CSS tienen la posibilidad de forzar el establecimiento de su valor predeterminado.

Texto en negrita

La propiedad **font-weight** especifica si el texto irá en negrita o no, con diferentes grados:

- **bold**: negrita.
- **bolder** y **lighter**: valor relativo que aplica un efecto más pronunciado en el primer caso o menos en el segundo.
- **100..900**: también podemos indicar con un valor entre 100 y 900 el grado de "negrita" que queremos.
- **normal**: hace que el texto no se muestre en negrita, saltándose cualquier otra directriz marcada por otras normas.

Mayúsculas y minúsculas

La propiedad **font-variant** se emplea para forzar que un texto concreto se muestre en mayúsculas con tamaño de minúsculas, lo que se conoce como **versalitas** ("versal" es un sinónimo de mayúsculas).

Sus valores son **normal** o **small-caps**, con los resultados que se muestran en la figura:



Ahora todos juntos

Todas las propiedades anteriores tienen algo en común; pueden ser agrupadas mediante una propiedad única denominada **font**. Los valores que podemos asignarle son todos los anteriores, separados por espacios, siguiendo este orden:

1. **font-style** (opcional)
2. **font-weight** (opcional)
3. **font-variant** (opcional)
4. **font-size**
5. **font-family**

Los valores opcionales no es necesario indicarlos. De hecho, aún se puede añadir un valor más justo antes de **font-family** denominado **line-height**, que veremos un poco más tarde.

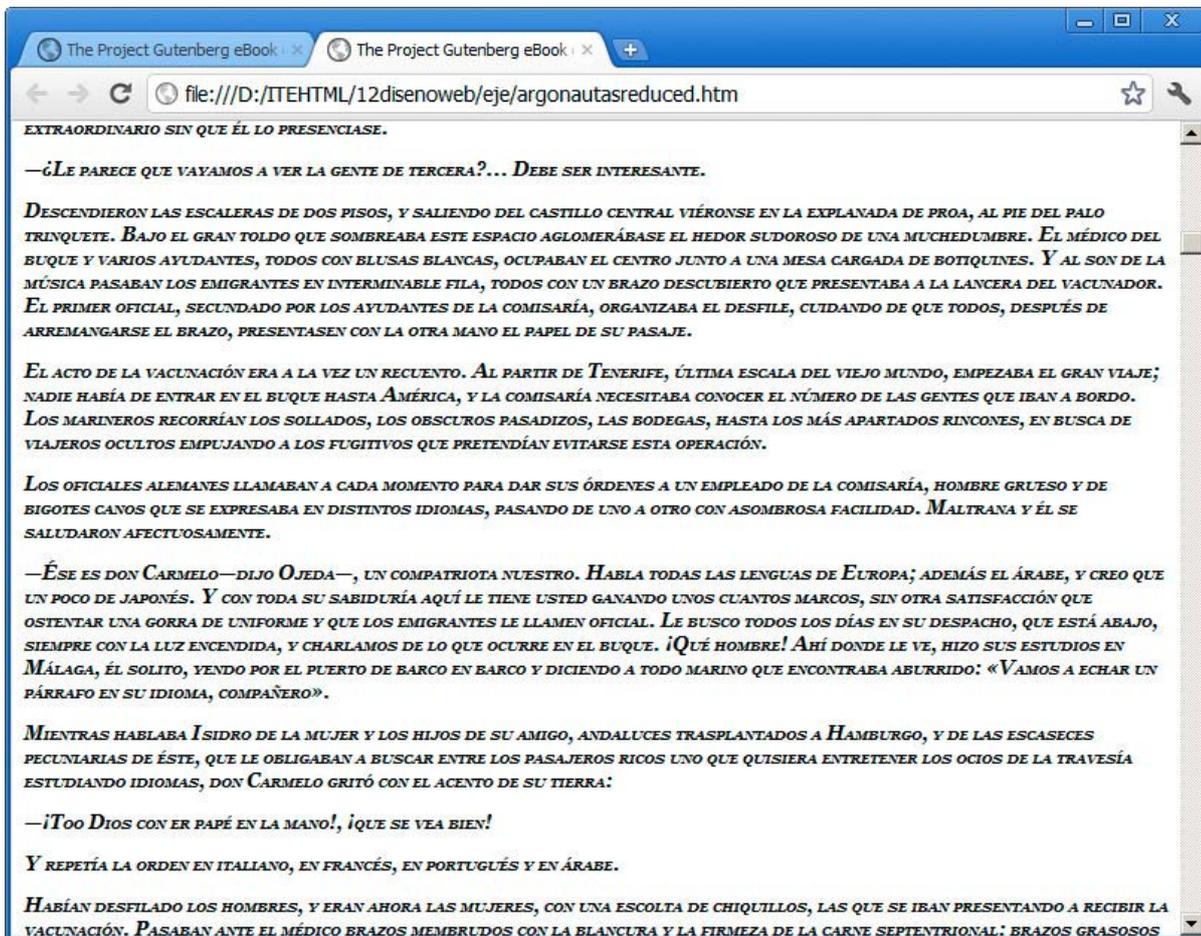
Así una línea mínima que definiese el estilo del cuerpo de la página podría quedar así:

```
body {font: 1em, TipoGeneral, Sans-serif; }
```

Un ejemplo más completo sería éste:

```
p {font:italic bolder small-caps 0.8em Georgia,"Times New Roman",Serif; }
```

Aplicado a nuestro ejemplo obtendríamos el resultado de la figura. Muy recargado, pero nos sirve para comprobar cómo se pueden condensar varias propiedades en una sola.



Pregunta de Elección Múltiple

Para definir el tipo de letra se utiliza el parámetro:

- font-variant.
- font-weight.
- font-style.
- font-size.
- font-family.

Más propiedades de texto

Veremos ahora algunas propiedades más, relacionadas en este caso más con el texto que con la apariencia de la tipografía.

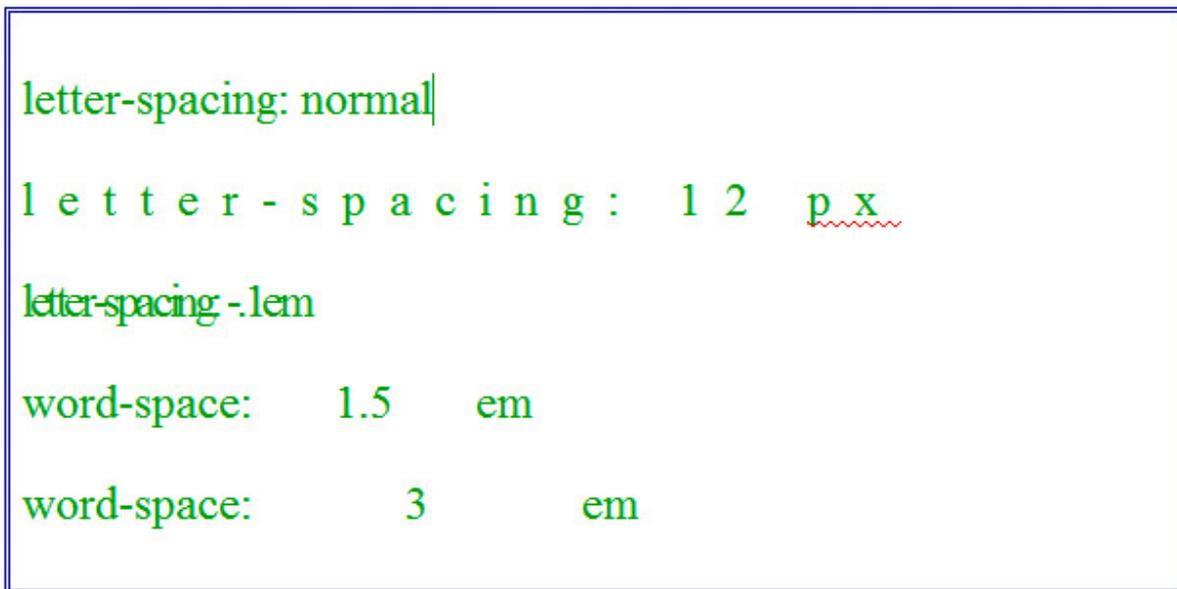
Espacios

La propiedad **letter-spacing** modifica el espacio entre los caracteres de una palabra y **word-spacing** hace lo mismo pero entre palabras.

Las medidas que se emplean en ambos casos son las mismas que ya vimos para el tamaño de letra. Además podemos emplear el valor normal para reestablecer cualquiera de los dos valores, si los hemos modificado en otro elemento. Aquí tenemos algunos ejemplos, que darían el resultado de la figura:

p {letter-spacing:normal}

p {letter-spacing: 12 px}
p {letter-spacing: -.1em}
p {word-space: 1.5 em}
p {word-space: 3 em}



En el eje vertical podemos emplear la propiedad **line-height** para incrementar o reducir el espacio entre líneas. Las dos líneas siguientes modifican la separación entre líneas de un párrafo:

p {line-height: 1em}
p {line-height: 2em}

Mientras que la primera elimina casi por completo el espacio entre líneas, la segunda pone el texto a doble espacio. En la figura se pueden comparar tres párrafos: el primero sería el modelo normal y los dos siguientes corresponden a las líneas anteriores.

Erguidos ante sus atriles con militar rigidez, entonaban los músicos una marcha solemne, que servía de acompañamiento a los pasajeros en su entrada al comedor. Los hombres vestían de frac o de *smoking*, guardando en una mano la gorra de viaje. Algunos se detenían en las puertas formando grupos para ver a las señoras que iban saliendo de los camarotes de preferencia o venían de los de abajo por la gran escalera de doble rampa, con un roce de finas ropas interiores.

Deslizábanse rápidas todas ellas, entre saludos y sonrisas, para sumirse, más allá de las mamparas de cristales, en un mar de luz en el que nadaban los colores de inquietas banderas. Una estela de polvos de tocador y vagas esencias de jardín artificial seguía el aleteo de las faldas desmayadas y flácidas, con brillantes pajuelas de oro o plata; el crujiente arrastre de los tejidos sedosos; el brillo de las espaldas desnudas suavizadas con una capa de blanquete; la tersura de las nuca, sobre las que se elevaba el edificio de un peinado extraordinario, el primero de una navegación que únicamente se había prestado hasta entonces a exhibir sombreros de paseo y velos de odalisca.

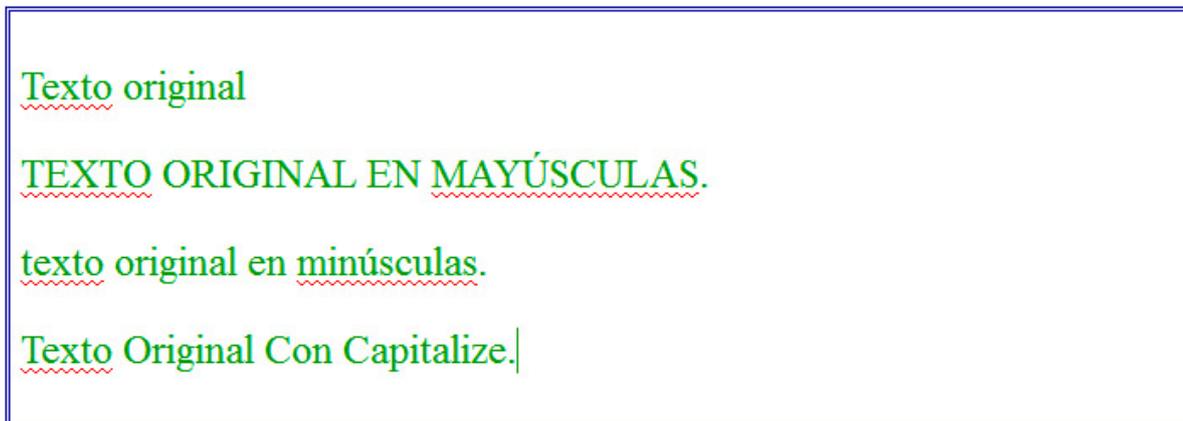
En el antecomedor lucía un gran cartel pintarrajeado con una pareja danzante y una inscripción gótica en alemán y en español: «Esta noche baile.» Y el anuncio parecía esparcir por todo el buque un regocijo de colegio en libertad. «Esta noche baile», repetían las personas de grave aspecto, como si se prometiesen un sinnúmero de misteriosas satisfacciones.

Mayúsculas y minúsculas

Con la propiedad **text-transform** podemos convertir un bloque de texto completo en:

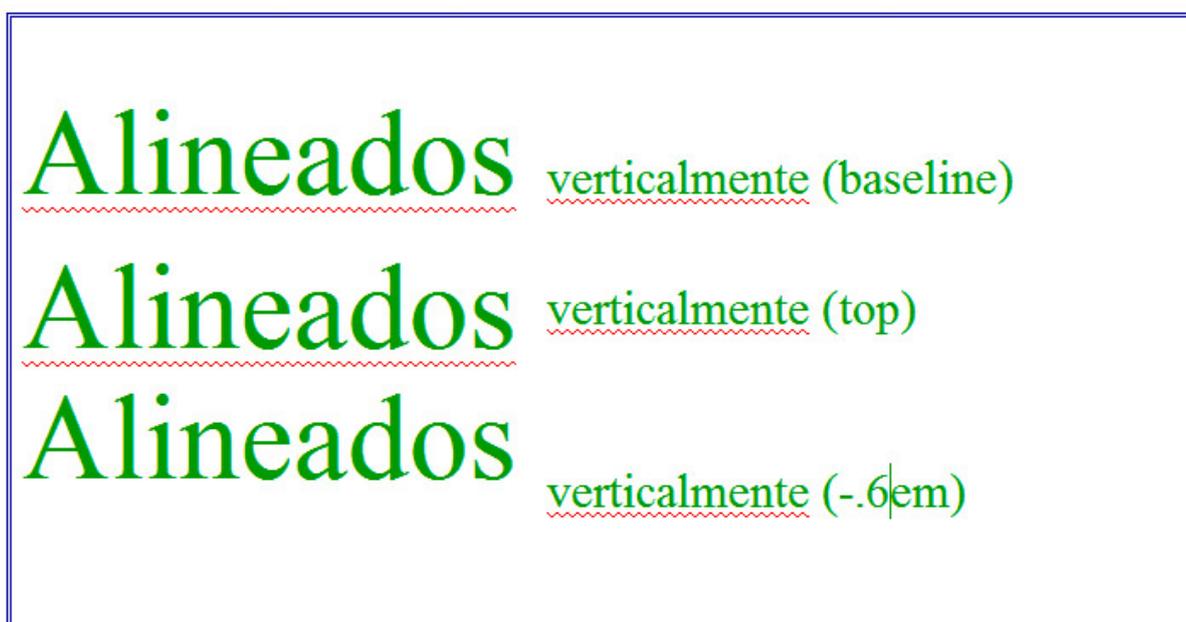
- Mayúsculas: con el valor **uppercase**.
- Minúsculas: valor **lowercase**.
- Primera en mayúsculas, resto en minúsculas: **capitalize** convierte la primera letra de cada palabra a mayúscula.
- Dejar el texto como está: emplearemos el valor **none**.

La figura nos presenta las diferentes modalidades:



Alineación entre elementos

La propiedad **vertical-align** nos será útil para definir la alineación entre dos o más elementos que no tengan la misma altura, por ejemplo, un texto y una imagen o textos de diferente tamaño, como los que se muestran en la figura:



Su composición, si la segunda parte fuese, por ejemplo, una clase aplicada a un valor **span**, quedaría así:

```
span.parte2 {vertical-align:baseline}  
span.parte2 {vertical-align:top}  
span.parte2 {vertical-align:-.6em}
```

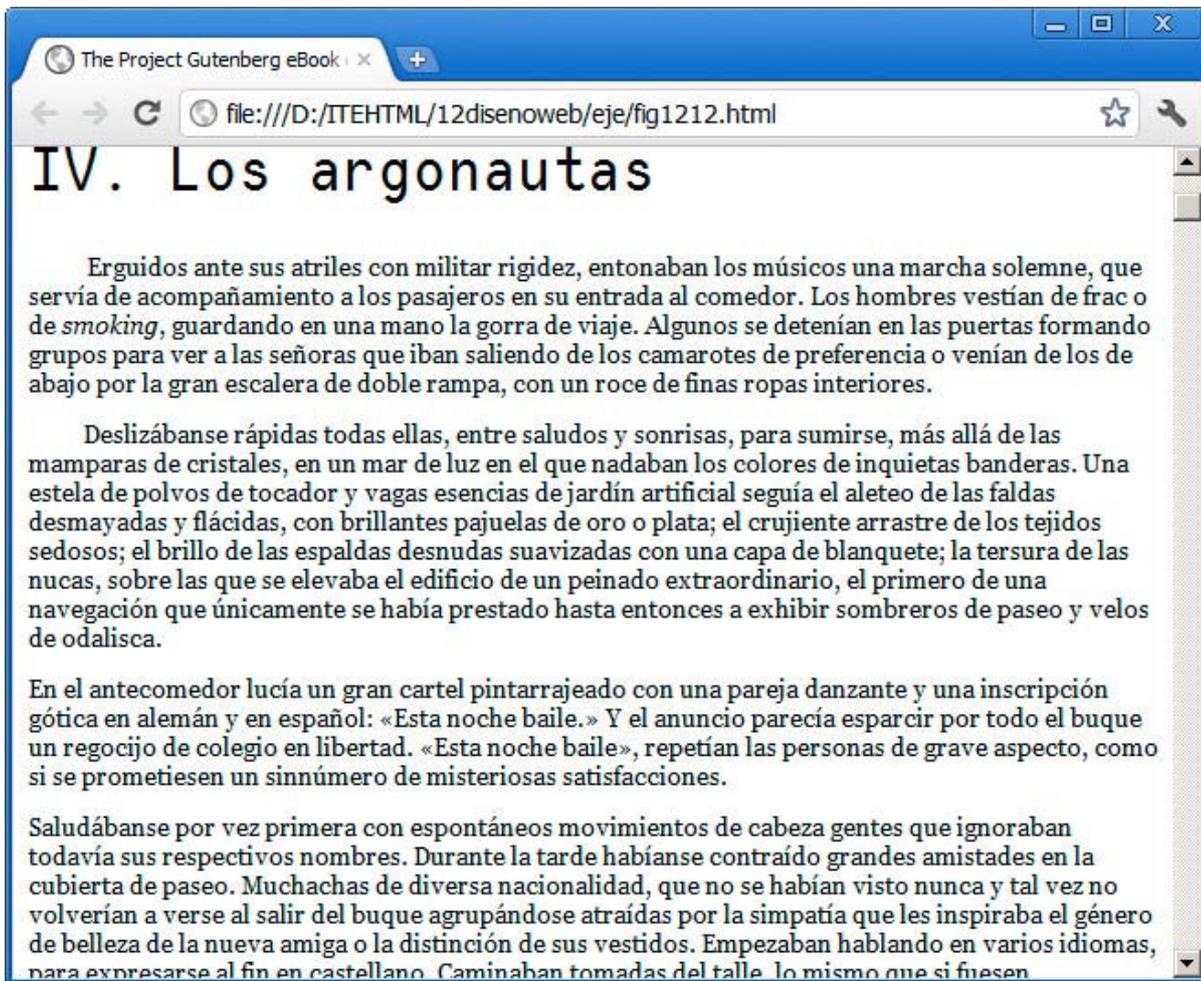
Contamos con un amplio muestrario de parámetros, como se puede observar. Podemos definir valores relativos, absolutos o utilizar algunas palabras clave como **top** (alineación superior), **bottom** (inferior), **baseline** (se alinean sus líneas base), etc. Este tipo de propiedades se suele modificar con algún tipo de editor, por lo que no es necesario recordar todos sus valores.

Sangrado del texto

El sangrado (a veces denominado *indentación*, en referencia al término inglés) consiste en desplazar a la izquierda o a la derecha la primera línea de un párrafo. Con CSS se puede realizar mediante la propiedad **text-indent**, seguida de algún valor numérico absoluto o relativo.

La figura nos muestra el resultado que se produce con la siguiente línea:

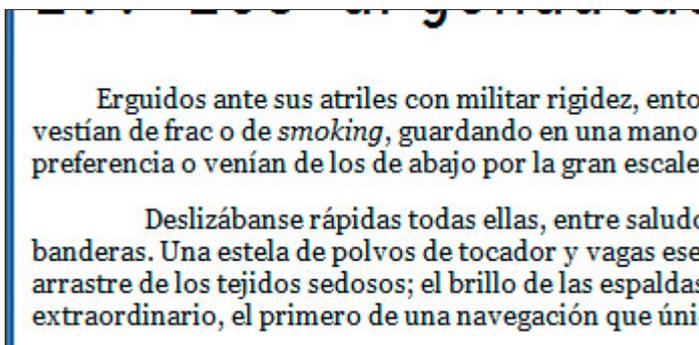
```
p {text-indent: 2.5em}  
p {text-indent: 3%}
```



Valores relativos o absolutos

Este ejemplo ilustra perfectamente la diferencia entre valores relativos y absolutos. El primer párrafo tiene aplicada la primera línea, es decir, el valor exacto y absoluto. Por el contrario, la segunda tiene el valor porcentual, el relativo. Se desplaza la línea un 3% del tamaño horizontal de la ventana del navegador.

Ambas líneas parecen iguales pero, si ampliamos el tamaño de la ventana del navegador, observaremos que la primera línea se mantiene en la misma distancia, mientras que la segunda ha variado, ya que el 3% de un valor mayor será una distancia notablemente mayor.



Dependiendo del sitio web que estemos diseñando, nos resultará más conveniente usar valores relativos o absolutos. Por lo general los valores relativos aportarán flexibilidad a nuestra página web.

Alineación del texto

Otro factor clásico de cualquier editor de texto es su capacidad de alinear el texto horizontalmente a la izquierda, derecha, centrado o justificado. Con CSS se realiza mediante la propiedad **text-align**.

Es tan simple como añadir alguna regla de este tipo:

```
body {text-align: justify; }
h1 {text-align: right;}
```

Los valores posibles son **left** (izquierda, el predefinido), **right** (derecha), **center** (centrado) o **justify** (justificado).

Espacios en blanco

La propiedad **white-space** controla el tratamiento que hace el navegador de los espacios en blanco. Los navegadores tienden a comprimir o ampliar esos espacios para ajustar las palabras en las líneas. Para evitarlo, podemos utilizar el valor **pre** con la propiedad **white-space** y así conseguir que éstos no se modifiquen. Esto es útil si tenemos un texto exacto, fórmulas, un poema, etc. Las rupturas de línea que tuviese el texto original se respetarán escrupulosamente.

Igualmente, si indicamos la opción **nowrap**, el navegador no cortará las líneas, sino que mantendrá cada párrafo o bloque de texto en la misma línea. La línea se prolongará hacia la derecha todo lo que sea necesario.

Ajuste de palabras

La propiedad **word-wrap** permite, con el valor **break-word**, la ruptura de palabras que no serían divisibles en circunstancias normales, como palabras muy largas en cuadros de texto pequeños. Su valor predeterminado es normal, permite la ruptura cuando sea posible y si no, se traslada la palabra a la línea siguiente.

Decoración del texto

La propiedad **text-decoration** cuenta con algunos valores para decorar una línea de texto. Por ejemplo, para un rótulo **<h1>**:

- **h1{text-decoration:none;}** elimina cualquier decoración que pueda tener un elemento.
- **h1{text-decoration:underline;}** el texto aparece subrayado.
- **h1{text-decoration:overline;}** el texto se presenta con una línea por encima.
- **h1{text-decoration:line-through;}** el texto se muestra tachado.
- **h1{text-decoration:blink;}** el texto parpadea, distrayendo la atención sobre el resto de la página. Se debe usar con mucho criterio.

Estas opciones se emplean con frecuencia para tachar un texto que ya no vale o para resaltar los estados de un enlace, mediante sus pseudoselectores. Por ejemplo, el enlace de la figura se resalta mediante un **underline** y un **overline** al pasar el ratón sobre el.



Para conseguirlo, hemos añadido estas dos líneas a nuestro archivo de estilos:

```
a:link {text-decoration: none; }
a:hover {text-decoration: overline underline;}
```

Sombras

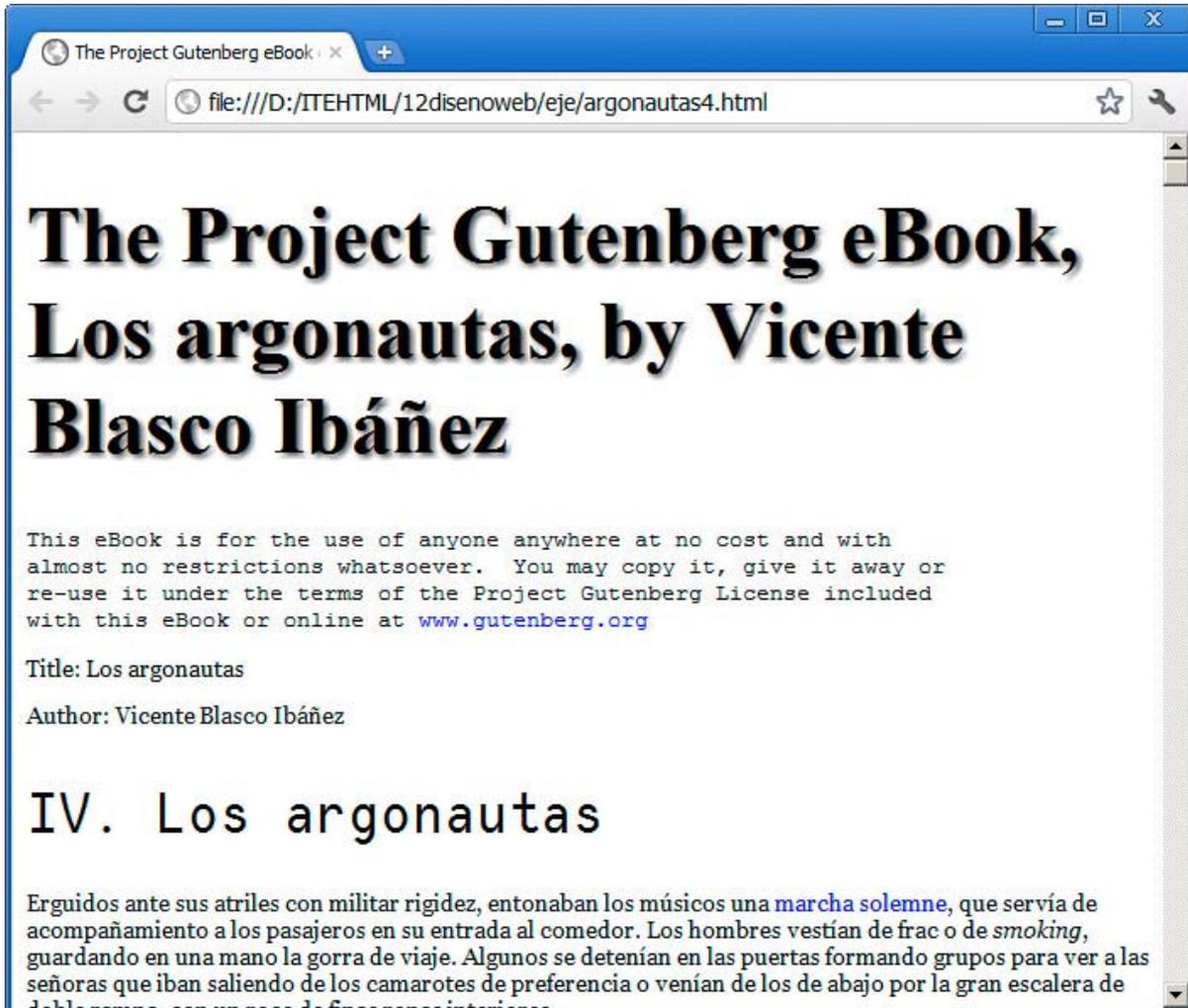
Con CSS3 es posible añadir sombras a un texto con la propiedad **text-shadow**; así de fácil:

```
h1 { text-shadow: rgb(90,90,90) 2px 2px 4px };
```

Lleva cuatro parámetros:

- **Color**: expresado de diferentes maneras, es además opcional.
- **X e Y**: los dos valores siguientes son obligatorios e indican la distancia de la sombra.
- **Suavizado**: el último valor es opcional y especifica lo difuminada que se mostrará la sombra.

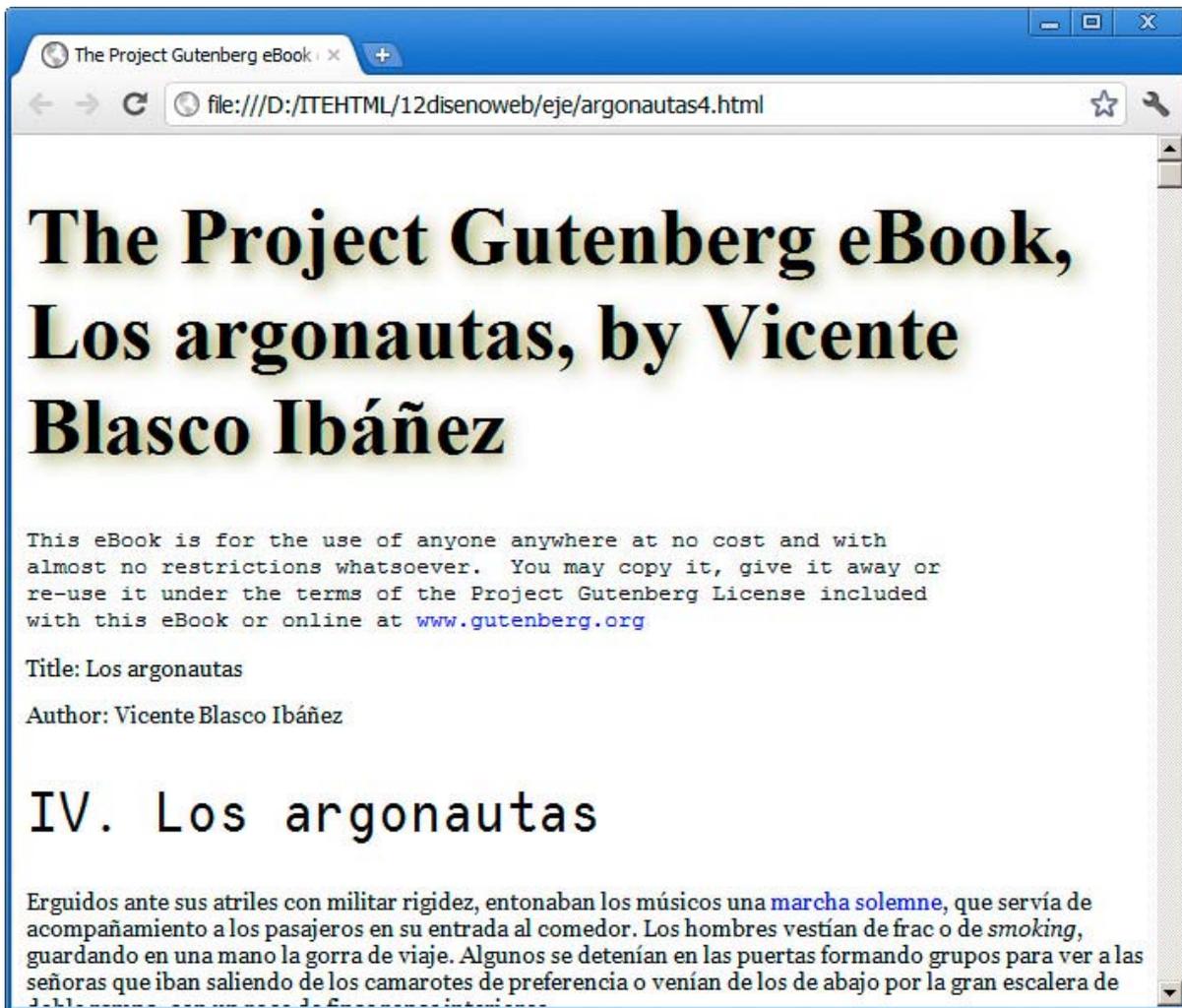
El resultado se muestra en la figura:



Para obtener unas sombras apropiadas, es necesario realizar algunas pruebas e intentar conseguir el efecto que buscamos.

La figura siguiente muestra esta otra combinación, mucho más difuminada y con un toque de color.

```
h1 {text-shadow:rgb(160,160,90) 4px 4px 16px;}
```



Actividad 3

En un documento con varios títulos realizaremos diferentes incorporaciones a la hoja de estilos, para conseguir que proyecten algunas sombras. Debemos conseguir al menos una sombra muy marcada y precisa y otra tan sutil que sea casi inapreciable. ¿Cuál consideramos que proporciona más elegancia al texto?



Pregunta Verdadero-Falso

Las siguientes afirmaciones, ¿son verdaderas o falsas?

La propiedad *letter-spacing* modifica el espacio entre los caracteres de una palabra y *word-spacing* hace lo mismo pero entre palabras.

Verdadero Falso

Con la propiedad *text-transform* podemos convertir un bloque de texto completo en mayúsculas utilizando el valor *lowercase*.

Verdadero Falso

La propiedad *vertical-align* nos será útil para definir la alineación entre dos o más elementos que no tengan la misma altura, por ejemplo, un texto y una imagen o textos de diferente tamaño.

Verdadero Falso

Cualquier editor de texto tiene la capacidad de alinear el texto horizontalmente a la izquierda, derecha, centrado o justificado mediante la propiedad *style-align*.

Verdadero Falso

Colores

Hasta el momento hemos obviado la gestión y aplicación del color, exceptuando pequeños ejemplos puntuales. Son varias las propiedades en las que podemos aplicar un color determinado, ya sea como fondo, como color de texto, para una sombra, etc. pero en todos los casos las formas de definir el color son similares.

Las propiedades que se emplean para definir el color del texto son las siguientes:

- **color**: para indicar el color del texto.
- **background-color**: para indicar el color de fondo de diferentes elementos, entre ellos de un texto.

Formas de definición del color

Al empezar a trabajar con hojas de estilo, encontraremos que el método más sencillo para definir un color es mediante su nombre en inglés (especialmente si tenemos unas nociones básicas de inglés; en caso contrario encontraremos este método muy complicado).

Por ejemplo, para definir un texto de color rojo de una clase determinada utilizaríamos esta regla:

```
.parrafodestacado {color:red; }
```

Un título azul y con el fondo amarillo se conseguiría así:

```
h1 {color:blue; background-color:yellow; }
```

Y que toda la página tuviese un color lila, con fondo verde y sombra azul oscura, se consigue así:

```
body {  
    color:purple;  
    background-color: green;  
    text-shadow: midnightblue 2px 2px 4px;  
}
```

Sería digno de verse. La figura lo muestra:

This eBook is for the use of anyone anywhere at no cost and with almost no restrictions whatsoever. You may copy it, give it away or re-use it under the terms of the Project Gutenberg License included with this eBook or online at www.gutenberg.org

Title: Los argonautas

Author: Vicente Blasco Ibáñez

IV. Los argonautas

Engruados ante sus ariles con millares rígidos, entonaban los músicos una **marcha solemne**, que servía de acompañamiento a los pasajeros en su entrada al comedor. Los hombres vestían de lino o de smoking, guardando en una mano la garter de virje. Algunos se detenían en las puertas formando grupos para ver a las señoras que iban saliendo de las camarotes de primera o venían de las de abajo por la gran escalera de doble rampa, con un roce de lince ropas interiores.

Dedicábanse rápidas volutas ellas, entre salidos y sonrisas, para quitarse, más allá de las mangas de cristal, en un haz de luz en el que nadaban los colores de pequeñas banderas. Una cascada de pedruzcos de frigidor y vapor escapaba de jardín artificial según el alfiler de un falda de mariposa y florecidas, con brillantes puntas de oro o plata, el orgulloso arquetipo de las bordas norteamericanas, el collar de las espaldas deprimidas apoyadas con una capa de blancopunto, la hermosa de las curvas, sobre las que se elevaba el edificio de un pómulo, sobre el marino, el primero de una navegación que intrínsecamente se había presentado frente entonces a estallar simultáneo de pasos y voces de solución.

En el antecomedor había un gran cartel pintado a guisa con una pareja danzante y una inscripción gótica en alemán y en español: «Esta noche baile.» Y el anuncio parecía escapar por todo el buque un resaca de colegio en libertad. «Esta noche baile», repetían las personas de grave aspecto, como si se prometiesen un



Nota

Algunos nombres de colores son más intuitivos que otros, ¿de dónde hemos sacado el color *midnightblue*? La lista de colores de nombres proviene de una serie denominada *colores X11*, que son admitidos en su mayoría por todos los navegadores. Se puede consultar un listado en la página http://en.wikipedia.org/wiki/X11_color_names, con sus equivalencias en formato RGB y hexadecimal.

El ejemplo anterior también nos arroja con crueldad otra realidad fundamental. Hasta ahora, usando negro sobre blanco, hemos podido elaborar páginas web con un cierto equilibrio estético pero, al comenzar a utilizar el color, todo esto se volverá mucho más complejo. Elegir unos colores que armonicen correctamente y que refuercen nuestro mensaje, en lugar de desvirtuarlo, es un verdadero arte. Para alcanzar el equilibrio necesario, el mejor consejo posible es practicar, practicar y practicar. Y nunca conformarnos con el primer resultado.

Definir un color numéricamente

El sistema de nombres de color se nos quedará corto rápidamente y necesitaremos buscar alguna solución para definir colores de una forma precisa.

En los ordenadores, en las pantallas, se emplea un sistema de representación del color basado en tres parámetros. Un ordenador es capaz de representar varios millones de colores con tan sólo realizar combinaciones de tres colores básicos: el rojo, el verde y el azul.

Hay otros sistemas de representación de colores, pero el basado en estos tres, conocido como sistema RGB (son las siglas de los tres colores en inglés, *red*, *green* y *blue*) nos servirá perfectamente para diseñar nuestros propios colores.

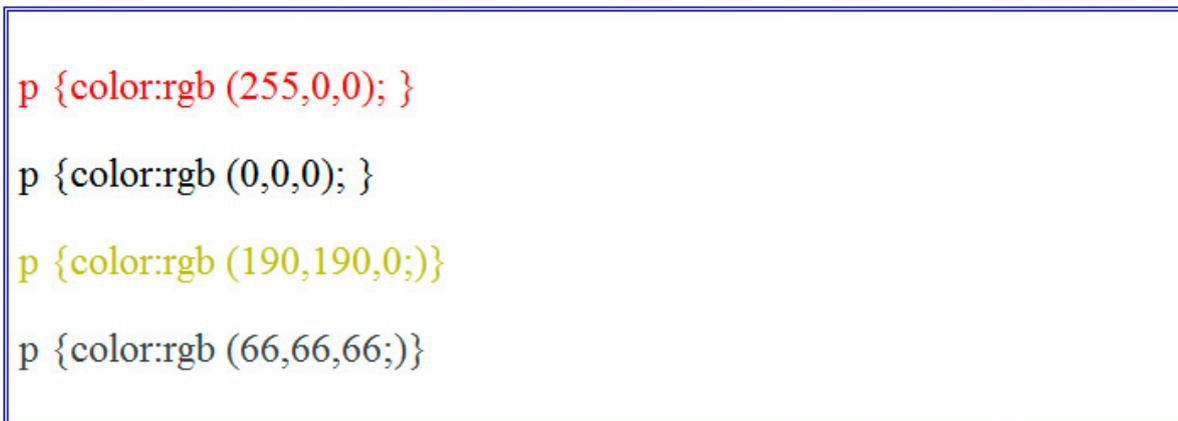
La definición de un color en formato RGB consiste en indicar qué cantidad de cada uno de esos colores debe llevar el color resultante, sabiendo que el valor más bajo representa la ausencia de color y el color más alto el blanco absoluto.

El color más alto dependerá de la medida que estemos empleando. Podemos hacerlo mediante números decimales que van del 0 al 255 o mediante valores porcentuales.

Veamos algunos ejemplos:

- `p {color:rgb (255,0,0); }` representa el rojo puro. El valor de `r` está al máximo mientras los demás están al mínimo.
- `p {color:rgb (0,0,0); }` este color sería el negro; todos los valores al mínimo.
- `p {color:rgb (190,190,0);}` éste sería un amarillo claro, pero no el más claro posible, que se obtendría con (255,255,0). La combinación de rojo y verde en una pantalla proporciona el amarillo.
- `p {color:rgb (255,255,255);}` es el blanco.
- `p {color:rgb (66,66,66);}` cuando los tres tonos se mezclan a partes iguales, obtenemos siempre un color gris, más o menos oscuro, dependiendo de lo elevados que sean los valores.

Observemos algunos de esos ejemplos en la imagen:



Todos los ejemplos anteriores también pueden ser expresados en valores porcentuales. Por ejemplo:

```
p {color: rgb (100%, 0%, 0%); }  
p {color: rgb (80%, 80%, 0%); }  
p {color: rgb (33%, 33%, 33%); }
```



Actividad 4

Identificaremos a qué colores se corresponden las reglas anteriores.

Realicemos además la incorporación de otros tres colores más. Pensaremos en un color y a continuación intentaremos formularlo mediante sus combinaciones de rojo, verde y azul. Al principio resulta difícil acertar pero, cuantas más veces probemos, más sencillo nos resultará.

Colores hexadecimales

Empleando la lógica del formato RGB, se emplea con frecuencia la definición de colores basada en valores hexadecimales, en lugar de decimales. Es decir, en vez de utilizar valores que van de 0 a 255, utilizamos valores que van de 00 a FF. Los códigos hexadecimales emplean los valores del 0 al 9, más la A, B, C, D, E y F.

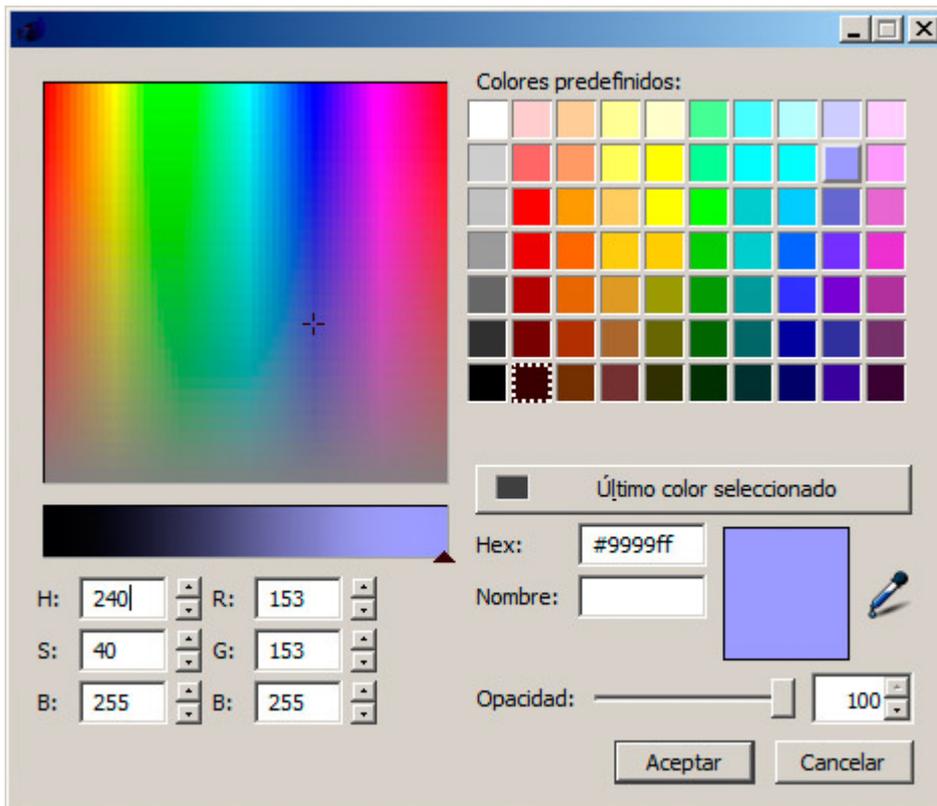
En este caso el número se precede del signo `#`. Los ejemplos anteriores quedarían así:

```
p {color: #ff0000; }  
p {color: #bbbb00; }  
p {color: #424242; }
```

Encontraremos este tipo de notación en muchos sitios. Produce unos códigos muy breves y que se descifran con rapidez, una vez que nos hemos acostumbrado a leerlos. De hecho, aún se pueden abreviar más, empleando un solo valor para definir cada color, pero reduce mucho el número de colores que podemos obtener, por lo que es mejor mantener el sistema de dos cifras para cada componente de color.

Definir un color con un editor web

Elegir los colores mediante códigos puede resultar costoso al principio. En este caso, los editores web serán de gran ayuda. Por ejemplo, si empleamos *BlueGriffon* para escoger un color determinado, desde el panel **Propiedades de estilo**, en la sección **Colores**, veremos que desde el cuadro desplegable de la figura podemos seleccionar un color, utilizando todos los métodos que hemos visto hasta ahora. En la parte inferior se encuentran los valores en RGB, en hexadecimal e incluso por su nombre, mientras que en la parte superior podemos seleccionar el color que más se aproxime a nuestras necesidades.



Ayuda para la definición de colores

En Internet encontramos muchas herramientas para definir conjuntos de colores, lo que se conoce como una **paleta de colores**. Más que mezclar colores al azar, es conveniente elegir colores siguiendo algunas reglas básicas. Hay toda una teoría de uso del color que nos permite saber qué colores combinan bien con cuáles.

Además de tomar como referencia aquellas páginas que nos gusten, podemos recurrir a herramientas, como este par de ejemplos:

- *Adobe Kuler*: <http://kuler.adobe.com/>
- *Color Blender*: <http://www.colorblender.com/>

La figura siguiente muestra un esquema concreto aplicado al documento. Hemos elegido un conjunto de colores y lo hemos ido aplicando a ciertos elementos de la página. Nos podrá gustar más o menos, pero el conjunto mantiene cierta armonía.



Pregunta Verdadero-Falso

Las siguientes afirmaciones, ¿son verdaderas o falsas?

El parámetro `color` indica el color del texto y `background-color` indica el color de fondo de diferentes elementos, entre ellos de un texto.

Verdadero Falso

Empleando la lógica del formato `RGB`, se emplean valores que van de 0 a 255, se utilizan los valores del 0 al 9, más la A, B, C, D, E y F.

Verdadero Falso



Actividad 5

Utilizando la propiedad `color` y `background-color`, definiremos los colores básicos para las cabeceras de nuestro texto, para el `<body>` y para los párrafos, buscando un modelo lo más armónico posible. Podemos definir los colores a mano o utilizando un editor web.

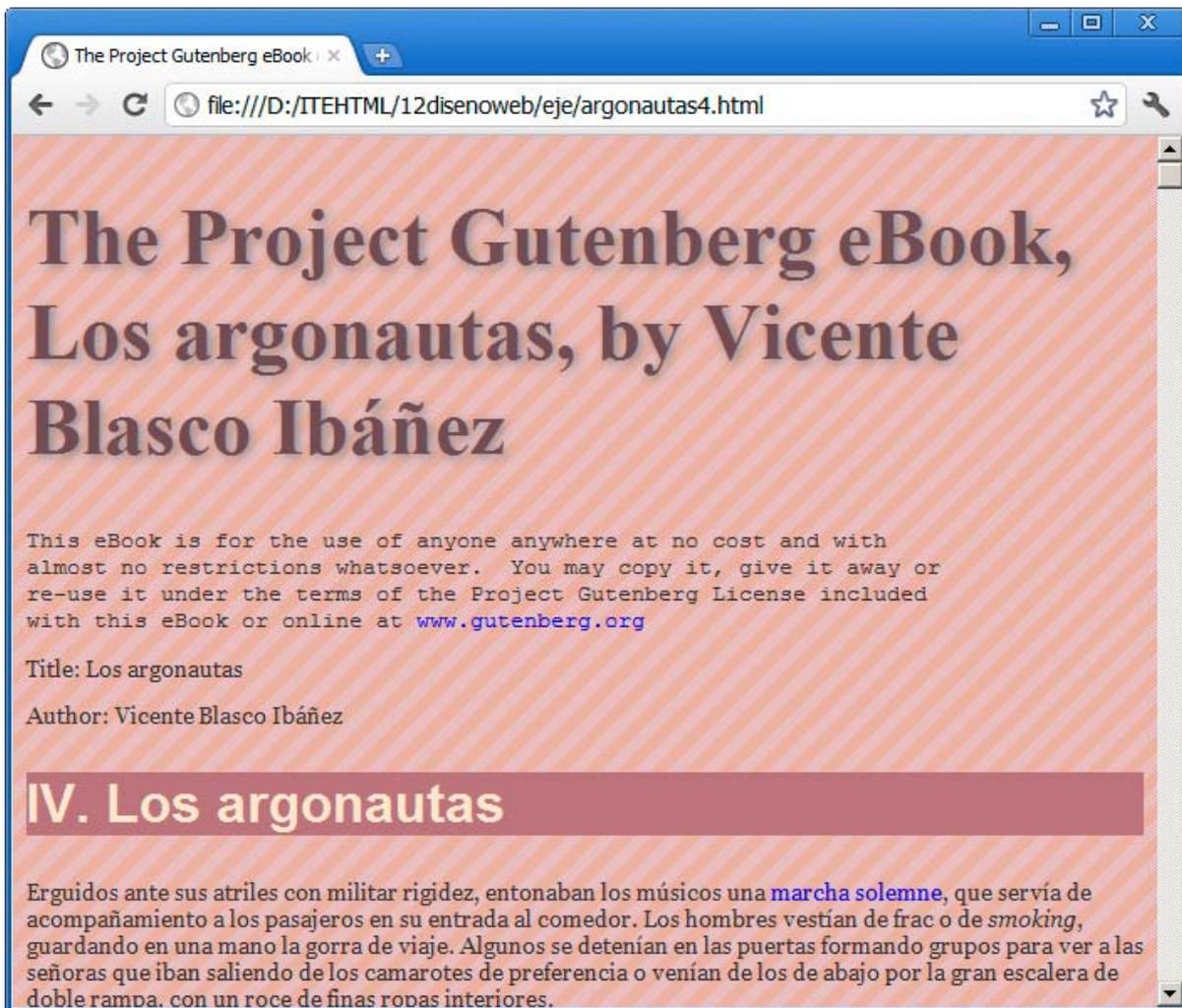
Imágenes de fondo

La propiedad **background-image** se puede aplicar a una gran cantidad de etiquetas y consigue que se muestre una imagen de fondo.

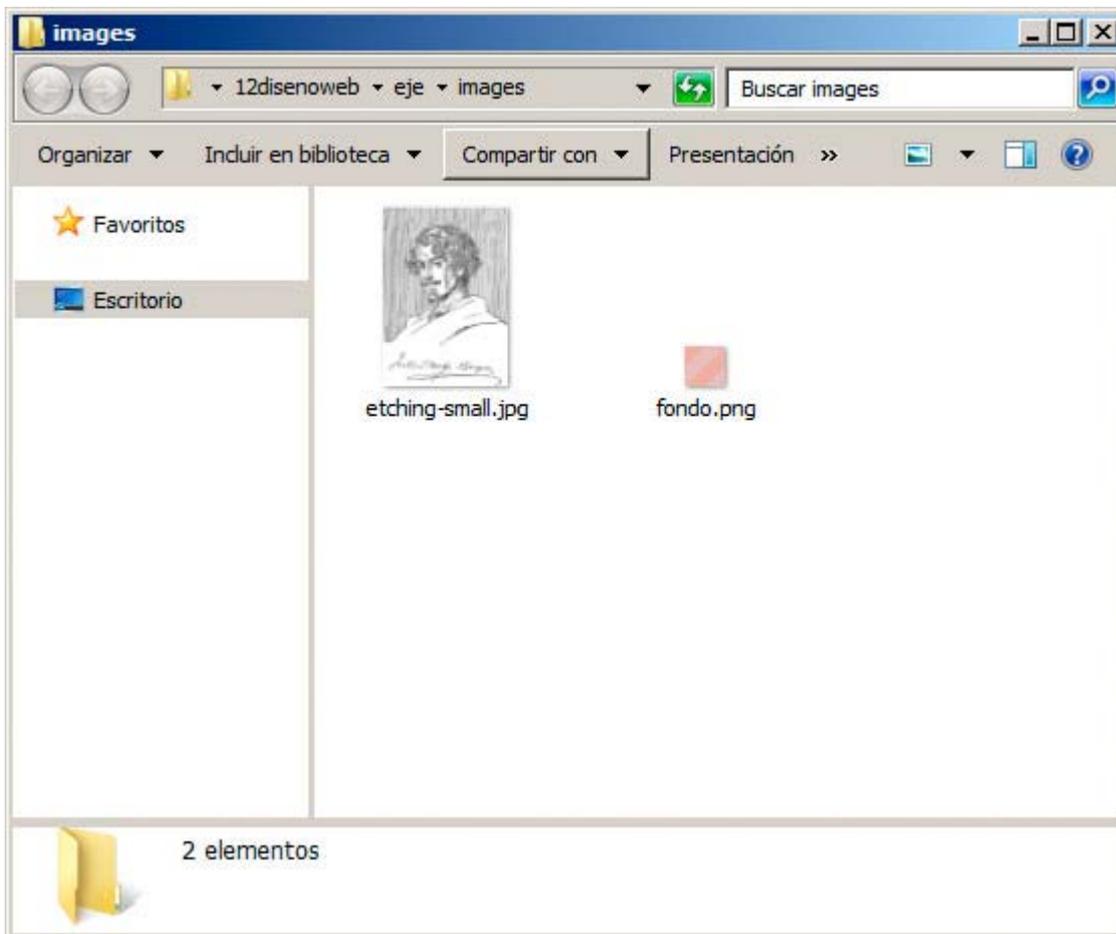
Su formato es el siguiente, aplicada por ejemplo al cuerpo de la página:

```
body { background-image: url("images/fondo.png"); }
```

El resultado sería el que se muestra en la figura.



Una imagen, en este caso tan pequeña como se puede observar en la figura siguiente, se toma como fondo de todo el documento, repitiéndose una y otra vez hasta cubrir toda la página.



Repeticiones

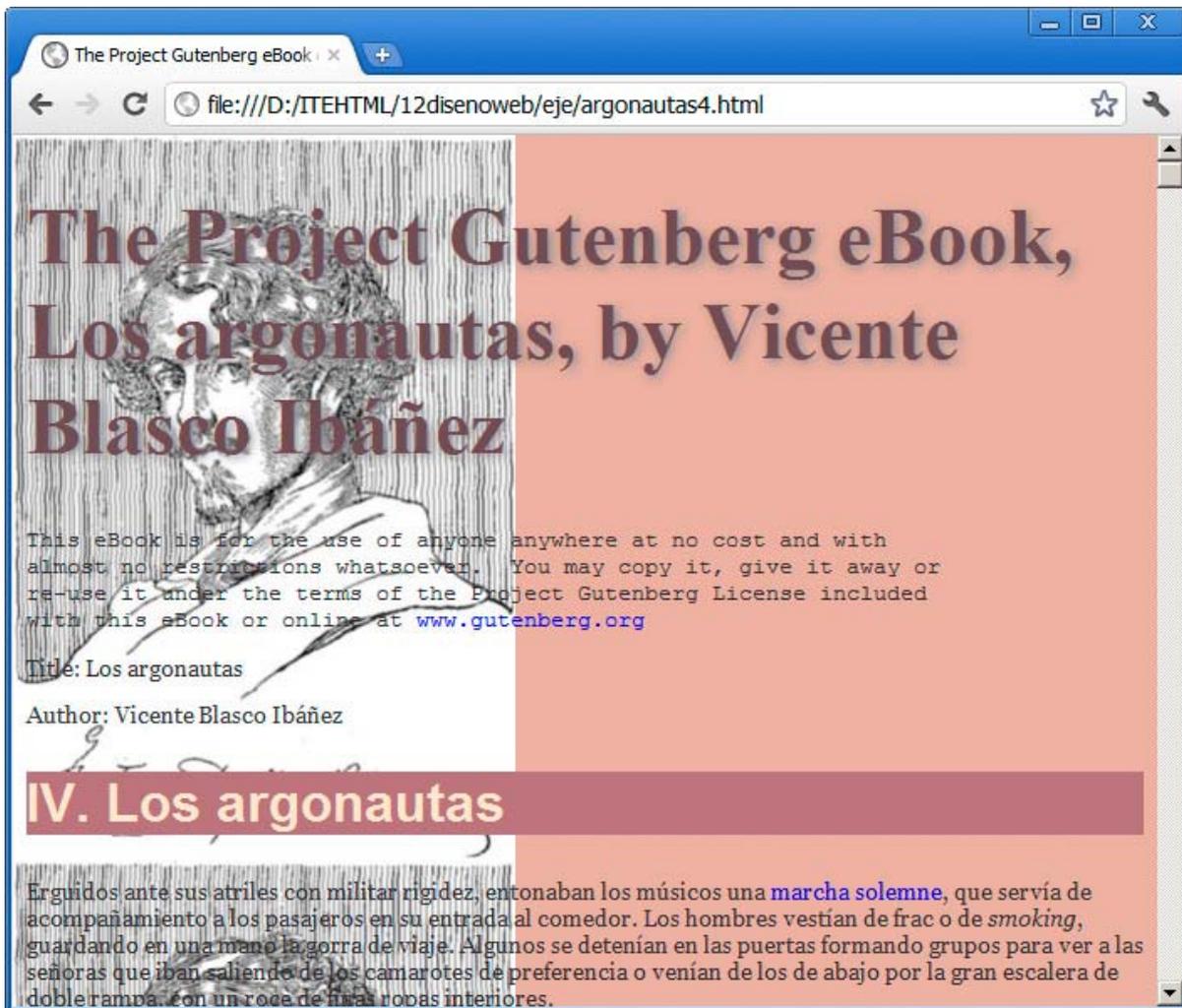
No siempre nos va a interesar que la imagen se reproduzca indefinidamente. Por ello, a continuación de la url podemos indicar el modo de repetición a través de la propiedad **background-repeat**:

- **repeat**: para que se repita. Es el valor predeterminado, por lo que lo podemos omitir.
- **repeat-x**: para que sólo se repita horizontalmente, no verticalmente.
- **repeat-y**: para que se repita en el eje vertical.
- **no-repeat**: para que no se repita nunca.

Por ejemplo, para que sólo se repita verticalmente añadiríamos esta línea:

```
body {  
    background-color: #DEB0A0;  
    background-image: url("images/etching-small.jpg");  
    background-repeat: repeat-y;  
}
```

El resultado es que la imagen sólo aparece en el eje vertical.



Fondo fijo

Todavía podemos mejorar el conjunto. En los casos en los que la imagen no debe ocupar todo el fondo podemos emplear la propiedad **background-attachment** para controlar la posición de la imagen, con los siguientes valores:

- **scroll**: la imagen se desplazará al desplazarnos arriba y abajo cuando el documento sea largo. Es el valor predeterminado.
- **fixed**: la imagen se queda fija aunque nos desplazemos hacia abajo.
- **local**: similar al anterior, aunque la imagen está fijada al contenido del elemento en lugar de al propio elemento.

Para comprender el uso de esta propiedad, lo mejor es probar esta regla:

```
body {  
    background-color: #DEB0A0;  
    background-image: url("images/etching-small.jpg");  
    background-repeat: no-repeat;  
    background-attachment: fixed;  
}
```

Al bajar en el documento veremos como la imagen se queda fija en el fondo, en lugar de desaparecer, que sería el comportamiento estándar.

Colocación del fondo

La propiedad **background-position** se emplea para ubicar el fondo a nuestro gusto, mediante los valores x e y, ya sea en formato de píxeles, mediante porcentajes o mediante indicaciones de posición (**center**, **top**, **bottom**, **right** o **left**).

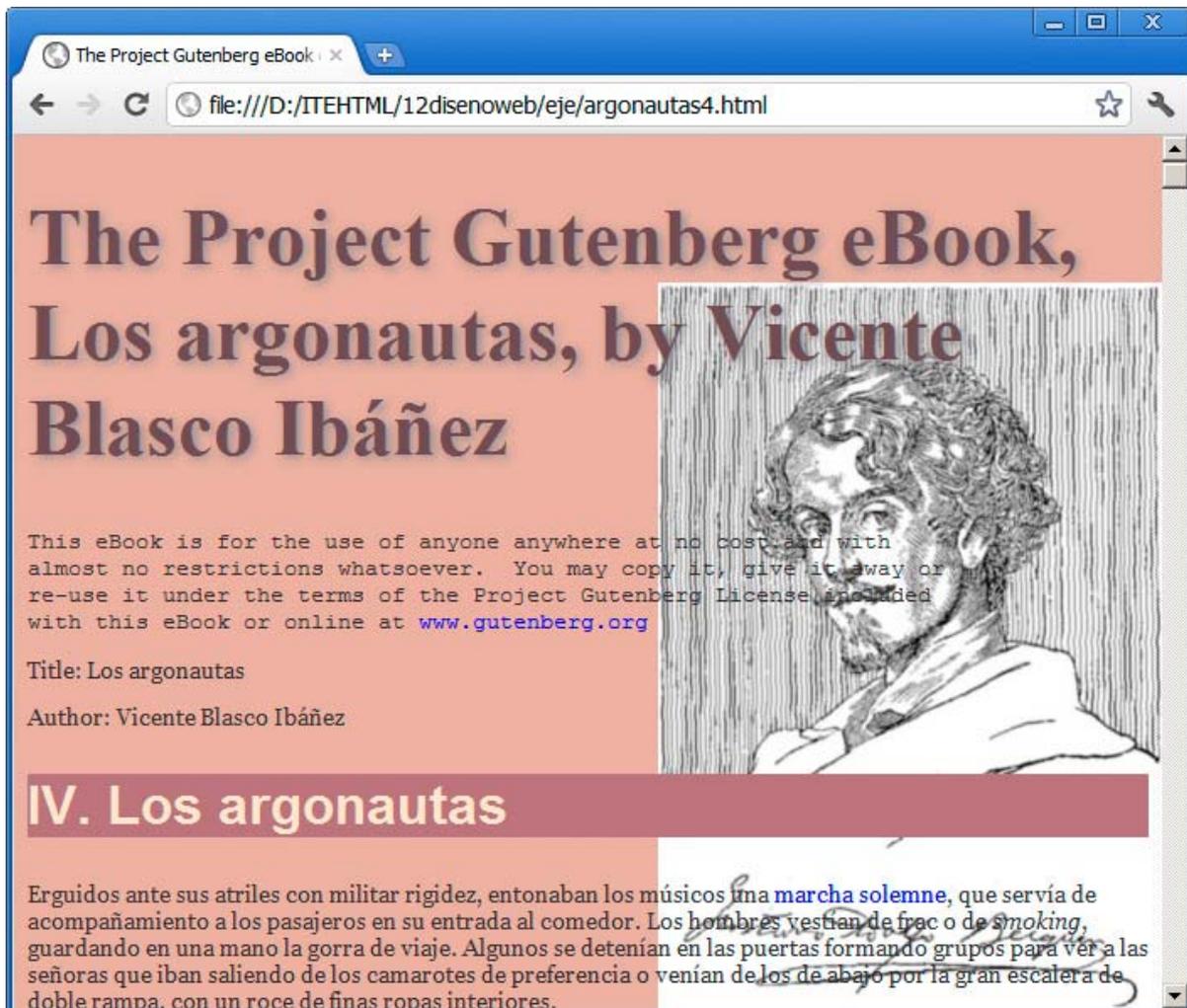
Por ejemplo, probaremos esta regla:

```

body {
    background-color: #DEB0A0;
    background-image: url("images/etching-small.jpg");
    background-repeat: no-repeat;
    background-attachment: fixed;
    background-position: bottom right;
}

```

La imagen se ha desplazado a la esquina inferior derecha (**bottom right**), como se muestra en la figura.



Si estamos empleando una imagen como fondo de un bloque o un elemento similar y la imagen es más grande que el elemento que la va a contener podemos emplear los valores `cover` (ajusta la imagen al bloque sin deformarla) o `contain` (reduce la imagen para ajustarla al bloque, también sin deformarla).

La propiedad `background`

Como sucedía con la propiedad `font`, podemos fusionar todas las propiedades relacionadas con los fondos en una sola mediante `background`.

El ejemplo que hemos venido realizando se podría expresar también de esta manera:

```

body {
    background: #DEB0A0 url("images/etching-small.jpg") no-repeat fixed bottom right;
}

```

En primer lugar indicamos el color de fondo, si no queremos que aparezca ningún color indicaremos el término **transparent**. Luego viene la dirección de la imagen, seguida de los diferentes parámetros que hemos visto.



Actividad 6

Incorporaremos nuestro propio fondo, probando a colocarlo en diferentes lugares y comprobando lo que sucede al cambiar de tamaño la ventana del navegador y al desplazarnos con el documento. Intentaremos decidir qué tipo de ajuste favorece más la legibilidad del texto y cuál menos.

No sólo para body

Las prácticas que hemos realizado se han centrado en aplicar fondos a la etiqueta `<body>`; tenemos que tener en mente que esto no tiene por qué ser así. Es muy habitual aplicar imágenes de fondo a pequeños bloques definidos por un `<div>`, e incluso a cualquier tipo de cabecera del documento.



Pregunta de Elección Múltiple

Para utilizar una imagen como fondo y que ésta no se repita, debemos utilizar las etiquetas:

- `background-image`.
- `background-no-repeat`.
- `background-attachment`.
- Ninguna de las anteriores es correcta.

Opacidad

Todos los navegadores actuales pueden aplicar la propiedad **opacity**. Ésta va seguida de un valor entre cero y uno, que indica el nivel de transparencia que tendrá el elemento al que se le aplique (cero sería transparente y uno sería completamente opaco). Por ejemplo:

```
div.grupouno {opacity:.2}
div.grupodos {opacity:.6}
div.grupotres {opacity:.8}
```

La figura muestra el resultado de aplicar diferentes niveles de opacidad a un grupo de elementos.



Apariencia del cursor

Un último recurso gráfico lo obtenemos con la propiedad **cursor**. Asignando diferentes valores, podremos conseguir que el puntero del ratón tome la apariencia que queramos entre una extensa lista de posibilidades:

```
img.botones {cursor:hand;}  
span.ayuda {cursor:help;}  
div.esperar {cursor:wait;}
```

En los tres ejemplos anteriores conseguimos que el cursor se convierta, respectivamente, en una mano al pasar sobre una imagen de la clase **botones**, el icono de **ayuda** al pasar sobre un **span** etiquetado como tal y el **reloj** de arena habitual, al pasar sobre un bloque de la clase **esperar**.

De todas las opciones posibles, sin duda la de **hand** es la más empleada para simular botones con imágenes, pero también tenemos otras muchas posibilidades que se describen por si mismas, como **text**, **default** (el predeterminado), **progress**, **move**, **crosshair** (una cruz), etc. Incluso podemos definir nuestro propio puntero indicando una URL, como hacíamos con las imágenes o con las tipografías.

Resumen



En este módulo hemos revisado una abundante cantidad de propiedades. Con todas ellas podremos definir la apariencia de nuestras páginas web y controlar hasta los más mínimos aspectos.

No nos resultará necesario recordar todas las propiedades, ya que contaremos con el apoyo de diferentes aplicaciones para diseñar nuestras hojas de estilo, pero sin duda es muy necesario conocerlas, para poder realizar los ajustes y modificaciones que surgen de cuando en cuando.

Eso sí, en el día a día podemos hacer uso de alguna aplicación como *BlueGriffon* para aplicar los estilos. Las opciones que hemos visto a lo largo de este módulo se encuentran recogidas en su mayoría en las pestañas **General** y **Colores** del panel **Propiedades de estilo**, recogido en la figura y que se despliega mediante la opción **Paneles>Propiedades de estilo**.



Actividades y ejemplos



Actividad 1. Propiedades de texto

Localizaremos una tipografía libre en Internet y se la aplicaremos a varias etiquetas del texto, empleando el método visto anteriormente. Comprobaremos su funcionamiento en al menos dos navegadores.

Se pueden encontrar tipografías libres mediante cualquier buscador. También podemos recurrir a algún sitio web concreto dedicado a ello, como <http://openfontlibrary.org/> o <http://www.theleagueofmoveabletype.com>.



Actividad 2. Propiedades de texto

Utilizando valores **em** definiremos diferentes parámetros para nuestra web de ejemplo. Comprueba lo que sucede si a un elemento le asignamos 0em como tamaño o 1em.



Actividad 3. Más propiedades de texto

En un documento con varios títulos realizaremos diferentes incorporaciones a la hoja de estilos, para conseguir que

proyecten algunas sombras. Debemos conseguir al menos una sombra muy marcada y precisa y otra tan sutil que sea casi inapreciable. ¿Cuál consideras que proporciona más elegancia al texto?



Actividad 4. Colores

Identificaremos a qué colores se corresponden las siguientes reglas:

```
p {color: rgb (100%, 0%, 0%); }
```

```
p {color: rgb (80%, 80%, 0%); }
```

```
p {color: rgb (33%, 33%, 33%); }
```

Realicemos además la incorporación de otros tres colores más. Pensemos en un color y a continuación intentaremos formularlo mediante sus combinaciones de rojo, verde y azul. Al principio resulta difícil acertar pero, cuantas más veces probemos, más sencillo nos resultará.



Actividad 5. Colores

Utilizando la propiedad **color** y **background-color**, definiremos los colores básicos para las cabeceras de nuestro texto, para el **<body>** y para los párrafos, buscando un modelo lo más armónico posible. Podemos definir los colores a mano o utilizando un editor web.



Actividad 6. Imágenes de fondo

Incorporaremos nuestro propio fondo, probando a colocarlo en diferentes lugares y comprobando lo que sucede al cambiar de tamaño la ventana del navegador y al desplazarnos con el documento. Intentaremos decidir qué tipo de ajuste favorece más la legibilidad del texto y cuál menos.



Ejemplos

Las diferentes prácticas, recursos y ejemplos, realizadas en este módulo, están disponibles para realizar pruebas:

Ejemplos del módulo

Aplicación al aula

Diseño web con estilos

El alumnado tomará un texto y lo transformará de diferentes maneras, empleando las hojas de estilo.



Programación dirigida al alumnado

Objetivos

- Aplicar los conceptos básicos de las hojas de estilo en cascada.
- Modificar una hoja de estilos externa.
- Transformar un documento mediante los estilos.

Contenidos

- Hojas de estilo.
- Estilos de texto y tipografías.
- Estilos de colores y fondos.
- Creación de estilos con un editor web.

Materiales y recursos

- Ordenador con acceso a Internet.

Temporalización

- Dos sesiones.

Planificación



El alumnado aplica los conceptos básicos de la creación de estilos a través de la modificación de una página web predeterminada, probando e investigando los resultados de la utilización de cada estilo.

Organización del aula

Trabajaremos en un aula con ordenadores con un agrupamiento individual o por parejas.

Desarrollo de la actividad

- Se repasan los conceptos básicos de creación de hojas de estilos en cascada.
- Se toma el texto indicado y se realizan las modificaciones.
- Se prueban diferentes estilos, obteniendo resultados variados.
- Se comparten los resultados.

Presentación y evaluación de los resultados

La evaluación se realizaría mediante la revisión del resultado y la observación del proceso. Se pueden evaluar varios

aspectos a lo largo de todo el proceso:

- Capacidad para aplicar los estilos.
- Inserción apropiada de los estilos en los lugares indicados.
- Apariencia de los cambios aplicados en el documento.

Sugerencias metodológicas



La metodología empleada es la de **proyecto**.

Para su aplicación proponemos:

Primera sesión

Explicamos el objetivo de la actividad y describimos los conceptos necesarios.

Les entregaremos dos documentos con un texto largo.

Generaremos una hoja de estilos externa y comenzaremos a trabajar con el primer texto.

Aplicaremos algunos estilos generales a toda la página.

Segunda sesión

Retomando el ejemplo anterior, continuamos aplicando estilos.

Definimos un conjunto de colores armónicos y los aplicamos al documento.

Probamos algunas técnicas avanzadas más.

Para terminar, se aplica la hoja de estilos al segundo documento y se revisan los resultados.

Atención a la diversidad



Actividad de refuerzo

Para aquellos alumnos/as que puedan tener más dificultad, se les facilita, junto al documento, una hoja de estilos vacía.



Actividad de ampliación

La profundización en esta actividad se basaría en que el alumnado probase algunas propiedades más complejas, como cambiar las tipografías, imágenes de fondo, etc.
