Formación en Red

Ministerio de Educación, Cultura y Deporte.

HTLM5 en la educación

Módulo 13: Técnicas avanzadas.

Instituto Nacional de Tecnologías Educativas y de Formación del profesorado 2012

GOBIERNO DE ESPAÑA MINISTERIO DE EDUCACIÓN, CULTURA Y DEPORTE
SECRETARÍA DE ESTADO DE EDUCACIÓN, FORMACIÓN PROFESIONAL Y UNIVERSIDADES
intef INSTITUTO NACIONAL DE TECNOLOGÍAS EDUCATIVAS Y DE FORMACIÓN DEL PROFESORADO

Técnicas avanzadas

Una solución para cada problema

Decir que vamos a tratar algunas técnicas avanzas es, a todas luces, un atrevimiento. El mundo de las páginas web está avanzando tanto y a tal velocidad, que es muy difícil determinar si una técnica es o no avanzada. Con este epígrafe pretendemos, más bien, agrupar algunos elementos que no han encontrado acomodo en otros lugares, técnicas algo más complejas y etiquetas que están emprendiendo un camino prometedor y de las que merece la pena hablar al menos.

Programación

Objetivos específicos

- Crear enlaces internos.
- Crear imágenes de sustitución.
- Crear mapas sensibles.
- Diseñar imágenes con varias técnicas.

Contenidos

- Enlaces internos.
- Imágenes de sustitución.
- Mapas sensibles.
- Canvas.
- Imágenes SVG.
- Favicon.

Criterios de evaluación

- Crear enlaces internos.
- Diseñar imágenes de sustitución.
- Añadir mapas sensibles a una página web.
- Añadir imágenes con canvas o SVG.

Requisitos mínimos

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

Recurso TIC: Técnicas avanzadas

Seguiremos avanzando en nuestro conocimiento de las propiedades más empleadas de las hojas de estilo, centrándonos ahora en aquellas que conllevan posicionamiento de objetos, incluyendo algunas que definen su apariencia, pero más en un ámbito espacial.

Enlaces internos

La creación de enlaces internos dentro de una página web nos permite realizar saltos, dentro de una misma página, a diferentes lugares de la misma. Es una técnica imprescindible si creamos páginas muy largas y simplemente innecesaria para páginas cortas. En todo caso es necesario conocerla.

Hay un ejemplo muy popular de enlaces internos, que es el uso que hace de ellos la archiconocida web http://www.wikipedia.es. Cada artículo de esta enciclopedia *online* suele comenzar con un pequeño índice de secciones, que nos sirve para saltar a diferentes partes del documento que estamos leyendo. Además, al final de cada sección encontramos un enlace para regresar de nuevo al principio.

Posiciones enlazables

En primer lugar, antes de poder saltar a un punto específico de una página, necesitamos definir qué puntos serán susceptibles de esos saltos. Para ello emplearemos la etiqueta **<a>**, pero seguida del parámetro **name**, así:

```
<a name="capitulo3"></a>
```

En la etiqueta podemos colocar algún otro elemento HTML, por ejemplo:

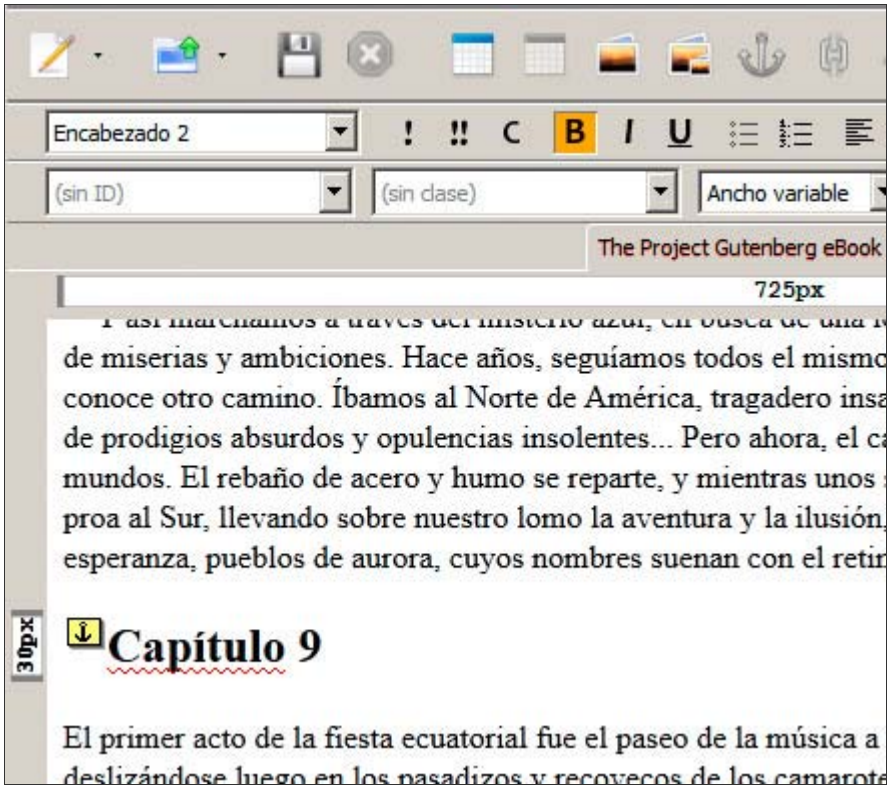
```
<a name="capitulo3"><h2>Capítulo 3</h2></a>
```

o de la siguiente manera:

```
<h2>Capítulo 3<a name="capitulo3"></a></h2>
```

Las dos son válidas.

En un editor web como *BlueGriffon* este tipo de elementos se insertan mediante el icono en forma de ancla (la etiqueta **a** viene de *anchor*, "ancla" en inglés). La figura muestra una etiqueta **<a name>** insertada en el editor.



Salto a enlaces internos

Ahora necesitamos saber cómo podremos acceder a esos enlaces internos. Es similar a cualquier referencia de las que hemos hecho hasta ahora, pero añadiendo el nombre del enlace interno precedido del signo **#**. Por ejemplo, para acceder a uno de estos enlaces desde la misma página, añadiríamos algo así:

```
<a href="#capitulo3">Saltar al capítulo 3</a>
```

o desde una página diferente, añadiríamos primero el nombre de la página:

```
<a href="organizacion.html#equipo">Consultar el equipo</a>
```

Saltos a la parte superior

Es muy habitual añadir en páginas largas un enlace similar a éste:

```
<a name="inicio">
```

y a lo largo del documento añadir alguna etiqueta de este tipo:

```
<a href="#inicio">Regresar a la parte superior</a>.
```

Así el usuario tiene un método rápido para subir hasta la parte de arriba de la página.

Pregunta Verdadero-Falso

La siguiente afirmación, ¿es verdadera o falsa?

Para poder saltar a un punto específico de una página, necesitamos definir qué puntos serán susceptibles de esos saltos, para ello emplearemos la etiqueta **<a>**, pero seguida del parámetro *name*.

Verdadero ○ Falso ○

Actividad 1

En un documento largo crearemos un par de enlaces a lo largo del documento, para regresar a su parte superior.

Imágenes de sustitución

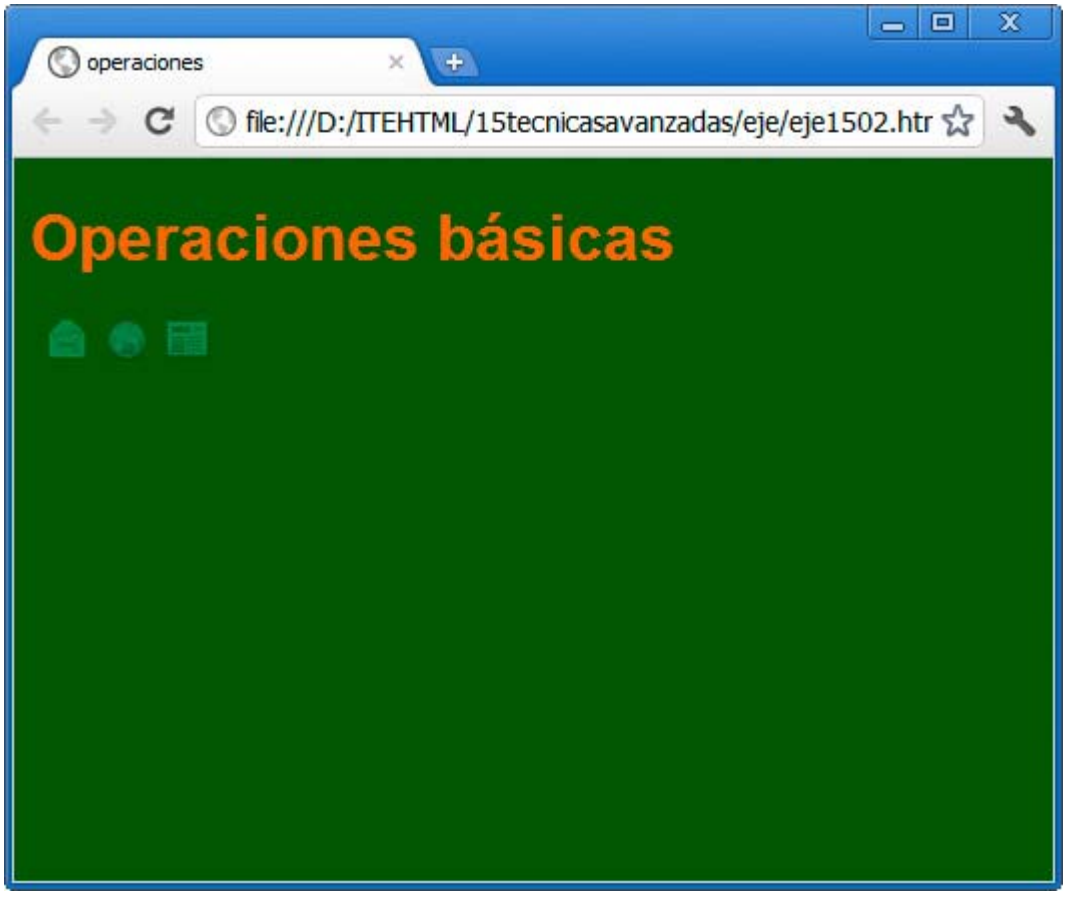
Ésta es otra técnica muy empleada, que consiste en presentar una imagen determinada en una página web y en reemplazarla por otra diferente, al pasar el puntero del ratón sobre ella. Se puede utilizar para destacar elementos, para revelar recursos ocultos, etc.

Su uso nos ayuda a profundizar en algunos conceptos que ya hemos comentado, como la gestión de eventos.

Veamos su funcionamiento mediante un ejemplo. La figura nos muestra seis imágenes que vamos a emplear para este efecto. Tres de ellas están ligeramente difuminadas, mientras que las otras tienen todo su color.

Vamos a dejar en una página las tres difuminadas y hacer que, cuando el usuario pase el ratón sobre ellas, parezca que se resalten, reemplazando las primeras por las segundas.

Ésta es la página de partida:



Y éste es el código incluido en el **body:**

```
<body>
```

```
        <h1>Operaciones b&aacute;sicas</h1>
        <p>
         <img alt="" src="resources/Internet-mail-off.png" />
         <img alt="" src="resources/Internet-web-browser-off.png" />
         <img alt="" src="resources/Internet-news-reader-off.png" />
        </p>

    </body>
```

Todo es conocido y sencillo para nosotros. Las imágenes están guardadas en una carpeta llamada **resources** y en su nombre hemos añadido un **-off** para diferenciarlas de las que se ven con todos los colores.

Como sabemos, podemos añadir eventos a cualquier etiqueta HTML. En este caso, para poder saber si el ratón pasa por encima, tenemos el evento **onmouseover**, así como **onmouseout** para saber cuándo deja de estar encima. Por tanto, debemos añadir a cada imagen esos dos eventos:

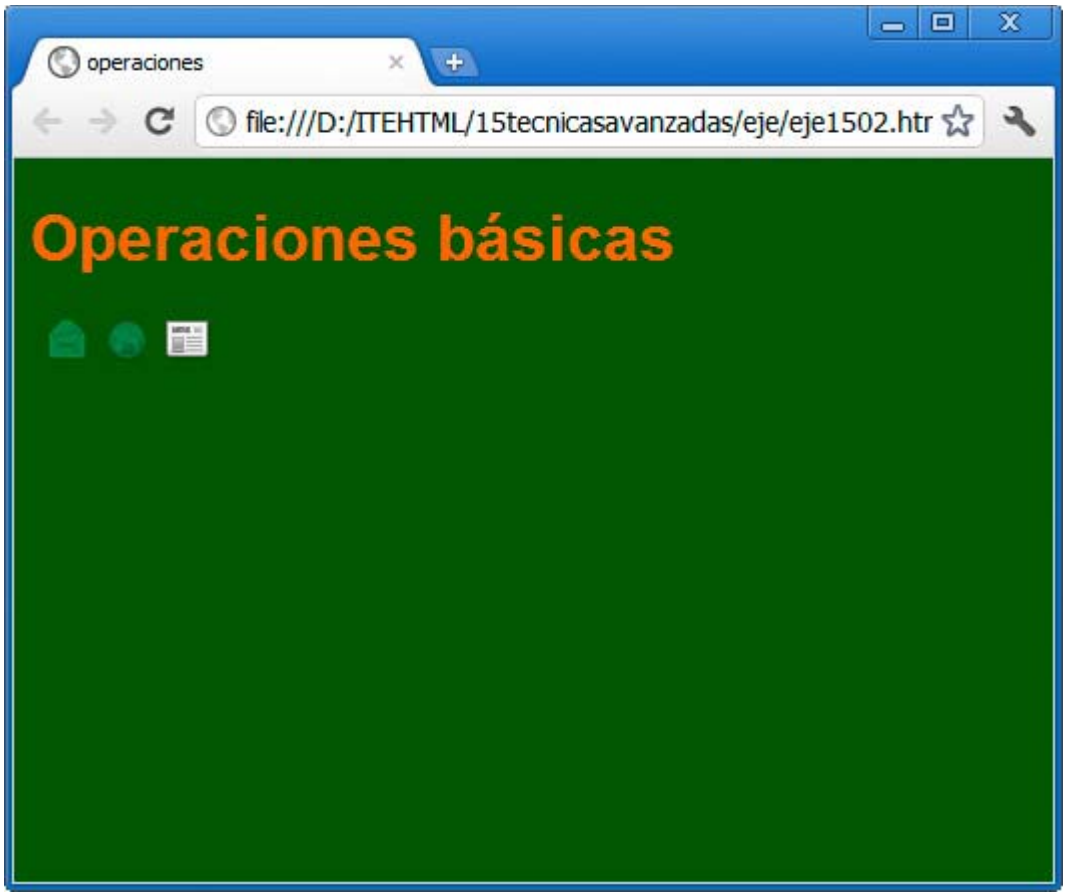
Las tres líneas de las imágenes quedarían así:

```
<img alt="" src="resources/Internet-mail-off.png" onmouseover="this.src = 'resources/Internet-mail.png'" onmouseout="this.src = 'resources/Internet-mail-off.png'" />
<img alt="" src="resources/Internet-web-browser-off.png" onmouseover="this.src = 'resources/Internet-web-browser.png'" onmouseout="this.src = 'resources/Internet-web-browser-off.png'" />
<img alt="" src="resources/Internet-news-reader-off.png" onmouseover="this.src = 'resources/Internet-news-reader.png'" onmouseout="this.src = 'resources/Internet-news-reader-off.png'" />
```

Son largas, pero muy sencillas de entender: con ese **this.src** le estamos indicando al navegador que, al pasar el ratón por encima o abandonarlo (según el evento), debemos sustituir el valor **src** de ese elemento por la imagen que se indica.

this (traducido al español sería "éste") es un elemento del lenguaje *JavaScript* que nos sirve para indicar el elemento en el que nos encontramos.

Si probamos de nuevo el ejemplo, veremos que cada icono se destaca al pasar el ratón por encima. La siguiente figura muestra ese momento.



Hay otras muchas técnicas para hacer esta sustitución, aunque ésta es una de las más rápidas y sencillas.

Pregunta Verdadero-Falso

La siguiente afirmación es, ¿verdadera o falsa?

Con el código **this.src** estamos indicando al navegador que, al pasar el ratón por encima o abandonarlo, debemos sustituir el valor **src** de ese elemento por un archivo en pdf que se indica.

Verdadero ○ Falso ○

Actividad 2

Tomaremos dos fotografías y colocaremos una en la parte central de la página. Debemos conseguir que, cuando el usuario haga clic sobre ella, se reemplace por la segunda. El evento que se emplea para esto se llama **onclick**.

Mapas sensibles

Con HTML podemos utilizar una única imagen y definir zonas en su interior de tal modo que, cuando el usuario pase el ratón sobre esas zonas, podamos enlazar con varios sitios. Se emplea mucho para crear mapas o grandes imágenes desde las que saltar a diferentes lugares.

Su realización se basa en usar una imagen normal y corriente, pero añadiéndole el parámetro **usemap** seguido del nombre de un mapa. Por ejemplo:

```
<img src="panoramicaparis" usemap="#paris" />
```

Esa imagen se acompañará de una serie de coordenadas, que se engloban con la etiqueta **<map>**. Observemos este mapa típico:

```
<map name="paris">
<area shape="rect" coords="60,250,140,400" alt="Torre Eiffel" href="http://es.wikipedia.org/wiki/Torre_Eiffel" />
<area shape="rect" coords="240,290,260,350" alt="Arco de la Defénse" href="http://es.wikipedia.org
/wiki/Arco_de_la_Defensa" />
<area shape="circle" coords="100,450,40" alt="Campo de Marte" href="http://es.wikipedia.org
/wiki/Campo_de_Marte_%28Par%C3%ADs%29" />
</map>
```

La etiqueta **<area>** se acompaña de varios parámetros:

- **shape** para indicar el tipo de área que se va a definir. Usaremos **rect** para rectángulos (seguido de cuatro coordenadas x1,y1,x2 e y2), **circle** para círculos (más x1,y1 y el radio) o **poly** (seguido de una serie de coordenadas x1,y1,...xn,yn).
- **coords** para indicar las coordenadas, según el tipo de figura.
- **alt** para añadir un texto alternativo que no se mostrará; sólo se introduce por motivos de accesibilidad.
- **href** para establecer una dirección web a la que accederemos al hacer clic sobre el área.

Si queremos que se muestre algún rótulo sobre el área, debemos añadir junto a **alt** un parámetro **title**, como hemos hecho en la figura.

Pregunta de Elección Múltiple

La etiqueta **<area>** se acompaña de varios parámetros:

- *shape*, *rect*, *circle* o *poly*.
- *coords, alt* y *href*.
- Ninguna de las anteriores es correcta.
- A y b son correctas.

Nota

Para el cálculo de las coordenadas podemos emplear cualquier programa de dibujo, trazando las diferentes áreas y viendo qué valores **x** e **y** nos arrojan.

Actividad 3

Realizaremos un pequeño mapa basado en la imagen de un reloj y convertiremos algunas horas en enlaces a páginas de un calendario. Para redondear la actividad, podríamos hacer que todos los enlaces fuesen a diferentes secciones de un mismo documento, empleando los enlaces internos.

Canvas

HTML5 incorpora un interesante recurso denominado **<canvas>**. Esta etiqueta genera un espacio en la página web en el que se puede dibujar, empleando instrucciones creadas con *JavaScript*.

El **lienzo** (*canvas* en inglés) se define así:

```
<canvas id="canvas01" width="400" height="300"></canvas>
```

Con el sistema habitual de dimensiones especificaremos su tamaño. Al probar esa página web, veremos una gran mancha transparente, es decir, nada de nada.

Aquí cobra importancia el identificador, porque es la parte que nos va a permitir hacer referencia a ese **<canvas>** concreto desde *JavaScript*, de la siguiente manera:

```
var milienzo = documentgetElementByID("canvas01");
```

Esta instrucción en *JavaScript* averigua qué elemento de todos los presentes en el *DOM* (el modelo de objetos del documento) se corresponde con el identificador *canvas01* y guarda la referencia en la variable *milienzo*. Éstos son detalles de programación en *JavaScript* que se repiten siempre de la misma manera y que nos permitirán operar con ese valor, con *milienzo*. Ahora vendría una línea como ésta:

```
var micontexto = milienzo.getContext("2d");
```

que de nuevo es una línea que se repite siempre igual, cada vez que queramos manipular un lienzo. Define un espacio de dibujo en dos dimensiones, en previsión de que en un futuro (ahora no) podamos realizar dibujos en 3D dentro del navegador.

A dibujar

Tenemos por tanto una pequeña función que va a comenzar siempre con las dos líneas anteriores; así:

```
function dibujar( ) {
    var milienzo = document.getElementById("canvas01");
    var micontexto = milienzo.getContext("2d");
}
```

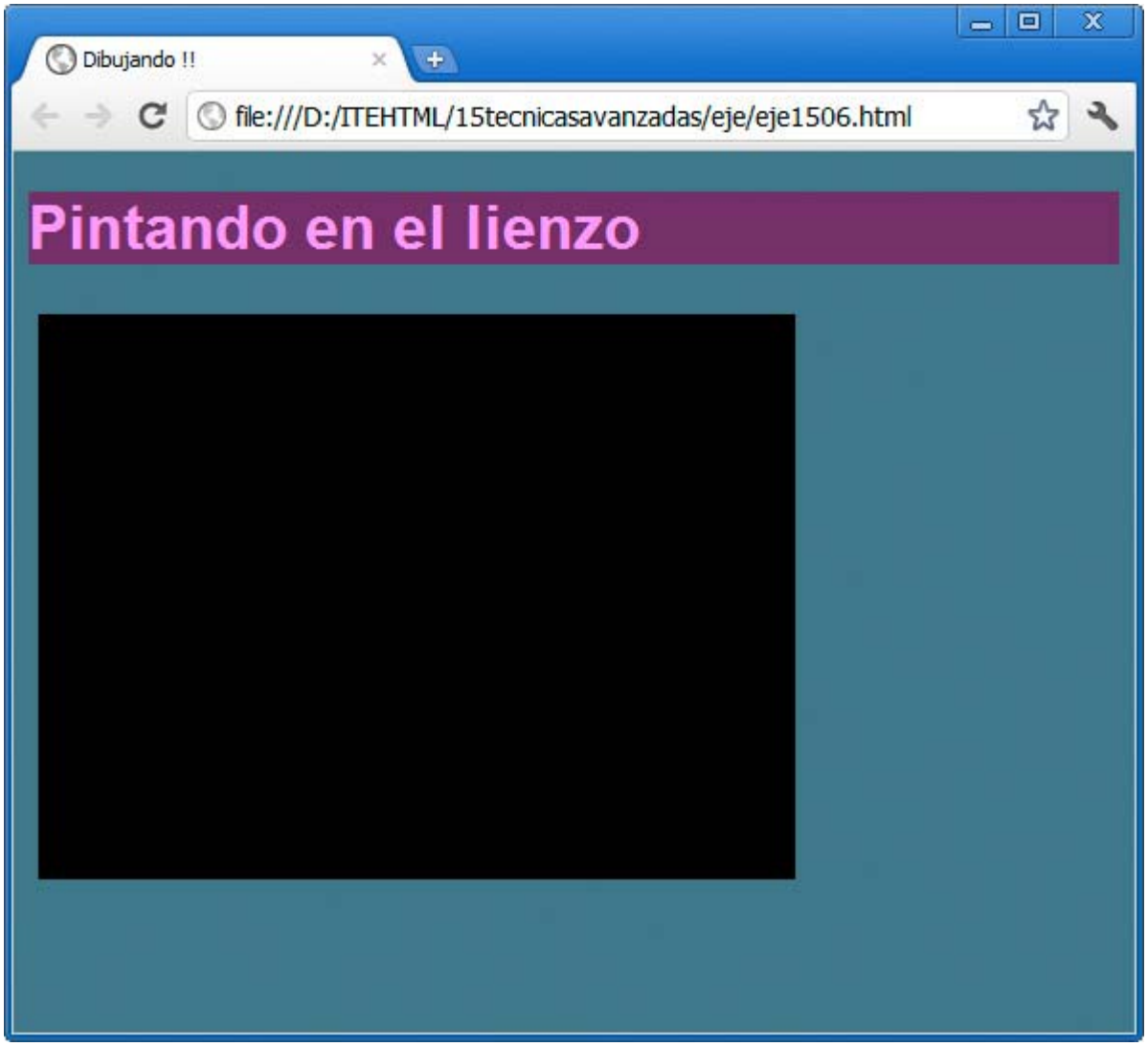
A la que sólo falta que le incorporemos algo visual. Tenemos varias funciones a nuestra disposición. Por ejemplo, el método **fillRect** permite dibujar un rectángulo que va de una posición inicial (x1,y1) seguido de un ancho y un alto. Teniendo en cuenta que la esquina superior izquierda del lienzo es la posición 0,0, es fácil deducir que nuestro lienzo concreto acaba en la posición 400,300. Probaremos a realizar un rectángulo que vaya casi rodeando el lienzo:

```
micontexto.fillRect(5,5,390,290);
```

Nota

Ojo con las mayúsculas. En *JavaScript* no es lo mismo escribir **fillrect** que **fillRect**. La primera no funcionará. Sí. Así de difícil les gusta hacer las cosas a los creadores de los lenguajes de programación.

El resultado del ejemplo anterior se muestra en la figura:



Para llegar a ese punto, debemos poner cada cosa en su sitio. Veamos el código de la página completo:

```
<!DOCTYPE html>
<html>
<head>
<meta content="text/html; charset=UTF-8" http-equiv="Content-Type">
<title>Dibujando !!</title>
<script type="text/JavaScript">

        function dibujar() {
        var milienzo = document.getElementById("canvas01");
        var micontexto = milienzo.getContext("2d");
        micontexto.fillRect(5,5,390,290);
        }

</script>
</head>

<body onload="JavaScript:dibujar();">

        <h1>Pintando en el lienzo</h1>
        <canvas id="canvas01" width="400" height="300"></canvas>

</body>
</html>
```

Sólo hemos retirado del ejemplo algunos estilos; lo demás está todo ahí. Nos interesa prestar atención a dos partes:

- Por un lado está la función completa, llamada **dibujar**, que hemos definido en la cabecera de la página con el método estándar, usando la etiqueta **<script>**
- Por otro lado, para que la función se ejecute al cargar la página, hemos establecido una llamada en la etiqueta **<body>**, para cuando se produzca el evento **onload**, al cargar.

En la función ya sólo necesitaremos ir añadiendo nuevas líneas debajo de **fillRect,** para modificar el dibujo y realizar más operaciones.

Más opciones de dibujo

- **fillStyle**: establece el formato del relleno de la figura, indicado como un color, un diseño o incluso un degradado de color.
- **strokeRect (x,y,ancho,alto)** dibuja un rectángulo vacío; sólo su contorno.
- **strokeStyle**: establece el formato de línea del elemento anterior.
- **clearRect (x,y,ancho, alto)**: borra los píxeles de las dimensiones indicadas.

El dibujo, por tanto, es muy diferente si cambiamos el **fill** por un **stroke**; probamos esta secuencia:

```
function dibujar() {

    var milienzo = document.getElementById("canvas01");
    var micontexto = milienzo.getContext("2d");
    micontexto.strokeStyle="#3333ff";
    micontexto.strokeRect(5,5,390,290);

}
```

Tanto **fillStyle** como **strokeStyle** son lo que se conoce como **propiedades**, es decir, que internamente no son funciones y por tanto su valor no se le indica mediante paréntesis, sino que se le asigna un valor mediante el signo igual. Todo esto se debe a que *JavaScript* es un lenguaje orientado a objetos; *micontexto* es un objeto que tiene algunas funciones (llamadas *métodos*) que realizan operaciones y también *propiedades*, es decir, características que podemos modificar asignándole algún valor.

Nota

La propiedad de **relleno** se debe establecer antes de realizar el dibujo, para que así el dibujo aparezca con el color indicado. Así podemos definir distintos objetos con colores variados.

Actividad 4

Probaremos a realizar mezclas con varios rectángulos rellenos y sin rellenar con diferentes colores. Podemos añadir tantas líneas como sea necesario a la función.

Como se puede observar, se abren un montón de posibilidades. El límite lo impondrá nuestro conocimiento de *JavaScript*. Cuanto más profundicemos en este lenguaje, más se abrirá nuestro horizonte. Observemos este otro pequeño código:

```
function dibujar() {

    var milienzo = document.getElementById("canvas01");
    var micontexto = milienzo.getContext("2d");
    var i=0;
    var x=5;
    var y=5;
    var ancho = 390;
    var alto = 290;

    for (i=0;i<=25;i++) {

        micontexto.strokeStyle="#"+Math.floor(Math.random()*16777215).toString(16);
        micontexto.strokeRect(x,y,ancho,alto);
        x=x+5;
        y=y+5;
        ancho=ancho-10;
        alto=alto-10;

    }

}
</script>
```

Es una variación de nuestra función; podemos copiarla y pegarla. En ella se emplean una serie de técnicas imprescindibles:

- Usamos variables para almacenar valores que van a cambiar a lo largo de la función, como **x**, **y**, etc. Luego variamos su valor con lineas como **x=x+5;**
- Usamos una estructura de repetición **for**, que hará que el bloque de su interior se repita tantas veces como indiquemos; en este caso 25 veces.
- Empleamos el objeto **Math**, que nos aporta un montón de funciones matemáticas, como **floor** para obtener un número

entero o **random**, para un número aleatorio. Con todo ello empleamos una pequeña fórmula para generar colores aleatorios.

El resultado sería éste:

Actividad 5

Realizaremos algunas modificaciones a la función anterior, cambiando números y variando valores. Observaremos el resultado de cada cambio, intentando sacar conclusiones. Sobre todo no nos desesperaremos si algo sale mal. Podemos comenzar de nuevo copiando y pegando la función.

Todavía más opciones de dibujo

El trabajo con la etiqueta **<canvas>** es muy extenso, pero no podemos dejar de citar algunas de las propiedades y métodos que podemos emplear mediante JavaScript:

- **moveTo (x,y)**: imaginemos que el canvas tiene una especie de puntero interior. Con moveto lo desplazaremos a una posición determinada. Es como desplazar un lapiz hasta un punto.
- **lineTo (x,y)**: teniendo ese lapiz en una posición concreta, con **lineTo** lo desplazaremos a otro punto, pero dejando dibujando una línea por el camino.

Estos dos métodos trazan una figura invisible. Hasta que no definamos un color de trazo con **strokeStyle** y lo apliquemos no

se verá nada. Para aplicarlo recurriremos al siguiente método:

- **stroke()**: traza la figura diseñada con las herramientas **moveTo** y **lineTo**.

Texto en el canvas

También podemos añadir texto a un <canvas>. Emplearemos las siguientes funciones:

- **font**: usando las reglas CSS que ya conocemos definiremos la apariencia del texto.
- **textAlign** y **textBaseline** se emplean para definir alineación (izquierda, derecha, etc) y punto de inicio del texto. Probablemente no empleemos estas propiedades en los primeros compases de trabajo con texto.
- **fillText (texto, x, y)**: como parámetros indicaremos el mensaje a mostrar y la posición inicial del texto.

La figura siguiente muestra un ejemplo sencillo.

Se obtiene añadiendo estas tres líneas al ejemplo anterior:

```
micontexto.textAlign="center";
micontexto.font="20px arial";
micontexto.fillText("Cuadrados",200,155);
```

Aún nos quedamos más cosas en el tintero, que no trataremos para no alargarnos demasiado. Con un **<canvas>** podemos realizar círculos (**arc**), degradados de color (**createLinearGradient**) e insertar imágenes en su interior (**drawImage**).

Pregunta de Elección Múltiple

Para que aparezca la función dibujar es necesario intruducir:

function dibujar( ) {

- *var milienzo = document.getElementById("canvas01");*
  var micontexto = milienzo.getContext("2d");

}

- *var milienzo = document.getElementById("canvas01");*
  var micontexto = milienzo.getContext("2d");
- *Function dibujar.*

Actividad 6

Intentaremos realizar una imagen con las diferentes herramientas aprendidas. Es muy importante prestar mucha atención a cómo escribimos las partes de *JavaScript*. El más mínimo error hará que deje de funcionar el script completamente.

Crear imágenes SVG

Desde *BlueGriffon* también podemos diseñar imágenes en formato SVG (abreviatura de *Scalable Vector Graphics*). Éste es un formato de imágenes vectorial basado en el formato XML. Ésto quiere decir que los archivos en este formato pueden ser modificados con un editor de texto sencillo y que su código se puede visualizar e interpretar con un poco de entrenamiento. Para terminar de añadir ventajas, es un formato abierto desarrollado por el *Word Wide Web Consortium*, el organismo regulador de HTML5 y de CSS, por lo que su uso es muy recomendable.

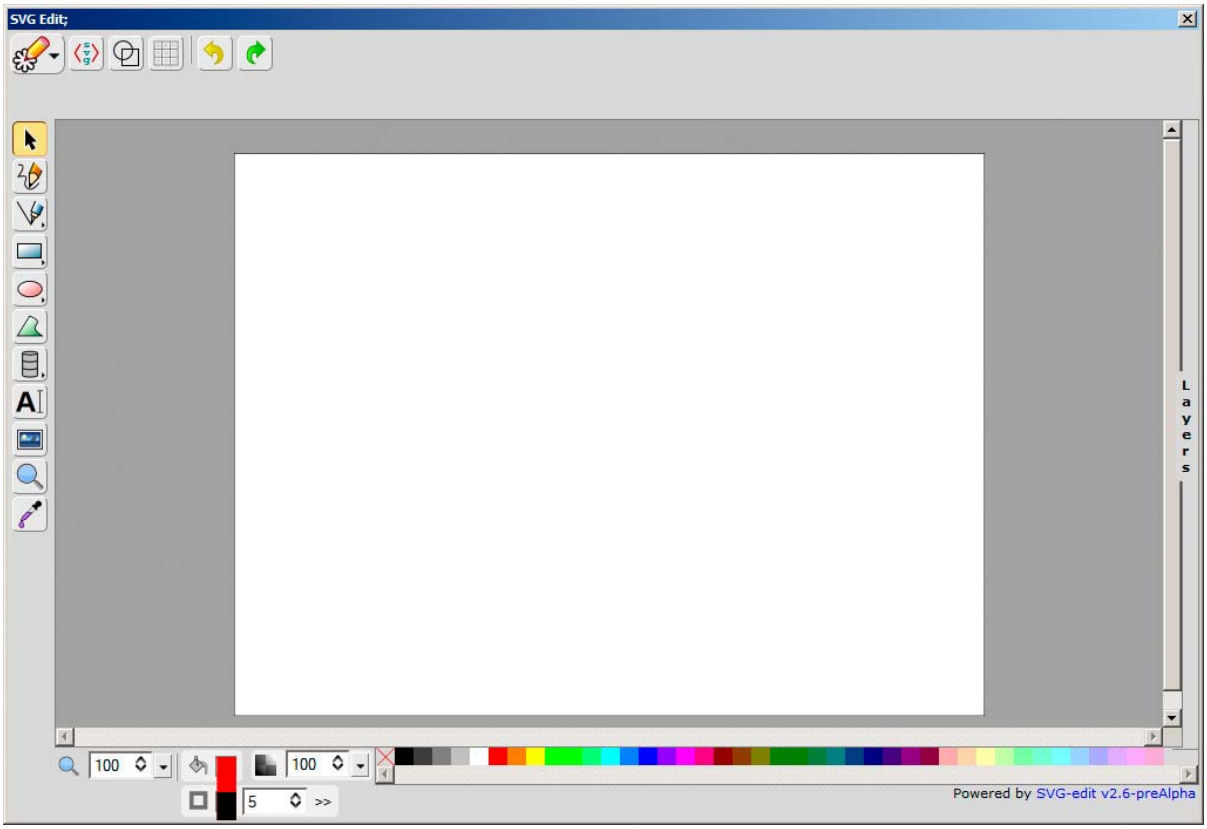
No vamos a ver cómo se modifican y se crean los archivos SVG, ya que en *BlueGriffon* contamos con una interesante herramienta que nos ayudará a crearlos.

Sigamos estos pasos para conseguirlo.

1. Abriremos *BlueGriffon* con un archivo nuevo.

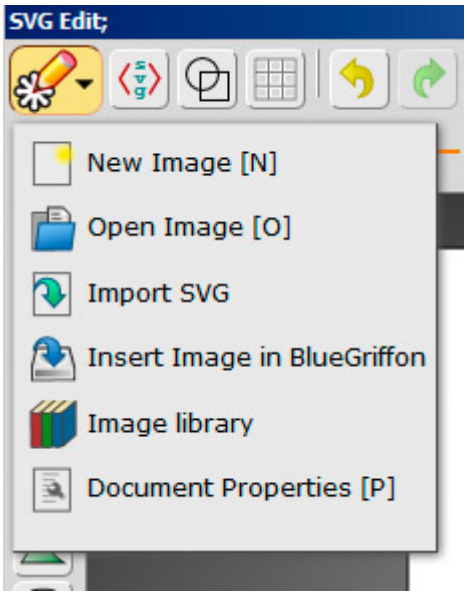
2. Haremos clic en el icono **Editor SVG**, recogido en la figura:

3. En la ventana de la figura iremos seleccionando las herramientas de la izquierda y realizando nuestro dibujo. Es un editor de imágenes sencillo, similar a otros de su gama. En la parte superior aparecen diferentes modificadores, dependiendo de la herramienta seleccionada.

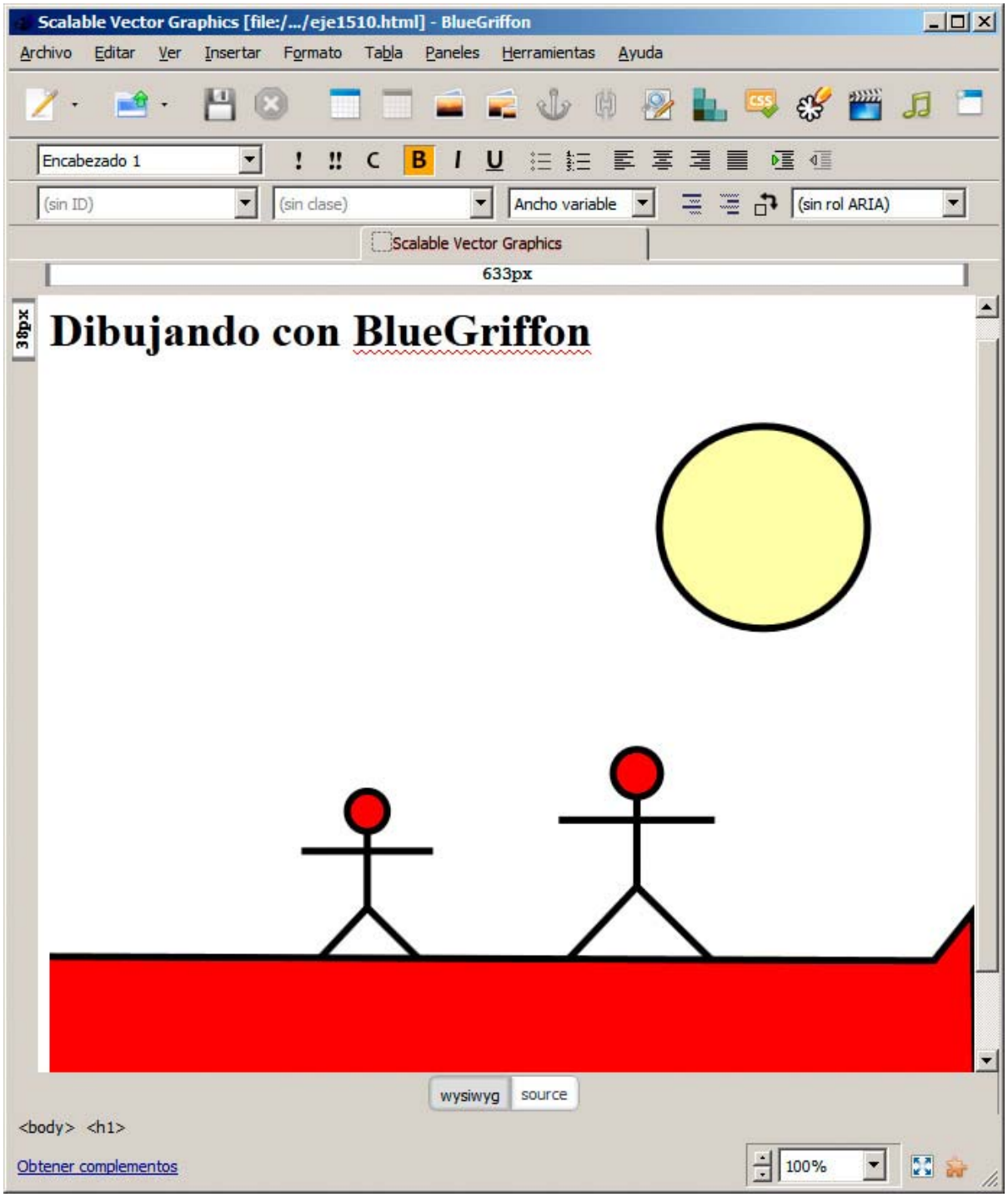


4. Si nuestra imagen va a ser complicada, podemos utilizar las capas para distribuir las partes. Se despliega haciendo clic en la parte derecha.

5. Para terminar, en el menú de la esquina superior izquierda seleccionaremos la opción **Insertar imagen** en *BlueGriffon*.



Tras ese paso, veremos que la imagen ha pasado a formar parte de la página web. Podríamos volverla a editar en cualquier momento, con tan sólo hacer doble clic sobre ella.



Lo más interesante lo encontramos al hacer clic en el botón **source**, donde podremos ver qué apariencia toma nuestra imagen. Es una simple etiqueta **<svg>** seguida de los elementos necesarios.

```
<body>
  <p></p>
  <h1>Dibujando con BlueGriffon</h1>
  <p><svg xmlns="http://www.w3.org/2000/svg" height="480" width="640">
      <!-- Created with SVG-edit - http://svg-edit.googlecode.com/ -->
      <g>
        <title>Layer 1</title>
        <path id="svg_1" d="m174.332016,320.016815190.999252,0m-45.290833,39.4

          stroke-width="5" stroke="#000000" fill="#FF0000"></path>
        <path id="svg_2" d="m352.431641,298.4697881108,0m-53.756287,46.3561711

          stroke-width="5" stroke="#000000" fill="#FF0000"></path>
        <path id="svg_3" d="m612,395.8999941-623,-2.89999417,841645,21-2,-1171

          stroke-width="5" stroke="#000000" fill="#FF0000"></path>
        <ellipse ry="70" rx="72" id="svg_4" cy="95.900002" cx="494" stroke-wid

          stroke="#000000" fill="#ffffaa"></ellipse> </g></svg></p>
  <p><br>
  </p>
</body>
</html>
```

Nota

Este editor es un complemento muy interesante desde el que podremos disfrutar de muchas opciones, aunque la implementación de este tipo de contenidos en los diferentes navegadores aún no está excesivamente madura, por lo que podemos encontrarnos con errores según el navegador y la complejidad del dibujo.

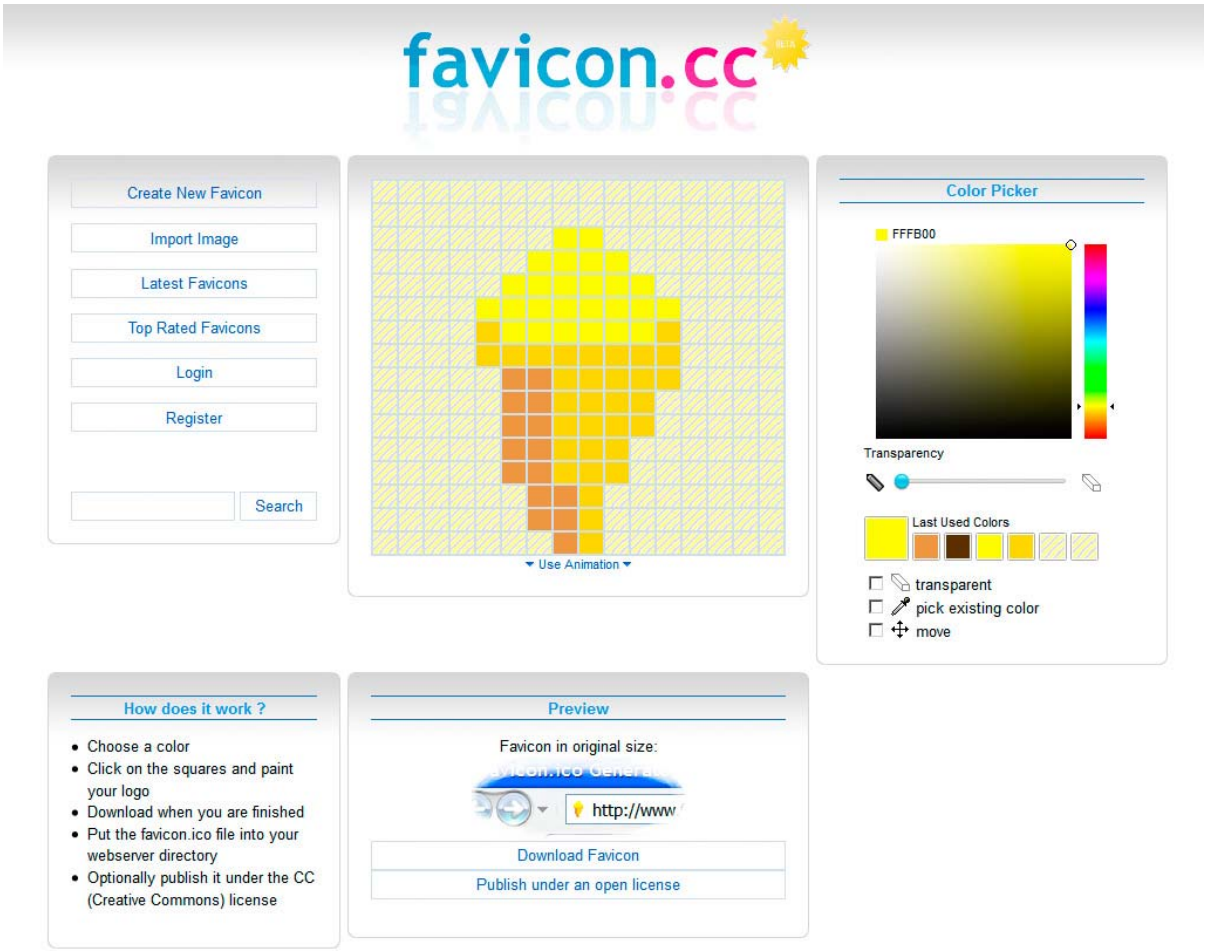
Añadir un icono a nuestra web

Mientras que todas las personas que trabajan en la creación de páginas web tienen clara la importancia de añadir un título a la página, para que sea fácil identificarla entre un grupo de pestañas o una lista de marcadores, no siempre se hace tanto hincapié en lo apropiado que resulta que nuestra web se pueda diferenciar de las demás con un icono.

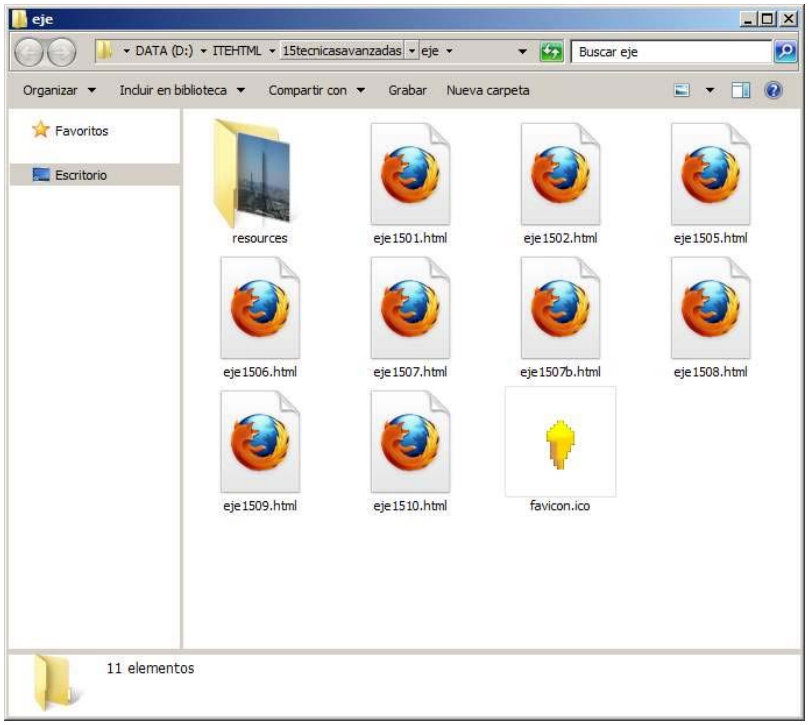
Los *favicons*, como se suele llamar a este tipo de iconos, son los pequeños dibujos que aparecen en la barra de navegación del marcador, en las pestañas y en los marcadores. Son muy útiles para ayudar al usuario a distinguir nuestra web entre todas las demás.

La imagen suele ser un archivo en formato .ico, de 16x16 píxeles o más, aunque en algunos casos también se pueden emplear imágenes en formato .png. Finalmente el archivo se guardará en la carpeta principal de nuestro sitio web con el nombre **favicon.ico**

Para crearlos, podemos emplear cualquier editor de imágenes que pueda generar iconos o incluso algún editor de iconos *online*, como *favicon.cc* (http://www.favicon.cc), que se muestra en la figura.



Una vez diseñado el icono, haremos clic en el enlace **Download Icon** para guardarlo en nuestro ordenador. Se verá como el de la figura:



Pregunta Verdadero-Falso

La siguiente afirmación, ¿es verdadera o falsa?

Los *favicons* son los pequeños dibujos que aparecen en la barra de navegación del marcador, en las pestañas y en los marcadores, los cuales, son imprecindibles para ayudar al usuario a distinguir nuestra web entre todas las demás.

Verdadero ○ Falso ○

Actividad 7

Diseñaremos un icono para nuestro sitio web y lo descargaremos en la carpeta principal del sitio web. Cuando realicemos la transferencia a Internet y probemos las páginas, veremos si se muestra en los diferentes lugares donde suele hacerlo.

Resumen

A través de este módulo hemos revisado algunas técnicas que nos pueden resultar útiles ante determinadas tareas y otras que, sobre todo, comienzan a abrir puertas para que profundicemos en todas las vertientes del diseño web que nos quedan por descubrir.

En concreto hemos visto:

- Creación de **enlaces internos**, para poder realizar saltos dentro de una página web y para poder acceder a un punto concreto de una página desde otra.
- **Mapas sensibles** que nos permiten interactuar con una imagen.
- Cómo crear y utilizar un ***canvas***.
- Crear **imágenes SVG** desde *BlueGriffon*.
- Añadir un **icono personalizado** a nuestra web.

Actividades y ejemplos

Actividad 1. Enlaces internos

En un documento largo crearemos un par de enlaces a lo largo del documento para regresar a su parte superior.

Actividad 2. Imágenes de sustitución

Tomaremos dos fotografías y colocaremos una en la parte central de la página. Debemos conseguir que, cuando el usuario haga clic sobre ella, se reemplace por la segunda. El evento que se emplea para esto se llama **onclick**.

Actividad 3. Mapas sensibles

Realizaremos un pequeño mapa basado en la imagen de un reloj y convertiremos algunas horas en enlaces a páginas de un calendario. Para redondear la actividad, podríamos hacer que todos los enlaces fuesen a diferentes secciones de un mismo documento, empleando los enlaces internos.

Actividad 4. Canvas

Probaremos a realizar mezclas con varios rectángulos rellenos y sin rellenar con diferentes colores. Podemos añadir tantas líneas como sea necesario a la función.

Actividad 5. Canvas

Realizaremos algunas modificaciones a la función anterior, cambiando números y variando valores. Observaremos el resultado de cada cambio, intentando sacar conclusiones. Sobre todo no nos desesperaremos si algo sale mal. Podemos comenzar de nuevo copiando y pegando la función.

Actividad 6. Canvas

Intentaremos realizar una imagen con las diferentes herramientas aprendidas. Es muy importante prestar mucha atención a cómo escribimos las partes de *JavaScript*. El más mínimo error hará que deje de funcionar el *script* completamente.

Actividad 7. Añadir un icono a nuestra web

Diseñaremos un icono para nuestro sitio web y lo descargaremos en la carpeta principal del sitio web. Cuando realicemos la transferencia a Internet y probemos las páginas, veremos si se muestra en los diferentes lugares donde suele hacerlo.

Ejemplos

Las diferentes prácticas, recursos y ejemplos realizados en este módulo están disponibles para realizar pruebas.

Ejemplos del módulo

Aplicación al aula

Técnicas avanzadas

Dada la mezcla de técnicas que hemos revisado, deberíamos intentar generar una práctica que se centrase en una o dos de las opciones revisadas. Haremos una propuesta basada en los mapas sensibles y los enlaces internos.

Programación dirigida al alumnado

Objetivos

- Crear un mapa sensible.
- Crear enlaces internos en un formulario.
- Vincular ambas prácticas.

Contenidos

- Mapas sensibles.
- Enlaces.

Materiales y recursos

- Ordenador con acceso a Internet.

Temporalización

- Sesión única.

Planificación

El alumnado tomará un texto sugerido por nosotros, dividido en diferentes apartados, y una imagen relacionada con el texto que nos pueda servir de mapa sensible. Podemos pensar en algún tipo de tarea de anatomía, un mapa geográfico, una representación de la fauna de una determinada zona, etc.

Organización del aula

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

Desarrollo de la actividad

- Se generan los enlaces internos en el documento.
- Se diseñan las áreas de la imagen.

- Se aplica un formato basado en estilos.
- 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:

- Creación de enlaces.
- Diseño de las zonas en el mapa.
- Apariencia general del documento.

Sugerencias metodológicas

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

Para su aplicación proponemos:

Sesión única

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

Añadimos los enlaces en las zonas indicadas del documento.

Diseñamos las zonas del mapa. Si no vamos a emplear ninguna herramientas específica, podemos hacer esta parte en gran grupo, sugiriendo resultados y probándolos.

Aplicamos al conjunto alguna plantilla de estilos ya creada. Si es necesario la modificaríamos.

Atención a la diversidad

Actividad de refuerzo

Para aquellos alumnos/as que puedan tener más dificultad, el documento puede contar ya con las zonas del mapa predefinidas.

Actividad de ampliación

La profundización en esta actividad se basaría en que el alumnado añadiese zonas poligonales o mezclase enlaces internos con enlaces a otros lugares.