EFECTOS ESPECIALES EN PÁGINAS WEB .

INTRO .

En la interacción con cualquier página web, es interesante contar con elementos dinámicos que añadan movimiento y efectos visuales, dando un toque más estético, mejor experiencia al usuario y la diferencian del resto de webs.

Estos efectos especiales pueden consistir tanto en el uso de imágenes, efectos con los textos, sonidos e incluso videos, animaciones y juegos interactivos.

Para implementar todo esto en un sitio web se pueden emplear distintos lenguajes de programación, el más utilizado es *JavaScript*, con librerías como *JQuery, Canvas 3D JS* o *Paintbrush* entre otras.

Mediante *HTML5* y *CSS3* podemos crear también animaciones y transiciones que aporten movimiento y mejoren el aspecto de cualquier web.

TRABAJAR CON IMÁGENES .

Las imágenes son uno de los elementos más empleados en los sitios web, ya que además de poder ser funcionales en la interfaz de navegación, aportan color y vistosidad, mejorando el diseño gráfico, ampliando información, etc.

Para su uso en página web, es necesario prepararlas con anterioridad, ya que sus características deben ser diferentes a las necesarias para su impresión. El tratamiento de imágenes para su correcta visualización en pantallas pretende crear archivos ligeros, que disminuyen el tiempo de carga del sitio web. En este caso, serán las pantallas las encargadas de aportar la calidad en la visualización de las imágenes.

En el trabajo con imágenes podemos encontrarnos con software profesionales gratuitos y de pago. Entre estos últimos, los más utilizados son [*Adobe Photoshop*](https://www.adobe.com/es/products/photoshop/pricing-info.html?gclid=CjwKCAjwjqWzBhAqEiwAQmtgT6dpX725CYF87gY59XZP-vqiZ63-Z3ozBWFT2h9JkcNB9WDcTwd9ihoCZ38QAvD_BwE&mv=search&mv2=paidsearch&sdid=DMMD161V&ef_id=CjwKCAjwjqWzBhAqEiwAQmtgT6dpX725CYF87gY59XZP-vqiZ63-Z3ozBWFT2h9JkcNB9WDcTwd9ihoCZ38QAvD_BwE:G:s&s_kwcid=AL!3085!3!585410689838!!!g!412069089069!!16448564761!134133175579&gad_source=1) “*para editar fotos y crear composiciones, pinturas digitales, animaciones, diseños gráficos y mucho más”* y, [*Adobe Illustrator*](https://www.adobe.com/es/products/illustrator/pricing-info.html?gclid=CjwKCAjwjqWzBhAqEiwAQmtgTwAwgK-swleNXy43_80colzUDXFACno6pmfE9xSb23l06CdaRb13xxoCkSoQAvD_BwE&mv=search&mv2=paidsearch&sdid=DMMD161V&ef_id=CjwKCAjwjqWzBhAqEiwAQmtgTwAwgK-swleNXy43_80colzUDXFACno6pmfE9xSb23l06CdaRb13xxoCkSoQAvD_BwE:G:s&s_kwcid=AL!3085!3!585295178106!!!g!412399178104!!16448564761!134775561355&gad_source=1), es un editor de gráficos vectoriales que sirve, entre otras cosas, para la ilustración como rama del arte digital, técnica o de diseño gráfico.

Dentro de los software gratuitos y con las herramientas necesarias para un buen tratamiento profesional de las imágenes para web nos encontramos con *Gimp*, *Pinta* y *Paint.net*, entre otros. (logos)

[*Gimp*](http://www.gimp.org.es/) es un programa de edición de imágenes de código abierto creado en 1995, que se utilice sobre todo para la mejora y retoque de fotos, creación de gráficos para web y diseño de logos, edición de GIFs animados y la optimización de gráficos para web. Con numerosos plugins y compatible con multitud de formatos, como JPEG, PNG, TIFF y su formato propio XCF. Está disponible para Windows, macOS y Linux y su desarrollo se mantiene por la comunidad bajo la Fundación GNOME.

[*Pinta*](https://www.pinta-project.com/)es también un programa de código abierto, inspirado en *Paint.net* y creado en 2010, diseñado para ser simple y fácil de usar para ediciones básicas de imágenes, como recortar, redimensionar y ajustar colores; crear dibujos simples y bocetos digitales, y agregar textos y formas a las imágenes. Soporta formatos como PNG, JPEG, BMP, TIFF y su formato propio de PDN.

[Paint.net](https://www.getpaint.net/) es una herramienta de edición de imágenes fácil de usar y con una amplia gama de herramientas creada en 2004. Es ideal para ajustes básicos y avanzados de fotos, creación de gráficos simples y complejos (banners, logos), y para el dibujo digital. Soporta formatos como JPEG, PNG, BMP, GIF y su propio formato PDN, contando con plugins para soportar formatos extra.

Los formatos de imagen más utilizados en páginas web y para una óptima visualización en pantalla son PNG, JPEG, GIF y SVG:

* JPEG (Joint Photographic Experts Group) es el formato más utilizado por cualquier dispositivo. Aunque pierde calidad en su compresión, se puede elegir el grado de esta al generarlo, por lo que cuanto más ligero, más calidad adicional perderá.
* PNG (Portable Network Graphics) se caracteriza porque el tamaño final de compresión es ligero, admite mayor profundidad de contraste y es compatible con transparencias, muy utilizadas en el diseño web.
* GIF (Graphics Interchange Format) tiene una compresión limitada a 256 colores, dando imágenes de tamaño reducido pero con la calidad suficiente para su visualización en pantalla. También admite transparencias.
* SVG (Scalable Vector Graphics) permite elementos geométricos vectoriales y animaciones de gráficos vectoriales sencillas, con bibliotecas de *JavaScript* capaces de manejar sus datos para modificarlas.

Selección de las imágenes .

Según la finalidad de las imágenes, se necesitará cierta calidad o tamaño de archivo. Para las imágenes digitales, debe haber equilibrio entre la calidad de la imagen y el tamaño final del archivo.

Tipos de imágenes .

Lo primero a tener en cuenta al seleccionar una imágen para web son las dos maneras en las que se construyen las imágenes.

* Imágenes vectoriales: se construyen mediante formas geométricas independientes (segmentos, polígonos…). El resultado final es una colección de objetos geométricos simples, que se podrán cambiar de escala. Una vez generada esta imagen, hay que exportarla a un archivo de mapa de bits para poder visualizarla en el navegador. Un ejemplo de este tipo de imagen suelen ser los logotipos.
* Imágenes bitmap: imágenes constituídas mediante píxeles, cada uno con un color uniforme, con variaciones de color y luz en cada uno de los píxeles, formando una imagen en su conjunto. Este tipo de imagen no se puede ampliar ni reducir de escala, ya que pierde calidad.

Resolución .

Se refiere tanto a las dimensiones en píxeles de la imagen como al número de píxeles por pulgada. Cuanta más alta es la resolución, la dimensión y la densidad de píxeles, mayor será la calidad de la imagen. Para la visualización de imágenes en pantalla se suele emplear una resolución de 72 ppi.

Profundidad de color .

Se refiere a los bits necesarios para almacenar la tonalidad de color de cada píxel, los bits por píxel, y cada dispositivo puede utilizar medidas diferentes.

Cuantos más bits por píxel tenga una imagen, tendrá una mayor variedad de colores para su representación y, por lo tanto, mayor calidad. Esto implicará también mayor tamaño de archivo, lo que ralentizará su carga en el navegador.

Modo de color .

* HSB: clasifica los colores según su tono, saturación y luminosidad. A mayor luminosidad más claridad tendrá la imagen y a menos saturación tenga un color, más gris y descolorido parecerá. Ambos factores se miden en porcentajes. (rueda cromática)
* RGB: mdo de color que utilizan todos los dispositivos digitales con visualización de imágenes. Los colores se obtienen por mezcla aditiva de los colores primarios: rojo, verde y azul (Red, Green, Blue). (círculos color aditivo)
* CMYK: cian, magenta, amarillo y negro (Cyan, Magenta, Yellow, Key). Modelo de color substractivo que se basa en la absorción de color y se utiliza en impresión, por lo que hay que realizar una conversión a RGB para su visualización en pantalla. (círculos color substractivo)

Optimización de imágenes .

Es el proceso de tratamiento que se le aplica a una imagen para aligerar su tamaño y manteniendo la calidad suficiente para su correcta visualización en pantalla y con rápida velocidad de carga en web.

La optimización dependerá de la finalidad y el tamaño en el que se vaya a utilizar la imagen, buscando un archivo lo menos pesado posible. También se debe tener en cuenta el formato final de dicho archivo ya que tienen características diferentes.

Escalar .

Se refiere a redimensionar una imagen, ampliarla o reducirla.

Al reducir el tamaño de una imagen apenas se aprecia la pérdida de calidad, pero al aumentarlo, se necesita información de la que no se dispone, por lo que el programa que se utilice para este proceso tendrá que rellenar el espacio con píxeles repetidos o generados automáticamente según los colores adyacentes, sufriendo una considerable pérdida de calidad.

En la mayoría de programas este procedimiento es el mismo.

Implementación de código con varias imágenes .

Con el uso de imágenes en un sitio web se pueden conseguir efectos visuales llamativos aplicando eventos o acciones combinando varios lenguajes de programación, como son *HTML*, *CSS* o *JavaScript*.

Imágenes de sustitución .

Reemplazar una imagen del sitio web cuando el usuario realiza una acción determinada aporta dinamismo y funcionalidad a la página.

Para realizar este efecto de sustitución de imágenes contamos con diferentes técnicas según el lenguaje de programación.

Utilizando únicamente *HTML* habrá que implementar el siguiente código:

| <img src=”imagenes/play.png” onmouseover=”this.src=’imagenes/play\_over.png’” onmouseout=”this src=’play.png’”/> |
| --- |

Así, la imagen con nombre “play.png” y guardada en la carpeta “imagenes” se sustituirá por la imagen “play\_over.png”, de la misma carpeta, cuando el ratón pase por encima. El atributo *onmouseover* hace que la primera imagen vuelva a aparecer al apartar el cursor de la imagen.

Para conseguir el mismo efecto utilizando también *CSS*, se debe utilizar el siguiente código:

| HTML | <div id=”boton\_play”></div> |
| --- | --- |
| CSS | #boton\_play {  background: url(../imagenes/play.png) no-repeat;  height: 3em;  width: 3em;  cursor: default;  }  #boton\_play:hover {  background: url(../imagenes/play\_hover.png) no-repeat;  height: 3em;  width: 3em;  cursor: pointer;  } |

Esto además permitirá cambiar distintas propiedades, como el tamaño o los bordes.

La tercera opción es utilizando *JavaScript*:

| var x = $ (document);  x.ready(inicia);  function inicia(){  var x = $(“boton\_play”);  x.mouseover(sobre);  x.mouseout(fueta);}  function sobre(){  $(this).css(“background”, “url(imagenes/play\_hover.png)”)}  function fuera(){  $(this).css(“background”, “url(imagenes/play.png)”)} |
| --- |

Este lenguaje ofrece muchas otras posibilidades a la hora de aplicar efectos en las transiciones.

Imágenes secuenciales .

Son un conjunto de imágenes que se muestran en un orden específico para crear la ilusión de movimiento o para ilustrar un proceso paso a paso.

| HTML | <div id=”imagenes>  <div class=”imagen” id=”imagen1”></div>  <div class=”imagen” id=”imagen2”></div>  <div class=”imagen” id=”imagen3”></div>  <div class=”imagen” id=”imagen4”></div>  </div> |
| --- | --- |
| CSS | .imagen {  height: 15em;  width: 15em;  display: inline-block;  opacity: 0;  }  #imagenes {  margin: 0 auto;  width: 90%;  }  #imagen1 {background: url(../imagenes/imagen1.jpg) no-repeat; }  #imagen2 {background: url(../imagenes/imagen2.jpg) no-repeat; }  #imagen3 {background: url(../imagenes/imagen3.jpg) no-repeat; }  #imagen4 {background: url(../imagenes/imagen4.jpg) no-repeat; } |
| JS | var x = $(document);  x. ready(inicia);  function inicia(){  $(“imagenes .imagen”).each(function(secuencia){ $(this).delay((secuenci)\*1500).fadeTo)1500,1)}) |

Implementación de plugins y librerías .

Existen multitud de *plugins* que se pueden añadir a un sitio web y que ofrecen elementos funcionales y con distintos efectos que pueden ser de gran utilidad. Los más comunes son galerías de imágenes o *sliders*, ya que son muy dinámicos y vistosos.

Según el lenguaje utilizado, habrá que enlazar una o más bibliotecas en el documento *HTML*. El documento *CSS* servirá para modificar el diseño del *plugin* (color, tamaño), y los archivos *JavaScript* permiten cambiar parámetros como el tiempo entre transiciones o tipos de efectos.

Ejemplo de plugin de imágenes en movimiento

TRABAJAR CON TEXTO: EFECTOS ESTÉTICOS Y DE MOVIMIENTO .

En un sitio web, es posible aplicar distintos efectos especiales a los textos que en él se muestran, tanto para destacar palabras como para dar dinamismo a la página. Para ello, habrá que utilizar los distintos lenguajes de programación, *HTML, CSS* y *JavaScript*, pero siempre se recomienda cambiar cualquier aspecto del texto a través del documento CSS, ya que nos permite una amplia gama de modificaciones a las fuentes tipográficas, como puede ser el tamaño (font-size), el estilo de fuente (font-style) o su anchura (font-weight); y a los bloques de texto, como el color, el espacio entre los caracteres (letter-spacing), la capitalización del texto (text-transform) o algún tipo de decoración (text-decoration) entre otros.

Creación de textos mejorados y con movimiento .

Es frecuente encontrar sitios web con textos en movimiento, bien para llamar la atención hacia un elemento del sitio o para cualquier utilidad que pueda necesitar. Dependiendo del grado de complejidad, se utilizará un lenguaje de programación u otro. Podemos encontrar diferentes *plugins* JQuery que permiten la creación de estos efectos.

Un recurso muy común es el de dar efecto a los enlaces que se encuentran en bloques de texto para resaltarlos y diferenciarlos del resto de texto.

Se pueden aplicar estilos a cualquier texto mediante *CSS,* como decorarlos con degradados o darles profundidad aplicando sombras.

Implementación de efectos .

Para la implementación de efectos en los textos, la mejor opción será emplear el código *JavaScript*. Así se podrán crear sitios web dinámicos y llamativos.

*Un ejemplo de ello es este ejemplo*

Adecuación de los efectos a la página web .

Al aplicar los efectos a la página web hay que tener en cuenta las necesidades del sitio, así como su correcta visualización en todos los navegadores y dispositivos posibles mediante un diseño adaptativo o que responda a distintas resoluciones, bien modificando los *plugins* o implementando una consulta de medios, ya sea en el atributo media, con la regla @import o en la propia hoja de estilo. También se puede usar el lenguaje *JavaScript* para crear un *script* que detecte previamente la resolución de la pantalla del usuario.

TRABAJAR CON MARCOS .

Lenguajes como PHP permiten estructurar el documento del sitio web y, a través del comando include o require incluir, en las secciones necesarias, los módulos fijos que compartan todas las páginas añadiendo otros documentos externos. Esto además permite ahorrar codificación y espacio en el servidor.

TRABAJAR CON VENTANAS .

Las ventanas de un navegador son una parte fundamental ya que permiten la visualización de cada uno de los documentos de un sitio web. Mediante *JavaScript* se podrá controlar la aparición y desaparición de distintas ventanas adicionales y la interacción entre ellas. Pero se debe tener en cuenta que un uso abusivo de estas puede resultar tedioso para el usuario, pudiendo este tener bloqueada la aparición de ventanas secundarias en su navegador. En ese caso, habrá que avisar al usuario de que desactiva esa opción para el correcto funcionamiento del sitio web.

Creación de varias ventanas .

A través de la sentencia *JavaScript* “window.open()” se pueden abrir y modificar *pop-ups* mediante los parámetros URL, nombre de la ventana o atributo de destino y forma de la ventana.

* URL: dirección del contenido de la nueva ventana.
* Nombre de la ventana o atributo de destino: se determina un nombre para la nueva ventana para su posterior identificación, y se establece su vestido del mismo modo que los enlaces en *HTML*, con el atributo **target**:
  + \_blank: opción por defecto. La URL se abre en una nueva ventana.
  + \_parent: se carga en la ventana principal o ventana padre.
  + \_self: se sustituye el contenido de la página actual.
  + \_top: sustituye cualquier conjunto de marcos
* Forma de la ventana:
  + channelmode/fullscreen: la ventana se muestra en modo de pantalla completa.
  + height/width: establece el tamaño de la nueva ventana en píxeles de alto y ancho.
  + left/top: indica la posición inicial.
  + location: para no mostrar la barra de direcciones.
  + menubar: para no mostrar la barra de menú del navegador.
  + resizable: para permitir al usuario redimensionar la ventana.
  + scrollbar: muestra u oculta las barras de desplazamiento.
  + statusbar: agrega o quita la barra de estado.
  + titlebar: para agregar o quitar el título.
  + toolbar: muestra o habilita la barra de herramientas.
* Reemplazar: de forma opcional, se puede establecer su la nueva ventana creará una nueva entrada en el historial o sustituirá a la actual utilizando el parámetro **replace true** o **false.**

Deben escribirse del siguiente modo:

| window.open(“URL”, “nombre”, “forma”); |
| --- |

(Termina la ventana)

Para cerrar las ventanas emergentes creadas es necesario conocer su nombre o almacenarla en una variable para poder referirse a ella con la sentencia “window.close()” en *JavaScript*.

Interactividad entre varias ventanas .

Mediante *JavaScript* todas las ventanas creadas se pueden comunicar entre ellas y con la principal. Para ello habrá que almacenar las ventanas emergentes en variables que permitan referirse a ellas cuando sea necesario.

Se puede modificar el atributo **location** en una ventana abierta para que muestre un archivo *HTML* o que cargue una nueva dirección URL externa con la sentencia “windows.location”.

También es posible que la ventana secundaria se comunique con la principal para enviar información o modificar el contenido de algún elemento mediante la sentencia “windows.opener”.

OTROS EFECTOS .

Efectos con *HTML* .

Este lenguaje de programación se orienta a guardar y organizar el contenido de la página web para luego aplicar efectos visuales a través de *CSS* y otros lenguajes orientados a objetos.

Con la aparición de *HTML5* se intentan solventar las limitaciones que este lenguaje ofrece a la hora de aplicar efectos a los elementos del sitio web con la creación de nuevas etiquetas, por ejemplo <[canvas](https://www.w3schools.com/html/html5_canvas.asp)>. Es una herramienta poderosa para crear gráficos y animaciones en 2D mediante JavaScript. A diferencia de simplemente mostrar imágenes estáticas, el <canvas> te permite generar gráficos dinámicos y altamente interactivos, lo que es fundamental en el desarrollo moderno de aplicaciones web, pero su contenido no se puede representar como nodos de un objeto DOM, por los que los navegadores con accesibilidad para personas con diversidad funcional no podrán interpretar su contenido.

Ejemplo de canvas p. 182

Las posibilidades de esta nueva etiqueta son muy amplias, aportando grandes composiciones estáticas o animadas en las páginas web.

Efectos con *CSS* .

Mediante lenguaje *CSS* también es posible crear efectos que parecen muy elaborados, pero mucho más ligeros para los navegadores que utilizando otro tipo de lenguajes de programación.

Algunos de estos efectos son una galería de imágenes o el efecto de una pelota rebotando.

Efectos con capas .

En *HTML* la etiqueta <div> se utiliza para crear capas con cualquier tipo de contenido en su interior, pero cualquier objeto del documento que tenga contenido y se pueda situar en cualquier parte de la web se puede considerar una capa y se le pueden aplicar distintos efectos mediante *JavaScript*.

Estos efectos suelen estar orientados a pequeños elementos, como son botones o menús de navegación.

hide() show() .

Se utilizan para ocultar y mostrar elementos web y utilizarlos para crear efectos de desvanecimiento.

| $(selector).hide(velocidad, callback);  $(selector).show(velocidad, callback); |
| --- |

Estos métodos nos permiten, por ejemplo, ocultar una imagen mediante un botón y mostrarla de nuevo empleando otro botón.

También el evento **toggle** ofrece el mismo funcionamiento, pero a través de un solo botón.

Los eventos fade .

Estos eventos mantienen el objeto en el flujo normal de la web, pero modificando su opacidad.

Existen los siguientes métodos:

* fadeIn(): aparece un elemento no visible
* fadeOut(): desaparece un elemento visible
* fadeToggle(): alterna los métodos anteriores
* fadeTo(): permite especificar la opacidad para mostrar u ocultar

| $(selector).fadeIn(velocidad, callback); / $(selector).fadeOut(velocidad, callback); /  $(selector).fadeToggle(velocidad, callback);  $(selector).fadeTo(velocidad, opacidad, callback); |
| --- |

Un ejemplo similar al anterior sin mostrar el objeto al 100% de opacidad, si no al 70%.

Los eventos slide .

Estos métodos se utilizan para crear efectos de deslizamiento.

* slideDown(): deslizamientos hacia abajo
* slideUp(): deslizamiento hacia arriba
* slideToggle(): alterna entre los dos métodos anteriores.

| $(selector).slideToggle(velocidad, callback); |
| --- |

animate() .

Es un efecto muy utilizado pero también más complejo. Permite pequeñas animaciones. Se puede definir casi cualquier propiedad del elemento seleccionado (tamaño, color, etc) y cambiar su valor.

| $(selector).animate({ parámetros }, velocidad, callback); |
| --- |

stop() .

Sirve para detener una animación antes de que esta termine.

| $(selector).stop(stopAll, goToEnd); |
| --- |

En este caso, se podrá parar toda la cola de métodos “animate()” declarados con el parámetro “stopAll” con valor *true*, de los contrario este dera *false* y solo se parará el método en ejecución.

Con el parámetro “stopToEnd” con valor *true* se especifica si la animación se completa al ejecutarse el método “stop” o se detendrá con los valores asignados en ese momento.

Un ejemplo con la animación anterior.

callback .

Este parámetro es opcional y se puede incluir en cualquier efecto para para llamar a una función cuando este termina..

Con el ejemplo anterior, se puede establecer que cuando el logotipo termine el recorrido se abra una nueva ventana. De esta manera, es posible crear animaciones muy elaboradas que aporten dinamismo a cualquier sitio web.

Encadenamiento de efectos .

Todos los efectos que se apliquen a un objeto pueden ir encadenados en una única línea para facilitar su lectura a los navegadores y sin necesidad de hacer referencia a él y optimizando el código. Por ejemplo el deslizamiento hacia abajo, desaparecer y volver a aparecer.