

Contenidos multimedia en la Web: animaciones

Sitio: [Aula Virtual do IES de Teis](#)

Curso: Diseño de interfaces web (DAW-dual 2024-2025)

Libro: Contenidos multimedia en la Web: animaciones

Impreso por: Brais Bea Mascato

Data: venres, 16 de maio de 2025, 5:07 PM

Táboa de contidos

- 1. Las animaciones en la Web.**
- 2. Herramientas para crear animaciones**
- 3. Elementos de una animación**
- 4. Fotogramas y capas**
- 5. Animación de imágenes: tipos de animación**
- 6. Animación de imágenes. Interpolación de movimiento**
- 7. Animación de texto**
- 8. Integración de audio en una animación. Sincronización.**
- 9. Integración de vídeo en una animación**
- 10. Efectos y filtros de efectos**
- 11. Animaciones con CSS**
 - 11.1. Animaciones básicas con transition
 - 11.2. Animaciones continuas con keyframes

1. Las animaciones en la Web.

Según la definición de Wikipedia, una **animación** es un proceso utilizado para dar la sensación de movimiento a imágenes o elementos gráficos. Los cuadros se pueden generar dibujando, pintando o fotografiando pequeños cambios entre fotogramas, representando una transformación visual que puede simular desde un modelo real hasta un entorno totalmente virtual.

En la unidad de trabajo anterior vimos que el formato **GIF** fue uno de los primeros en emplearse para mostrar imágenes animadas en la Web. Aunque limitado a 256 colores, fue muy popular por su simplicidad. Hoy en día, sigue usándose en redes sociales y comunicación informal.

Sin embargo, para aplicaciones más avanzadas y eficientes, existen alternativas más modernas:

- **SVG:** Permite animaciones vectoriales escalables y ligeras mediante CSS o JavaScript, ideales para iconos e interfaces responsivas.
- **WebP:** Soporta animaciones y ofrece una mejor compresión que GIF.
- **Lottie (JSON):** Utiliza animaciones exportadas desde After Effects mediante Bodymovin, permitiendo integrarlas fácilmente en sitios web o apps con JavaScript o frameworks móviles.
- **APNG:** Es una extensión del formato PNG que soporta animaciones, ofreciendo mejor calidad y soporte de transparencia que el GIF.

También existen tecnologías específicas para crear animaciones interactivas:

- **CSS Animations y Transitions:** Ideales para efectos sencillos y eficientes, compatibles con todos los navegadores modernos.
- **JavaScript (con librerías como GSAP o Anime.js):** Permite mayor control y dinamismo en las animaciones, incluso sincronizándolas con eventos del usuario.
- **Canvas y WebGL:** Utilizados para animaciones complejas, videojuegos o visualizaciones 3D, mediante librerías como PixiJS o Three.js.

Con la evolución de la Web, tecnologías como Flash quedaron obsoletas. Adobe dejó de dar soporte oficial a Flash en diciembre de 2020, y todos los navegadores modernos lo han desactivado por completo por motivos de seguridad y compatibilidad.

Gracias a **HTML5** y sus tecnologías asociadas (CSS3, JavaScript, WebGL), hoy en día es posible crear animaciones ricas, interactivas y adaptables a cualquier dispositivo sin necesidad de complementos adicionales.

En el diseño actual de sitios web, las animaciones se han convertido en una herramienta clave para mejorar la experiencia del usuario, dirigir su atención, reforzar la identidad visual y transmitir información de manera intuitiva. No obstante, es importante usarlas con moderación y criterio, evitando sobrecargar la interfaz o comprometer la accesibilidad y el rendimiento del sitio.

2. Herramientas para crear animaciones

Las animaciones son un recurso habitual en la Web para mejorar la experiencia visual y facilitar la interacción del usuario, tanto en interfaces como en elementos multimedia. Se emplean, por ejemplo, en botones animados, desplazamientos suaves, banners, iconos interactivos, presentaciones y microinteracciones.

Actualmente, hay varias formas de añadir animación a una página web:

- **Mediante formatos de archivo animados:** como GIF, WebP o APNG, usados principalmente para elementos visuales simples o decorativos.
- **Mediante SVG animado:** ya sea con animaciones declarativas (SMIL) o usando CSS y JavaScript.
- **Utilizando reglas de estilo (CSS3):** que permiten animar propiedades como color, tamaño, opacidad, posición, etc.
- **Con JavaScript y librerías especializadas:** como GSAP, Anime.js o Lottie, que permiten animaciones complejas, controladas por eventos o interacciones del usuario.
- **A través de APIs avanzadas:** como Canvas o WebGL, para animaciones 2D y 3D de alto rendimiento.

Aunque la definición clásica de animación incluye el cambio visual progresivo de elementos, en la Web se amplía también a efectos visuales y de interfaz que responden a acciones del usuario.

El ecosistema de herramientas para crear animaciones ha evolucionado mucho. Actualmente existen soluciones más ligeras, compatibles con estándares abiertos y accesibles para todo tipo de usuarios. A continuación, una tabla con algunas herramientas populares:

Software actual de animación

Programa	Descripción
Adobe Animate (antes Flash)	Herramienta profesional para crear animaciones interactivas vectoriales y contenido multimedia. Exporta en formatos como HTML5, SVG, WebGL y video. Ya no usa Flash, y permite la integración con JavaScript.
Synfig Studio	Software libre para animación 2D vectorial basado en capas. Permite crear animaciones complejas sin necesidad de dibujar cada cuadro. Compatible con Linux, Windows y macOS.
Blender	Aunque es principalmente un software de animación y modelado 3D, Blender también permite crear animaciones 2D con su sistema Grease Pencil. Es gratuito y de código abierto.
Lottie + After Effects	Lottie permite integrar animaciones exportadas desde After Effects (con Bodymovin) en apps y sitios web usando JSON. Ideal para interfaces modernas.
Pencil2D	Herramienta de código abierto para animación 2D tradicional basada en fotogramas dibujados a mano. Ideal para estudiantes y artistas.
Rive (antes Flare)	Plataforma en línea para crear animaciones interactivas que pueden integrarse directamente en apps web y móviles. Compatible con Flutter y otros entornos modernos.

Nota: Herramientas como **Microsoft GIF Animator** o **Stykz** han quedado obsoletas frente a opciones más modernas que permiten animaciones más ricas, ligeras y compatibles con HTML5.

3. Elementos de una animación

En toda animación digital confluyen dos tipos principales de elementos que trabajan conjuntamente:

- **Elementos gráficos estáticos:** Son aquellos objetos visuales presentes en cada fotograma. Representan personajes, escenarios, objetos o componentes de interfaz que, en cada instante, mantienen una posición determinada.
- **Elementos dinámicos o de animación:** Son los cambios que se aplican a los elementos gráficos a lo largo del tiempo, generando la ilusión de movimiento, transformación o interacción.

Estos elementos están definidos por sus propiedades visuales (forma, color, textura, tamaño) y por cómo se transforman (posición, rotación, escala, opacidad, etc.), influidos por los principios del diseño y de la animación.

Importancia de las instancias de símbolos (en animación vectorial y digital)

Las **instancias de símbolos** (por ejemplo, en herramientas como Adobe Animate o Lottie) son representaciones reutilizables de elementos gráficos. Su uso aporta varias ventajas:

- **Ahorro de espacio:** Al almacenarse solo una vez en el archivo, permiten reducir el peso total de la animación.
- **Facilidad de edición:** Los cambios en el símbolo base se aplican automáticamente a todas sus instancias.
- **Coherencia visual:** Aseguran uniformidad en los elementos repetidos.

Consideraciones para una animación realista y efectiva

Más allá de los gráficos utilizados, una buena animación debe transmitir **credibilidad y fluidez**. Para ello, es importante tener en cuenta:

- **Claridad y propósito en la acción:** Las animaciones deben comunicar una idea clara sin generar confusión al espectador.
- **Sentido del tiempo y ritmo:** Es fundamental mantener una cadencia adecuada para que las acciones no parezcan artificiales ni aceleradas.
- **Deformación y flexibilidad:** Los elementos gráficos deben adaptarse a los movimientos, evitando rigidez excesiva.
- **Anticipación:** Incluir gestos previos a una acción principal mejora la legibilidad del movimiento.
- **Continuidad y fluidez:** Las acciones deben enlazarse con suavidad para evitar saltos o movimientos mecánicos.
- **Finalización natural:** Las acciones deben terminar de manera lógica y con una sensación de peso o gravedad acorde al objeto animado.

La animación debe reflejar todos los cambios que ocurren en una escena: la iluminación, la ambientación, la caracterización de personajes y la perspectiva del espectador. Estos elementos aportan profundidad y coherencia visual a la narrativa animada.

4. Fotogramas y capas

En animación digital, un **fotograma** es cada una de las imágenes individuales que, al reproducirse en secuencia, generan la ilusión de movimiento. Esta técnica se basa en principios similares a los del vídeo, donde una serie de imágenes estáticas se muestran rápidamente para simular acción.

En el contexto de la animación para la Web o para contenido multimedia, trabajamos principalmente con dos elementos clave:

- **Fotogramas:** Representan instantes específicos del tiempo en la animación.
- **Capas:** Permiten organizar los distintos elementos visuales y de sonido que se superponen en la escena.

Parámetros esenciales al crear una animación

Al comenzar una nueva animación en un software como Adobe Animate, Blender, Synfig o Pencil2D, es necesario definir:

- **FPS (Frames Per Second o fotogramas por segundo):** Define cuántos fotogramas se muestran cada segundo. A mayor FPS, más fluida será la animación (24 FPS es estándar cinematográfico).
- **Tamaño:** Establece las dimensiones del lienzo o escenario, normalmente en píxeles.
- **Color de fondo:** Determina el color base del escenario sobre el que se construirá la animación.

Línea de tiempo y organización

La línea de tiempo es el espacio donde se organizan los fotogramas y se distribuyen en distintas **capas**. Cada capa permite trabajar con un tipo específico de contenido: gráficos, texto, sonido, efectos especiales, etc. Esto facilita la edición, evita interferencias y permite una mayor flexibilidad.

Por ejemplo:

- Una capa para el personaje principal.
- Otra para el fondo.
- Una más para efectos sonoros o subtítulos.

Tipos de fotogramas

Los softwares de animación distinguen entre distintos tipos de fotogramas, cada uno con funciones específicas:

- **Fotograma clave:** Representa un cambio importante en la animación (posición, forma, color...). Es el punto de referencia para definir movimientos.
- **Fotograma en blanco:** Similar al anterior, pero no contiene ningún contenido gráfico; útil para preparar nuevas escenas o acciones.
- **Fotograma con contenido:** Deriva de un fotograma clave y mantiene sus elementos.
- **Intercalaciones (tweening):** Son los fotogramas generados automáticamente entre dos fotogramas clave. Pueden ser de:
 - **Movimiento:** Para cambios de posición.
 - **Forma:** Para transformaciones morfológicas.
 - **Color:** Para transiciones visuales.
- **Secuencia de fotogramas iguales:** Repite un mismo contenido durante un periodo sin cambios visibles.

Estos conceptos permiten crear animaciones fluidas, coherentes y profesionales, incluso en proyectos interactivos o destinados a la Web.

Consejos para organizar capas

- Usa una capa por tipo de elemento (personaje, fondo, texto, sonido...).
- Nombra cada capa con claridad.
- Bloquea capas terminadas para evitar cambios accidentales.
- Agrupa capas cuando trabajes en componentes complejos o reutilizables.

5. Animación de imágenes: tipos de animación

Una vez comprendido el uso de capas y fotogramas, es importante conocer los distintos **tipos de animación** que pueden utilizarse para dar vida a los elementos gráficos.

En animación digital, distinguimos principalmente dos enfoques:

- **Animación fotograma a fotograma (frame-by-frame)**
- **Interpolación (tweening)**

Animación fotograma a fotograma

Este método consiste en dibujar cada fotograma de manera individual, realizando pequeñas variaciones entre uno y otro para generar el movimiento. Es similar a la animación tradicional usada en el cine o la televisión.



Características:

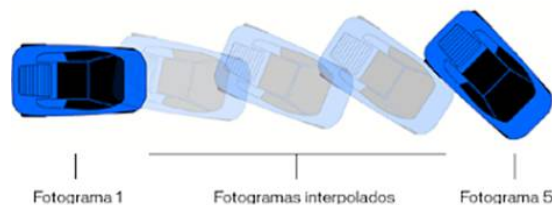
- Ideal para movimientos complejos o expresivos, como animaciones de personajes.
- Requiere más tiempo, ya que cada imagen se dibuja manualmente.
- Proporciona gran control artístico sobre el movimiento.

Pasos básicos:

1. Crear un fotograma clave inicial (ej. fotograma 1).
2. Dibujar el objeto o personaje en esa posición.
3. Crear un nuevo fotograma clave y redibujar el objeto con una leve variación.
4. Repetir el proceso tantas veces como sea necesario para completar el movimiento.

Interpolación (Tweening)

La interpolación permite generar automáticamente los fotogramas intermedios entre dos fotogramas clave. El software calcula los cambios progresivos en propiedades como posición, escala, color o forma.



Ventajas:

- Ahorra tiempo al no requerir dibujar cada paso.
- Útil para animaciones técnicas, efectos visuales o movimientos simples.
- Compatible con animaciones interactivas y responsivas en la Web.

Tipos comunes de interpolación:

- **Interpolación de forma:** Deforma un objeto entre dos formas distintas. Ej.: convertir un círculo en una estrella.
- **Interpolación de movimiento:** Mueve un objeto de un punto A a un punto B, pudiendo incluir rotación, escala o cambio de opacidad.

- **Interpolación de color:** Cambia gradualmente el color de un objeto entre dos valores definidos.
- **Interpolación de transformación (moderna):** Algunas herramientas como GSAP, Lottie o CSS permiten combinar múltiples propiedades (posición, opacidad, rotación, escala) con facilidad.

Este enfoque puede aplicarse también a **objetos agrupados o símbolos**, y se usa mucho en la animación web con tecnologías como **CSS3**, **JavaScript**, **SVG** o **Lottie/JSON**.

En proyectos interactivos y optimizados para la Web, lo más común es usar interpolaciones, ya que ofrecen animaciones suaves con menor esfuerzo y mejor rendimiento.

6. Animación de imágenes. Interpolación de movimiento

La **interpolación de movimiento** (motion tweening) es una técnica de animación que permite mover objetos de un punto a otro de forma fluida sin necesidad de dibujar cada fotograma. Esta técnica es ampliamente usada en herramientas de animación digital modernas como **Adobe Animate**, **Blender**, **Synfig**, **Lottie** o con **CSS** y **JavaScript** en la Web.

Tipos de interpolación de movimiento

Actualmente, se distinguen principalmente dos enfoques según el tipo de software o herramienta:

- **Interpolación clásica (o tradicional):** Es la forma básica de interpolación. Se define una posición inicial y otra final para un objeto (símbolo), y el software genera automáticamente los fotogramas intermedios. Se puede animar el desplazamiento, la escala, la rotación o la opacidad del objeto. Esta técnica sigue vigente, aunque es más limitada en control avanzado.
- **Interpolación avanzada (o basada en ruta):** En herramientas modernas como Adobe Animate o Lottie, se permite animar trayectorias complejas controlando la **ruta del movimiento**, la **velocidad**, los **giros** y los **efectos de aceleración y desaceleración** (easing). Esta técnica ofrece más precisión y naturalidad al movimiento, permitiendo también animaciones curvas, no lineales y con mayor expresividad.

¿Cómo se aplican estas interpolaciones?

- Para usar interpolaciones, el objeto debe estar convertido en un **símbolo** o **grupo animable**.
- Se establecen **fotogramas clave** al inicio y al final del movimiento.
- Luego, se genera la interpolación en la línea de tiempo y se ajustan sus parámetros (velocidad, dirección, efectos).
- En herramientas web como CSS o JavaScript (con librerías como GSAP o Anime.js), se pueden definir estas animaciones con precisión usando código o configuradores visuales.



Creación de una interpolación de movimiento

Los colores en la línea de tiempo suelen indicar el tipo de interpolación: azul o violeta para movimiento, verde para interpolaciones clásicas y grises para fotogramas estáticos o sin interpolación.

La interpolación de movimiento es una técnica poderosa que permite crear animaciones fluidas y eficientes, ideales para interfaces, personajes, gráficos o efectos en la Web.

7. Animación de texto

Además de animar objetos e imágenes, también es posible animar **texto**, lo cual es muy habitual en banners, presentaciones, interfaces web y publicidad digital. El texto puede aparecer, desaparecer, desplazarse, cambiar de forma o color, y más.

En la actualidad, existen múltiples formas de animar texto en la Web, gracias a tecnologías como:

- **CSS3** (para transiciones y animaciones básicas).
- **JavaScript** y librerías como **GSAP**, **Anime.js** o **Typed.js**.
- **SVG animado**, que permite efectos complejos con texto vectorial.
- **Lottie (JSON)**, para integrar texto animado desde After Effects.
- **Canvas y WebGL**, para animaciones más personalizadas o visualmente complejas.

Propiedades animables del texto

Tanto si el texto está compuesto por líneas simples o por cada letra individual, se pueden animar sus propiedades de forma independiente. Entre las más comunes están:

- **Posición (X, Y)**: Para mover el texto por la pantalla.
- **Tamaño**: Para aplicar efectos de escala o zoom.
- **Color**: Para crear efectos de cambio de tono, opacidad o iluminación.
- **Rotación**: Ideal para efectos de entrada o salida llamativos.
- **Transformación de forma o estilo**: Especialmente útil cuando el texto es convertido a trazado o SVG.

Modos de interacción con el texto

Dependiendo de la herramienta o tecnología, el texto puede ser:

- **Estático**: No se puede editar, solo animar como un bloque.
- **Seleccionable**: El usuario puede seleccionar texto, útil en contenido web accesible.
- **Editable**: El texto puede ser modificado por el usuario final (ej. campos de formularios).
- **Interactivo**: Responde a eventos (hover, clic, entrada de datos).

Animar texto letra por letra

Una técnica habitual consiste en separar el texto en letras individuales para aplicar animaciones diferenciadas (efecto máquina de escribir, aparición secuencial, rotación por carácter, etc.). Este proceso se puede automatizar con:

- Funciones integradas en **After Effects**.
- Plugins de **GSAP** como SplitText.
- Manipulación de nodos DOM con JavaScript.
- Edición de texto en SVG o herramientas vectoriales.

Consideraciones técnicas

- En algunos programas, al convertir texto a curvas (vectores), se pierde la capacidad de edición directa, pero se gana libertad para deformarlo.
- Es posible convertir el texto en **símbolos** o **componentes reutilizables**, como ocurre en Animate, Figma, Lottie o incluso herramientas web.
- Para animaciones responsivas o accesibles, se recomienda no abusar de textos convertidos en gráficos, ya que dificultan la indexación y lectura por lectores de pantalla.

Las animaciones de texto, bien aplicadas, mejoran la atención, el dinamismo y la comprensión visual del mensaje. No obstante, deben usarse con moderación para no sobrecargar la interfaz ni dificultar la lectura.

8. Integración de audio en una animación. Sincronización.

Una buena banda sonora potencia la emoción, el ritmo y el impacto de una animación. Un simple efecto de pasos sobre hojas secas o el chirrido de una puerta puede enriquecer la escena y conectar emocionalmente al espectador con lo que ocurre en pantalla.

El sonido es, por tanto, un componente narrativo esencial en animación.

Formatos de audio más comunes

En la Web o herramientas modernas de animación, los formatos más utilizados son:

- **MP3:** Ligero, ideal para la web.
- **WAV:** Alta calidad, usado para efectos sonoros puntuales o edición profesional.
- **OGG:** Alternativa libre al MP3, con buena compresión y calidad.

Cómo integrar audio en una animación

La mayoría de las herramientas modernas permiten importar y sincronizar sonido fácilmente:

- En **Adobe Animate**, **After Effects**, **Blender**, **Lottie**, o herramientas web como **Howler.js** o **Tone.js**, puedes:
 1. Importar el archivo de audio a la biblioteca del proyecto.
 2. Arrastrarlo a la línea de tiempo o pista correspondiente.
 3. Sincronizarlo con eventos visuales clave, ajustando su inicio, duración o tipo de reproducción.

Tipos de sincronización de audio (conceptos actuales)

- **Sincronización por evento (Event):** El audio comienza al llegar al fotograma correspondiente y continúa reproduciéndose aunque la animación avance o se repita.
- **Sincronización por flujo (Stream):** El sonido avanza sincronizado con los fotogramas. Si se detiene la animación, también se detiene el sonido. Ideal para animaciones de voz o sincronización labial.
- **Sincronización controlada por código:** Con JavaScript, puedes reproducir sonidos en respuesta a eventos como clics, desplazamientos o cambios de escena, lo cual es común en animaciones interactivas para la web.
- **Detención controlada:** Permite detener, pausar o reiniciar sonidos desde la línea de tiempo o mediante programación.

Buenas prácticas al trabajar con audio

- Utiliza efectos sonoros cortos para acciones clave.
- Ajusta el volumen y evita sobrecargar la animación con demasiados sonidos simultáneos.
- Comprueba que el sonido no interfiera con la comprensión del mensaje visual.
- Optimiza el formato y la compresión para mejorar el rendimiento web.
- Asegúrate de que los audios estén bien sincronizados con los elementos visuales (por ejemplo, abrir una puerta debe coincidir con el chirrido).

Una animación bien sincronizada con el audio transmite emoción, realismo y coherencia. El sonido no debe añadirse al final como un complemento, sino pensarse desde el principio como parte integral de la narrativa.

9. Integración de vídeo en una animación

La integración de vídeo en una animación es una técnica muy útil para enriquecer presentaciones, contenidos educativos, interfaces web o piezas multimedia interactivas. Gracias a los estándares actuales como **HTML5** y a herramientas modernas, es posible incorporar vídeo de forma eficiente, accesible y responsiva.

Formatos de vídeo recomendados hoy en día

Los formatos más utilizados para animaciones web y multimedia son:

- **MP4 (H.264 + AAC):** Compatible con la mayoría de navegadores, plataformas y dispositivos. Es el estándar recomendado para la web.
- **WebM:** Formato abierto, con buena compresión y calidad.
- **OGV (Ogg Video):** Alternativa libre, aunque con menos soporte en navegadores actuales.

¿Cómo incorporar vídeo en una animación?

Existen diferentes formas según la plataforma:

Herramientas como Adobe Animate o After Effects:

- Importan el archivo de vídeo a la biblioteca del proyecto.
- El vídeo se puede arrastrar a la línea de tiempo como un clip.
- Es posible sincronizarlo con otros elementos gráficos o de sonido.
- Se puede elegir entre incrustarlo (para vídeos cortos) o vincularlo por URL (recomendado para vídeos largos).

En entornos web (HTML5):

- Utilizamos la etiqueta `<video>` para incrustar vídeos directamente en la página.
- Se pueden controlar con JavaScript para sincronizarlos con animaciones, efectos o interacciones.
- `<video src="video.mp4" controls autoplay muted> </video>`

En Lottie, Unity, o herramientas interactivas:

- El vídeo se suele importar como recurso separado y reproducirse con eventos.
- No suele incrustarse directamente en el archivo de animación JSON o escena, sino como un complemento.

Buenas prácticas

- Para vídeos largos o de alta calidad, usa **referencias externas** en lugar de incrustarlos (mejora el rendimiento y reduce el tamaño del archivo).
- Evita formatos propietarios como FLV o SWF, que han quedado obsoletos y ya no son compatibles con navegadores modernos.
- Usa compresión adecuada para optimizar la carga sin perder calidad perceptible.
- Asegura la sincronización del vídeo con el audio y otros elementos visuales, especialmente si se activa por interacción del usuario.
- Considera la accesibilidad: añade **subtítulos** o una transcripción si el contenido es educativo o narrativo.

Actualmente, **HTML5 y los reproductores modernos** permiten una integración de vídeo más robusta, eficiente y sin necesidad de complementos externos como Flash. Es la forma más recomendada para la Web.

10. Efectos y filtros de efectos

Las animaciones pueden enriquecerse significativamente aplicando **efectos visuales** o **filtros** que mejoran su impacto, claridad o expresividad. Estas herramientas permiten dotar a los elementos gráficos de mayor dinamismo, profundidad o estilo profesional.

Los efectos pueden aplicarse tanto a gráficos como a texto, audio o vídeo.

Efectos de color (visuales)

Hoy en día, herramientas como **CSS**, **After Effects**, **Blender**, **Lottie**, **Photoshop**, **Figma** o **Canva** permiten aplicar los siguientes efectos de color:

- **Brillo:** Modifica la intensidad luminosa del objeto. Se puede animar con transiciones suaves en CSS o timeline en AE.
- **Contraste:** Realza la diferencia entre colores claros y oscuros.
- **Tinte (Hue):** Cambia la tonalidad general del color del objeto.
- **Opacidad (Alpha):** Controla la transparencia. Usado para efectos de entrada/salida o énfasis visual.
- **Saturación:** Ajusta la viveza de los colores.
- **Filtros avanzados (CSS filter):** Se pueden aplicar efectos como `blur()`, `brightness()`, `grayscale()`, `sepia()`, etc., incluso animables por CSS o JavaScript.

Efectos sonoros

Aunque las herramientas de animación no son editores de audio, permiten aplicar ciertas modificaciones básicas:

- **Volumen inicial y final:** Se pueden establecer valores distintos al inicio y fin del clip.
- **Desvanecimiento (fade-in/fade-out):** Transición progresiva del volumen.
- **Canales izquierdo/derecho:** Silenciar o equilibrar canales (útil en audio estéreo).
- **Compresión básica:** En algunos programas, es posible aplicar normalización del volumen.

En herramientas como **Adobe Audition**, **Audacity** o editores web, puedes aplicar efectos avanzados (eco, reverb, ecualización, pitch...).

Filtros visuales modernos

Actualmente, los filtros se aplican en tiempo real mediante CSS, SVG o frameworks de animación. Algunos ejemplos:

- **Sombra (box-shadow, drop-shadow):** Añade profundidad visual o efecto de elevación.
- **Desenfocado (blur()):** Simula foco/desenfoque, útil para transiciones suaves o fondos dinámicos.
- **Iluminado:** Simula bordes iluminados o halos, común en tipografía.
- **Biselado/Relieve:** Aplica sombras internas y externas para dar volumen.
- **Ajuste de color:** Permite cambiar brillo, saturación, contraste o tono dinámicamente.
- **Transformaciones combinadas:** Algunos programas permiten aplicar múltiples filtros en conjunto (como capas).

Estos efectos son esenciales para guiar la atención del usuario, destacar elementos clave o mejorar la estética general de la animación.

11. Animaciones con CSS

En CSS es posible realizar una gran variedad de animaciones visuales sin necesidad de utilizar JavaScript. Estas animaciones son útiles para mejorar la experiencia del usuario y aportar dinamismo a los elementos de una página web. A continuación, se describen las animaciones más comunes que se pueden crear con **CSS3**, organizadas por tipo.

11.1. Animaciones básicas con transition

Este grupo incluye animaciones que afectan a las siguientes propiedades:

- Cambio de color de fondo o del texto
- Cambio de tamaño o escala
- Desplazamiento (mediante margin, padding o transform: translate)
- Opacidad (efecto *fade in/out*)
- Cambio de borde o sombra

Antes de aplicar la animación, es importante establecer el **estado inicial** del elemento a través de la propiedad transform, por ejemplo:

```
transform: translateY(0) scale(1);
```

Esto significa que:

- translateY(0) → el elemento no se ha desplazado verticalmente.
- scale(1) → el elemento se muestra en su tamaño original (100%).

A continuación, se definen las propiedades que se animarán mediante transition, indicando **la duración** y **la curva de velocidad** (timing function). Esta propiedad permite suavizar los cambios en color, transformación, opacidad, sombra y bordes, entre otros.

Funciones de temporización (timing-function)

Las funciones de temporización determinan **cómo varía la velocidad** de la animación durante su desarrollo. Los valores más comunes son:

Valor	Descripción
ease	Comienza lento, acelera en el medio y termina lento.
linear	Velocidad constante desde el inicio hasta el final.
ease-in	Comienza lento y luego acelera.
ease-out	Comienza rápido y desacelera al final.
ease-in-out	Comienza y termina lentamente, acelera en el centro.
cubic-bezier()	Permite definir una curva de velocidad personalizada.

Ejemplo:

```
div {  
  transition: all 0.5s ease;  
}
```

Esto indica que **todos los cambios de propiedades** (all) se animarán en **0.5 segundos** siguiendo la curva de velocidad ease.

11.2. Animaciones continuas con keyframes

El uso de la regla @keyframes en CSS permite definir una secuencia completa de estados por los que pasará un elemento a lo largo del tiempo. Esta técnica es ideal para animaciones continuas, repetitivas o más elaboradas, que no se activan por interacción directa del usuario, sino que se reproducen automáticamente o en bucle.

Sintaxis general:

```
@keyframes nombre_animacion {  
  0% { /* estado inicial */ }  
  50% { /* estado intermedio */ }  
  100% { /* estado final */ }  
}  
  
.elemento {  
  animation: nombre_animacion duración tipo_repetición función_tiempo;  
}
```

Ejemplos comunes de animaciones con @keyframes:

- **Rotaciones (giros constantes):** Se usa habitualmente en iconos de carga o botones giratorios.

```
@keyframes girar {  
  from { transform: rotate(0deg); }  
  to { transform: rotate(360deg); }  
}  
  
.icono {  
  animation: girar 2s linear infinite;  
}
```

- **Pulsaciones o latidos (scaling pulsante):** Ideal para llamar la atención sobre botones, alertas o elementos activos.

```
@keyframes latido {  
  0%, 100% { transform: scale(1); }  
  50% { transform: scale(1.2); }  
}  
  
.boton {  
  animation: latido 1s ease-in-out infinite;  
}
```

- **Parpadeo o intermitencia (flicker/blink):** Puede aplicarse a textos, alertas o cursores.

```
@keyframes parpadeo {  
  0%, 100% { opacity: 1; }  
  50% { opacity: 0; }  
}  
  
.alerta {  
  animation: parpadeo 1s linear infinite;  
}
```

- **Movimiento horizontal, vertical o circular:** Movimiento continuo hacia un lado, como una cinta transportadora:


```
@keyframes moverDerecha {
  0% { transform: translateX(0); }
  100% { transform: translateX(200px); }
}

.caja {
  animation: moverDerecha 2s ease-in-out infinite alternate;
}
```

Desplazamiento de fondo (scroll infinito o parallax): Muy útil para fondos de nubes, estrellas o cintas de noticias.

```
@keyframes fondoScroll {
  0% { background-position: 0 0; }
  100% { background-position: -1000px 0; }
}

.fondo {
  background-image: url("fondo.png");
  background-repeat: repeat-x;
  animation: fondoScroll 10s linear infinite;
}
```

Animación de carga (loader animado): Círculos giratorios, barras de progreso, puntos que aparecen secuencialmente, etc.

```
@keyframes cargar {
  0% { transform: scale(0); opacity: 0; }
  50% { transform: scale(1); opacity: 1; }
  100% { transform: scale(0); opacity: 0; }
}

.loader div {
  animation: cargar 1.2s infinite ease-in-out;
}
```

Propiedades de control de animaciones

Propiedad	Función
animation-name	Nombre de la animación (@keyframes)
animation-duration	Duración total
animation-timing-function	Curva de velocidad (ease, linear, etc.)
animation-iteration-count	Número de repeticiones (infinite, 1, 2...)
animation-direction	Sentido de ejecución (normal, reverse, alternate)
animation-delay	Retraso antes de iniciar la animación

Estas animaciones permiten crear experiencias visuales atractivas y dinámicas sin necesidad de JavaScript, mejorando notablemente la interfaz de usuario cuando se utilizan de forma adecuada.