

INFORME TÉCNICO COMPLETO

Free Text Motion

Estudio de Texto Animado con 205 Presets de Motion Graphics y Exportación

con Canal Alfa

Responsable del Proyecto

Eduardo Andrés Fierro Duque

Santiago de Chile — Agosto 2026

Índice

1. Resumen Ejecutivo

2. Guía Rápida de Uso (5 pasos)

3. Stack Tecnológico

4. Arquitectura del Sistema

5. Catálogo Completo de los 205 Presets (30 Familias)

6. Fórmulas Matemáticas por Categoría de Efecto

7. Métodos Web Utilizados

8. Modelo de RAM y Rendimiento

9. Flujo de Datos Completo

10. Conclusiones

1. Resumen Ejecutivo

Free Text Motion es un estudio de animación de texto (kinetic typography) que funciona

completamente en el navegador. Su propósito es simple y concreto: el usuario escribe un texto, elige una de las 205 animaciones disponibles, ajusta unos pocos parámetros y descarga un video listo para usar en sus proyectos.

Ejemplo real de uso: un creador de contenido escribe "OFERTA 50%" y elige el preset "Rebote".

Ajusta la duración a 3 segundos, activa el bucle, desactiva el fondo y exporta en WebM con transparencia. En menos de un minuto obtiene un letrero animado con fondo transparente que puede superponer sobre cualquier video en su editor, como haría con un archivo de After Effects, pero sin instalar nada y sin costo.

Qué puede hacer concretamente la herramienta:

Escribir textos de una o varias líneas y animarlos con 205 presets organizados en 30 familias (entradas, rebotes, glitch, ondas, luces, barridos, física y más).

Cambiar la fuente al instante: 220 fuentes de Google en un único selector agrupado por categoría, donde cada nombre se muestra en su propia tipografía para reconocerla al instante, con carga dinámica por demanda.

Controlar el ritmo: duración total, duración de entrada, retraso entre letras (stagger), curva de aceleración (8 easing) y animación de salida espejo.

Animar propiedades con keyframes: tamaño, espaciado de letras, interlineado, opacidad, rotación y escala cambian a lo largo del tiempo con puntos arrastrables.

Personalizar el aspecto: relleno, contorno o ambos, colores, sombra, desenfoque, motion blur y fondo activable.

Editar cada letra por separado al estilo Photoshop/After Effects, integrado en la misma sección de tipografía: al tocar una letra sobre el escenario, los controles de fuente, peso, cursiva, tamaño y color pasan a editar solo esa letra (con aviso "Editando letra N" y botón de deseleccionar); se arrastra para moverla y un tirador naranja cambia su tamaño; al deseleccionar, todos los controles vuelven a editar el texto completo.

Exportar en WebM (con canal alfa real), MOV (ProRes 4444 con alfa, compatible con After Effects y Premiere), MP4 o GIF (con transparencia), a 24, 30 o 60 fps.

Guardar y abrir proyectos (.textmotion) y autoguardado automático en el navegador.

La herramienta está disponible en producción dentro del hub de Free Animation Power (<https://freeanimationpower.org/tools/textmotion/>) y como demo pública en GitHub Pages (https://freeanimationpower.github.io/FAP_TEXTMOTION/). El código es abierto y vive en https://github.com/freeanimationpower/FAP_TEXTMOTION.

2. Guía Rápida de Uso (5 pasos)

Paso 1 — Escribir el texto: en el panel izquierdo, sección "Texto", se escribe el contenido.

Cada salto de línea del cuadro de texto crea una línea separada en la animación.

Paso 2 — Elegir la animación: en el panel derecho se abre la familia deseada (por ejemplo "Rebote") y se hace clic en un preset. El buscador superior del panel encuentra cualquier efecto por nombre (pruebe con "lluvia" o "glitch"). Al seleccionar un preset se aplican automáticamente sus valores recomendados de duración de entrada, stagger y easing.

Paso 3 — Ajustar el ritmo: en la sección "Timing" se define la duración total (0.5 a 20 segundos), la duración de entrada, el stagger por letra (0 a 200 ms) y la curva de easing. La barra de tiempo inferior permite reproducir, pausar (ESPACIO) y arrastrar el cursor para revisar cualquier instante.

Paso 4 — Personalizar: tipografía y colores en el panel izquierdo; parámetros propios del preset (velocidad, amplitud, potencia) en "Parámetros del preset"; y cambios a lo largo del tiempo en la sección "Keyframes": se activa la propiedad, se hace clic en la barra para añadir puntos y se arrastran para moverlos en el tiempo.

Paso 5 — Exportar: el botón "Exportar" abre el diálogo con formato (WebM, MP4 o GIF), fps (24/30/60) y la opción "Fondo transparente". WebM y GIF conservan el canal alfa real; en MP4 la transparencia se renderiza en negro. El video resultante se descarga automáticamente.

3. Stack Tecnológico

Capa

Tecnología

Motor gráfico

Canvas 2D API nativa (1280×720), sin frameworks

Frontend

HTML5 + CSS3 + Vanilla JS (cero dependencias)

Backend

PHP 8 (solo el email gate del hub; todo el motor corre en el cliente)

Fuentes

Google Fonts API dinámica (css2) + Font Loading API

Exportación video

MediaRecorder + canvas.captureStream(fps); WebM VP9 con alfa o MP4 H.264

Exportación MOV

FFmpeg.wasm (@ffmpeg/core 0.12): ProRes 4444 con alfa (yuva444p10le), fallback qtrle (argb)

Exportación GIF

Encoder LZW propio embebido con transparencia por índice de color

Persistencia

localStorage (autoguardado) + archivos JSON

.textmotion (Blob/FileReader)

Distribución

Single-file PHP (producción) + demo estática en

GitHub Pages (docs/)

4. Arquitectura del Sistema

La herramienta se apoya en cuatro piezas: un estado global, un registro declarativo de presets,

un sistema de keyframes y un motor de render determinista por tiempo.

ESTADO GLOBAL (state)

Un único objeto reúne los ~35 parámetros editables: texto, fuente, peso, tamaño, espaciado, interlineado, alineación, modo de relleno, colores, fondo, preset activo, parámetros por preset, easing, duración de entrada, stagger, animación de salida, duración total, bucle, opacidad, rotación, escala, desenfoque, motion blur, sombra, keyframes y tiempo de reproducción. Cada control de la interfaz escribe en este objeto y el motor lo lee en cada frame.

REGISTRO DECLARATIVO DE PRESETS

Cada preset es un objeto {id, family, name, defStagger, defInDur, defEasing, forceMode, unit, params, fx}. La función fx recibe el contexto de cada carácter (índice, ancho, progreso, tiempo global, estado) y devuelve las transformaciones a aplicar: posición (dx, dy), escala (scX, scY), rotación, sesgado (skew), opacidad, desenfoque, kerning, espaciado extra, colores, sombra, duplicados (dupes) y recorte (clip, clipDir). Agregar un efecto nuevo es añadir una línea de código, sin tocar el motor. Un índice PRESET_BY_ID resuelve el preset activo en O(1).

SISTEMA DE KEYFRAMES

Seis propiedades admiten animación por keyframes: tamaño, espaciado de letras, interlineado, opacidad, rotación y escala. Cada propiedad guarda una lista ordenada de puntos {t, v} y un interruptor de activación. Durante el render, la función kfValue() interpola linealmente el valor de cada propiedad para el instante t antes de dibujar. Cuando una propiedad está animada, su slider estático se desactiva para evitar confusión.

SISTEMA DE LETRAS (PER-LETTER)

Cada carácter puede editarse de forma individual. state.letters guarda, por índice plano del carácter, overrides opcionales {fill, size, rot, dx, dy}. La selección se hace con clic sobre el escenario (hit-test contra lastLayout, las cajas estáticas de cada letra que render() reconstruye en cada frame) o desde la rejilla de fichas del panel izquierdo. En el pipeline del render, los overrides se componen SOBRE el resultado del fx del preset: el color reemplaza al relleno, el tamaño multiplica la escala, la rotación se suma y los desplazamientos se añaden a la posición. Así, una letra coloreada o agrandada sigue participando de la animación del preset y de los keyframes sin romper la composición.

MOTOR DETERMINISTA (render)

Una única función render(ctx, t) dibuja el estado completo de la animación para cualquier instante t: limpia el canvas, pinta el fondo opcional, interpola keyframes, calcula la fase de entrada/reposo/salida, mide cada carácter con measureText y, por cada uno, aplica stagger, easing, el fx del preset y la cadena de transformaciones (translate -> rotate -> scale -> skew -> alpha -> blur -> sombra), dibuja el glyph (fillText y/o strokeText) y sus duplicados. El preview en vivo (60 fps) y la exportación frame a frame usan esta misma función, lo que garantiza que lo que se ve es exactamente lo que se exporta.

5. Catálogo Completo de los 205 Presets (30 Familias)

ENTRADAS (10): fade (aparición suave); slideUp/Down/Left/Right (deslizamientos desde cada

borde); scalePop (escala con overshoot); rotateIn (giro de -45°); blurIn (desenfocado 20px->0); flipX (volteo horizontal); zoomIn (zoom desde 20%).

MÁQUINA (5): typewriter (revelado con cursor parpadeante); Flash letra a letra (destello y pop por carácter); Escritura glitch (las letras tiemblan al revelarse); Mezcla (convergen desde posiciones dispersas); Lluvia (caen desde arriba con estela). Ninguno sustituye los caracteres escritos: las letras reales se mantienen siempre.

ESPACIADO (5): kernExpand (tracking que se expande desde el centro); kernCollapse (nace amplio y se compacta); trackingOut (tracking creciente); stretchWide (letras estiradas al 190%); wordByWord (revelado por palabra).

ONDAS (5): wave (onda sinusoidal continua); waveCascade (onda con retraso acumulado); pendulum (rotación pendular); wobble (temblor aleatorio determinista); ripple (pulsación de escala en cadena).

3D (5): flip3D (volteo con cos()); spin3D (giro de 180°); perspectiveIn (entrada con desenfoque); fly3D (vuelo diagonal); cylinderRoll (rodado cilíndrico con sin()).

GLITCH (5): rgbSplit (canales rojo/azul desplazados en modo screen); ghostEcho (estela de copias); flicker (parpadeo cuadrado); shake (sacudida por frame); pixelate (escala cuantizada).

REBOTE (5): bounce (rebote de pelota); popcorn (escala alternada); stomp (aplastamiento con rebote); elasticDrop (caída elástica); jelly (squash & stretch continuo).

COLOR (5): hueCycle (tono giratorio); gradientSweep (barrido de degradado); rainbowFlow (arcoíris por carácter); neonPulse (halo neón pulsante); invertFlash (alterna relleno/fondo).

TRAZO (5): Trazo a mano (contorno que se dibuja progresivamente); Relleno progresivo; Brillo de contorno (grosor pulsante con halo); Doble tono (relleno + contorno); Estela de texto.

SALIDA Y AMBIENT (5): explode (explosión radial); gravityFall (caída con doble rebote); scatter (dispersión convergente); windBlow (soplo con ondulación); marqueeLoop (marquesina infinita con copia envolvente).

REVELADOS (5): wipeRight/Left/Down/Up (barridos de revelado en las 4 direcciones); irisOpen (apertura desde el centro del carácter).

ELÁSTICOS (5): rubberBand (estiramiento oscilante amortiguado); heartBeat (pulsación doble); dizzy (rotación de 220° con desenfoque); sneeze (expansión súbita con blur); tickTock (vaivén angular amortiguado).

LÍQUIDO (5): meltDown (derretido: estiramiento vertical + blur); waveSkew (onda sesgada); splash (salpicadura con squash); jellySkew (gelatina sesgada); liquidDrop (gota con rebote).

CAMINOS (5): spiralln (espiral decreciente); arcIn (parábola sobre la línea base); zigzag (entrada alternada); loopLoop (rizo antes de posarse); stairs (peldaños según hash del índice).

FÍSICA (5): pingPong (rebote horizontal continuo); springWobble (oscilación angular); ricochet (caída diagonal con rebotes); pendulumSwing (péndulo de cuerda con elevación); floatAmbient (deriva vertical lenta).

PROFUNDIDAD (5): extrudeIn (extrusión 3D con copias en grises); parallaxIn (profundidad aleatoria con blur); foldUp (pliegue horizontal); deepZoom (zoom pulsante continuo); tunnelIn (túnel: 20%->100% con blur).

LUCES (5): lightSweep (banda blanca en modo screen); strobe (alternancia de opacidad); sparkleTwinkle (destellos intermitentes); glowBreathe (halo pulsante); shineAcross (doble banda luminosa).

TIPOGRÁFICOS (5): baselineBounce (saltos sobre la línea base); swashIn (giro caligráfico alternado); kernDance (kerning oscilante); tallIn (crecimiento vertical 180%->100%); squishIn (aplastamiento con rebote).

AMBIENTALES (5): orbitLoop (órbita elíptica por letra); breathing (escala sinusoidal lenta); hoverFloat (deriva continua); windLoop (sesgado oscilante); colorShift (tono global rotatorio).

ATERRIZAJES (5): stampIn (impacto tipo sello desde 250%); helicopterLand (descenso con aspas); magnetSnap (llegada elástica desde la derecha); parachute (descenso con vaivén); crashLand (caída con impacto y rebote).

CINEMÁTICOS (10): cineFade (fundido con elevación y blur); cineTrack (tracking con blur); cineZoom (zoom suave); whipPan (paneo con rotación y desenfoque); glideIn; focusPull (desenfoque de foco); driftUp (deriva ascendente); chapterTitle (título de capítulo); microReveal (micro revelado); softPop (pop suave con Back Out).

KINETIC TYPE (10): volteo, salto, giro de 360°, deslizamiento alternado, pop, onda, golpe, rodado, estiramiento y zoom kinetic — lenguaje de la kinetic typography moderna.

EDITORIAL ELEGANTE (10): serif elegante, minimal, refinado, oro (tono HSL 48° con elevación), editorial, suizo limpio, ascenso de línea base, ajuste óptico, sobreimpulso sutil y fundido fino — movimientos sutiles para títulos y créditos.

WHIP Y OVERSHOOT (10): whip en 4 direcciones con desenfoque de movimiento; sobreimpulso de escala, giro y deslizamiento (easing Back Out); ajuste elástico; escala con puño; golpe de entrada con rebote (Bounce) — inspirado en los presets de Animation Composer (Mr. Horse).

TEXTURAS Y GRANO (10): grano, estencil (fuerza el modo contorno), grunge, VHS (RGB + jitter), fundido con blur, tinta (clipping central), papel, cinta adhesiva, fotocopia y sello de tinta — acabados analógicos y lo-fi.

FLUJO Y LIQUIDO (10): onda fluida con sesgado, ripple suave, gota, gelatina ondulada, cascada, surf, ondulación, burbujas, salpicadura doble y corriente — movimiento continuo.

MINIMAL MODERNO (10): máscaras de revelado en 4 direcciones + revelado dividido; fundido deslizante; escala suave; tracking de entrada; revelado con blur; ascenso y fundido.

3D PROFUNDO (10): inclinación 3D, rotación Y, campo de profundidad, escala en perspectiva, órbita 3D, vuelo a través, carrusel, zoom con rebote, paralaje continuo y rueda de cubo.

LETRAS OBJETO (10): caída, ascenso, barajado, vaivén, cascada de escala, blur, volteo, balanceo, pulso y brinco por letra — cada letra como objeto independiente.

ESPECTACULARES (10): explosión doble con destello, onda expansiva, estela con blur, neón parpadeante, holograma, ascenso de fuego, eléctrico, glitch de bloques, confeti y láser — para intros, trailers y momentos de alto impacto.

Cada preset expone además sus propios parámetros editables (velocidad, amplitud, potencia, ecos, profundidad, etc.) en la sección "Parámetros del preset" del panel derecho.

6. Fórmulas Matemáticas por Categoría de Efecto

6.1 Progreso de Fases del Timeline

La línea de tiempo se divide en tres fases según el tiempo absoluto t (segundos):

$\text{holdStart} = \text{inDur}$

$\text{holdEnd} = \max(\text{inDur}, \text{duration} - (\text{animOut} ? \text{outDur} : 0))$

Si $t < \text{holdStart}$: fase 'in', $p = t / \text{inDur}$

Si $t > \text{holdEnd}$ y animOut : fase 'out', $p = 1 - (t - \text{holdEnd}) / \text{outDur}$

En otro caso: fase 'hold', $p = 1$

p se acota siempre a $[0, 1]$ y representa el progreso global de la animación.

6.2 Stagger por Carácter y por Palabra

Para que la entrada se reparta entre todos los elementos:

$\text{delay} = \text{índice} \times (\text{stagger} / 1000)$

$\text{localP} = (p \times \text{inDur} - \text{delay}) / \max(0.05, \text{inDur} - (\text{total} - 1) \times (\text{stagger} / 1000))$

El denominador compensa el retraso del último elemento: todos terminan su animación exactamente cuando $p = 1$, sin importar el stagger. En presets con $\text{unit} = \text{'word'}$ el índice y el total se calculan por palabra.

6.3 Curvas de Easing (8 Curvas)

linear: $e = t$

easeIn: $e = t^3$

easeOut: $e = 1 - (1 - t)^3$

easeInOut: $e = 4t^3$ si $t < 0.5$, sino $1 - (-2t + 2)^3 / 2$

expoOut: $e = 1 - 2^{(-10t)}$

backOut: $e = 1 + (c + 1)(t - 1)^3 + c(t - 1)^2$, con $c = 1.70158$

elastic: $e = 2^{(-10t)} \cdot \sin((10t - 0.75) \cdot 2\pi/3) + 1$

bounce: $e = \text{tramos } n \cdot t^2$ con $n = 7.5625$ y cortes en $1/2.75, 2/2.75, 2.5/2.75$

6.4 Interpolación de Keyframes

El valor de una propiedad animada en el instante t es el punto medio lineal del segmento que contiene a t :

$$v(t) = k[i].v + (k[i+1].v - k[i].v) \times (t - k[i].t) / (k[i+1].t - k[i].t)$$

Antes del primer keyframe se usa el valor del primero; después del último, el del último. Con un solo keyframe el valor es constante. Las listas se mantienen ordenadas por tiempo.

6.5 Ruido Determinista hash(n)

Todos los efectos con comportamiento aleatorio (shake, wobble, scatter, explode, sparkle, stairs) usan ruido determinista, idéntico en preview y en exportación:

$\text{hash}(n) = \text{frac}(\sin(n \cdot 127.1) \cdot 43758.5453)$

El resultado está en $[0, 1)$. El argumento combina el índice del carácter con el tiempo cuantizado ($\text{Math.floor}(t \cdot \text{speed})$) para que el jitter cambie por frame, pero nunca dentro de un mismo frame.

6.6 Ondas y Efectos Ambientales

Onda vertical (wave): $dy = \sin(t \cdot \text{speed} + i \cdot 0.5) \cdot \text{amp}$

Péndulo (pendulum): $\text{rot} = \sin(t \cdot \text{speed} + i \cdot 0.4) \cdot \text{amp}$

Péndulo de cuerda (pendulumSwing): $dx = \sin(\phi) \cdot \text{amp}$, $dy = -(1 - \cos(\phi)) \cdot \text{amp} \cdot 0.5$

Rebote de pelota (bounce): $dy = -|\sin(e \cdot \pi \cdot 2)| \cdot \text{amp} \cdot (1 - e \cdot 0.6)$

Los ambientales multiplican su amplitud por $\min(1, e \cdot 2)$ para que la onda aparezca con la entrada en lugar de surgir de golpe.

6.7 Simulación 3D en Canvas 2D

La perspectiva se simula comprimiendo un eje:

flip3D: $\text{scX} = \cos((1 - e) \cdot \pi/2) \rightarrow$ la letra gira sobre su eje vertical.

cylinderRoll: $\text{scX} = \sin(e \cdot \pi) \rightarrow$ la letra rueda sobre un cilindro.

spin3D: $\text{rot} = (1 - e) \cdot 180$, $\text{scX} = 0.2 + 0.8 \cdot e$.

foldUp: $\text{scY} = \cos((1 - e) \cdot \pi/2) \rightarrow$ pliegue horizontal.

La compresión se combina con rotación y desenfoque para reforzar la profundidad.

6.8 Barridos y Clipping Direccional

Los presets de revelado recortan el glyph con un rectángulo según clipDir (cl = clip en $[0,1]$, h = alto de línea, w = ancho del carácter):

lr: $\text{rect}(x - w/2, y - h, w \cdot \text{cl}, 2h)$ — de izquierda a derecha.

rl: $\text{rect}(x + w/2 - w \cdot \text{cl}, y - h, w \cdot \text{cl}, 2h)$ — de derecha a izquierda.

tb: $\text{rect}(x - w/2, y - h, w, 2h \cdot \text{cl})$ — de arriba hacia abajo.

bt: $\text{rect}(x - w/2, y + h - 2h \cdot \text{cl}, w, 2h \cdot \text{cl})$ — de abajo hacia arriba.

center: $\text{rect}(x - w \cdot \text{cl}/2, y - h \cdot \text{cl}, w \cdot \text{cl}, 2h \cdot \text{cl})$ — desde el centro.

6.9 Sistema de Duplicados (Dupes)

Los efectos de glitch, estela, luces y extrusión dibujan el mismo glyph varias veces. Cada dupe es {dx, dy, color, alpha, gco} y se dibuja dentro del contexto transformado del carácter:

rgbSplit: copias roja y azul en $dx = \pm(1-e) \cdot \text{amp}$ con gco 'screen'.

ghostEcho: por $k = 1..echoes$, copia en $dx = -(1-e) \cdot \text{spread} \cdot k$ con alpha $0.3/k$.

extrudeIn: por $k = 1..depth$, copia gris en $dx = -k \cdot 7 \cdot (1-e)$.

sparkleTwinkle: la copia blanca solo existe cuando $\sin(t \cdot \text{speed} \cdot 3 + i \cdot 2) > 0.6$.

6.10 Degradado en Barrido (Gradient Sweep)

El preset gradientSweep crea un degradado lineal por línea (relleno -> trazo -> relleno) y lo desplaza en el tiempo:

`span = totalW + 800`

`off = (t · speed) mod span - 400`

`grad = createLinearGradient(lineX + off, 0, lineX + off + span, 0)`

El período del módulo coincide con el ancho del degradado, así el barrido es continuo y sin saltos.

6.11 Encoder GIF LZW con Canal Alfa

El GIF se genera con un encoder propio (GIF89a, paleta local por frame, LZW de 8 a 12 bits):

Cuantización: cada píxel se reduce a 5-6-5 bits (r & 0xF8, g & 0xFC, b & 0xF8) y se indexa en una paleta de hasta 255 colores. El índice 255 queda reservado para transparencia.

Canal alfa: si $a < 128$ el píxel es transparente; si no, se cuantiza su RGB. El Graphic

Control Extension se escribe con flag 0x09 e índice de transparencia 255.

LZW: diccionario con códigos de 8->12 bits, emisión de CLEAR al alcanzar 4096

entradas, sub-bloques de 255 bytes y EOI. Bucle NETSCAPE2.0 con repetición infinita.

6.12 Captura WebM con Canal Alfa (VP9)

`stream = renderCanvas.captureStream(fps)`

`mime = primer soportado de ['video/webm;codecs=vp9', 'video/webm']` (con alfa) o

`['video/mp4;codecs=avc1', ...]` (sin alfa)

`recorder = new MediaRecorder(stream, {mimeType, videoBitsPerSecond: 8000000})`

Por cada frame i ($0..duración \times fps$) se llama `render(ctx, (i/frames)·duración, mostrarFondo)` y se espera 1000/fps ms. Con el fondo desactivado el canvas queda transparente y VP9 codifica el canal alfa real. En MP4 la transparencia se renderiza en negro por limitación del códec.

6.13 Exportación MOV ProRes 4444 con FFmpeg.wasm

El MOV con transparencia (el formato nativo de After Effects y Premiere) se genera con FFmpeg compilado a WebAssembly, cargado bajo demanda desde un CDN (~30 MB, solo la primera vez) mediante el patrón:

1. Cada frame se dibuja con render() en el canvas offscreen y se exporta como PNG con

canal alfa (`canvas.toBlob 'image/png'`), escribiéndose en el sistema de archivos virtual del

WASM como `frame0001.png`, `frame0002.png`, etc.

2. Se ejecuta: `ffmpeg -y -framerate fps -i frame%04d.png -c:v prores_ks`

-profile:v 4444 -pix_fmt yuva444p10le out.mov

3. Si el códec ProRes no está disponible en la compilación, se reintentará con el códec

QuickTime Animation: -c:v qtrle -pix_fmt argb out.mov (también legible por AE).

El límite por exportación es de 300 fotogramas para proteger la memoria del WASM (10 segundos a 30 fps); para clips más largos se recomienda reducir fps o duración. Los fotogramas temporales se eliminan al iniciar una nueva exportación.

7. Métodos Web Utilizados

API

Uso en la herramienta

CanvasRenderingContext2D

fillText, strokeText, measureText, save/restore,
translate, rotate, scale, transform (skew), globalAlpha,
filter (blur), sombras, clip, createLinearGradient,
globalCompositeOperation

Font Loading API

document.fonts.load() para precargar los pesos

400/700/900 de cada fuente

HTMLCanvasElement.captureStream

Alimenta MediaRecorder en la exportación de video

MediaRecorder

isTypeSupported negocia VP9/AVC; grabación a
tiempo real

Blob + URL.createObjectURL

Descarga de video, GIF y proyectos

FileReader

Carga de proyectos .textmotion (JSON)

localStorage

Autoguardado continuo del estado

requestAnimationFrame

Bucle de previsualización a 60 fps

Pointer Events

Scrubbing del timeline y arrastre de keyframes

Google Fonts CSS API

Inyección dinámica de <link> css2 con pesos variables

8. Modelo de RAM y Rendimiento

BUFFERS DE MEMORIA

Los canvas de previsualización y exportación son de 1280×720×4 bytes = 3.69 MB cada uno. getImageData solo se usa en la exportación GIF, un frame a la vez.

El GIF se construye en un array de bytes con paleta local de 256 entradas por frame; el pico de memoria es el canvas + el array de salida.

El video no acumula memoria: los chunks del MediaRecorder se liberan al cerrar el Blob.

COSTO POR FRAME

El bucle de caracteres ejecuta por carácter: 1 measureText + 1 fx + 1-2 drawText +

dupes. Con textos típicos (<100 caracteres) el costo es menor a 1 ms, sobrando margen para los 60 fps.

filter blur es el único atributo costoso; solo se aplica cuando blur > 0 y los presets lo usan

únicamente durante la fase de entrada.

Los duplicados multiplican los drawText (máximo: 5 ecos, 10 de profundidad, 6 de estela); los límites están fijados en las definiciones de los presets.

EXPORTACIÓN

GIF: render inmediato por frame, limitado solo por el encoder.

Video: grabación a tiempo real; exportar 3 segundos tarda unos 3 segundos más la codificación.

9. Flujo de Datos Completo

1. INICIALIZACIÓN: se construye la lista de 205 presets en 30 familias (UI con buscador), se

restaura el autoguardado de localStorage y se precarga la fuente activa.

2. RENDER CONTINUO: requestAnimationFrame invoca render(ctx, state.time). El tiempo

avanza solo con playing = true; al llegar a duration se reinicia si el bucle está activo.

3. EDICIÓN: cada control escribe en el objeto state y dispara saveLS() (autoguardado). El

siguiente frame refleja el cambio.

4. KEYFRAMES: los puntos {t, v} se guardan en state.kf por propiedad; kfValue() interpola

en cada render. Los puntos se añaden con clic en la barra y se arrastran en el tiempo.

5. SELECCIÓN DE PRESET: selectPreset(id) aplica los defaults del preset (stagger, duración

de entrada, easing y modo de trazo forzado con restauración posterior), inicializa sus

parámetros y reconstruye la UI de parámetros dinámicos.

6. GUARDAR/ABRIR: JSON.stringify(state) -> Blob -> descarga .textmotion. Al abrir:

FileReader -> JSON.parse -> Object.assign -> syncUI().

7. EXPORTAR: modal con formato (WebM/MP4/GIF), fps y fondo transparente. Se renderiza

frame a frame en el canvas offscreen con la MISMA función render del preview.

8. ACCESO: en producción, el bloque PHP valida \$_SESSION['email'] y redirige al login del

hub; en localhost (127.0.0.1) el gate se omite para desarrollo. La demo de GitHub Pages es

una copia estática sin PHP.

10. Conclusiones

Free Text Motion demuestra que un estudio de motion graphics para texto puede construirse

sin frameworks, en un solo archivo y con calidad de exportación profesional: 105 presets parametrizables, keyframes para animar la propia tipografía, canal alfa real en WebM y GIF, y un motor determinista que garantiza que la previsualización y el archivo exportado son idénticos.

La arquitectura declarativa de presets permite escalar el catálogo indefinidamente: cada efecto es una función pura del contexto del carácter y del tiempo, sin estado mutable entre frames.

Esto también hace que cualquier animación sea reproducible exactamente desde su archivo de proyecto JSON.

El patrón de doble distribución (PHP con email gate para el hub + demo estática en GitHub Pages) mantiene la coherencia con el resto del ecosistema Free Animation Power y permite probar la herramienta sin instalación desde cualquier navegador moderno.