INFORME TÉCNICO COMPLETO

FAP Desktop Web

Estudio de Animación 2D Profesional para PC con Tableta Gráfica

Responsable del Proyecto

Eduardo Andrés Fierro Duque

Santiago de Chile — Agosto 2026

Índice

1. Resumen Ejecutivo

FAP Desktop Web es la versión para computadoras de escritorio del estudio de animación 2D de Free Animation Power. Comparte el mismo motor de dibujo y la misma arquitectura de buffers que FAP Mobile Web, pero eleva las capacidades significativamente en tres dimensiones: (1) resolución de canvas cinematográfica de 1920×1080 píxeles (Full HD), (2) soporte completo para tabletas gráficas Wacom, Huion y XP-Pen mediante la API de Pointer Events con captura de presión y ángulo de inclinación (tilt) del lápiz, y (3) un sistema completo de atajos de teclado que permite a los animadores profesionales trabajar sin soltar el lápiz.

La interfaz abandona el modo "teléfono" de FAP Mobile y adopta un diseño de aplicación de escritorio completa que ocupa el 100% del viewport, con una toolbar unificada en la parte superior, strips de pinceles y colores, canvas central con zoom y paneo, y timeline en la parte inferior. El espaciado y los tamaños de los controles están optimizados para interacción con mouse de alta precisión.

2. Stack Tecnológico

Capa	Tecnología
Frontend	HTML5 + CSS3 embebido + JavaScript vanilla ES5
Canvas	HTML5 Canvas 2D API — resolución fija 1920×1080, doble buffer ctx + fbCtx
Entrada	Pointer Events API con pressure, tiltX, tiltY. getCoalescedEvents() para precisión. Keyboard Events para atajos. Wheel Events para zoom.
Viewport	CSS transform: scale(zoom) translate(panX, panY) con origen en 0,0
Animación	requestAnimationFrame con control de FPS (8-60)
Exportación	gif.js (LZW Web Workers) + MediaRecorder API (WebM VP9)
Almacenamiento	Canvas en memoria + FileReader API + Blob/URL para .fap
Autenticación	PHP session_start() con verificación de $_SESSION['email']

3. Arquitectura del Sistema

3.1 Diferencias Clave con FAP Mobile

- Canvas fijo a 1920×1080 (vs. proporcional al viewport en Mobile)
- Viewport con zoom de 0.15× a 64× (vs. 0.3× a 5× en Mobile)
- Soporte de presión del stylus con suavizado EWMA (Exponentially Weighted Moving Average)
- Soporte de tilt (inclinación) del stylus para caligrafía dinámica
- getCoalescedEvents() para trazos de alta precisión en tabletas
- 36 atajos de teclado documentados en panel de ayuda
- Space + arrastrar = paneo temporal (preSpaceTool)
- Grid de composición (48px) y Safe Zone (5% y 10% de margen)
- Toolbar unificada con todos los controles en una sola barra horizontal
- Panel de atajos desplegable con diseño de 5 secciones

3.2 Sistema de Viewport

```
const viewport = {
zoom: 1, // escala actual (0.15 a 64)
panX: 0, // desplazamiento horizontal en px
panY: 0, // desplazamiento vertical en px
minZoom: 0.15, // mínimo: ver canvas completo a ~15%
maxZoom: 64 // máximo: zoom extremo para pixel art
};

function applyViewport() {
drawCanvas.style.transform =
'scale(' + viewport.zoom + ') ' +
'translate(' + viewport.panX + 'px, ' + viewport.panY + 'px)';
}
```

El zoom se aplica mediante CSS transform en el elemento canvas. Esto es deliberado: el buffer interno del canvas siempre es 1920×1080 (resolución nativa), y el zoom solo afecta la visualización. El paneo se calcula en coordenadas de pantalla divididas por el zoom actual para mantener la sensación de "arrastrar el lienzo".

4. Análisis Línea por Línea

4.1 Toolbar Unificada y Atajos de Teclado

La toolbar unificada (líneas 313-433) condensa todos los controles de la interfaz en una sola barra horizontal de 40px de altura, organizada con separadores visuales (tool-sep) y espaciadores flexibles (spacer). Los controles se agrupan lógicamente: logo + undo/redo → playback → frame counter + add/remove → FPS → herramientas → sliders de tamaño/opacidad → zoom → exportación → ayuda.

El sistema de atajos de teclado (implementado en el event listener keydown global) soporta 36 combinaciones documentadas en el panel desplegable de ayuda, organizadas en 5 categorías: Tools (B, E, G, H, O), Navigation (Space, Left/Right, Home/End), Brush ([,], Shift+[, Shift+], 1-0-=-), Zoom (Wheel, +, -, Ctrl+0), Edit & Export (Ctrl+Z, Ctrl+Shift+Z, Ctrl+S, Ctrl+O, Ctrl+Shift+G, Ctrl+Shift+V).

4.2 Canvas 1920×1080 y Ajuste de Relación de Aspecto

A diferencia de Mobile donde el canvas se adapta al contenedor, Desktop utiliza una resolución fija de 1920×1080. La función fitCanvasSurface() (líneas 780-803) calcula las dimensiones CSS del contenedor para mantener la relación de aspecto 16:9 dentro del área disponible, usando la fórmula:

Si (area_width / area_height) > (1920 / 1080):

altura_css = area_height

anchura_css = altura_css × (1920 / 1080)

Si no:

anchura_css = area_width

altura_css = anchura_css × (1080 / 1920)

Esta lógica garantiza que el canvas siempre se muestre completo (sin recortes) independientemente de las dimensiones de la ventana del navegador, manteniendo la relación de aspecto cinematográfica 16:9.

4.3 Captura de Presión y Tilt del Stylus

Esta es la característica diferenciadora más importante de FAP Desktop. La API de Pointer Events expone las propiedades pressure (0.0 a 1.0), tiltX (-90° a 90°) y tiltY (-90° a 90°) en cada evento. La función getPos() (líneas 891-900) extrae estas propiedades:

```
function getPos(e) {
var rect = drawCanvas.getBoundingClientRect();
return {
x: (e.clientX - rect.left) * (drawCanvas.width / rect.width),
y: (e.clientY - rect.top) * (drawCanvas.height / rect.height),
clientX: e.clientX,
```

clientY: e.clientY,
pressure: e.pressure || 0.5 // default 0.5 si no hay stylus
};
}

4.4 Grid y Safe Zone

Dos overlays de composición que se renderizan sobre el canvas después del dibujo: el grid (drawGrid, líneas 695-708) dibuja líneas grises semitransparentes cada 48px en ambas direcciones (12×12 = 144 subdivisiones en horizontal, 22 en vertical). La safe zone (drawSafeZone, líneas 710-732) dibuja dos rectángulos con bordes discontinuos (setLineDash) al 5% y 10% de margen desde los bordes, indicando las zonas seguras para contenido que no será recortado en diferentes formatos de exportación.

4.5 Motor de Pinceles con Presión Dinámica

La función moveDraw() (líneas 929+) incorpora el procesamiento de presión del stylus en el cálculo del ancho de línea. A diferencia de Mobile donde el ancho es fijo, Desktop utiliza el valor de presión para modular dinámicamente el grosor del trazo. Además, implementa getCoalescedEvents() (línea 944) para obtener todos los eventos de puntero que ocurrieron entre frames de renderizado, fundamentales para trazos rápidos en tabletas de alta frecuencia de muestreo.

5. Fórmulas Matemáticas de FAP Desktop

5.1 Suavizado de Presión — Filtro EWMA

La presión del stylus se suaviza mediante un filtro de media móvil exponencialmente ponderada (EWMA). Este filtro es esencial porque los valores de presión crudos del stylus pueden fluctuar rápidamente, causando variaciones bruscas en el ancho del trazo que resultan en líneas temblorosas. La fórmula es:

presión_suavizada[n] = presión_anterior × α + *presión_actual* × *(1 -* α*)*

donde α *= 0.3 (factor de suavizado, valores más altos = más inercia)*

En el código:

*lastPressure = lastPressure * 0.3 + pressure * 0.7*

Con α = 0.3, el peso de la historia es 30% y el peso de la nueva medición es 70%. Esto produce una respuesta rápida a cambios de presión (70% de la nueva lectura) pero con suficiente inercia (30% del historial) para suavizar el ruido de alta frecuencia. La constante de tiempo del filtro es τ = -1/ln(α) = -1/ln(0.3) ≈ 0.83 muestras, lo que significa que la señal alcanza ~63% de su valor final en menos de 1 evento de puntero.

5.2 Cálculo de Ángulo de Tilt para Caligrafía Dinámica

Cuando el pincel de caligrafía detecta datos de tilt del stylus, calcula el ángulo de inclinación a partir de los componentes tiltX y tiltY:

ángulo_efectivo = atan2(tiltY, tiltX) × 180 / π

Si no hay datos de tilt (mouse o stylus sin soporte):

ángulo_efectivo = 45° (slante predeterminado)

ratio = 0.3 (elipse 3.3× más ancha que alta)

La función atan2(y, x) retorna el ángulo en radianes del vector (x, y) respecto al eje X positivo. Al rotar el pincel según este ángulo, la caligrafía responde naturalmente a la inclinación física del lápiz: inclinar el stylus hacia la derecha produce trazos más anchos en horizontal, inclinarlo hacia arriba produce trazos más anchos en vertical.

5.3 Fórmula de Presión Escalada en Pinceles

La presión suavizada se escala para evitar que los trazos desaparezcan completamente con presión cero (el stylus siempre ejerce algo de presión al hacer contacto):

factor_presión = 0.2 + presión_suavizada × 0.8

ancho_línea = tamaño_pincel × escala_pincel × factor_presión × dpr

donde:

tamaño_pincel = valor del slider (1-80px)

escala_pincel = propiedad scale del pincel (0.15 a 3.0)

dpr = drawCanvas.width / 200 = 1920/200 = 9.6

El factor de presión está acotado entre 0.2 (presión = 0) y 1.0 (presión = 1). El valor mínimo de 0.2 garantiza que el trazo nunca desaparezca, mientras que el rango efectivo de 0.8 proporciona sensibilidad completa a la presión del artista. El dpr (device pixel ratio lógico) escala el ancho para que 1px en la UI del slider corresponda a un grosor visualmente consistente en el canvas 1920×1080.

5.4 Zoom con Rueda del Mouse

zoom_nuevo = zoom_actual + (deltaY > 0 ? -0.1 : +0.1)

zoom = clamp(zoom_nuevo, 0.15, 64)

Cada "tick" de la rueda del mouse cambia el zoom en ±0.1 (equivalente a ±10%). Para un zoom ×2 (200%) se requieren 10 ticks hacia arriba. Con zoom máximo de 64×, el canvas de 1920×1080 se puede ampliar hasta ver regiones de 30×17 píxeles en pantalla completa, ideal para trabajo de pixel art.

6. Métodos Web Utilizados

- Canvas 2D API: Renderizado principal en 1920×1080 nativo. Operaciones: fillRect, arc, beginPath, stroke, fill, getImageData, putImageData, drawImage, globalCompositeOperation, setLineDash, createRadialGradient, imageSmoothingEnabled.
- Pointer Events (extendido): Captura de pressure, tiltX, tiltY, pointerType. getCoalescedEvents() para muestreo de alta frecuencia en tabletas Wacom/Huion.
- Keyboard Events: 36 atajos de teclado con prevención de comportamientos por defecto del navegador. Manejo especial de Space: tap = play/pause, hold = paneo temporal (líneas de spaceHeld/spaceDidPan/preSpaceTool).
- Wheel Events: Zoom del canvas con la rueda del mouse. Evento passive: false para prevenir el scroll de la página.
- requestAnimationFrame: Playback de animación con control de FPS. Uso de timestamp para calcular delta time.
- MediaRecorder API: Exportación de video mediante canvas.captureStream(30) a 30 FPS. Codec VP9 en contenedor WebM.
- Web Workers (gif.js): Encoder LZW para GIF ejecutado en hilos separados. 2 workers para paralelismo.
- FileReader API: Carga de proyectos .fap desde el sistema de archivos. Lectura como texto JSON.
- Blob / URL.createObjectURL: Descarga de GIF, video WebM, y archivos .fap (JSON).
- CSS Transform: Zoom y paneo del canvas mediante transform: scale(zoom) translate(panX, panY).

7. Modelo de RAM y Rendimiento

■■ *Cálculo de Memoria (Canvas 1920×1080)* ■■

Bytes por frame (RGBA): 1920 × 1080 × 4 = 8,294,400 bytes ≈ 7.91 MB

Cada frame: frameCanvas + strokeCanvas = 15.82 MB

24 frames: 24 × 15.82 MB = 379.7 MB (máximo teórico)

Undo stacks (30 por frame activo): 30 × 7.91 MB = 237.3 MB adicionales

ESTRATEGIAS DE MITIGACIÓN: (1) Los frameCanvases se crean bajo demanda — un frame sin dibujar no consume memoria. (2) En la práctica, raramente se dibujan los 24 frames completos; una animación típica usa 4-8 frames clave. (3) El garbage collector del navegador libera los OffscreenCanvas cuando se eliminan las referencias. (4) clearAllFrameData() se llama al redimensionar, liberando toda la memoria de frames. (5) El límite de undo (MAX_UNDO=30) acota el crecimiento de los stacks.

8. Flujo de Datos Completo

[ENTRADA] PointerEvent (stylus/mouse) + KeyboardEvent
■
■■■ getPos(e) → {x, y, clientX, clientY, pressure}
■ ■■■ x = (clientX - rect.left) × (1920 / rect.width)
■
■■■ startDraw(e)
■ ■■■ tool == 'hand' → isPanning = true
■ ■■■ spaceHeld → preSpaceTool = activeTool; tool = 'hand' (paneo temporal)
■ ■■■ tool == 'fill' → floodFill(x, y)
■ ■■■ tool == 'brush'/'eraser' → drawDot(x, y, pressure)
■
■■■ moveDraw(e)
■ ■■■ getCoalescedEvents() → array de eventos de alta precisión
■ ■■■ Para cada evento:
■ ■■■ lastPressure = lastPressure×0.3 + pressure×0.7 (EWMA)
■ ■■■ pressureScale = 0.2 + lastPressure×0.8
■ ■■■ lw = brushSize × brushScale × pressureScale × dpr
■ ■■■ Si br.slant && tiltX/tiltY → effSlant = atan2(tiltY,tiltX)×180/PI
■ ■■■ [ctx, fbCtx].forEach → algoritmo del pincel
■
■■■ endDraw(e)
■■■ Si spaceDidPan → restaurar preSpaceTool; spaceDidPan = false
■■■ commitFrame → frameCanvases['frame_N'] = fbCtx

[ZOOM]
WheelEvent → zoom ±= 0.1 → clamp(0.15, 64) → applyViewport()
Teclas +/- → zoomIn/zoomOut() → ×1.25 / ÷1.25
Ctrl+0 → resetViewport() → zoom=1, panX=0, panY=0

[SAVE/LOAD]
Save: frameCanvases → canvas.toDataURL('image/png') → JSON { frames, state }
→ Blob → URL.createObjectURL → <a download="proyecto.fap">
Load: FileReader → JSON.parse → restaurar frameCanvases y state

9. Conclusiones

FAP Desktop Web representa la evolución lógica del motor de FAP Mobile hacia un entorno de producción profesional. La resolución 1920×1080, el soporte de presión y tilt del stylus, y el sistema de atajos de teclado transforman la herramienta de un juguete creativo móvil a un instrumento de trabajo viable para animadores independientes.

El filtro EWMA para suavizado de presión es una solución matemáticamente elegante a un problema práctico real: los datos crudos de presión del stylus contienen ruido de alta frecuencia que produce trazos visualmente desagradables. Con solo 2 operaciones por muestra (una multiplicación y una suma), el filtro logra una mejora dramática en la calidad percibida del trazo sin costo computacional significativo.

La decisión de mantener el canvas en resolución nativa 1920×1080 y aplicar el zoom exclusivamente mediante CSS transform es una optimización deliberada: evita redimensionar el buffer de píxeles (operación costosa que requiere reasignar 8MB de memoria), preserva la calidad de los trazos existentes (no hay reescalado de ImageData), y permite zoom extremo de 64× para trabajo de precisión sin pérdida de fidelidad.

— Fin del Documento 03 — FAP Desktop Web —