Motion UI Library: Kit de Animaciones Modulares

Fabian Campoverde

9 de diciembre de 2025

Tema: Motion UI y Micro-interacciones
Entregable: Construcción de una librería de UI Motion

1. Propósito y Filosofía de Diseño

El objetivo de este proyecto no fue decorar una interfaz, sino utilizar el movimiento como una herramienta de comunicación funcional. Basado en el principio de que "la latencia mata la experiencia", se desarrolló un **Sistema de Diseño Modular** enfocado en micro-interacciones que informan al usuario sobre el estado del sistema (carga, éxito, atención) de manera fluida y moderna.

Para el estilo visual, se eligió una estética **"Dark Glassmorphism"**, utilizando transparencias, desenfoques (*blur*) y luces de neón, lo cual exige una alta precisión en las animaciones para mantener la legibilidad y la sensación "premium".

2. Ingeniería y Modularidad (Nivel Senior)

Cumpliendo con el criterio de la rúbrica sobre encapsulamiento y reutilización, el código no contiene estilos "hardcodeados.en elementos específicos. Se implementó una arquitectura basada en variables CSS y clases utilitarias.

2.1. Variables Globales (Design Tokens)

Se definieron tokens en el `:root` para garantizar consistencia en tiempos y colores:

```
:root {
    --primary: #8b5cf6;       /* Color sem ntico principal */
    --accent: #06b6d4;        /* Color de acento */
    --glass-bg: rgba(255, 255, 255, 0.05);
    --transition-fast: 0.2s ease;
    --transition-medium: 0.3s ease;
}
```

2.2. Desarrollo de Módulos

2.2.1. Módulo A: Feedback de Acción (Interacción)

Se diseñó un botón con estado "Neón"que responde a dos principios:

- **Anticipación (Hover):** Elevación y brillo (`translateY(-2px)`) para invitar al clic.
- **Respuesta Háptica Visual (Active):** Escala negativa (`scale(0.98)`) al presionar, simulando un botón físico.

2.2.2. Módulo B: Estados de Espera (Loaders)

En lugar de usar el spinner predeterminado del navegador, se creó un componente `.spinner-neon` utilizando `conic-gradient`. Esto permite una carga visualmente continua sin cortes bruscos, reforzando la estética tecnológica.

2.2.3. Módulo C: Entradas (Transitions)

Para la aparición de elementos, se evitó el movimiento lineal robótico. Se implementó la clase utilitaria `.glass-card` combinada con una animación de entrada escalonada (*staggering*):

```
@keyframes fadeInUp {
    from { opacity: 0; transform: translateY(40px); }
    to { opacity: 1; transform: translateY(0); }
}
/* Retrasos para efecto cascada */
.delay-1 { animation-delay: 0.1s; }
.delay-2 { animation-delay: 0.2s; }
```

3. Estética del Movimiento (Look & Feel)

Para alcanzar el nivel de "sensación premium.exigido en la rúbrica:

1. **Timing:** Se limitaron las micro-interacciones a un rango de **200ms - 400ms**. Tiempos superiores a 500ms generan sensación de lentitud en la interfaz.
2. **Easing (Aceleración):** Se descartó `linear` en favor de curvas `ease-out` para las entradas (el objeto entra rápido y frena suavemente) y `ease-in-out` para los loaders cíclicos.
3. **Interactividad 3D:** Mediante JavaScript, se agregó un efecto de "Tilt"(inclinación) que calcula la posición del mouse sobre las tarjetas, rotándolas sutilmente en los ejes X e Y para dar profundidad.

4. Showcase / Demostración

Se ha construido una **Pantalla de Prueba (Playground)** que integra los tres módulos en un grid responsivo. Esta demostración permite testear:

- La carga inicial en cascada de las tarjetas.
- La interactividad de los botones con simulación de carga asíncrona (cambio de texto y color mediante JS).

- El comportamiento responsivo del Loader y las Notificaciones.

5. Enlaces del Proyecto

- **Repositorio de Código:** `http://github.com/FabianSSJ/Interfaces-y-Multimedia/tree/main/Actividad%201.6`