

Diseño de Interfaz con Micro-interacciones Accesibles y Alto Rendimiento

Fabian Campoverde

Contexto

La universidad requiere un tablero informativo web moderno, pero accesible para una población diversa: estudiantes jóvenes, adultos mayores, personas con sensibilidad al movimiento y usuarios con dispositivos de bajo rendimiento. El objetivo no es impresionar con animaciones complejas, sino comunicar estado, cambios y acciones de forma clara, sin generar mareos ni sobrecargar el navegador.

Micro-interacción #1: Buscador Reactivo

Qué hace: El campo de búsqueda muestra feedback inmediato cuando el usuario escribe. Se activa un spinner discreto y un texto de estado que indica que el sistema está procesando la búsqueda.

Por qué ayuda:

- Confirma que el sistema recibió la entrada del usuario.
- Reduce la ansiedad en adultos mayores que podrían pensar que la página no responde.
- Evita múltiples intentos innecesarios de escritura.

Decisión de diseño: No se usan movimientos bruscos ni rebotes. El feedback es visual y contextual (borde, texto e icono), lo que es más estable para usuarios sensibles.

Micro-interacción #2: Aparición progresiva de tarjetas

Qué hace: Las tarjetas de noticias aparecen suavemente cuando entran en la pantalla del usuario al hacer scroll.

Por qué ayuda:

- Facilita la lectura progresiva del contenido.
- Evita mostrar demasiada información de golpe.
- Refuerza la jerarquía visual sin distraer.

Decisión de diseño: Se anima únicamente la entrada inicial. Una vez visibles, las tarjetas permanecen estáticas para no generar distracción continua.

Accesibilidad: Reducción de Movimiento

Se implementa la media query `prefers-reduced-motion` para usuarios que han indicado preferencia por interfaces con menos movimiento.

Estrategia aplicada:

- Se elimina el desplazamiento físico (translate).
- Se mantiene el feedback mediante cambios de opacidad y color.
- El spinner animado se reemplaza por un indicador estático.

Esto evita mareos y problemas vestibulares sin eliminar la información de estado.

Rendimiento

Técnica 1: Debounce

Aplicación: Buscador de trámites.

Problema que evita: Llamadas excesivas a la lógica de búsqueda o al servidor en cada pulsación de tecla.

Beneficio: Reduce consumo de CPU y mejora la experiencia en celulares de gama baja.

Técnica 2: IntersectionObserver

Aplicación: Aparición de tarjetas informativas.

Problema que evita: Escuchar el evento `scroll` constantemente, lo cual sobrecarga el hilo principal del navegador.

Beneficio: El navegador decide cuándo ejecutar la lógica, mejorando eficiencia y fluidez.

Evidencia Técnica (Fragmento de Código)

CSS: Accesibilidad y Animación Controlada

```
.card {
  opacity: 0;
  transform: translateY(30px);
  transition: opacity 0.6s ease, transform 0.6s ease;
}

.card.visible {
  opacity: 1;
  transform: translateY(0);
}

@media (prefers-reduced-motion: reduce) {
  .card {
    transform: none;
    transition: opacity 0.5s ease;
  }
}
```

JavaScript: Debounce e IntersectionObserver

```
function debounce(func, wait) {
```

```
let timeout;
return (...args) => {
  clearTimeout(timeout);
  timeout = setTimeout(() => func(...args), wait);
};
}

const observer = new IntersectionObserver(entries => {
  entries.forEach(entry => {
    if (entry.isIntersecting) {
      entry.target.classList.add("visible");
      observer.unobserve(entry.target);
    }
  });
});
```

Explicación del Código

El CSS controla animaciones suaves y elimina movimiento físico cuando el usuario lo solicita. El debounce optimiza la interacción del buscador evitando ejecuciones innecesarias. IntersectionObserver activa animaciones solo cuando los elementos son visibles, reduciendo el trabajo del navegador.

Conclusión

Las decisiones de diseño priorizan claridad, accesibilidad y rendimiento sobre efectos visuales llamativos. En un entorno universitario con usuarios reales y condiciones diversas, una interfaz responsable comunica estado y cambios sin generar fatiga visual ni problemas de accesibilidad.