

# ANIMACIONES COMPLETAS DEL SISTEMA

## Sistema de Inventario - Guía de Animaciones

### TECNOLOGÍAS DE ANIMACIÓN

Tecnología	Uso	Archivos
<b>Framer Motion</b>	Transiciones de página, modales, listas	React components
<b>CSS Animations</b>	Iconos, loaders, hover effects	styles.css, lottieicons.css
<b>Styled Components</b>	Switch tema, componentes dinámicos	ThemeSwitch.jsx
<b>LDRS</b>	Spinners de carga	Dashboard loading states
<b>Recharts</b>	Gráficos animados	Dashboard charts

### 1 PÁGINA DE LOGIN/REGISTRO (Auth.jsx)

#### Transición entre Login ↔ Registro

Código:

```
import { motion, AnimatePresence } from 'framer-motion'

export default function Auth() {
  const [isLogin, setIsLogin] = useState(true)

  return (
    <div className="auth-container">
      {/* Tabs animados */}
      <div className="auth-tabs">
        <button
          className={isLogin ? 'active' : ''}
          onClick={() => setIsLogin(true)}
        >
          Iniciar Sesión
          {isLogin && (
            <motion.div
              className="active-indicator"
              layoutId="activeTab"
              transition={{ type: 'spring', stiffness: 500, damping: 30 }}
            />
          )}
        </button>

        <button
          className={!isLogin ? 'active' : ''}
```

```
    onClick={() => setIsLogin(false)}
  >
  Registrarse
  {!isLogin && (
    <motion.div
      className="active-indicator"
      layoutId="activeTab"
      transition={{ type: 'spring', stiffness: 500, damping: 30 }}
    />
  )}
</button>
</div>

/* Formularios con transición */
<AnimatePresence mode="wait">
  {isLogin ? (
    <motion.div
      key="login"
      initial={{ x: -50, opacity: 0 }}
      animate={{ x: 0, opacity: 1 }}
      exit={{ x: 50, opacity: 0 }}
      transition={{ duration: 0.3 }}
    >
      <LoginForm />
    </motion.div>
  ) : (
    <motion.div
      key="signup"
      initial={{ x: 50, opacity: 0 }}
      animate={{ x: 0, opacity: 1 }}
      exit={{ x: -50, opacity: 0 }}
      transition={{ duration: 0.3 }}
    >
      <SignupForm />
    </motion.div>
  )}
</AnimatePresence>
</div>
)
}
```

## Animaciones incluidas:

1. **Indicador activo:** Barra naranja que se desliza entre tabs (layoutId)
2. **Formulario Login:** Entra desde la izquierda, sale a la derecha
3. **Formulario Registro:** Entra desde la derecha, sale a la izquierda
4. **Fade:** Opacidad 0 → 1 al entrar

## CSS del indicador:

```
.active-indicator {  
  position: absolute;  
  bottom: -2px;  
  left: 0;  
  right: 0;  
  height: 3px;  
  background: linear-gradient(90deg, #f97316, #fb923c);  
  border-radius: 2px 2px 0 0;  
}
```

## Animación de botones al hover

CSS:

```
.btn-primary {  
  background: linear-gradient(135deg, #f97316 0%, #ea580c 100%);  
  transition: all 0.3s ease;  
}  
  
.btn-primary:hover {  
  transform: translateY(-2px);  
  box-shadow: 0 8px 20px rgba(249, 115, 22, 0.4);  
}  
  
.btn-primary:active {  
  transform: translateY(0);  
}  
  
/* Efecto ripple al hacer click */  
.btn-primary:active::after {  
  content: '';  
  position: absolute;  
  top: 50%;  
  left: 50%;  
  width: 0;  
  height: 0;  
  border-radius: 50%;  
  background: rgba(255, 255, 255, 0.5);  
  transform: translate(-50%, -50%);  
  animation: ripple 0.6s ease-out;  
}  
  
@keyframes ripple {  
  to {  
    width: 300px;  
    height: 300px;  
    opacity: 0;  
  }  
}
```

## Indicador de fortaleza de contraseña

Código:

```
const [passwordStrength, setPasswordStrength] = useState(0)

const checkPasswordStrength = (password) => {
  let strength = 0
  if (password.length >= 6) strength++
  if (password.length >= 10) strength++
  if (/^[a-z]/.test(password) && /[A-Z]/.test(password)) strength++
  if (/^\d/.test(password)) strength++
  if (/[^a-zA-Z\d]/.test(password)) strength++
  setPasswordStrength(strength)
}

// Render
<div className="password-strength">
  <div className="strength-bar">
    <motion.div
      className={`strength-fill strength-${passwordStrength}`}
      initial={{ width: 0 }}
      animate={{ width: `${(passwordStrength / 5) * 100}%` }}
      transition={{ duration: 0.3 }}
    />
  </div>
  <span className="strength-label">
    {['Muy débil', 'Débil', 'Regular', 'Fuerte', 'Muy fuerte'][passwordStrength]}
  </span>
</div>
```

CSS:

```
.strength-bar {
  height: 6px;
  background: #333;
  border-radius: 3px;
  overflow: hidden;
}

.strength-fill {
  height: 100%;
  transition: width 0.3s ease;
}

.strength-0 { background: #ef4444; }
.strength-1 { background: #f59e0b; }
.strength-2 { background: #eab308; }
```

```
.strength-3 { background: #84cc16; }
.strength-4 { background: #22c55e; }
```

## 2 DASHBOARD - Sidebar Animado

### Expansión/Contracción del Sidebar

Código:

```
const [sidebarOpen, setSidebarOpen] = useState(true)

return (
  <motion.aside
    className="sidebar"
    initial={false}
    animate={{
      width: sidebarOpen ? 260 : 80
    }}
    transition={{ duration: 0.3, ease: 'easeInOut' }}
  >
  {/* Toggle button */}
  <button
    className="sidebar-toggle"
    onClick={() => setSidebarOpen(!sidebarOpen)}
  >
    <motion.div
      animate={{ rotate: sidebarOpen ? 0 : 180 }}
      transition={{ duration: 0.3 }}
    >
      <img alt="Close icon" />
    </motion.div>
  </button>

  {/* Menu items */}
  {menuItems.map((item) => (
    <motion.div
      key={item.id}
      className={`menu-item ${activeView === item.id ? 'active' : ''}`}
      onClick={() => setActiveView(item.id)}
      whileHover={{ x: 5, backgroundColor: 'rgba(249, 115, 22, 0.1)' }}
      whileTap={{ scale: 0.95 }}
    >
      <LottieIcon name={item.iconName} size={24} />

      {/* Label con fade */}
      <AnimatePresence>
        {sidebarOpen && (
          <motion.span
            initial={{ opacity: 0, width: 0 }}
            animate={{ opacity: 1, width: 'auto' }}
          >
            {item.name}
          </motion.span>
        )}
      </AnimatePresence>
    </motion.div>
  ))}
)
```

```

        exit={{ opacity: 0, width: 0 }}
        transition={{ duration: 0.2 }}
      >
      {item.label}
    </motion.span>
  )}
</AnimatePresence>
</motion.div>
))}
```

</motion.aside>

)

### Animaciones del sidebar:

1. **Ancho:** 260px ↔ 80px (transición suave)
  2. **Flecha:** Rota 180° al contraer
  3. **Labels:** Aparecen/desaparecen con fade
  4. **Hover:** Item se desplaza 5px a la derecha + fondo naranja suave
  5. **Click:** Escala a 0.95 (efecto push)
- 

## 3 MODALES ANIMADOS

### Apertura/Cierre de Modales

#### Código genérico:

```

import { AnimatePresence, motion } from 'framer-motion'

export default function Modal({ open, onClose, children }) {
  return (
    <AnimatePresence>
      {open && (
        <>
          {/* Backdrop (fondo oscuro) */}
          <motion.div
            className="modal-backdrop"
            initial={{ opacity: 0 }}
            animate={{ opacity: 1 }}
            exit={{ opacity: 0 }}
            transition={{ duration: 0.2 }}
            onClick={onClose}
          />

          {/* Contenido del modal */}
          <motion.div
            className="modal-content"
            initial={{ scale: 0.9, opacity: 0, y: 20 }}
            animate={{ scale: 1, opacity: 1, y: 0 }}
            exit={{ scale: 0.9, opacity: 0, y: 20 }}
            transition={{ duration: 0.2 }}
          >
            {children}
          </motion.div>
        </>
      )}
    </AnimatePresence>
  )
}
```

```

        type: 'spring',
        stiffness: 300,
        damping: 25
    )}
    onClick={(e) => e.stopPropagation()}
>
    {children}
</motion.div>
</>
)
</AnimatePresence>
)
}

```

### Secuencia de animación:

1. **Backdrop:** Fade in del fondo oscuro (0.2s)
  2. **Modal:** Escala de 0.9 → 1 + fade + sube 20px (spring)
  3. **Salida:** Inverso (escala 1 → 0.9 + fade out + baja)
- 

## 4 NOTIFICACIONES TOAST

### Código (ToastProvider.jsx):

```

<AnimatePresence>
  {toasts.map((toast) => (
    <motion.div
      key={toast.id}
      className={`toast toast--${toast.type}`}
      initial={{ opacity: 0, y: 20, scale: 0.8 }}
      animate={{ opacity: 1, y: 0, scale: 1 }}
      exit={{ opacity: 0, scale: 0.8, x: 100 }}
      transition={{
        type: 'spring',
        stiffness: 400,
        damping: 25
      }}
      layout // Smooth reordering
    >
      {toast.message}
    </motion.div>
  )))
</AnimatePresence>

```

### Animaciones:

1. **Entrada:** Sube 20px, escala 0.8 → 1, fade in
2. **Salida:** Se desliza 100px a la derecha + fade out
3. **Reordenamiento:** Si hay múltiples toasts, se repositionan suavemente (layout)

## 5 SWITCH DE TEMA (ThemeSwitch.jsx)

### Animaciones CSS Complejas

**Sol → Luna:**

```
.sun-moon {
  background-color: yellow;
  transition: all 0.4s ease;
}

input:checked + .slider .sun-moon {
  transform: translateX(26px);
  background-color: white;
  animation: rotate-center 0.6s ease-in-out both;
}

@keyframes rotate-center {
  0% { transform: translateX(26px) rotate(0); }
  100% { transform: translateX(26px) rotate(360deg); }
}
```

**Manchas de luna:**

```
.moon-dot {
  fill: gray;
  opacity: 0;
  transition: opacity 0.4s ease;
}

input:checked + .slider .moon-dot {
  opacity: 1;
}

#moon-dot-1 {
  left: 10px;
  top: 3px;
  width: 6px;
  height: 6px;
}
```

**Estrellas parpadeantes:**

```
.stars {
  transform: translateY(-32px);
  opacity: 0;
  transition: all 0.4s ease;
```

```

}

input:checked + .slider .stars {
  transform: translateY(0);
  opacity: 1;
}

.star {
  animation: star-twinkle 2s ease-in-out infinite;
}

@keyframes star-twinkle {
  0%, 100% { transform: scale(1); }
  40% { transform: scale(1.2); }
  80% { transform: scale(0.8); }
}

```

### Nubes flotantes:

```

.cloud-dark {
  animation: cloud-move 6s ease-in-out infinite;
}

@keyframes cloud-move {
  0%, 100% { transform: translateX(0px); }
  40% { transform: translateX(4px); }
  80% { transform: translateX(-4px); }
}

```

## ⑥ ICONOS ANIMADOS (LottieIcon + CSS)

### Dashboard Icon:

```

.icon-dashboard {
  animation: dashboard-pulse 3s ease-in-out infinite;
}

@keyframes dashboard-pulse {
  0%, 100% {
    transform: scale(1);
    filter: drop-shadow(0 0 0 transparent);
  }
  50% {
    transform: scale(1.05);
    filter: drop-shadow(0 0 8px rgba(249, 115, 22, 0.5));
  }
}

```

## Products Icon (Caja 3D):

```
.icon-products {
  animation: products-rotate 4s linear infinite;
  transform-origin: center;
}

@keyframes products-rotate {
  0% { transform: rotateY(0deg); }
  100% { transform: rotateY(360deg); }
}
```

## Sales Icon (Carrito):

```
.icon-sales {
  animation: sales-bounce 2s ease-in-out infinite;
}

@keyframes sales-bounce {
  0%, 100% { transform: translateY(0); }
  50% { transform: translateY(-4px); }
}
```

## Hover en menú:

```
.menu-item:hover .animated-icon-container {
  animation: icon-wiggle 0.5s ease;
}

@keyframes icon-wiggle {
  0%, 100% { transform: rotate(0deg); }
  25% { transform: rotate(-10deg); }
  75% { transform: rotate(10deg); }
}
```

## 7 LOADERS Y SPINNERS

### Loader de página (LDRS):

```
import { ring } from 'ldrs'

ring.register()

export default function Loader() {
```

```

return (
  <div className="loader-container">
    <l-ring
      size="60"
      stroke="5"
      bg-opacity="0"
      speed="2"
      color="#f97316"
    />
    <p>Cargando...</p>
  </div>
)
}

```

## Skeleton loading:

```

const SkeletonProductCard = () => (
  <div className="skeleton-card">
    <div className="skeleton-image"></div>
    <div className="skeleton-title"></div>
    <div className="skeleton-text"></div>
    <div className="skeleton-text short"></div>
  </div>
)

```

## CSS:

```

.skeleton-card {
  animation: skeleton-pulse 1.5s ease-in-out infinite;
}

@keyframes skeleton-pulse {
  0%, 100% { opacity: 0.6; }
  50% { opacity: 1; }
}

.skeleton-image {
  width: 100%;
  height: 150px;
  background: linear-gradient(
    90deg,
    rgba(255, 255, 255, 0.1) 0%,
    rgba(255, 255, 255, 0.2) 50%,
    rgba(255, 255, 255, 0.1) 100%
  );
  background-size: 200% 100%;
  animation: skeleton-shimmer 1.5s infinite;
}

```

```
@keyframes skeleton-shimmer {
  0% { background-position: -200% 0; }
  100% { background-position: 200% 0; }
}
```

## 8 TRANSICIONES DE PÁGINA

### App.jsx - Rutas animadas:

```
import { useLocation } from 'react-router-dom'
import { AnimatePresence } from 'framer-motion'

export default function App() {
  const location = useLocation()

  return (
    <AnimatePresence mode="wait">
      <Routes location={location} key={location.pathname}>
        <Route path="/" element={
          <PageTransition>
            <Auth />
          </PageTransition>
        } />
        <Route path="/dashboard" element={
          <PageTransition>
            <Dashboard />
          </PageTransition>
        } />
      </Routes>
    </AnimatePresence>
  )
}
```

### PageTransition wrapper:

```
const PageTransition = ({ children }) => (
  <motion.div
    initial={{ opacity: 0, x: 20 }}
    animate={{ opacity: 1, x: 0 }}
    exit={{ opacity: 0, x: -20 }}
    transition={{ duration: 0.3 }}
  >
    {children}
  </motion.div>
)
```

## 9 GRÁFICOS ANIMADOS (Recharts)

**Los gráficos de Recharts vienen con animaciones integradas:**

1. **LineChart:** Líneas se dibujan de izquierda a derecha
2. **BarChart:** Barras crecen desde 0 hasta su valor
3. **PieChart:** Sectores aparecen girando desde 0°

**Configurar duración:**

```
<Line
  dataKey="total"
  animationDuration={1000} // 1 segundo
  animationEasing="ease-in-out"
/>
```

**Deshabilitar animaciones:**

```
<LineChart data={data}>
  <Line dataKey="total" isAnimationActive={false} />
</LineChart>
```

## 10 EFECTOS ADICIONALES

**Parallax en fondos:**

```
import { useScroll, useTransform, motion } from 'framer-motion'

const { scrollY } = useScroll()
const y = useTransform(scrollY, [0, 300], [0, 100])

<motion.div
  className="background-gradient"
  style={{ y }}
/>
```

**Efecto glass (glassmorphism):**

```
.glass-card {
  background: rgba(255, 255, 255, 0.1);
  backdrop-filter: blur(10px);
  border: 1px solid rgba(255, 255, 255, 0.2);
  border-radius: 16px;
```

```
    box-shadow: 0 8px 32px rgba(0, 0, 0, 0.1);
}
```

## Gradientes animados:

```
.animated-gradient {
  background: linear-gradient(
    -45deg,
    #f97316,
    #fb923c,
    #ea580c,
    #c2410c
  );
  background-size: 400% 400%;
  animation: gradient-shift 15s ease infinite;
}

@keyframes gradient-shift {
  0%, 100% { background-position: 0% 50%; }
  50% { background-position: 100% 50%; }
}
```

## RESUMEN DE ANIMACIONES

Elemento	Tipo	Duración	Librería
Login/Registro tabs	Slide	0.3s	Framer Motion
Sidebar	Width change	0.3s	Framer Motion
Modales	Scale + Fade	Spring	Framer Motion
Toast notifications	Slide up	0.2s	Framer Motion
Theme switch	Multiple	0.4s	CSS + Styled Components
Iconos	Rotate/Pulse	2-4s	CSS
Botones hover	Transform	0.3s	CSS
Gráficos	Draw	1s	Recharts
Page transitions	Fade + Slide	0.3s	Framer Motion
Skeleton loading	Pulse	1.5s	CSS

## EXPLICAR AL PROFESOR

### ¿Cómo funcionan las animaciones?

## 1. Framer Motion (React):

- Librería especializada en animaciones React
- Usa componente <motion.div> en lugar de <div>
- Props: initial, animate, exit, transition
- AnimatePresence: Para animar montaje/desmontaje
- Layout animations: Transiciones automáticas de posición

## 2. CSS Animations:

- @keyframes: Define secuencia de cambios
- animation: Aplica a elementos
- transition: Cambios suaves entre estados
- transform: Mover, rotar, escalar sin afectar layout

## 3. ¿Por qué usar ambas?

- Framer Motion: Componentes React, eventos, complejas
- CSS: Simples, performance, hover effects
- Combinadas: Mejor resultado y experiencia

---

## FIN DE LA DOCUMENTACIÓN COMPLETA

Has completado la documentación técnica exhaustiva del sistema de inventario. Ahora tienes 6 documentos que cubren:

1.  Introducción y arquitectura general
2.  Estructura completa de archivos
3.  Todos los endpoints de la API
4.  Componentes frontend detallados
5.  Gráficos y reportes
6.  Animaciones completas

¡Listo para cualquier pregunta del profesor! 