

Propósito y Funcionalidad

Sistema completo de indicadores de carga con múltiples variantes y modos de uso.

Componentes Disponibles

`createLoadingIndicator()`

- Spinner CSS animado
- Texto personalizable
- Tres tamaños: small, medium, large

`createSkeletonCard()`

- Efecto skeleton con gradiente animado
- Simula estructura de card

`createSkeletonTable()`

- Tabla esqueleto para datos tabulares
- Número configurable de filas

Sistema de Overlay

- `createLoadingOverlay()`: Modal bloqueante
- `showLoadingOverlay()/hideLoadingOverlay()`: Control programático

Características Técnicas Detalladas

Indicador de Carga Básico

```
javascript
```

```
export function createLoadingIndicator(text = "Cargando...", size =  
"medium") {
```

```
const sizeClasses = {  
  
  small: "loading-small",  
  
  medium: "loading-medium",  
  
  large: "loading-large"  
  
};
```

```
const container = el("div", {  
  
  class: `loading-container ${sizeClasses[size]}`,  
  
  style: `  
  
    display: flex;  
  
    flex-direction: column;  
  
    align-items: center;  
  
    justify-content: center;  
  
    padding: 20px;  
  
    color: #666;  
  
  `
```

```
},
```

```
// Spinner CSS puro
```

```
el("div", {
```

```
  class: "loading-spinner",
```

```
  style: `
```

```
    width: 40px;
```

```
    height: 40px;
```

```
    border: 4px solid #f3f3f3;
```

```
    border-top: 4px solid #007bff;
```

```
    border-radius: 50%;
```

```
    animation: spin 1s linear infinite;
```

```
    margin-bottom: 10px;
```

```
  `
```

```
  })),
```

```
// Texto configurable
```

```
el("div", {
```

```
        class: "loading-text",

        style: "font-size: 0.9em; text-align: center;"

    }, text)

);
```

```
return container;
```

```
}
```

Skeleton Card para Placeholders

```
javascript
```

```
export function createSkeletonCard(title = "Cargando...") {

    return el("div", {

        class: "card skeleton-card",

        style: `

            animation: pulse 1.5s ease-in-out infinite;

            background: linear-gradient(90deg, #f0f0f0 25%, #e0e0e0 50%, #f0f0f0
75%);

            background-size: 200% 100%; /* Para animación de shimmer */

        `
```

```
},
```

```
// Barras de carga simuladas
```

```
el("div", {
```

```
  style: `
```

```
    height: 20px;
```

```
    background: #ddd;
```

```
    border-radius: 4px;
```

```
    margin-bottom: 15px;
```

```
    width: 60%;
```

```
  `
```

```
}),
```

```
el("div", {
```

```
  style: `
```

```
    height: 15px;
```

```
    background: #ddd;
```

```
    border-radius: 4px;
```

```

        margin-bottom: 10px;

        width: 80%;
    },
    // ... más barras progresivamente más cortas

);
}

```

Skeleton Table para Datos Tabulares

javascript

```

export function createSkeletonTable(rows = 5) {

    // Generación de filas esqueleto

    const tableRows = Array.from({ length: rows }, () =>

        el("tr", {},

            el("td", {

                style: `

                    height: 20px;

                    background: #ddd;

```

```

        border-radius: 4px;

        margin: 5px 0;
        ,
    })),

    // Repetir para cada columna...

)

);

return el("div", { class: "card" },

    el("h3", {}, "Cargando datos..."),

    el("div", { style: "overflow-x: auto;" },

        el("table", { style: "width: 100%; border-collapse: collapse;" },

            el("thead", {},

                el("tr", { style: "background-color: #f5f5f5;" },

                    el("th", { style: "padding: 10px; text-align: left;" },
"Columna 1"),

                    // ... más headers

```

```

        )

    ),

    el("tbody", {}, ...tableRows)

)

)

);
}

```

Sistema de Overlay Modal

```
javascript
```

```

export function createLoadingOverlay(text = "Procesando...") {

    const overlay = el("div", {

        class: "loading-overlay",

        style: `

            position: fixed;

            top: 0;

            left: 0;

            width: 100%;

```



```
height: 100%;

background: rgba(0, 0, 0, 0.5); /* Fondo semitransparente */

display: flex;

align-items: center;

justify-content: center;

z-index: 9999; /* Por encima de todo */
`
},

el("div", {

  class: "loading-modal",

  style: `

    background: white;

    padding: 30px;

    border-radius: 8px;

    text-align: center;

    box-shadow: 0 4px 20px rgba(0, 0, 0, 0.3);
```

```
,  
  
},  
  
// Spinner grande para overlay  
  
el("div", {  
  
    class: "loading-spinner",  
  
    style: `  
  
        width: 50px;  
  
        height: 50px;  
  
        border: 5px solid #f3f3f3;  
  
        border-top: 5px solid #007bff;  
  
        border-radius: 50%;  
  
        animation: spin 1s linear infinite;  
  
        margin: 0 auto 20px;  
  
    `,  
  
    }),  
  
el("div", {
```

```
        style: "font-size: 1.1em; color: #333;"

        }, text)

    )

);

return overlay;

}
```

```
// Funciones de control del overlay
```

```
export function showLoadingOverlay(text) {

    const overlay = createLoadingOverlay(text);

    document.body.appendChild(overlay);

    return overlay; // Para control posterior

}
```

```
export function hideLoadingOverlay(overlay) {
```

```
if (overlay && overlay.parentNode) {  
  
    overlay.parentNode.removeChild(overlay);  
  
}  
  
}
```

Sistema de Estilos CSS Integrado

javascript

// Los estilos se inyectan automáticamente

```
const style = document.createElement('style');
```

```
style.textContent = `
```

```
@keyframes spin {
```

```
    0% { transform: rotate(0deg); }
```

```
    100% { transform: rotate(360deg); }
```

```
}
```

```
@keyframes pulse {
```

```
    0% { opacity: 1; }
```

```
    50% { opacity: 0.5; }
```

```
    100% { opacity: 1; }  
  
}
```

```
@keyframes fadeIn {  
  
    from { opacity: 0; }  
  
    to { opacity: 1; }  
  
}
```

```
/* Variantes de tamaño */  
  
.loading-small .loading-spinner {  
  
    width: 20px;  
  
    height: 20px;  
  
    border-width: 2px;  
  
}
```

```
.loading-large .loading-spinner {
```

```
width: 60px;
```

```
height: 60px;
```

```
border-width: 6px;
```

```
}
```

```
`;
```

```
document.head.appendChild(style);
```