

Propósito y Funcionalidad

Sistema de componentes para secciones hero y tarjetas de características con variantes configurables.

Componentes Exportados

`heroSection()`

- Título y subtítulo configurables
- Sistema de acciones (botones)
- Variantes de fondo y tamaño

`featureCard()`

- Card de características con icono
- Opcionalmente clickeable

`statusCard()`

- Display de métricas con tendencias
- Indicadores de estado visuales

`loadingCard()`

- Placeholder de carga con skeleton

Características Técnicas

Componente Hero Principal

```
javascript
```

```
export function heroSection({  
  
  title = "Bienvenido",
```

```

    subtitle = "Descripción del sistema",

    actions = [],

    background = "gradient",

    size = "large"

} = {})) {

    // Sistema de clases dinámicas

    const heroClasses = [

        "hero",

        background === "gradient" ? "" : `hero-${background}`,

        size === "small" ? "hero-sm" : size === "large" ? "hero-lg" : ""

    ].filter(Boolean).join(" ");

    const titleElement = el("h1", {

        class: size === "small" ? "text-3xl" : "text-4xl",

        style: "margin-bottom: 1rem;"

    }, title);

    const subtitleElement = el("p", {

        class: "text-lg opacity-90",

        style: "max-width: 600px; margin: 0 auto 2rem;"

    }, subtitle);

```

```

// Sistema de acciones (botones)

const actionsContainer = actions.length > 0 ?

    el("div", { class: "flex flex-wrap justify-center gap-4 mt-6" },
...actions) : null;

return el("section", { class: heroClasses },

    titleElement,

    subtitleElement,

    actionsContainer

);
}

```

Componente Feature Card

javascript

```

export function featureCard({

    icon = "📊",

    title = "Característica",

    description = "Descripción de la característica",

    onClick = null

} = {}) {

    const cardClasses = [

```

```

    "feature-card",

    onClick ? "card-interactive" : "" // Clase condicional para
    interactividad

].filter(Boolean).join(" ");

const iconElement = el("div", { class: "feature-icon" }, icon);

const titleElement = el("h3", { class: "text-xl font-semibold mb-3"
}, title);

const descriptionElement = el("p", { class: "text-gray-600" },
description);

const card = el("div", { class: cardClasses },

    iconElement,

    titleElement,

    descriptionElement

);

// Event handler opcional

if (onClick) {

    card.addEventListener('click', onClick);

    card.style.cursor = 'pointer'; // Feedback visual

}

```

```
        return card;
    }
}
```

Componente Status Card para Métricas

javascript

```
export function statusCard({
    title = "Estado",
    value = "0",
    unit = "",
    status = "online",
    trend = null,
    icon = null
} = {}) {
```

```
// Mapeo de estados a clases CSS
```

```
const statusClasses = {
    online: "status-online",
    offline: "status-offline",
    warning: "status-warning"
};
```

```
const statusClass = statusClasses[status] || "status-online";
```

```
    const iconElement = icon ? el("div", { class: "text-2xl mb-2" },
icon) : null;

    const titleElement = el("h3", { class: "text-sm font-medium mb-1"
}, title);

    // Valor con unidad opcional

    const valueElement = el("div", { class: "text-3xl font-bold mb-1"
},

    value,

    unit ? el("span", { class: "text-lg" }, unit) : ""

);

    // Indicador de estado visual

    const statusElement = el("div", { class: `status-indicator
${statusClass}` },

    el("span", { class: "status-dot" }),

    status.charAt(0).toUpperCase() + status.slice(1)

);

    // Tendencia opcional (↑, ↓, →)

    const trendElement = trend ? el("div", { class: "text-sm mt-2" },
trend) : null;

    return el("div", { class: "card-stats" },
```

```

        iconElement,

        titleElement,

        valueElement,

        statusElement,

        trendElement

    );
}

```

Componente Skeleton Loading

javascript

```

export function loadingCard({ lines = 3 } = {}) {

    // Generación dinámica de líneas esqueleto

    const skeletonLines = Array.from({ length: lines }, (_, i) =>

        el("div", {

            class: "skeleton h-4 mb-2",

            style: `width: ${100 - (i * 10)}%;` // Líneas progresivamente
            más cortas

        })

    );

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

        el("div", { class: "skeleton h-6 mb-4 w-3/4" }), // Título
        esqueleto

        ...skeletonLines
    );
}

```

);

}

