**Análisis de Componentes del Alert Dialog en SvelteKit**

**1. alert-dialog-title.svelte**

**📌 Análisis del código**

* Extiende DialogTitle de bits-ui, utilizado para definir el título del diálogo.
* Permite personalización mediante la propiedad className.
* Usa <slot /> para contenido dinámico.

**✅ Puntos positivos**

✔ Reutilizable y flexible. ✔ Uso de <slot /> para contenido dinámico. ✔ Fácil personalización mediante className.

**🔧 Posibles mejoras**

* Agregar un valor por defecto al título (<slot>Diálogo</slot>).
* Validar className como una cadena (export let className: string = "").
* Mejorar accesibilidad con aria-label.

**2. alert-dialog-action.svelte**

**📌 Análisis del código**

* Extiende DialogAction de bits-ui, utilizado para acciones de confirmación.
* Permite personalización mediante className.
* Usa <slot /> para definir el contenido del botón.

**✅ Puntos positivos**

✔ Código minimalista y reutilizable. ✔ Uso de <slot /> para contenido flexible. ✔ Personalización con className.

**🔧 Posibles mejoras**

* Agregar un texto por defecto (<slot>Confirmar</slot>).
* Asegurar que className siempre sea una cadena.
* Agregar type="button" para evitar problemas en formularios.

**3. alert-dialog-cancel.svelte**

**📌 Análisis del código**

* Extiende DialogCancel de bits-ui, usado para cerrar el diálogo sin confirmar.
* Permite personalización mediante className.
* Uso de <slot /> para definir el contenido del botón.

**✅ Puntos positivos**

✔ Sencillo y flexible. ✔ Uso de <slot /> para personalización. ✔ Fácil de integrar con DialogCancel.

**🔧 Posibles mejoras**

* Agregar un texto por defecto (<slot>Cancelar</slot>).
* Validar className como cadena (export let className: string = "").
* Agregar type="button" para mejorar accesibilidad.

**4. alert-dialog-content.svelte**

**📌 Análisis del código**

* Extiende DialogContent de bits-ui, usado para contener el contenido del diálogo.
* Permite personalización mediante className.
* Uso de <slot /> para contenido dinámico.

**✅ Puntos positivos**

✔ Código limpio y reutilizable. ✔ Uso de <slot /> para flexibilidad. ✔ Fácil personalización con className.

**🔧 Posibles mejoras**

* Agregar una clase por defecto (export let className: string = "dialog-content-default").
* Incluir role="dialog" aria-modal="true" para accesibilidad.

**5. alert-dialog-description.svelte**

**📌 Análisis del código**

* Extiende DialogDescription de bits-ui, proporcionando una descripción accesible.
* Permite personalización mediante className.
* Uso de <slot /> para contenido dinámico.

**✅ Puntos positivos**

✔ Accesibilidad integrada. ✔ Uso de <slot /> para flexibilidad. ✔ Código limpio y modular.

**🔧 Posibles mejoras**

* Agregar una descripción por defecto (<slot>Descripción del diálogo</slot>).
* Asegurar que aria-describedby esté correctamente implementado.

**6. alert-dialog-footer.svelte**

**📌 Análisis del código**

* Extiende DialogFooter de bits-ui, usado para contener botones de acción.
* Permite personalización mediante className.
* Uso de <slot /> para contenido dinámico.

**✅ Puntos positivos**

✔ Código claro y reutilizable. ✔ Uso de <slot /> para personalización. ✔ Facilita la organización del diálogo.

**🔧 Posibles mejoras**

* Agregar una clase por defecto.
* Definir una estructura base (display: flex;).
* Incluir aria-label="Opciones del diálogo" para accesibilidad.