ANEXO · TAREAS DEL ALUMNADO

LADRILLOS QUE HABLAN: EVENTOS, SNIPPETS Y MÉTODOS

Nombre: _____ Fecha: _____

CUATRO FORMAS DE HABLAR (5 MIN)

Copia la tabla del archivo de Figma (**Communication map**): qué mecanismo usa cada dirección y para qué sirve.

DIRECCIÓN	MECANISMO	SIRVE PARA...
Padre → hijo		
Hijo → padre		
Padre → hijo		
Padre → hijo		
En los dos sentidos		

NOTEROW AVISA (15 MIN)

Individual, en **tu propio proyecto**:

OK	HAZ	ESTÁ HECHO CUANDO...
☐	Añade `onopen` y `ondelete` a tu NoteRow, con JSDoc. Sin `ondelete`, no hay botón Delete.	Una fila sin `ondelete` no muestra el botón; una fila sin `onopen` no se rompe al hacer clic.
☐	Añade la historia **Deletable** con `fn()`.	Al hacer clic en Delete, el panel **Actions** registra `ondelete`.
☐	En una página de tu app, pinta una lista de filas con `{#each}`. Delete debe quitar el elemento de un array `$state`.	Al pulsar Delete, la nota desaparece de la página.

Si algo falla:

- **Todas las notas desaparecen al cargar.** Escribiste `ondelete={remove(note.id)}`: eso **llama** a `remove` al pintar. Tiene que ser una función: `() => remove(note.id)`.
- **Le pasas a la fila todo el array `notes` «para que se borre sola».** ¿Podrías usar esa fila en otra lista? La fila avisa; quien tiene los datos decide.
- **Copiaste `createEventDispatcher` de un tutorial viejo.** Es de Svelte 3–4 y está obsoleto. Bórralo y usa una prop con una función.
- **Al pulsar Delete también se abre la nota.** Metiste el botón Delete dentro del botón de la fila. Deben ser dos botones hermanos.

Antes del descanso: el panel **Actions** registra `ondelete`.

BORRAR, BARRA, DESHACER (27 MIN)

Individual, en **tu propio proyecto**. El circuito completo del mapa de Figma:

OK	HAZ	ESTÁ HECHO CUANDO…
☐	Construye `UndoBar.svelte` a partir del plano de Figma: una prop (`duration`), un evento (`onundo`), un snippet (`action`) y dos métodos (`show`, `hide`).	`show` y `hide` llevan `export`.
☐	Conecta tu página: Delete quita el elemento y llama a `undoBar.show(...)`; `onundo` lo devuelve.	Borras la **primera** nota, pulsas Undo y vuelve a salir la primera.
☐	Una historia para UndoBar. La barra es invisible hasta que se llama a `show()`, así que la historia necesita un botón.	En Storybook, al pulsar el botón aparece la barra.

Si algo falla:

- **`undoBar is undefined`.** Falta `bind:this`, o llamas a `show` antes de que exista el componente (arriba del script en lugar de dentro de `remove`).
- **Undo pone la nota al final de la lista.** Usaste `push` en lugar de `splice(index, 0, note)`.
- **La barra nunca se esconde en la app, pero sí en la demo.** Copiaste `duration: 0` de la historia a la página.
- **«undoBar.show is not a function».** No exportaste `show`.
- **Le pasas `notes` a UndoBar.** Recuerda el juego de rol: la barra mira a la pared, no tiene la lista.

Si acabas: usa el snippet `action` en una segunda historia (una barra con un botón **Close** en lugar de Undo). Después construye ModeSwitch con su CSS a partir de la anatomía de Figma.

ENTREGA

Haz commit y push. En Classroom: el enlace del repositorio y una grabación corta de pantalla (o dos capturas) de **borrar → barra → deshacer** en tu propia app.

PARA CASA

- Termina el circuito borrar → barra → deshacer en tu propia app.
- Rellena la tabla **Your turn** (tu copia del archivo de Figma) para un componente de tu app: sus eventos, snippets y métodos.
- Ejercicio de transferencia (un párrafo). Un componente `SearchBox` tiene un input y un botón Clear. La página necesita (a) saber el texto mientras el usuario escribe y (b) poner el cursor en la caja cuando el usuario pulsa Ctrl+K. ¿Qué mecanismo usas para cada cosa y por qué?