ANEXO · TAREAS DEL ALUMNADO

FORMULARIOS Y VALIDACIÓN: EL FORMULARIO QUE DICE POR QUÉ

Nombre: _____ Fecha: _____

PRIMERA HORA · PROGRAMA CON EL PROFESOR (45 MIN)

En tu `notebox`, con el archivo de Figma 13112026 · Forms and validation. Cada regla tiene que responder tres preguntas al usuario: **qué campo**, **qué está mal** y **cómo arreglarlo**.

OK	HAZ	ESTÁ HECHO CUANDO…
☐	Crea `src/lib/NoteForm.svelte`: título, nota con contador `/ 280`, etiqueta (`select`) y casilla «Pin this note», con `bind:value` y `bind:checked`. Estilos del frame 1.	Al hacer clic en la palabra «Title», el cursor salta al campo.
☐	Pon `<NoteForm onsave={addNote} />` encima de la búsqueda y borra el botón «New note».	El formulario sale en la página.
☐	Añade `onsubmit={submit}` con `event.preventDefault()` y `onsave(...)`. `addNote(data)` recibe los datos.	Al guardar, la página no se recarga, la nota aparece en la lista y los campos se vacían.
☐	Prueba `required minlength="3" maxlength="40"` en el título; después quítalos y pon `novalidate` en el `<form>`.	Has visto la burbuja del navegador y ya no sale.
☐	Crea `errors` y `valid` con `$derived`, con los textos exactos del frame **4 · The rule book**. `submit` no guarda si `!valid`.	Con un título de dos letras no se guarda nada.

Pregunta (bloque 4): ¿qué problemas tiene la validación propia del navegador? Escribe al menos dos.

SEGUNDA HORA · MOSTRAR BIEN LOS ERRORES (10 MIN)

OK	HAZ	ESTÁ HECHO CUANDO…
☐	Añade `let submitted = $state(false);`: `true` al empezar `submit`, `false` después de guardar bien.	El formulario vacío no sale en rojo al abrirlo.
☐	Título y nota con `aria-invalid`, `aria-describedby` y un `<p class="error">` debajo (frame 2).	Tras pulsar Save con el título corto, el campo se marca en rojo y sale el mensaje debajo.
☐	El contador se pone rojo con la clase `over` si pasas de 280.	El error de la nota sale en cuanto te pasas, sin pulsar Save.

EJERCICIO (35 MIN)

Individual, a partir del frame **5 · Exercise** de Figma.

OK	HAZ	ESTÁ HECHO CUANDO…
☐	Muestra el error del título en cuanto el usuario **sale** del campo (`onblur`), no solo después de Save. Estado nuevo `touched`.	Escribes dos letras, pulsas Tab y sale el error.
☐	Regla entre dos campos: fijada + «No tag» → bajo Tag: «A pinned note needs a tag.»	El formulario no es válido hasta que cambia uno de los dos.
☐	Botón «Clear» a la izquierda de Save (fondo paper, borde de 1 px ink, 110 px de ancho).	Vacía todo y esconde todos los errores, **sin enviar** el formulario.

☐ Si acabas pronto: cuando Save falla, mueve el foco al primer campo con error (`document.getElementById('title').focus()`).

Si algo falla:

- **Clear envía el formulario.** Un `<button>` dentro de un formulario es `type="submit"` si no dices otra cosa. Ponle `type="button"`.
- **Clear vacía los campos pero los errores siguen.** No has puesto `submitted` y `touched` otra vez a `false`.
- **La regla de la etiqueta es un `if` dentro de `submit`.** Mira dónde viven las otras reglas: una clave más en `errors` y una condición más en `valid`.
- **`onblur={touched = true}` sin flecha.** No se ejecuta al dibujar: al salir del campo cambia el estado y lanza `handler?.apply is not a function` (el error del martes). El handler debe ser una función: `() => (touched = true)`.
- **El error de la etiqueta no tiene `id` ni `aria-describedby`.** Piensa qué oiría alguien con un lector de pantalla.
- **Has reescrito el mensaje** («Tag required!»). El frame **4 · The rule book** manda: texto exacto.

ENTREGA

Commit **«Session 5: validated form»** y push. En Classroom: el enlace y una **captura de tu formulario mostrando dos errores a la vez**.

PARA CASA

- Termina el ejercicio.
- En **tu propia app**: construye tu formulario principal (el que crea la cosa principal de tu app) con al menos cuatro campos de tres tipos distintos, al menos tres reglas con mensajes que respondan las tres preguntas, y una regla entre campos si tu diseño la tiene. Haz commit.
- Pregunta de transferencia: un formulario de registro tiene e-mail, contraseña y repetir contraseña. Escribe las claves de su objeto `errors`, un mensaje para cada una, y di cuál es una regla entre campos.