ANEXO · TAREAS DEL ALUMNADO

TALLER: TU APP, DESDE TU DISEÑO DE FIGMA

Nombre: _____ Fecha: _____

LAS REGLAS DE HOY

- Trabajas en **tu** app, solo. Pregunta a un compañero antes que al profesor, y al profesor antes que a internet. IA: solo `cs50.ai`, para que te explique, nunca para que escriba tu app.
- Cuatro checkpoints, con la hora en la pizarra. En cada uno el profesor pasa y tú le enseñas **la app funcionando**. No el código.
- Los últimos trece minutos revisas la app de un compañero y él revisa la tuya.
- Una app más pequeña que funciona vale más que una más grande que no. Recorta pantallas, nunca checkpoints.

LA HOJA DE TRASPASO (15 MIN)

Duplica el archivo de Figma 23112026 · Workshop: from Figma to app y rellena el frame **1 · Handoff sheet**: una fila por pantalla de tu flujo principal (de tres a cinco filas), con el nivel de detalle de la fila de NoteBox. Todavía no hay código.

OK	HAZ	ESTÁ HECHO CUANDO…
☐	Una fila por **pantalla**, no por estado («lista vacía» y «lista con error» van en la columna de estado).	Tienes de tres a cinco filas.
☐	Rellena la columna de datos de cada pantalla: ¿de dónde salen?	Ninguna columna de datos está vacía («un archivo JSON» vale).
☐	Cada evento tiene su acción y cada acción, su evento.	No hay «clic en la tarjeta» sin acción.
☐	Marca con una estrella tu **pantalla principal**: la que enseñarías diez segundos a tu familia.	Los checkpoints 2 a 4 tratan de esa pantalla.

LOS CUATRO CHECKPOINTS

OK	HORA	ESTÁ HECHO CUANDO…
☐	**CP1** · 0:40	**Esqueleto.** Cada pantalla de tu hoja tiene su ruta y su archivo. La navegación de la cabecera llega a todas y marca la página actual. Tus tokens están en `src/app.css` (con **tus** nombres, no `--nb-...`).
☐	**CP2** · 1:15	**Una pantalla honesta.** Tu pantalla principal coincide con tu frame de Figma a 390 px: mismos espacios, tamaños y colores, con tokens. Y no se rompe a 1280 px.
☐	**CP3** · 1:35	**Reacciona.** Una interacción con estado y un valor derivado (un filtro, un contador, un interruptor). Un formulario con al menos dos reglas de validación y mensajes de error visibles.
☐	**CP4** · 1:45	**Tiene datos.** Tu pantalla principal recibe sus datos de un archivo JSON o de un endpoint, con `load`. `npm run build` pasa. Commit y push.

Para comprobar el CP2: navegador a la izquierda con la barra de dispositivo a 390; frame de Figma a la derecha al 100 %. Ve región por región, de arriba abajo: **tamaño, espacio, color, tipografía**. Si un frame es imposible en el navegador, puedes cambiar **primero el Figma** y después el código: diseño y código tienen que coincidir.

Para el CP4, al final:

```
npm run build
git add -A
git commit -m "Workshop 23/11"
git push
```

Si algo falla:

- **«Más o menos igual» que el Figma.** Busca un número: ¿cuánto padding tiene la tarjeta en Figma? ¿Y en tu CSS? Si no coinciden, ese es el trabajo.
- **Colores hexadecimales y píxeles sueltos.** Cada color es un token; los espacios que se repiten, también.
- **Tu diseño no tiene versión de escritorio.** No le hace falta: solo no romperse. Una columna centrada con `max-width` es una respuesta honesta; una cuadrícula de tarjetas con `auto-fill`, mejor.
- **Un contador llevado a mano** (`count++` en tres handlers). ¿Qué pasa cuando un cuarto handler se olvida? Usa `$derived`.
- **Botones que no hacen nada.** Un control sin acción, o la consigue o sale de la pantalla.
- **Errores que solo salen en la consola.** El mensaje es para el usuario.
- **Un formulario copiado de NoteBox con las reglas de NoteBox.** Tus reglas salen de **tus** datos: un precio no puede ser negativo, una fecha no puede ser pasada.
- **`npm run build` falla.** Lee el primer error: dice el archivo y la línea. Lo típico: un import de un archivo renombrado, un `{#each}` sin cerrar, una variable que usas en el marcado y nunca declaraste.

REVISIÓN ENTRE COMPAÑEROS (13 MIN)

Cambiad de **sitio**, no de portátil: te sientas delante de la app de tu compañero y de su copia del Figma, frame **3 · Review criteria**. Un tick o un comentario para cada criterio, en el recuadro amarillo de al lado. Cada comentario dice **dónde**, **qué** y **qué esperabas**: no «el formulario está mal», sino «Nuevo producto: he escrito un precio de −5 y lo ha aceptado».

OK	CRITERIO
☐	Funciona: `npm install` y `npm run dev` van en un clon limpio, y `npm run build` pasa.
☐	La pantalla principal se parece al frame de Figma a 390 px (espacios, tamaños de letra, colores).
☐	Ningún color hexadecimal dentro de un componente: los colores salen de los tokens de `app.css`.
☐	No se rompe entre 390 y 1280 px: nada se sale, nada se solapa.
☐	Cada ruta de la hoja se abre, y la página actual está marcada en la navegación.
☐	Cada botón y enlace hace algo. Apunta cualquier control muerto que encuentres.
☐	El formulario rechaza datos erróneos y dice por qué, junto al campo, en palabras sencillas.
☐	Algo de la pantalla está calculado (`$derived`), no escrito a mano: un contador, un filtro, un total.
☐	Los datos llegan con `load` desde un archivo JSON o un endpoint; una URL desconocida muestra una página de error.
☐	Solo con el teclado (Tab, Enter, Espacio) puedes completar la tarea principal.

Vuelve a tu sitio y lee la revisión. Escoge el **único** comentario que vas a arreglar primero y escríbelo arriba de la hoja:

ENTREGA

En Classroom: el enlace a tu copia del archivo de Figma (hoja rellenada, revisión recibida) y el enlace a tu repositorio, con el último commit **«Workshop 23/11»**.

PARA CASA

- Arregla el comentario que escogiste de tu revisión. Commit «Fix after review».
- Estudia para el viernes con el frame **0 · UT3 on one page** del archivo de mañana (estará en Classroom esta noche).
- Opcional, para el proyecto: lleva todas las pantallas de tu hoja al checkpoint 2. La UT4 (desde el 30/11) empieza desde este código.