ANEXO · TAREAS DEL ALUMNADO

TESTS UNITARIOS: VITEST + TESTING LIBRARY

Nombre: _____ Fecha: _____

INSTALAR Y CONFIGURAR (12 MIN)

Todos a la vez con el profesor, en **tu propio proyecto**. Cuatro pasos:

OK	HAZ	ESTÁ HECHO CUANDO…
☐	**1 · Instala:** `npm install -D vitest jsdom @testing-library/svelte @testing-library/jest-dom @testing-library/user-event`	El comando termina sin errores.
☐	**2 · `vite.config.js`:** `defineConfig` viene de `vitest/config`, añade el plugin `svelteTesting()` y el bloque `test` (abajo). La llamada `sveltekit(...)` no se toca.	El archivo tiene `svelteTesting()` en `plugins` y el bloque `test`.
☐	**3 · `vitest-setup.js`**, junto a `package.json`, con una línea: `import '@testing-library/jest-dom/vitest';`	El archivo existe.
☐	**4 · Script** en `package.json`, dentro de `"scripts"`: `"test": "vitest run"`.	`npm test` dice «No test files found» (correcto: aún no hay ninguno).

```
import { svelteTesting } from '@testing-library/svelte/vite';
import { defineConfig } from 'vitest/config';
// plugins: [ sveltekit({ ... }), svelteTesting() ],
test: {
  environment: 'jsdom',
  setupFiles: ['./vitest-setup.js'],
  include: ['src/**/*.test.js']
}
```

Si tu `vite.config.js` es distinto (un proyecto antiguo con `svelte.config.js`), conserva tu llamada `sveltekit()` y añade solo el import, `svelteTesting()` y el bloque `test`.

AHORA RÓMPELO TÚ (6 MIN)

Individual, con **tu** Button:

OK	HAZ	ESTÁ HECHO CUANDO…
☐	Copia el archivo `Button.test.js` del profesor y ejecuta `npm test`.	Ves cuatro tests en verde.
☐	**Sabotea una cosa**: cambia la variante por defecto a `'secondary'`, o borra `{...rest}`. Ejecuta `npm test`.	Un test sale en rojo. Lees el mensaje.
☐	Repara el componente y vuelve a ejecutar.	Todo en verde. Le dices a tu compañero qué test lo detectó.

Test que detectó tu sabotaje: _____

EL PLAN ANTES DEL CÓDIGO (10 MIN)

En parejas. Duplica el archivo de Figma. En Plan · NoteRow, completa las seis celdas amarillas: para cada promesa del plano, qué props preparas, qué hace el usuario y qué debe ser verdad. Pista: el nombre del botón Delete no es «Delete».

PROMESA	ARRANGE · ACT	ASSERT
Se ve el título; por defecto, sin etiqueta Pinned ni botón Delete		
`pinned` muestra la etiqueta		
Delete avisa, y no abre la nota		

TESTS DE NOTEROW (23 MIN)

Individual, en **tu propio proyecto**:

OK	HAZ	ESTÁ HECHO CUANDO…
☐	`src/lib/components/NoteRow.test.js`: un test por fila de tu plan.	`npm test` los muestra en verde.
☐	Un test para un componente de **tu propia app** (el plan Your turn).	`npm test` lo muestra en verde.
☐	Sabotea el componente una vez.	Un test sale en rojo; después lo reparas.

Si algo falla:

- **`getByRole('button', { name: 'Delete' })` no encuentra nada.** El nombre accesible es el `aria-label`: «Delete Shopping list». Lee el error: Testing Library imprime todos los roles y nombres que **sí** ve.
- **El test falla con «0 calls».** Olvidaste `await` antes de `user.click`: la comprobación se ejecuta antes de que acabe el clic.
- **`getByText('Pinned')` dentro de un `not.` da error.** `getBy` falla antes de que `not` actúe. Usa `queryByText`.
- **El test llama a `onDelete()` él mismo** y luego comprueba que se llamó. Eso prueba el test. Tiene que hacer clic el **usuario**.
- **El test copia la implementación**: `expect(component.pinned).toBe(true)`. Comprueba lo que se ve en la página.

Si acabas: un cuarto test. Con `preview: ''` no hay ningún elemento con la clase `.preview` (usa `container.querySelector`).

ENTREGA

Haz commit y push. En Classroom: el enlace del repositorio, una captura de `npm test` todo en verde y **una captura de una ejecución en rojo** con una línea: qué línea del componente rompiste y qué test lo detectó.

PARA CASA

- Termina los tres tests de NoteRow y escribe tres tests para tu propio componente (tu plan Your turn). Entrega el plan (enlace de Figma) con el código.
- Ejercicio de transferencia. Un `LikeButton` muestra un corazón y un número. Al hacer clic suma 1 y llama a `onLike` con el número nuevo. Escribe el plan de tests: tres filas (promesa, arrange · act, assert), con la misma tabla que el plan de NoteRow.