# ANEXO · TAREAS DEL ALUMNADO

## STORYBOOK Y EL PRIMER COMPONENTE

Nombre: _____ Fecha: _____

## CINCO HERRAMIENTAS, CINCO TRABAJOS (8 MIN)

Copia la tabla de la pizarra: qué trabajo hace cada herramienta y si sirve para **diseñar** o para **probar**.

| HERRAMIENTA | SU TRABAJO | ¿DISEÑAR O PROBAR? |
|---|---|---|
| **Figma** | | |
| **Storybook** | | |
| **Vitest** + **Testing Library** | | |
| **Playwright** | | |
| `svelte-package` + **npm** | | |

## LEE EL PLANO (10 MIN)

En parejas. Abre el enlace de Figma de Classroom, página **Button (spec)**. Haz clic en una instancia del component set y mira el panel derecho: **Variant**, **Size**, **Disabled**, **Label**.

Sin bajar hasta la tabla, escribid: ¿qué puede cambiar en este botón y qué valores puede tomar? Después comprobadlo con la tabla **Figma property → Svelte prop**.

## INSTÁLALO TÚ (10 MIN)

Individual, en **tu propio proyecto**:

```
npm create storybook@latest
npm run storybook
```

| OK | HAZ | ESTÁ HECHO CUANDO… |
|---|---|---|
| ☐ | Ejecuta los dos comandos en la carpeta de tu proyecto. Si te ofrece «Testing», desmárcalo. | Storybook se abre en `http://localhost:6006`. |
| ☐ | Borra la carpeta `src/stories/` (son ejemplos de otro diseño). | La barra lateral de Storybook queda vacía. |
| ☐ | Si acabas: crea `src/lib/tokens.css` con los colores de tu sistema de diseño y los nombres de la tabla de Figma. | El archivo existe y usa nombres como `--nb-ink`. |

Si algo falla:

- **`ERESOLVE unable to resolve dependency tree.`** npm eligió dos versiones distintas de Storybook. Ejecuta `npm install -D @storybook/svelte@X`, donde X es la versión de `storybook` en `package.json`.

- **El puerto 6006 está ocupado.** Otro Storybook sigue abierto en una terminal olvidada. Ciérralo.
- **Lo ejecutaste en la carpeta equivocada.** Junto a ti debe haber un `package.json` (compruébalo con `ls`).

## TU BUTTON (20 MIN)

Individual, en **tu propio proyecto** y con **tus propios tokens**:

| OK | HAZ | ESTÁ HECHO CUANDO… |
|---|---|---|
| ☐ | `src/lib/tokens.css` con los nombres de la tabla de Figma y los colores de tu sistema de diseño de la UT1. | Está importado en `+layout.svelte` y en `.storybook/preview.js`. |
| ☐ | `src/lib/components/Button.svelte` con las tres variantes: `primary`, `secondary`, `danger`. | El componente no tiene ningún código hex: solo `var(--...)`. |
| ☐ | `Button.stories.svelte`, junto al componente, con las tres historias. | Ves **Primary**, **Secondary** y **Danger** en la barra lateral. |
| ☐ | Sustituye **un** `<button>` escrito a mano en una de tus páginas por `<Button>`. | La página funciona igual que antes. |

Si algo falla:

- **Pusiste códigos hex dentro del componente.** Si cambia tu color de marca, ¿cuántos archivos abres? Debería ser uno: `tokens.css`.
- **La historia muestra un botón sin estilo.** Falta importar `tokens.css` en `.storybook/preview.js`: el catálogo no carga tu layout.
- **El botón sale vacío.** Te olvidaste `{@render children()}`. Si escribiste `{children}`, imprime una función.
- **Storybook no encuentra las historias.** El archivo debe llamarse `Button.stories.svelte` y estar junto al componente, no en `src/stories/`.

Si acabas: duplica el archivo de Figma a tus borradores y rellena la tabla **Your turn** para un segundo componente de tu app (propiedades, tipos y valores por defecto).

## ENTREGA

Haz commit y push (`git add -A && git commit -m "Button component + stories"`). En la tarea de Classroom: el enlace del repositorio y **una captura** de Storybook con las tres historias en la barra lateral.

## PARA CASA

- Termina **Tu Button** si no lo acabaste: tres historias visibles en tu Storybook.
- Rellena la tabla **Your turn** en tu copia del archivo de Figma para un componente de tu app.
- Un párrafo: tu app tiene una «tarjeta de nota» copiada en cuatro páginas, cada una con un padding un poco distinto. Explícale a un compañero, con la analogía del LEGO, qué harías y qué gana.