ANEXO · TAREAS DEL ALUMNADO

LA APP GANA SENTIDOS: NUI + MEDIAPIPE + PRIMERA DEMO

Nombre: _____ Fecha: _____

TAREA 1 · CLASIFICA LAS TARJETAS (10 MIN)

En parejas. Abre el enlace de Figma 08012027 · Natural user interfaces, página **Students**, y haz **Duplicate to your drafts**. Tablero **A**: nueve tarjetas de productos y cuatro columnas, una por familia: **voz**, **manos y cara**, **cuerpo entero** y **realidad aumentada (AR)**.

OK	HAZ	ESTÁ HECHO CUANDO…
☐	Arrastra cada tarjeta a su familia.	Las nueve tarjetas están en una columna.
☐	Añade una palabra a cada tarjeta: el sensor.	Cada tarjeta dice qué sensor usa (micrófono, cámara…).
☐	Busca la tarjeta que va en dos columnas.	Esa tarjeta aparece en las dos familias.

TAREA 2 · EL PIPELINE DE TU APP (FIGMA, TABLERO B)

Cuatro cajas: **sensor → modelo → regla → feedback**. Completa la fila **Your app** del tablero **B** con una interacción natural para tu app. Antes, piensa y responde:

1 ¿Quién decide que un pulgar arriba fija una nota: la cámara, el modelo o tú?

2 Un hospital quiere una pantalla sin manos en el quirófano. ¿El modelo en el navegador o en un servidor? ¿Por qué?

3 ¿Por qué no podemos ejecutar un chatbot enorme de la misma manera, dentro del navegador?

TAREA 3 · TU PRIMERA DEMO (28 MIN)

Individual, en **tu propio proyecto** de la UT4.

OK	HAZ	ESTÁ HECHO CUANDO…
☐	En tu proyecto: `npm install --save-exact @mediapipe/tasks-vision@1.0.1`	El paquete aparece en `package.json`.
☐	Descomprime los tres archivos de Classroom en `src/lib/components/` y `src/lib/nui/`.	Tienes `CameraStage.svelte`, `camera.ts` y `vision.ts` en su sitio.
☐	Crea `src/routes/demo/+page.svelte` y **escribe** las veinte líneas (no las pegues).	El archivo compila sin errores.
☐	`npm run dev`, abre `localhost:5173/demo` en **Chrome o Edge** y permite la cámara.	La línea bajo el vídeo muestra el gesto, p. ej. `Thumb_Up (93 %)`.

Si algo falla:

- **«This page cannot use the camera» nada más empezar.** Has abierto la página por IP o con `http://` desde otro equipo. La cámara solo funciona en `https://` o en `localhost`.
- **No pasa nada durante diez segundos.** El modelo se está descargando (8,4 MB). La segunda vez es instantáneo.

- **Errores de versión en la consola.** `vision.ts` carga el runtime de la versión `1.0.1`: `package.json` debe tener la misma: por eso se instala con `--save-exact`. Nunca uses `@latest` en esa URL.
- **`Cannot find module '#lib/nui/camera.js'` en `npm run check`.** El comprobador de tipos no adivina extensiones después de `#lib`: escribe `#lib/nui/camera.js`, aunque el archivo sea `camera.ts`. La página compila igualmente. `CameraStage.svelte`, de la app de referencia (SvelteKit 3), ya lo importa así.
- **`recognizer` es undefined.** Has puesto la lógica de `frame` arriba del script y no dentro de la función.
- **Sin webcam.** Siéntate con un compañero que tenga una, pero escribe la página en tu propio proyecto: el estado «No camera found» también cuenta.

Si acabas pronto: reconoce el gesto de **las dos** manos (`numHands: 2` en `vision.ts` y un `{#each}` sobre `result.gestures`).

TAREA 4 · RÓMPELO A PROPÓSITO (7 MIN)

En parejas. El tablero **C** de Figma es la especificación del componente de cámara. Marca cada estado cuando lo hayas **visto** en tu demo:

OK	ESTADO	CÓMO LLEGAR
☐	**Off**	Carga la página.
☐	**Loading**	Pulsa el botón la primera vez.
☐	**On**	Permite la cámara.
☐	**Blocked**	Icono de la cámara en la barra de direcciones → bloquear → recargar → pulsar el botón.
☐	**No camera**	Desconecta la webcam (o usa un PC sin ella).

1 ¿Qué heurística de Nielsen es el mensaje de «Blocked»?

2 ¿Y la frase «never uploaded» del estado Off?

ENTREGA

En la tarea de Classroom: (1) un **enlace de solo lectura** a tu copia de Figma con los tableros A y C hechos y la fila **Your app** del tablero B rellenada; (2) una **captura** de tu página de demo reconociendo un gesto; (3) una línea: ¿qué interacción natural tendría sentido en TU app y de qué familia es?

PARA CASA

- Lee `CameraStage.svelte` de principio a fin y escribe un comentario con tus palabras encima de `start`, `loop` y `stop`.
- Termina el tablero B de Figma (fila **Your app**) y las marcas del tablero C.
- Ejercicio de transferencia (un párrafo): el salpicadero de un coche permite cambiar de emisora de radio moviendo la mano. Nombra las cuatro cajas de su pipeline y una razón por la que el gesto no debe ser la única forma de hacerlo.