ANEXO · TAREAS DEL ALUMNADO

CÁMBIALO, EXPÓRTALO: INFORMES INTEGRADOS + PREPARACIÓN DEL EXAMEN

Nombre: _____ Fecha: _____

El plato vuelve a la cocina: el cliente lo ha probado y pide cambios. Hoy cambias el informe **sin romperlo**, dejas que el usuario se lo lleve (en papel, como fichero, como PDF) y preparas el examen del lunes.

TAREA 1 · TRES PETICIONES DE CAMBIO (22 MIN)

En parejas. Duplica el fichero de Figma y, en la página **Change requests**, escribe en cada ticket **dónde va el cambio** (qué función o fichero). Todavía sin código.

CR	EL CLIENTE DICE…	EL CAMBIO VA EN…
1	«Muestra qué notas están fijadas, y cuántas por carpeta y en total.»	
2	«Quiero solo las notas de noviembre: puedo elegir el primer día pero no el último.»	
3	«School siempre gana. Muestra la **media** de palabras por nota, de mayor a menor.»	

Después, solo y en orden, programa cada cambio. Está hecho cuando la pantalla muestra estos números:

OK	CR	ESTÁ HECHO CUANDO…
☐	**1**	Sin filtro: Personal 1 · School 2 · Shopping 2 · Work 4 · total **9 pinned**.
☐	**2**	Desde 01/11/2026 hasta 30/11/2026: **9 notes · 1189 words · 10 min**, y sin grupo School.
☐	**3**	Barras, de izquierda a derecha: School 435 · Work 259 · Personal 182 · Shopping 41. Título «Average words per note».

Si algo falla:

- **En CR1 escribes el código del recuento dos veces** (grupo y total). No hace falta: `groupByFolder` llama a `totals`, así que una línea nueva en `totals` sirve para cada grupo **y** para el report footer.
- **En CR2 el filtro no se puede resetear.** Te olvidaste de `until` en `reset`.

TAREA 2 · INTEGRADO, NO ADJUNTO (10 MIN)

Un **informe integrado** vive dentro de la aplicación: llegas desde la navegación de la app, lee los datos de la app y se parece al resto de la app. Abre **tu propio proyecto**, marca y arregla ahora lo que falte:

OK	MI INFORME TIENE…
☐	**Un enlace** al informe en la navegación (`<a href="/report">Report</a>` en `+layout.svelte`), no una URL que haya que saberse de memoria.
☐	**Un título** en la pestaña del navegador (`<svelte:head><title>...</title></svelte:head>`).
☐	**Los datos de la app**: lee la misma fuente que el resto de la app, no una copia privada.
☐	**El aspecto de la app**: usa tus componentes de la UT4 (botones, inputs) y tus estilos, no el gris por defecto del navegador.

TAREA 3 · EXPORTAR: IMPRIMIR, CSV Y PDF (36 MIN)

Sigue al profesor. Mira la página de Figma **Design spec · Export**: tres botones debajo de la tabla.

OK	HAZ	ESTÁ HECHO CUANDO...
☐	**Print**: un botón con `window.print()` y una regla `@media print` que oculta los filtros y los botones (y la barra de la app en `+layout.svelte`).	La vista previa de impresión solo muestra el informe.
☐	**CSV**: `exportCsv()` construye el texto a partir de `rows`, lo mete en un `Blob` y hace clic en un enlace temporal. El título va entre comillas.	El fichero descargado tiene las filas filtradas y la columna `minutes`.
☐	**PDF · fichero Typst**: en `reports/notes-report.typ`, si llega `--input rows=...` usa esas filas; el detalle escribe `str(n.words)`.	Sigue compilando desde el terminal como el martes.
☐	**PDF · endpoint y botón**: crea `src/routes/report/pdf/+server.js`, que ejecuta `typst compile` con `execFile` y devuelve el PDF. En la página, `exportPdf()` envía `rows` y los filtros por `POST`. Tres estados: normal, Creating PDF… (desactivado) y error.	Con **Folder = School**, el PDF descargado tiene solo School.

Responde:

1 ¿Por qué el CSV se construye con `rows` y no con `data.notes`?

2 ¿Por qué el navegador no puede ejecutar Typst por sí mismo?

TAREA 4 · PREPARACIÓN DEL EXAMEN (22 MIN)

LUNES 01/02/2027	11:15-13:05
Calentamiento	Kahoot, 5 minutos, sin nota.
Parte A · teoría	20 minutos, en papel, 5 preguntas, **2,5 puntos**. Nada encima de la mesa.
Parte B · práctica	80 minutos, en el PC, **7,5 puntos**. Un proyecto inicial de NoteBox, un CSV que no has visto y un diseño de Figma. Construyes el informe integrado y su PDF con Typst.
Permitido en la parte B	Tus apuntes y proyectos de esta unidad, y la documentación oficial: svelte.dev, echarts.apache.org, typst.app/docs.
No permitido	**IA de cualquier tipo**, chats, mensajería, trabajo de otros. Copiar = 0 para todos los implicados.
Entrega	Un `.zip` del proyecto sin `node_modules`, más el PDF, en Classroom antes de las 13:05.

#	TAREA DEL EXAMEN	RA5	PUNTOS
1	Estructura: report header, column header, grupos con header y footer, report footer, como en el diseño.	a	1,0
2	Filtros: un selector de carpeta alimentado desde el JSON, un checkbox y el estado vacío.	b, c	1,0
3	Dos columnas calculadas, y recuentos y totales por grupo y del informe.	d	1,5
4	Un gráfico ECharts que reacciona a los filtros.	e	1,25
5	El PDF con Typst: tabla con grupos, columnas calculadas, subtotales, total general.	f	1,25
6	Una petición de cambio sobre el código de plantilla dado.	g	0,75
7	Integrado y exportado: un enlace en la navegación, un botón de imprimir y una vista de impresión limpia.	h	0,75

Ejercicio en papel (5 min, solo). Para cada fila: pendientes y progreso en %. Para cada carpeta y para el total: tareas, hechas, pendientes y progreso.

FILA	TAREAS	HECHAS	PENDIENTES	PROGRESO
A · Trip	10	5		
A · Party	2	2		
Carpeta A				
B · Exams	8	4		
Carpeta B				
Total				

Lista para el fin de semana:

☐ Tu proyecto NoteBox **compila** (`npm run build`) y `typst --version` responde en el PC que usarás.

☐ Encuentras en tu proyecto, en menos de un minuto: `toNote`, `filterNotes`, `totals`, `groupByFolder`, `Chart.svelte`, el bucle de grupos de Typst.

☐ Puedes entrar en Figma y en Classroom.

☐ Técnica de examen: **antes de programar, suma un grupo a mano**. Si tu pantalla dice otra cosa, sabes dónde mirar.

ENTREGA

En la tarea de Classroom **UT6 · Changes and export**: (1) una captura de NoteBox `/report` después de CR1, CR2 y CR3, sin filtro; (2) el `notes-report.pdf` exportado desde el botón con **Folder = School**; (3) una línea: ¿qué petición de cambio tocó menos líneas, y por qué?

PARA CASA

- Aplica **una petición de cambio propia** al informe de tu proyecto (pide a un compañero que te la escriba, con una línea está hecho cuando), y añade al menos una exportación. Tu informe debe ser accesible desde la navegación de tu app.
- Prepara el examen: repite el ejercicio en papel con tus propios números y repasa la lista del fin de semana.
- Ejercicio de transferencia (un párrafo): una app de biblioteca tiene un informe «Préstamos por mes». Llegan tres peticiones: (1) «muestra también cuántos préstamos se devolvieron tarde»; (2) «solo los préstamos de la sección infantil»; (3) «el gráfico debería ser préstamos por día de la semana». Para cada una: ¿qué capa cambia, y qué número calcularías a mano antes de programar?