ANEXO · TAREAS DEL ALUMNADO

FIRMAR, INSTALAR EN SILENCIO, ACTUALIZAR

Nombre: _____ Fecha: _____

Durante la sesión, en parejas, resuelve también en tu copia del archivo de Figma los tableros **2 · Two signatures** y **3 · Silent commands** cuando te lo indiquen. Guarda la salida del terminal de las tareas 1, 2 y 3: va en la entrega.

TAREA 1 · SELLA EL PAQUETE Y RÓMPELO (15 MIN)

Individual, en la carpeta del `.deb`. La clave que creas es de prueba: vale para clase, no para el público.

OK	HAZ	ESTÁ HECHO CUANDO…
☐	Crea una clave de prueba: `gpg --batch --passphrase '' --quick-generate-key "Tu Nombre <tu@example.com>" default default never`	`gpg` termina sin preguntar nada.
☐	Firma y comprueba (comandos de abajo).	Ves `Good signature from "Tu Nombre..."` y `NoteBox_1.0.0_amd64.deb: OK`.
☐	Rómpelo: trabaja sobre una copia, añádele una letra y un salto de línea, y verifica.	Ves `BAD signature`.
☐	Crea la clave del updater de Tauri y firma el paquete con ella.	Tienes `~/notebox.key`, `~/notebox.key.pub` y un `.sig` junto al paquete.

```
gpg --detach-sign --armor NoteBox_1.0.0_amd64.deb
gpg --verify NoteBox_1.0.0_amd64.deb.asc NoteBox_1.0.0_amd64.deb
sha256sum NoteBox_1.0.0_amd64.deb > SHA256SUMS && sha256sum -c SHA256SUMS

cp NoteBox_1.0.0_amd64.deb tampered.deb && echo x >> tampered.deb
gpg --verify NoteBox_1.0.0_amd64.deb.asc tampered.deb

npm run tauri signer generate -- -w ~/notebox.key
npm run tauri signer sign -- -f ~/notebox.key src-tauri/target/release/bundle/deb/
NoteBox_1.0.0_amd64.deb
```

La clave privada se queda en tu carpeta personal, fuera del repositorio. Lee el aviso que imprime la herramienta.

TAREA 2 · INSTALA SIN DECIR NADA (6 MIN)

Una instalación **desatendida** (o **silenciosa**) no abre ventanas ni hace preguntas.

```
sudo apt install -y ./NoteBox_1.0.0_amd64.deb
dpkg -s note-box | head -3        # Status: install ok installed
which notebox                     # /usr/bin/notebox
notebox &                         # la app se abre, con sus notas de bienvenida
```

Si algo falla:

- **Sin el `./`**, `apt` busca en los repositorios un paquete que se llame como el archivo.
- **El nombre del paquete es `note-box`**, no el del archivo.
- **Sin `-y`**, `apt` se para a preguntar.

TAREA 3 · NO DEJES RASTRO (8 MIN)

OK	HAZ	ESTÁ HECHO CUANDO…
☐	Añade dos notas en el NoteBox instalado y ciérralo.	Hay algo que perder.
☐	Lista, desinstala y mira (comandos de abajo).	`dpkg -s note-box` dice «is not installed» y los archivos del programa ya no están.
☐	Vuelve a instalar y abre la app.	Compruebas qué ha pasado con tus dos notas.

```
dpkg -L note-box
sudo apt remove -y note-box
dpkg -s note-box
ls /usr/lib/NoteBox /usr/bin/notebox /usr/share/applications/NoteBox.desktop
ls ~/.local/share/com.iessonferrer.notebox
```

TAREA 4 · ABRE EL PAQUETE (11 MIN)

`dpkg-deb` es la herramienta de empaquetado de Debian, no de Tauri. Un `.deb` tiene dos partes: `control` (la etiqueta) y `data` (los archivos).

OK	HAZ	ESTÁ HECHO CUANDO…
☐	Inspecciona: `dpkg-deb -I NoteBox_1.0.0_amd64.deb` (etiqueta) y `dpkg-deb -c NoteBox_1.0.0_amd64.deb` (contenido).	Encuentras tres cosas que empaquetaste el lunes.
☐	Reconstruye el paquete a mano cambiando el `Maintainer` (comandos de abajo).	`dpkg-deb -I notebox-external.deb` muestra tu nombre.

```
cd src-tauri/target/release/bundle/deb
mkdir -p pkg/DEBIAN && cp -r NoteBox_1.0.0_amd64/data/* pkg/
cp NoteBox_1.0.0_amd64/control/control pkg/DEBIAN/
# edita pkg/DEBIAN/control: Maintainer: Tu Nombre <tu@example.com>
dpkg-deb --build --root-owner-group pkg notebox-external.deb
dpkg-deb -I notebox-external.deb
```

TAREA 5 · EL PIPELINE Y LA LISTA DE PUBLICACIÓN

OK	HAZ	ESTÁ HECHO CUANDO…
☐	En parejas, coloca las ocho etiquetas en el tablero **1 · Release pipeline**.	Puedes contar la historia de izquierda a derecha, de la etiqueta `v1.0.1` a la app instalada.
☐	Tablero **4 · Release checklist**: arrastra una tarjeta a **Done** solo si puedes enseñar hoy su prueba, para una de las dos apps de la pareja.	Cada tarjeta en Done tiene una prueba.

TAREA 6 · EL STARTER DEL LUNES (4 MIN)

Cuando se publique en Classroom, descárgalo, descomprímelo en tu carpeta personal y ejecuta:

```
npm install && npm run tauri build -- --no-bundle
```

Si falla con un mensaje sobre `com.tauri.dev`, ya sabes qué hacer. Arréglalo, vuelve a lanzarlo y déjalo compilando.

PREGUNTAS DE LA SESIÓN

1 Has añadido una letra y un salto de línea a un archivo de 3 MB. ¿Cómo lo ha notado la firma?

2 La app está desinstalada pero `~/.local/share/com.iessonferrer.notebox` sigue ahí. ¿De quién es esa carpeta y qué contiene?

3 Escribe tres cosas que empaquetaste el lunes y que has encontrado con `dpkg-deb -c`.

4 ¿En qué estación del pipeline sería un desastre que te robaran el portátil? ¿Por qué?

5 En el workflow de CI, ¿por qué `ubuntu-22.04` y no la versión más nueva?

ENTREGA

En la tarea de Classroom: el enlace a la copia de Figma de la pareja con el tablero 4 relleno + un archivo de texto con la salida del terminal de las tareas 1, 2 y 3 (la verificación, la instalación, el listado de la desinstalación). Texto, no fotos.

PARA CASA

- Termina el tablero de la lista de publicación para tu propia app.
- Añade a tu manual de instalación de la UT7 una sección «Instalación y desinstalación desatendidas» con los dos comandos de tu paquete y la ruta de los datos del usuario.
- Ejercicio de transferencia (un párrafo): un instituto instala tu app en silencio en 60 PCs compartidos. En junio la desinstala en silencio, también en los 60. En Windows, las notas de todo el alumnado siguen en los discos. Explica por qué y qué añadirías al instalador para que una desinstalación silenciosa las borre sin romper las actualizaciones.