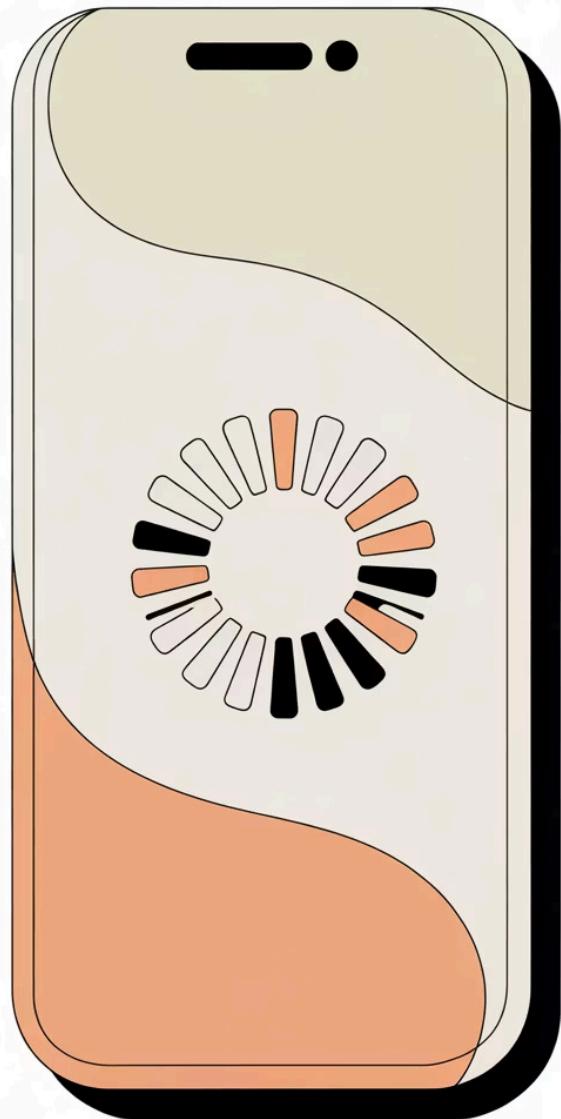


Arquitectura s Móviles

**Del "Cloud-First" (Nuestra App)
al "Local-First" (El Futuro)**

Un caso de estudio con nuestra App de Evaluación y Roble.





El "Spinner de la Muerte"

¿Alguna vez has abierto una app en un ascensor o en el metro y te quedas atascado mirando esto?

Este "spinner" no es un bug, es un síntoma de una arquitectura: "Cloud-First".

Arquitectura Actual: Nuestra App + Roble (Cloud-First)

La "Fuente de Verdad" (los datos maestros) está en la nube, en el servidor de Roble.

Cómo Funciona: La app *debe* conectarse a internet para funcionar. Para ver cursos, debe *preguntarle* a Roble.



La Alternativa: "Local-First" (Primero lo Local)

La "Fuente de Verdad" está en el propio dispositivo (en una base de datos local).

Cómo Funciona: La nube (Roble) se convierte en un simple respaldo o sincronizador. La app funciona *primero* con el dispositivo y *después* sincroniza cuando puede.



Local-First ≠ Offline-First

OFFLINE-FIRST

- Es "Cloud-First con un caché".
- La nube sigue mandando.
- Es un **Plan B** (un *fallback* de solo lectura).
- (Nuestra app intenta esto al mostrar datos viejos si Roble falla).

LOCAL-FIRST

- La base de datos local manda.
- La nube es un respaldo.
- Es el **Plan A**.
- (Funciona para leer Y escribir offline).

Ventaja Local-First: Rendimiento Instantáneo



Las acciones son instantáneas (0 latencia de red) porque ocurren localmente. No hay *spinners* para el usuario.

Hoy (con Roble)

Click en 'Crear Curso' → Spinner de Carga → Esperar respuesta de Roble → Listo.

Local-First

Click en 'Crear Curso' → Listo (Instantáneo) → (Sincronización en segundo plano).

Ventaja Local-First: 100% Funcional Offline



La app funciona perfectamente (*leer y escribir*) sin conexión.

Hoy (con Roble)

Un estudiante sin datos intenta guardar una evaluación. La app da 'Error de Red' y el estudiante pierde su trabajo.

Local-First

El estudiante (offline) guarda la evaluación. Se almacena localmente y se sincroniza con Roble más tarde cuando tenga WiFi.



Otras Ventajas: Privacidad y Colaboración



Privacidad 🔒

"Tus datos son tuyos". Viven en tu dispositivo, no en el servidor de un tercero. Facilita el cifrado.



Colaboración 🤝

Es la base de apps "multijugador" (como Figma) donde varios editan a la vez, incluso sin un servidor central.



La Desventaja: ¿Por qué no lo hacen todos? 🤯

Respuesta (Grande): Es MUCHO MÁS DIFÍCIL de programar.

El Problema N°1: La Sincronización y los Conflictos de Datos.

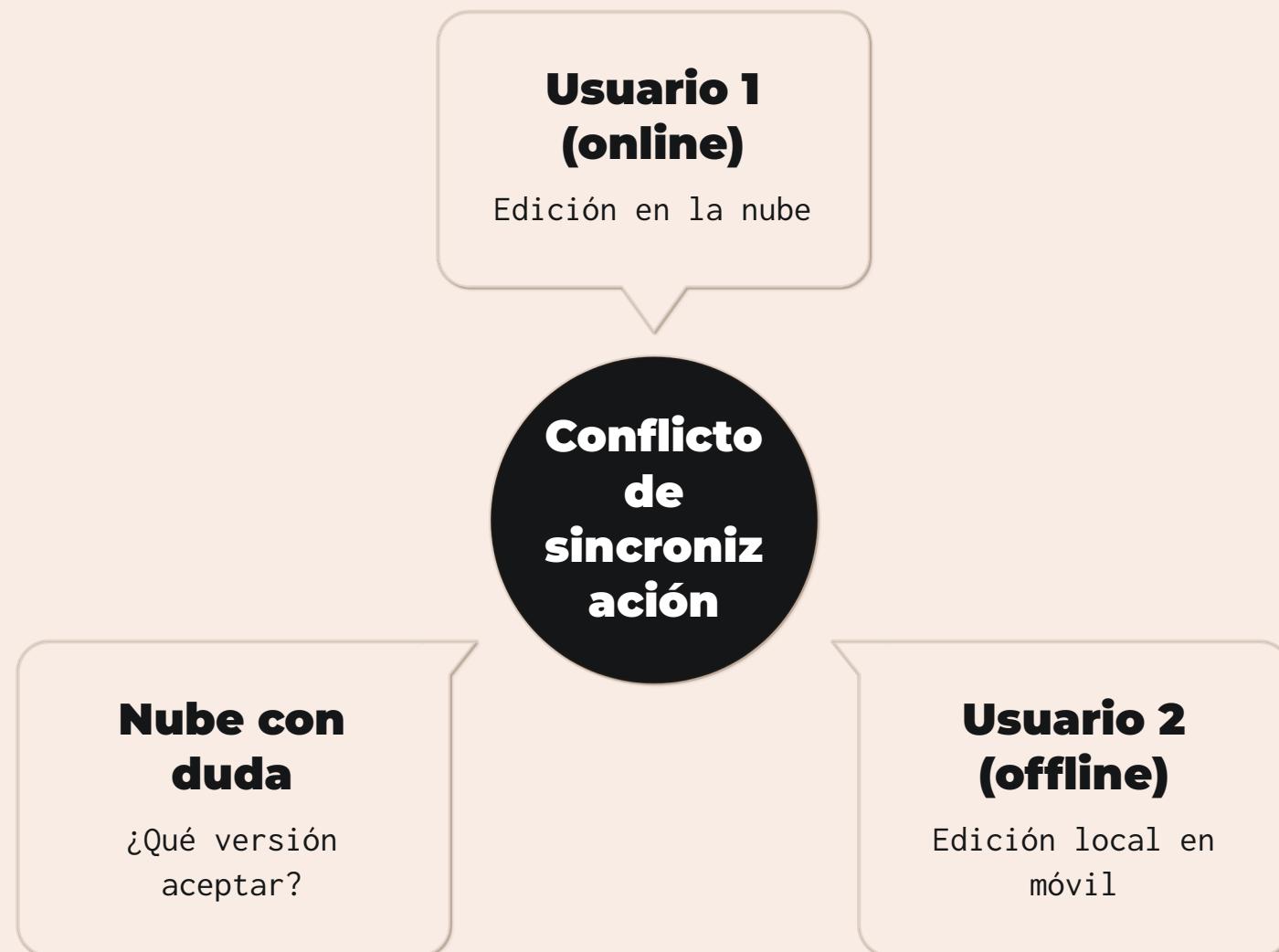
El Reto: Conflictos de Sincronización

Pregunta: ¿Qué pasa si dos usuarios editan lo mismo al tiempo (uno online y otro offline)?

Ejemplo:

- Profesor 1 (online en Roble) cambia el nombre de un curso.
- Profesor 2 (offline en el móvil) cambia el *mismo* nombre de curso.

El Conflicto: Cuando el móvil se conecta, ¿cuál nombre gana? En el modelo de Roble, esto no pasa (el último que guarda, gana). En Local-First, es un gran problema.



Otros Retos del Modelo Local-First

Otros Retos: Almacenamiento y Autenticación



Almacenamiento

¿Qué pasa con 10 años de cursos de un profesor? La app pesaría Gigabytes en su teléfono.



Autenticación

¿Cómo haces "login" si no tienes internet para verificar tu contraseña con Roble?

Conclusión: El Veredicto

Cloud-First (Roble)

- Más fácil de programar.
- Lógica centralizada.
- Ideal para prototipos.
- Peor Experiencia de Usuario (lento, depende de la red).

Local-First

- La mejor Experiencia de Usuario (UX).
- Rápido y 100% confiable.
- Mucho más complejo de sincronizar.

La Decisión: Es un trade-off (intercambio) entre: Facilidad de Desarrollo vs. Experiencia de Usuario.