

1. Falta de reglas estrictas de validación en la base de datos

Si en las tablas de PostgreSQL no se definen adecuadamente *constraints* (como claves foráneas, *unique*, *not null*), pueden **insertarse datos incompletos, duplicados o inconsistentes**, provocando registros incoherentes o errores en consultas posteriores.

2. Transacciones concurrentes sin aislamiento adecuado

PostgreSQL soporta niveles de aislamiento, pero si la aplicación no gestiona transacciones correctamente puede haber **anomalías de lectura o escritura entre registros relacionados**, lo que rompe la consistencia global entre tablas vinculadas por relaciones lógicas.

3. Ausencia de manejo de errores y validaciones en la lógica del negocio

Si el backend Java no valida antes de insertar o actualizar, pueden **quedarse estados incongruentes** (por ejemplo, registrar un progreso de entrenamiento sin que exista el usuario asociado, o con valores de ejercicio fuera de lógica esperada).

4. Conflictos entre datos de usuarios y rutinas de sincronización

Aplicaciones móviles tienden a sincronizar datos locales con el servidor. Sin controles de versión o *timestamps* robustos, se puede **sobrescribir datos más recientes con versiones antiguas**, afectando la consistencia de históricos.

5. Dependencia de scripts externos sin controles de integridad

Si se ejecutan migraciones o scripts sin revisiones o *rollbacks*, cambios estructurales o poblaciones masivas pueden **corromper estructuras o relaciones**, dejando la base de datos en estados inconsistentes.

6. Ausencia de pruebas automatizadas de integridad de datos

Sin pruebas que aseguren que cada operación mantiene las reglas de negocio y las restricciones de la base de datos, existe riesgo de introducir errores no detectados que afecten la consistencia a largo plazo.