

Shortcut, Mirroring, Pipeline, Dataflow o

Notebook

¿Cuándo usar cada uno?

Kawaii BI

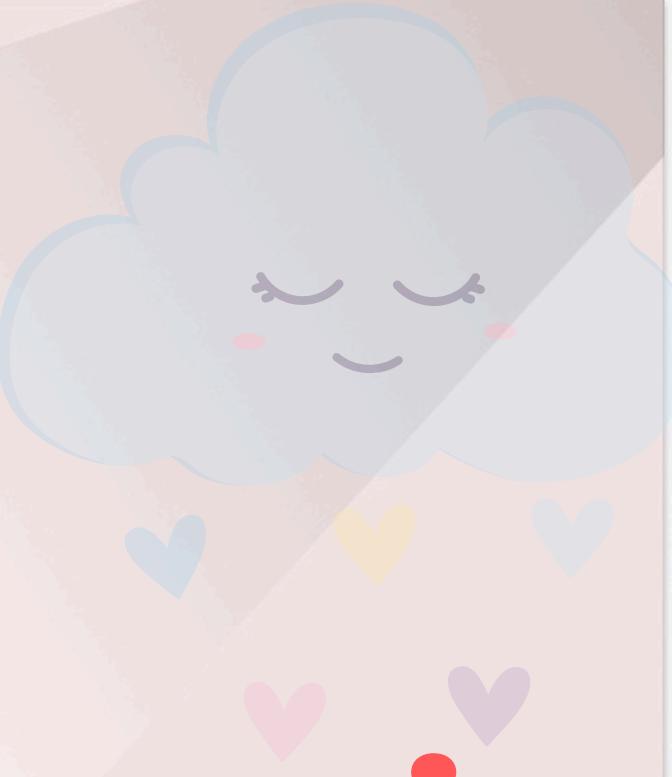
¿Qué usar y cuándo?

Necesitas...	Lo mejor	¿Mueve/copía datos?	Latencia	Escala/ETL	¿SQL directo / Direct Lake?	Casos típicos
Evitar duplicar y "ver" datos que ya existen en otro lago (ADLS/S3/otro Lakehouse)	Shortcut (File/Table)	No (cero-copia)	N/A (depende del origen)	No transforma por sí mismo	Table shortcut: Sí (SQL); File: no (lee por ruta)	Reusar datos sin copiarlos; pruebas; compartir datasets entre equipos
Analítica casi en tiempo real sin golpear la BD operativa + Direct Lake	Mirroring	Sí (replica a OneLake Delta)	Baja (CDC/sync)	No transforma; ingestión gestionada	Sí (tablas nativas, Direct Lake)	Offload del OLTP, alta concurrencia BI, simplificar ingestión
Orquestar cargas por lotes desde muchas fuentes + gobernanza + horarios/dependencias	Pipeline (Data Factory)	Sí (normalmente)	Batch (programado)	ETL/ELT vía actividades; escalable	Sí, tras aterrizar a Delta	Ingestiones periódicas; copias incrementales; control de fallos y reintentos
ETL self-service con UI (Power Query) para analistas	Dataflow Gen2 (FlowGen2)	Sí (escribe resultados)	Batch	Transformaciones ricas (M), fáciles	Sí, al escribir a Tables (Delta)	Curado "silver/gold" por analistas; limpieza, uniones, reglas
Transformaciones pesadas / big data / ML en paralelo	Notebook (PySpark)	Opcional (tú decides si escribes)	Batch o interactivo	Máxima escala (Spark)	Sí, si escribes a Delta	Curado avanzado, joins masivos, time series, ML, compresión/OPTIMIZE



¿Cuando estoy copiando mis
datos en Fabric y cuando
no?

Kawaii BI



Objetivo: elegir la herramienta correcta y no morir en el intento



¿Quieres evitar copias? [Shortcut \(File/Table\)](#)

Como un marcapáginas a datos que ya existen. Ves y usas sin copiar. La “vieja del visillo de Fabric”

¿Necesitas muchos lectores y casi en tiempo real? [Mirroring](#)

Como una fotocopia sincronizada de tu base de datos. Lista para BI rápido y muchos usuarios, se evitan sobrecargas en el origen.

¿Carga por lotes con control total? [Pipeline \(Data Factory\)](#)

Como un planificador de cargas por lotes. Horarios, dependencias, control. Voy trayendo mis datos sin prisa pero sin pausa.

¿ETL para negocio? [Dataflow Gen2](#)

ETL sin código para analistas. Limpia, combina y deja bonito ¡Mi favorita! Power Query 4eva!

¿Big data/ML o cálculos pesados? [Notebook \(PySpark\)](#)

Turbo para datos enormes o modelos. Procesa en paralelo muy rápido ¡Datos a lo bestia!

Reglitas exprés 🚀:

- No duplicar → **Shortcut** ✂️
- Casi real-time/Direct Lake → **Mirroring** 🏡
- Lotes → **Pipeline** 🌇
- Self-service → **Dataflow** 😍
- Big data/ML → **Notebook** 🐾

Kawaii BI



Nos vemos en Kawaii Power BI



<https://www.youtube.com/@KawaiiPowerBI>

Kawaii BI