



**“Canalizaciones de
implementación” o
“Deployment Pipelines” de**

Fabric



“Flow sin Fricción”



Kawaii BI





¿Esto te suena?



"Subo el PBIX y cruzo los dedos 🤞"



"Cambios manuales de orígenes de datos en cada entorno, esperando no cometer errores 🤪"



El versionado de archivos se basa en nombres como `Reporte_Final_v3_AHORASÍ.pbix`

"Producción se rompió... otra vez 💥"



Inconsistencias y errores humanos que generan desconfianza en los datos.

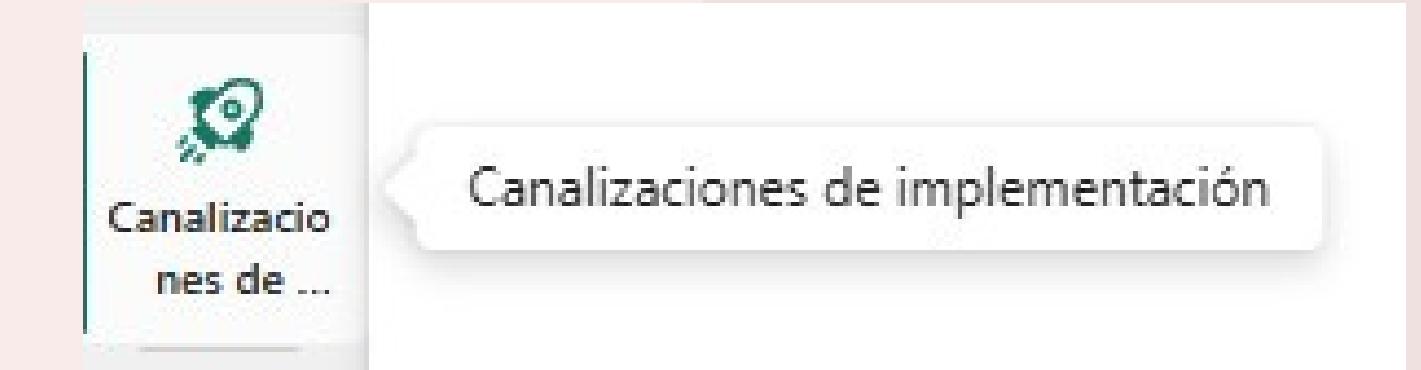


¿Qué es una Deployment pipeline (en fácil)?



Un Deployment Pipeline es una funcionalidad nativa de Fabric que permite a los equipos de BI construir un proceso de lanzamiento eficiente y reutilizable. Actúa como una “cinta transportadora” 🚂🍩 para mover tu contenido por etapas:

Development → Test → Production ✓



¿Por qué es importante? 🤔💡

Porque te da:

- Control 🛡️
- Calidad (pruebas antes de Prod) 💧
- Orden (entornos claros) 💯
- Velocidad (deploy en clicks) ⚡

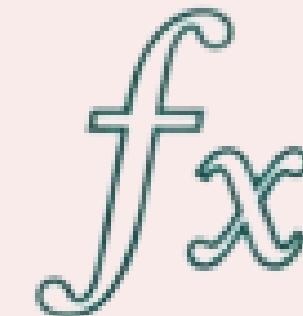
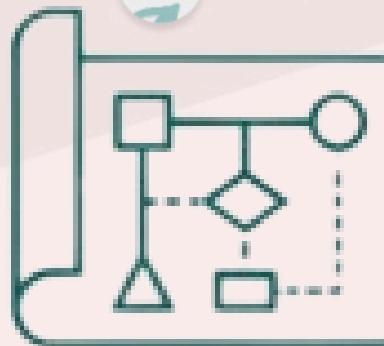
¿Qué es una Deployment pipeline (en fácil)?

Cada etapa – workspace distinto:

- Dev: trasteo libre 🐾 Dar permisos “full” en Dev
- Test: validación 💊 En Test solo a quien valida
- Prod: permisos estrictos 🔒 En Prod... poquita gente
😈🔒 (te evitas el “me lo cargué sin querer” 💀)

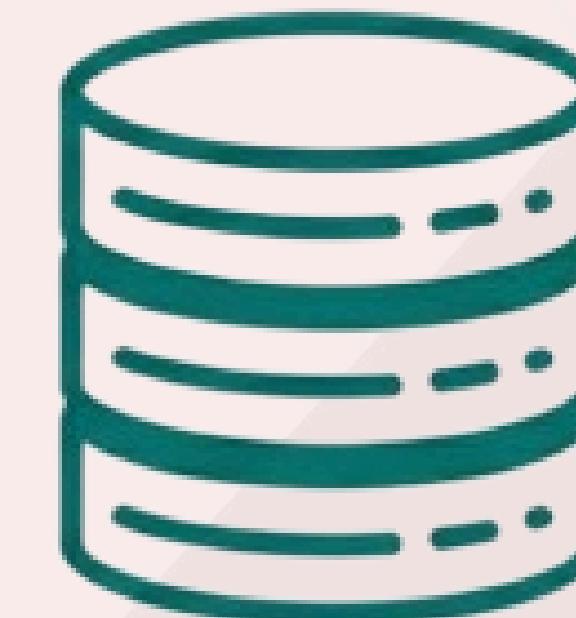
¡OJO! que se despliega y que no se despliega...

QUÉ SE DESPLIEGA (Metadata)



El proceso de despliegue copia la *definición* de los artefactos (esquemas, informes, relaciones), pero no los datos contenidos en ellos.

QUÉ NO SE DESPLIEGA (Datos)



Implicación Clave: Después de desplegar un modelo semántico o un dataflow, siempre es necesario ejecutar una actualización (refresh) para cargar los datos del nuevo entorno.

La magia: Deployment Rules



Aquí es donde el pipeline se pone pro (y menos sufrimiento), la verdadera potencia reside en configurar reglas para que los parámetros y orígenes de datos se actualicen automáticamente en cada etapa. Esto elimina la necesidad de cambios manuales post-despliegue:

Puedes decirle:

- En Dev usa servidor DEV 
- En Test usa servidor TEST 
- En Prod usa servidor PROD 

Con reglas como:

- Parameter rules  (ej: ServerName, Database, Environment, LakehouseName)
- Data source rules  (cambiar el origen por etapa, sin tocar el modelo a mano)

¿Cómo creo este sarao?



Paso a paso la propia interfaz de Fabric te guiará

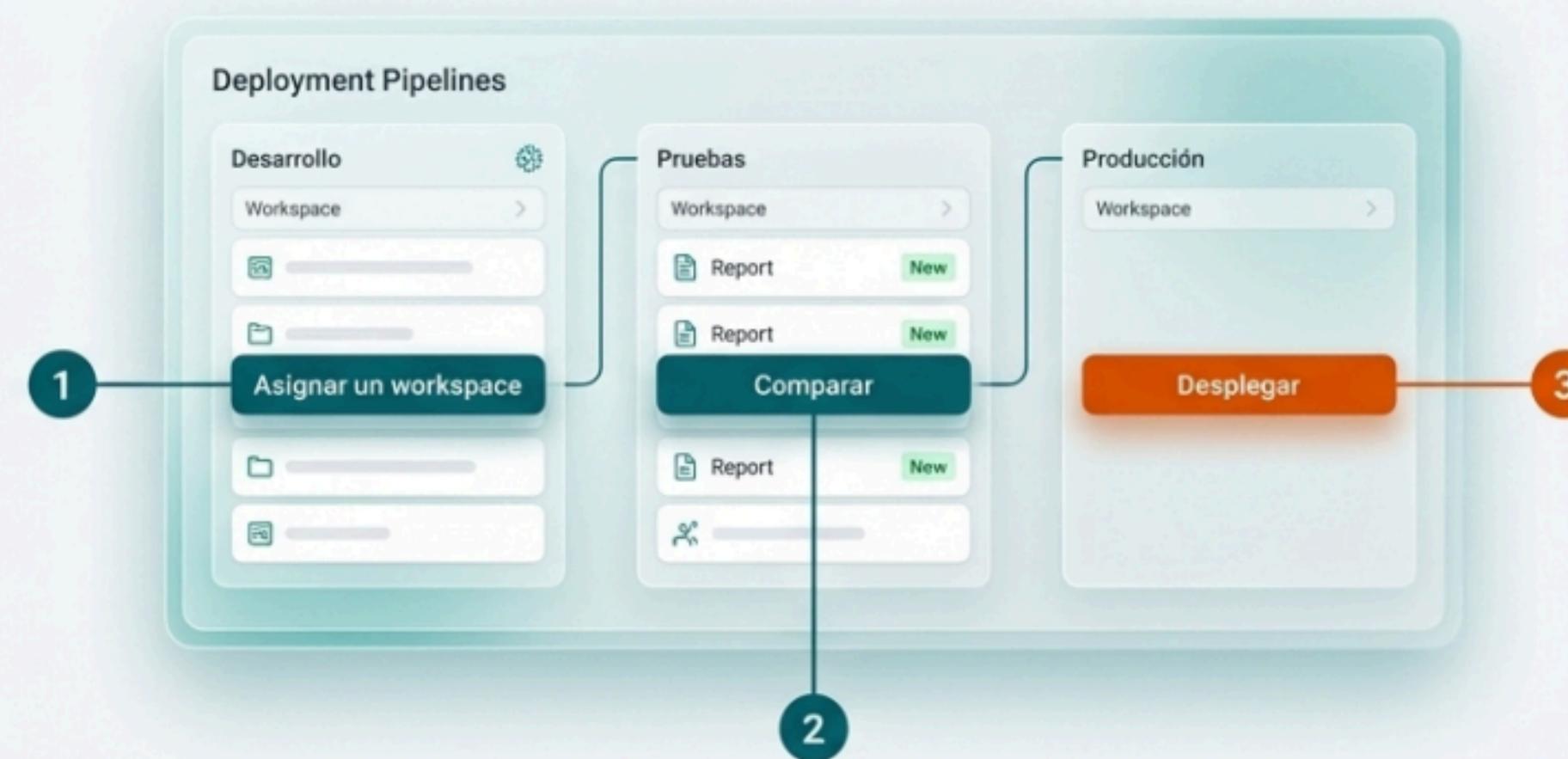
1. Ve a Power BI Service
2. En el menú, entra en Deployment pipelines
3. Clic en Create pipeline +
4. Ponle nombre (ej: Proyecto_X - ALM)
5. Elige el nº de etapas (2 a 10) (tip: lo normal es 3: Dev/Test/Prod)
6. Asigna un workspace a cada etapa (Dev, Test, Prod)
7. Configura Deployment rules (parámetros/orígenes)
8. ¡A promover! Deploy to Test → Deploy to Prod



¿Cómo creo este sarao?



1. **Asignar:** Asigna un workspace a cada etapa (de conterero (Desarrollo, Pruebas, Producción).



2. **Comparar:** Compara el contenido entre etapas para ver las diferencias de un vistazo.

3. **Desplegar:** Promueve todo el contenido o selecciona ítems específicos con un solo clic.

¿Y puedo hacer control de versiones con Git?

✓ Por supuesto! La integración nativa de Fabric con repositorios Git permite aplicar prácticas de desarrollo de software profesionales a tu contenido de BI 🚀

- Control de versiones
- Colaboración entre varios desarrolladores
- Trazabilidad y revisiones

iPero esta parte ya la dejó mejor para un video en mi [canal de Youtube Kawaii Power BI!](#) 😊

Deployment Pipelines en Fabric y Power BI – llevar tu contenido de forma ordenada:

DEV 🧰 → TEST 🔎 → PROD 🔒

¿Por qué mola?

- Menos sustos en Prod 🛡️
- Cambios controlados y repetibles ✅
- Validación antes de publicar 🧰

Requisitos clave:

- Fabric capacity ⚡
- 1 workspace por etapa 🏠🏠🏡

Lo más importante:

- Usa Deployment rules (parámetros/orígenes) 📋🔌
- El pipeline no mueve datos → luego toca refresh 🔄
- Combo pro: GitHub + PBIP para versionado + pipeline para promover 🐙📦



Menos caos, más colorinchis 🌈✨



Nos vemos en Kawaii Power BI



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



Kawaii BI