Manual de operación de los ETLs POT

Proyecto: Contrato de prestación de servicios n° 202500136 de 2025 celebrado entre la corporación Ruta N - Medellín y Guane Enterprises S.A.S
Fecha de elaboración: 2026-07-28

¿Cómo usar este manual? 3
1. Introducción 4
Qué es este manual 4
Audiencia 4
Qué cubre 4
Qué no cubre 4
2. Resumen ejecutivo 5
¿Qué hace este sistema? 5
Roles 6
3. Descripción general del proceso 6
Tres momentos de la operación 7
4. Arquitectura operativa 7
Ambientes 7
5. Roles y responsabilidades 7
6. Requisitos previos 8
Documentos necesarios 8
Identidad del operador 9
7. Procedimiento general de operación 9
Paso 1 — Preparar el ambiente (una vez) 9
Paso 2 — Recibir y organizar los insumos del periodo 9
Paso 3 — Cargar a Bronce 9
Paso 4 — Validar en Plata 9
Paso 5 — Publicar en Oro y verificar 9
8. Procedimientos operativos detallados 9
8.1 Recepción de archivos 9
8.2 Cargas a Bronce 10
Flujo de Carga de Variables 11
Flujo de Carga de Capas 11
8.3 Validaciones (Plata) 12
Validación de Variables tabulares 12
Validación de Capas 12
8.4 Carga a Oro (variables) 13

8.5 Carga a Oro (capas) 14
8.6 Carga de histórico de indicadores 15
9. Monitoreo de la ejecución 16
Estado de pipelines 16
Logs en datalake 16
Verificación de datos publicados 16
Duración típica 17
Mapa de artefactos 17
10. Manejo de errores por categoría 17
Errores de archivos (Excel / landing zone) 17
Errores de validación (reglas de negocio) 17
Errores de infraestructura (SHIR / conexión) 17
Errores de catálogo 18
Errores de integración (usuario / locks / aprobación) 18
11. Procedimientos de recuperación 18
Árbol rápido: síntoma → acción → reejecutar 18
Re-ejecución parcial 19
12. Buenas prácticas de operación 20
13. Glosario 21
14. Anexos 25
14.1 Tabla maestra Synapse ↔ Pipeline 25
14.2 Catálogo completo de errores 25
Variables (ERR-CV) 25
Capas (ERR-CC) 28
14.3 Checklists por escenario 30
Setup de ambiente (una vez) 30
Setup de usuario (una vez por operador) 30
Carga de periodo — solo variables 30
Carga de periodo — solo capas 31
14.4 Escalamiento 31
14.5 Matriz de decisiones 31
14.6 Historial de cambios del documento 32
14.7 Convención de ambientes y parámetros técnicos 32
14.8 Configuración de dependencias y capas (config.json) 32
14.9 Setup técnico de ambiente (una vez por ambiente) 33
14.10 Estructura técnica del Data Lake y bases de datos 33

¿Cómo usar este manual?

Este manual describe la **carga de variables, capas e indicadores (histórico)** del POT Medellín. Está pensado para operadores que ejecutan, supervisan y verifican cargas periódicas.

Si es tu primera vez: lee la Sección 2 (resumen), la Sección 5 (tu rol) y la Sección 7 (vista general en cinco pasos). Luego revisa los requisitos previos antes de ejecutar cualquier carga.

Si vas a ejecutar algo específico: ve directo a la Sección 8. Cada procedimiento incluye una ficha al inicio y criterios de cierre al final.

Si tuviste un error: consulta la Sección 10 (por categoría), la Sección 11 (qué reejecutar) y el catálogo completo de errores en anexos.

Si necesitas un término: usa el Glosario (Sección 13). Todo término técnico usado en este manual está definido allí.

1. Introducción

Qué es este manual

Este documento describe el procedimiento que deben seguir los **operadores** para:

1. **Preparar los insumos** del periodo: archivos de avances (Excel) por dependencia y de las capas geográficas a cargar.
2. **Ejecutar y supervisar** los procesos de carga de variables y de capas, incluyendo la comprobación previa de conexión cuando aplica.
3. **Monitorear** el avance de la carga a través de sus distintas etapas (Bronce, Plata, Oro) y revisar los registros (logs) generados.
4. **Verificar resultados**: confirmar que la información quedó disponible en el repositorio oficial.
5. **Identificar errores** y escalar al equipo técnico con evidencia (ver Sección 11 y Anexo 14.4).

Audiencia

Personas que **operan** los ETLs del POT con rol **Editor**, **Operador** o **Visualizador** (ver Sección 5).

Qué cubre

- Carga recurrente de **variables tabulares** (Excel de dependencias) y **capas geográficas**.
- Setup único por ambiente: catálogo POT, histórico de indicadores, registro de operadores.
- Monitoreo, verificación y recuperación ante fallos.

Qué no cubre

- Configuración de la infraestructura tecnológica del sistema (servidores de proceso, conexiones, despliegue de la plataforma).
- Modificación del diseño interno de los procesos de carga.
- Mantenimiento del catálogo más allá de la ejecución de los procedimientos de setup descritos en este manual.

Estos temas corresponden al equipo técnico o soporte DAP.

2. Resumen ejecutivo

El presente manual describe la operación de los procesos de carga de información (ETL) implementados para soportar la gestión del Plan de Ordenamiento Territorial (POT) de Medellín, representada mediante variables, indicadores y capas geográficas. Su propósito es orientar a los operadores responsables de la ejecución, monitoreo y control de estos procesos, para garantizar una carga de información confiable, trazable y consistente.

Los procesos automatizan la incorporación de la información reportada por las dependencias del Distrito —tanto archivos de avances (variables) como capas geográficas—, validando previamente su calidad y consistencia antes de integrarla al repositorio oficial de publicación (Oro). **Estos ETL no calculan indicadores**: cargan y publican variables (`TBL_VALORVARIABLE`) y capas resultado (`TBL_CAPARESULTADO`) para que **otros procesos** del POT evalúen fórmulas y calculen indicadores. La arquitectura se organiza en tres capas sucesivas (Bronce, Plata y Oro) que conservan los datos originales, aplican controles de calidad y dejan la información publicada en Oro.

Este manual describe las actividades que debe realizar el operador durante el ciclo de ejecución: preparación de los archivos de entrada, ejecución de los procesos, verificación de resultados, interpretación de mensajes de error, acciones de recuperación ante fallos y criterios para la reejecución cuando sea necesario.

El documento no explica el diseño interno de los procesos ni su implementación técnica; es una guía práctica para la operación diaria del sistema, presentada desde la perspectiva del operador: qué debe hacer en cada etapa, cuáles son los resultados esperados y cómo actuar frente a las situaciones más comunes.

La correcta aplicación de estos procedimientos contribuye a garantizar la calidad de la información utilizada para el seguimiento del POT, la trazabilidad de los datos procesados y la confiabilidad de las **variables publicadas** que alimentan procesos posteriores de cálculo de indicadores.

¿Qué hace este sistema?

1. **Recepción de la información:** se identifican y reciben los archivos de avances (variables) y las capas geográficas que cada dependencia tiene configuradas para reportar en el periodo.
2. **Validación:** se verifica la estructura, integridad y consistencia de los datos reportados, tanto de archivos como de capas.
3. **Transformación:** la información se normaliza de acuerdo con el modelo de datos del sistema; las capas geográficas, además, se procesan para obtener **variables derivadas** (escalares geo y capas resultado).
4. **Carga:** los datos validados —variables reportadas y variables resultantes del procesamiento de capas— se publican en Oro (`TBL_VALORVARIABLE`, `TBL_CAPARESULTADO`) como insumo para procesos externos de cálculo de indicadores.
5. **Registro y auditoría:** el sistema genera bitácoras de ejecución, registra los errores detectados y conserva la trazabilidad completa del procesamiento realizado.

Roles

El manual distingue tres roles operativos — **Editor**, **Operador** y **Visualizador** — definidos en la Sección 5.

3. Descripción general del proceso

El flujo completo de **carga de variables y capas** sigue esta cadena:

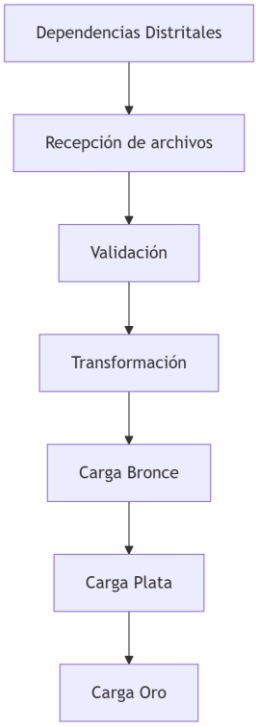


Figura 1: Flujo del proceso de variables y capas

Recepción: las dependencias entregan los archivos de avances y las capas geográficas que les corresponden en el periodo.

Validación: se revisa que la información cumpla las reglas de calidad establecidas antes de continuar.

Transformación: los datos se normalizan según el modelo del sistema; las capas geográficas se procesan para obtener variables.

Bronce → Plata → Oro: cada zona del Data Lake cumple un rol específico (ver Sección 4), hasta dejar variables y capas resultado publicadas en Oro para consumo por procesos posteriores (incluido el cálculo de indicadores, fuera de estos ETL).

Tres momentos de la operación

- **Configuración inicial del entorno** (una vez por ambiente): catálogo, histórico y configuración de dependencias y capas.
- **Registro del operador** (una vez por persona que va a ejecutar cargas).
- **Operación periódica** (cada periodo): carga de variables y/o capas, validación y consulta.

Variables y capas **no comparten un orden obligatorio** entre sí: una vez completada la configuración inicial, puedes cargar solo variables, solo capas, o ambas.

4. Arquitectura operativa

El sistema organiza la información en tres capas sucesivas, cada una con un propósito específico para el operador:

Bronce — Zona de ingesta. Aquí queda una copia fiel de lo que las dependencias reportaron (archivos y capas), tal como se recibió desde la landing zone (Servidor de origen), sin ninguna transformación. Sirve como respaldo y punto de partida para la validación.

Plata — Zona de validación y procesamiento. Aquí queda la información que superó las reglas de calidad, ya normalizada y lista para publicarse a la zona oro.

Oro — Zona de publicación. Aquí queda la información definitiva en PostgreSQL (`vars.*`): valores de variables (`TBL_VALORVARIABLE`), capas resultado (`TBL_CAPARESULTADO`) y demás hechos consultables. **No se calculan indicadores en esta etapa**; las variables publicadas alimentan procesos downstream (cálculo de indicadores, consultas de otras áreas).

Ambientes

El sistema opera en tres ambientes independientes — desarrollo, pruebas y producción — para permitir pruebas sin afectar la información oficial. El operador debe confirmar siempre en cuál ambiente está trabajando antes de ejecutar una carga (ver Sección 6). El detalle técnico de nombres de recursos por ambiente está en el Anexo 14.10.

5. Roles y responsabilidades

Tres roles operativos: **Editor**, **Operador** y **Visualizador**.

Editor — Deposita los insumos del periodo (archivos de avances y configuración de capas) en el servidor de origen del DAP. Verifica que la configuración esté al día y que existan las carpetas del periodo para cada dependencia.

Operador — Dispara los procesos de carga de variables y de capas, y la comprobación previa de conexión cuando aplica. Supervisa la ejecución: revisa el avance por etapas (Bronce, Plata, Oro), consulta los registros (logs) y reintenta la etapa que haya fallado. **No** deposita archivos en el servidor de origen del DAP.

Visualizador — No ejecuta procesos de carga. Consulta la información ya publicada mediante los reportes o consultas analíticas disponibles. **No** accede a las zonas Bronce ni Plata.

Fuera de estos tres roles (equipo técnico): configuración inicial del entorno, registro de operadores, despliegue de los procesos y mantenimiento de la infraestructura tecnológica.

6. Requisitos previos

Antes de ejecutar cualquier carga, el operador debe verificar:

- **Servidor:** tener acceso al recurso compartido donde las dependencias depositan sus archivos, y a la plataforma donde se ejecutan los procesos.
- **Permisos:** contar con una cuenta activa y los permisos correspondientes a su rol (ver Sección 5).
- **Carpetas:** que existan las carpetas del periodo para cada dependencia que va a reportar en ese periodo, aunque estén vacías.
- **Archivos:** que los archivos de avances (Excel) sigan la plantilla oficial y que la configuración de capas geográficas esté completa, al igual que el diccionario de capas geográficas esté actualizado.
- **Periodo:** confirmar que el periodo a cargar (mes y año) es el correcto antes de iniciar cualquier proceso.
- **Catálogos:** si es la primera vez, consultar al rol editor si la carga inicial del sistema con el catálogo indicadores (instrumento de caracterización del POT) y el archivos de histórico de indicadores han sido cargados. Esto se hace una sola vez por ambiente (ver Sección 8.6 y Anexo 14.9).

Contacto soporte: *grupo de soporte asignado ETL DAP*.

Documentos necesarios

- **Instrumento de recolección de informe de avances (Excel):** reporte de avances por dependencia y periodo.
- **Instrumento de caracterización del POT (Excel):** catálogo maestro de variables e indicadores, cargado una sola vez por ambiente.

Identidad del operador

Todo operador que vaya a ejecutar cargas debe estar registrado en el sistema (ver Sección 8.6). El registro se hace una sola vez por persona.

El detalle técnico de ambientes, parámetros y configuración de dependencias/capas está en los Anexos 14.7 a 14.9.

7. Procedimiento general de operación

Vista general de la **carga de variables, capas e indicadores (histórico)** en cinco pasos. El detalle operativo de cada paso está en la Sección 8.

Paso 1 — Preparar el ambiente (una vez)

Confirmar que el entorno esté configurado: catálogo cargado, histórico cargado, configuración de dependencias y capas lista, y cada operador registrado. Ver Sección 6 y Sección 8.6.

Paso 2 — Recibir y organizar los insumos del periodo

El Editor deposita los archivos de avances del periodo y verifica que la configuración de capas esté al día. Opcionalmente se puede comprobar la conexión antes de continuar. Ver Sección 8.1.

Paso 3 — Cargar a Bronce

Se dispara la carga de variables y/o de capas geográficas para el periodo. La información llega, sin transformar, a la zona Bronce. Ver Sección 8.2.

Paso 4 — Validar en Plata

La información pasa por las reglas de calidad y, en el caso de las capas, por su procesamiento para obtener variables derivadas. Si algo se rechaza, queda registrado en los logs. Ver Sección 8.3.

Paso 5 — Publicar en Oro y verificar

Los datos validados quedan disponibles en el repositorio oficial. El operador confirma que la carga fue exitosa consultando los datos publicados; si algo falló, consulta la Sección 11. Ver Secciones 8.4, 8.5 y 9.

8. Procedimientos operativos detallados

Esta sección detalla la **carga de variables, capas e indicadores (histórico)**.

8.1 Recepción de archivos

Ficha del procedimiento

- **Objetivo:** dejar listos en landing zone los insumos del periodo antes de disparar pipelines de carga.
- **Responsable:** Editor.

- **Entradas:** Excel de avances por dependencia; config.json actualizado; carpetas de periodo creadas.
- **Salidas:** archivos en {share}/dependencias/{DEP}/{PER}/variables/; configuración de capas en config.json.
- **Tiempo estimado:** 30 min – 2 h según número de dependencias.
- **Frecuencia:** cada periodo (YYYY_MM).

Pasos:

1. Confirmar que config.json lista todas las dependencias que reportarán en el periodo.
2. Crear carpeta {share}/dependencias/{DEP}/{YYYY_MM}/variables/ para **cada** dependencia en archivos_variables, aunque aún no haya Excel.
3. Depositar archivos .xlsx en las carpetas correspondientes.
4. Verificar diseño del instrumento (filas 3/5/6/7 según plantilla POT).
5. Si hay capas nuevas, agregar entrada en capas_estandarizadas con query validado en PostgreSQL origen en archivo config.json. Ver sección 14.8.
6. *(Opcional)* Disparar VALIDAR-CONEXION con p_env correcto para comprobar SHIR, config y conexión a BD.

Criterios de alerta:

- Dependencia en config.json sin carpeta de periodo → fallo en Copy (CopyVariables).
- Carpeta vacía → la corrida puede continuar con 0 archivos (revisar logs-carga/{PER}/).

Resultado esperado

Carpetas de periodo existentes para todas las dependencias en config.json
Excel depositados (donde aplique)
config.json coherente con estructura física del share
(Opcional) Preflight VALIDAR-CONEXION con checks OK

8.2 Cargas a Bronce

Ficha del procedimiento

- **Objetivo:** copiar insumos del servidor a la zona Bronce del datalake (variables y/o capas).
- **Responsable:** Operador (disparo); Editor (insumos previos).
- **Entradas:** insumos en landing zone; config.json; setup de ambiente completado.
- **Salidas:** archivos en Bronce bajo servidor/pot/dependencias/{DEP}/{PER}/.
- **Tiempo estimado:** 15 – 45 min por flujo.

- **Frecuencia:** cada periodo.

Flujo de Carga de Variables

Pipeline: ETL-CARGA-VARIABLES, sub-pipeline 01-CARGA-ARCHIVOS-BRONCE
Parámetros pipeline padre: p_periodo (YYYY_MM; vacío = mes anterior), **p_env** (dev | test | prod), **p_correo_usuario** (su correo registrado).

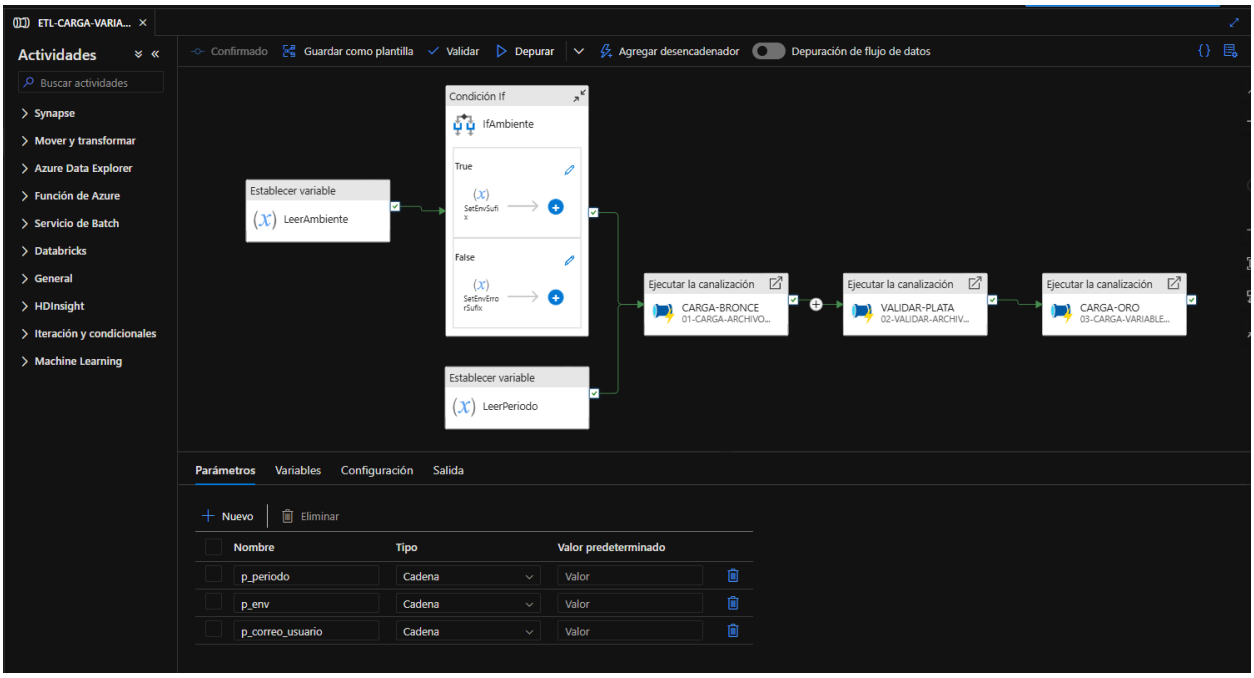


Figura 1: ETL Carga de Variables en Synapse

Actividades ejecutadas por el sub-pipeline 01-*:

1. CopiarConfiguracionArchivosServidor — Copy SHIR config.json → {bronce}/config/
2. ProcesarConfiguracion — Data flow VariablesDataFlow → CSV manifiesto en Bronce
3. ListarArchivos — Lookup del manifiesto
4. CopiarArchivosServidorBronce — ForEach + Copy de .xlsx → pot/dependencias/{DEP}/{PER}/variables/

Path destino de archivos: servidor/pot/dependencias/{DEP}/{PER}/variables/*.xlsx

Logs de actividades de copia quedan en: {bronce*}/servidor/pot/logs-carga/{PER}/ — alerta si FilesWritten=0.

Flujo de Carga de Capas

Pipeline: ETL-CARGA-CAPAS, sub-pipeline 01-CARGA-CAPAS-BRONCE
Parámetros pipeline padre: p_periodo (YYYY_MM; vacío = mes anterior), **p_env** (dev | test | prod), **p_correo_usuario** (su correo registrado), **p_approval_status** (approved)

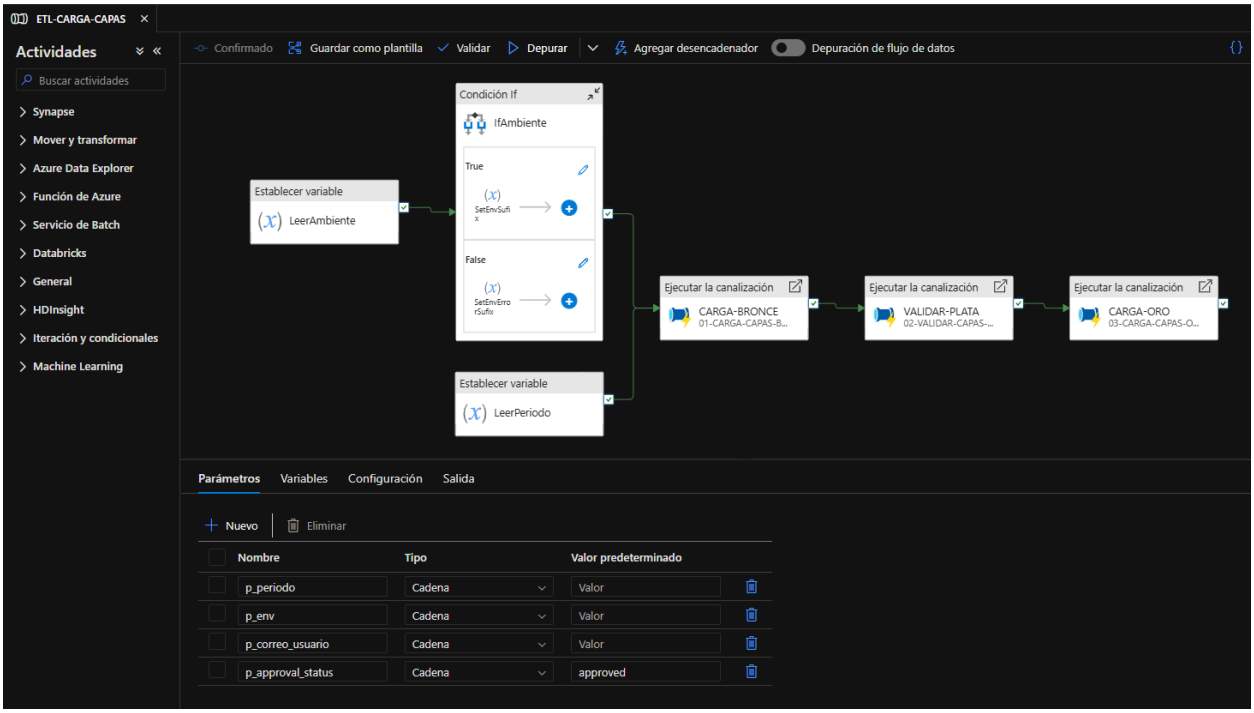


Figura 2: ETL Carga de Capas en Synapse

Actividades ejecutadas por el sub-pipeline 01-*:

1. CopiarConfiguracionCapasServidor — Copy config capas SHIR → Bronce
2. ProcesarConfiguracion — Data flow CapasDataflow → CSV capas/queries
3. ListarCapas — Lookup del manifiesto
4. ForEach1 / CopyCapaPostgreSQL — Query → pot/dependencias/{DEP}/{PER}/capas/{capa}.parquet

Path destino: servidor/pot/dependencias/{DEP}/{PER}/capas/{capa}.parquet

Disparo manual: Integrate → pipeline → Trigger now → p_periodo, p_env. Monitor → Pipeline runs.

Nota: si no confías en los datos, ejecuta sub-pipelines 01-* → 02-* → 03-* **uno a uno** y revisa logs entre pasos.

Resultado esperado

Pipeline/cuaderno ejecutado sin errores
Archivos en Bronce (variables/*.xlsx y/o capas/*.parquet)
Log de ejecución generado en logs-carga/{PER}/
Conteo de archivos copiados > 0 (o confirmado vacío intencional)

8.3 Validaciones (Plata)

Ficha del procedimiento

- **Objetivo:** Validar reglas de negocio sobre insumos en Bronce y generar salida procesada en Plata.
- **Responsable:** Operador.
- **Entradas:** archivos en Bronce del periodo; catálogo POT cargado; usuario registrado en pot.TBL_USUARIO.
- **Salidas:** Parquet en Plata; logs.json de validación; posible lock.json.
- **Tiempo estimado:** 20 – 60 min por flujo.
- **Frecuencia:** cada periodo (automático tras 01-* en pipeline padre).

Validación de Variables tabulares

Sub-pipeline 02-VALIDAR-ARCHIVOS-PLATA, cuaderno ETLValidarArchivosVariables
Parámetros cuaderno: p_periodo, p_env, p_postgres_connection_string, p_correo_usuario.

Bloques lógicos:

1. Periodo, ETL_ENV, bootstrap p_correo_usuario (ERR-CV-002/003 si no registrado)
2. Archivo de histórico de indicadores — JSON remoto → SQL por periodo (ERR-CV-006 si falta archivo)
3. Incremental (default) vs p_procesamiento_completo=true
4. Validación RN-CV-*; Parquet Plata; logs.json

Salidas Plata: Parquet bajo plata/archivos/...; logs en pot/logs/archivos/{PER}/logs.json.

Locks: {bronce}/servidor/pot/logs/archivos/{PER}/lock.json. Si existe, el periodo ya fue validado — confirmar con soporte antes de reprocesar.

Validación de Capas

Sub-pipeline 02-VALIDAR-CAPAS-PLATA, cuaderno ETLValidarCapasGeograficas
Bloques lógicos:

1. Periodo, rutas lake, usuario, locks
2. Catálogo geoproceso — JSON → SQL (ERR-CC-002/020 si falta)
3. Incremental vs completo
4. Validación diccionario, CRS, geometrías, slug
5. SCD2, geoproceso, escalares; _pipeline_out

Salidas de geoprocesos en Plata: capas_procesadas/tablas_maestras/, periodos/{slug}/{periodo}/; logs en pot/logs/capas/{PER}/**/logs.json.

Lock (bloqueo) de geo: {bronce}/servidor/pot/logs/capas/{PER}/lock.json.

Resultado esperado

Sub-pipeline 02-* completado sin errores
Parquet procesado visible en Plata
logs.json sin errores críticos
Usuario p_correo_usuario reconocido en BD

8.4 Carga a Oro (variables)

Ficha del procedimiento

- **Objetivo:** publicar valores validados en vars.TBL_VALORVARIABLE.
- **Responsable:** Operador.
- **Entradas:** Parquet en plata del periodo; validación Plata OK.
- **Salidas:** filas insertadas en Oro; auditoría en TBL_LOGAUDITORIA.
- **Tiempo estimado:** 10 – 30 min.
- **Frecuencia:** cada periodo (automático tras 02-* en pipeline padre).

Sub-pipeline: 03-CARGA-VARIABLES-ORO — cuaderno ETLCargaVariablesOro.

Actividades ADF:

1. WebDBSecret — Key Vault → PostgreSQL
2. CargaOro — cuaderno upload a vars.TBL_VALORVARIABLE

Pasos lógicos del cuaderno:

1. Lee Parquet en plata/pot/variables_procesadas/{periodo}/ — requiere 02-* OK
2. Lock carga Oro; respeta p_can_upload_variables (ERR-CV-022 si lock existe)

3. INSERT ON CONFLICT DO NOTHING; auditoría
4. JSON de salida: inserted_rows, execution_id

Resultado esperado

Sub-pipeline 03-CARGA-VARIABLES-ORO completado sin errores
inserted_rows > 0 o lock coherente con carga previa
VISUALIZAR-DATOS con p_coleccion_a_listar=variables muestra filas del periodo

8.5 Carga a Oro (capas)

Ficha del procedimiento

- **Objetivo:** publicar capas resultado y escalares geográficos en Oro (TBL_CAPARESULTADO, TBL_VALORVARIABLE).
- **Responsable:** Operador.
- **Entradas:** salida procesada en Plata; aprobación QA (Asegurador de calidad) para el periodo geo.
- **Salidas:** capas y escalares en PostgreSQL Oro; lock Oro geo.
- **Tiempo estimado:** 15 – 45 min.
- **Frecuencia:** cada periodo (automático tras 02-* en pipeline padre).

Pipeline maestro: ETL-CARGA-CAPAS — sub-pipeline 03-CARGA-CAPAS-ORO — cuaderno ETLCargarGeodatosOro.

Parámetros adicionales: p_approval_status (debe ser approved), p_correo_usuario, p_env.

Pasos lógicos:

1. Auto-descubre último run Plata con scalars/rutas_disposicion
2. p_approval_status=approved o approval_status.json (ERR-CC-021 si pending)
3. INSERT capas resultado — rollback en manifiesto si falla
4. INSERT escalares geo en VVA — misma auditoría que variables
5. Lock Oro geo tras carga exitosa — re-ejecución bloqueada; contactar soporte

Resultado esperado

Sub-pipeline 03-CARGA-CAPAS-ORO completado sin errores
Aprobación QA (Asegurador de calidad) registrada (approved)
Datos geo consultables en Oro

Lock Oro geo presente tras carga exitosa

8.6 Carga de histórico de indicadores

Ficha del procedimiento

- **Objetivo:** cargar series históricas de indicadores desde JSON en Bronce a vars.TBL_HISTORICOINDICADOR (setup único por ambiente).
- **Responsable:** equipo técnico / operador con permisos de cuaderno interactivo.
- **Entradas:** historico_indicadores.json en ADLS; catálogo POT cargado (ETLCargaInstrumentoPOT completado).
- **Salidas:** filas en vars.TBL_HISTORICOINDICADOR.
- **Tiempo estimado:** 15 – 30 min.
- **Frecuencia:** una vez por ambiente (re-ejecución sólo con autorización).

Cuaderno: ETLCargarHistoricoIndicadores (Synapse Studio interactivo).

Prerrequisitos:

1. Insumos en ADLS: abfss://bronce{suffix}/servidor/pot/insumos/indicadores/historico/historico_indicadores.json
2. Catálogo POT cargado vía **ETLCargaInstrumentoPOT** (esquemas pot/vars creados)
3. Spark Pool del ambiente configurado y disponible

Pasos:

1. Verificar JSON en ruta Bronce ({bronce}/servidor/pot/insumos/indicadores/historico/historico_indicadores.json)
2. Synapse Studio → cuaderno ETLCargarHistoricoIndicadores → ejecutar con p_env del ambiente
3. El cuaderno valida contra catálogo e INSERT en vars.TBL_HISTORICOINDICADOR
4. Consulta posterior: VISUALIZAR-DATOS con p_coleccion_a_listar=indicadores

Parámetros clave: p_historico_indicadores_json, p_postgres_connection_string. No requiere p_correo_usuario.

Cuadernos de setup relacionados (una vez por ambiente):

- **RegistrarUsuario** — una vez por operador de pipelines; garantiza fila en pot.TBL_USUARIO.

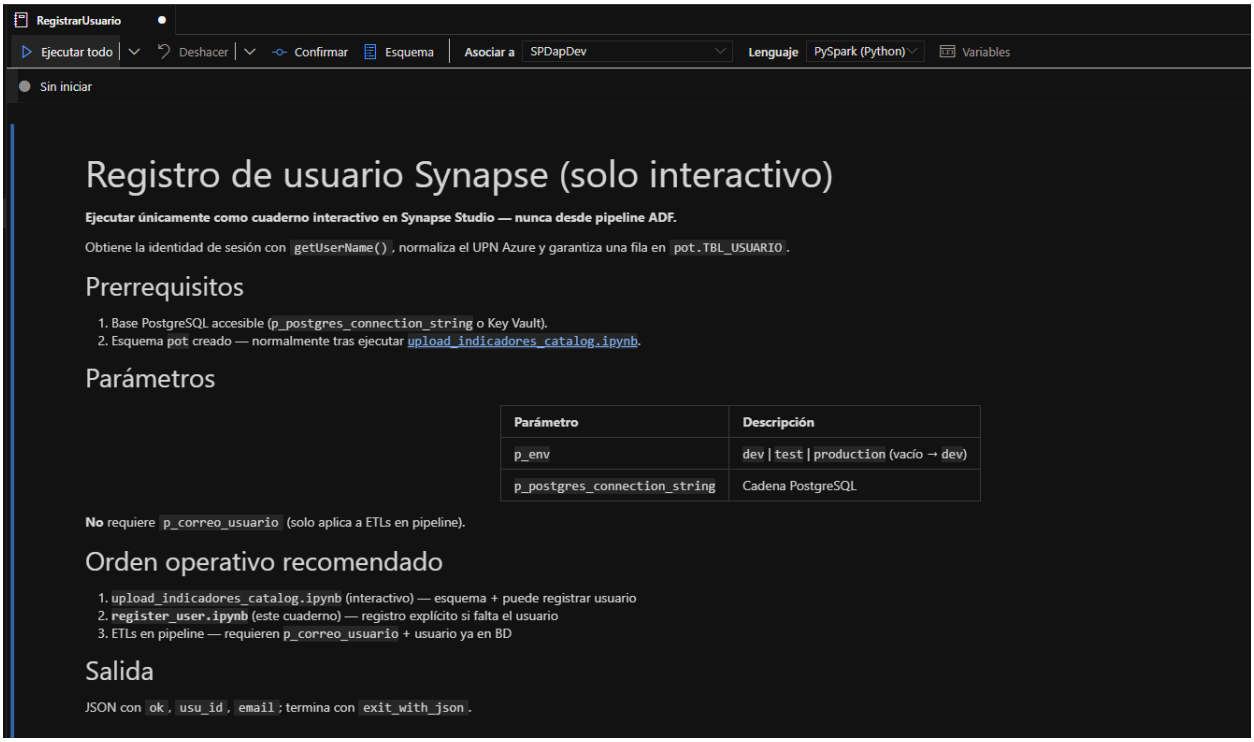


Figura 3: Cuaderno de RegistrarUsuario en Synapse

- **ETLCargaInstrumentoPOT** — crea esquemas, carga catálogo desde Excel instrumento; orden estricto de bloques; lock en instrumento tras éxito.

Resultado esperado

Cuaderno ETLCargarHistoricoIndicadores ejecutado sin errores
Filas visibles en vars.TBL_HISTORICOINDICADOR
VISUALIZAR-DATOS (indicadores) con conteo coherente con JSON fuente
Catálogo POT (ETLCargaInstrumentoPOT) completado previamente

9. Monitoreo de la ejecución

Estado de pipelines

- **Synapse Studio → Monitor → Pipeline runs:** estado Succeeded / Failed / In progress.
- Revisar sub-pipelines 01-, *02*-, 03-* individualmente si el padre falla.
- Usar **Rerun** en el sub-pipeline fallido (no forzar reproceso completo si Plata/Oro ya están OK).

Logs en datalake

Variables:

- Validación: {bronce}/servidor/pot/logs/archivos/{PER}/logs.json
- Copy Activity: {bronce}/servidor/pot/logs-carga/{PER}/*.txt

Capas:

- Validación/proceso: {bronce}/servidor/pot/logs/capas/{PER}/**/logs.json

Pipeline de consulta: VISUALIZAR-LOGS — cuaderno ETLVisualizarLogsBronce — consolida logs tras un fallo.

Verificación de datos publicados

Pipeline: VISUALIZAR-DATOS — cuaderno ETLVerDatosCargados.

Parámetros:

- p_periodo — YYYY_MM (vacío = mes anterior)
- p_coleccion_a_listar — variables | indicadores

Consultas:

- vars.TBL_VALORVARIABLE por VVA_PERIODO — conteo > 0 tras carga variables
- Histórico indicadores — filas tras ETLCargarHistoricoIndicadores

Duración típica

- Preflight (VALIDAR-CONEXION): 5 – 10 min
- Pipeline completo variables o capas: 30 – 90 min según volumen
- Sub-pipeline individual 01-* / 02-* / 03-*: 10 – 45 min

Mapa de artefactos

- **Synapse Studio interactivo:** RegistrarUsuario, ETLCargaInstrumentoPOT, ETLCargarHistoricoIndicadores
- **Pipelines ADF:** VALIDAR-CONEXION, ETL-CARGA-VARIABLES, ETL-CARGA-CAPAS, VISUALIZAR-LOGS, VISUALIZAR-DATOS

10. Manejo de errores por categoría

Ejemplos representativos por categoría. Catálogo completo ERR-CV / ERR-CC en Anexo 14.2.

Errores de archivos (Excel / landing zone)

- **Carpeta inexistente:** dependencia en config.json sin ruta {share}/dependencias/{DEP}/{PER}/variables/ → Copy falla con *Could not find a part of the path*. **Acción:** crear carpeta (aunque vacía) o retirar dependencia del config; re-ejecutar 01-*.
- **Bronce vacío con Copy OK:** carpeta vacía o periodo incorrecto (ERR-CV-007). **Acción:** depositar Excel o corregir p_periodo.
- **Excel corrupto o layout inválido** (ERR-CV-008, ERR-CV-010): **Acción:** solicitar corrección a la dependencia; re-ejecutar 02-*.

Errores de validación (reglas de negocio)

- **Variable no existe en catálogo** (ERR-CV-015): id variable/indicador incorrecto. **Acción:** verificar contra catálogo POT; actualizar catálogo si aplica.
- **Tipo de dato o campo obligatorio** (ERR-CV-011, ERR-CV-012): **Acción:** corregir celda en Excel; re-ejecutar 02-*.
- **Capa no en diccionario** (ERR-CC-004) o **CRS incorrecto** (ERR-CC-008): **Acción:** corregir capa/query; re-ejecutar 01-* → 02-*.

Errores de infraestructura (SHIR / conexión)

- **SHIR detenido o config ilegible:** Copy falla al leer landing zone. **Acción:** restaurar SHIR; verificar {share}/config.json; re-ejecutar 01-*.
- **PostgreSQL inaccesible** (ERR-CV-006): **Acción:** ejecutar VALIDAR-CONEXION; escalar a soporte.
- **Preflight falla** (checks.*.ok = false): revisar Key Vault, ADLS, SHIR, PostgreSQL.

Errores de catálogo

- **Catálogo vacío o desincronizado** (ERR-CV-006, ERR-CV-024, ERR-CC-015): **Acción:** re-ejecutar ETLCargaInstrumentoPOT; luego 02-* o 03-* según stage.
- **Geoproceso mal configurado** (ERR-CC-011, ERR-CC-012): **Acción:** escalar administrador catálogo POT.

Errores de integración (usuario / locks / aprobación)

- **Usuario no registrado** (ERR-CV-002, ERR-CV-003): **Acción:** ejecutar RegistrarUsuario; verificar p_correo_usuario en pipeline padre.
- **Lock de periodo** (ERR-CV-004, ERR-CV-022, ERR-CC-001): periodo ya procesado. **Acción:** confirmar con soporte; no forzar re-ejecución.

- **Aprobación geo pendiente** (ERR-CC-021): **Acción:** obtener aprobación QA (Asegurador de calidad); re-ejecutar 03-CARGA-CAPAS-ORO con approved.

11. Procedimientos de recuperación

Árbol rápido: síntoma → acción → reejecutar

Copy falla — carpeta no encontrada

- Revisar: config.json vs carpetas en landing zone
- Acción: crear carpeta del periodo o quitar dependencia del config
- Reejecutar: 01-CARGA-ARCHIVOS-BRONCE o 01-CARGA-CAPAS-BRONCE

Copy OK pero Bronce vacío

- Revisar: logs-carga/{PER}/; carpetas vacías; p_periodo
- Acción: depositar Excel o confirmar periodo
- Reejecutar: 01-*

Falla SHIR / config.json

- Revisar: estado SHIR; ruta {share}/config.json
- Acción: restaurar SHIR o corregir config
- Reejecutar: 01-*

Falla validación Plata (variables)

- Revisar: pot/logs/archivos/{PER}/logs.json (vía VISUALIZAR-LOGS)
- Acción: corregir Excel
- Reejecutar: 02-VALIDAR-ARCHIVOS-PLATA

Falla validación/proceso geo

- Revisar: pot/logs/capas/{PER}/logs.json
- Acción: corregir capa/diccionario/query
- Reejecutar: 02-VALIDAR-CAPAS-PLATA

Mensaje de lock / "ETL locked"

- Revisar: lock.json del stage
- Acción: confirmar con soporte si el periodo ya fue procesado

- Reejecutar: solo tras autorización de soporte

Error usuario no encontrado

- Revisar: pot.TBL_USUARIO; parámetro p_correo_usuario
- Acción: ejecutar RegistrarUsuario
- Reejecutar: 02-* / 03-* tras registro

Datos no aparecen en Oro

- Revisar: VISUALIZAR-DATOS; ¿llegó a 03-*?
- Acción: verificar Plata OK
- Reejecutar: solo 03-CARGA-*-ORO

Carga geo no inserta en Oro

- Revisar: p_approval_status, approval_status.json
- Acción: obtener aprobación QA (Asegurador de calidad)
- Reejecutar: 03-CARGA-CAPAS-ORO con approved

Re-ejecución parcial

En lugar del pipeline padre completo, ejecutar **en orden** y validar logs entre cada uno:

- **Variables:** 01-CARGA-ARCHIVOS-BRONCE → 02-VALIDAR-ARCHIVOS-PLATA → 03-CARGA-VARIABLES-ORO
- **Capas:** 01-CARGA-CAPAS-BRONCE → 02-VALIDAR-CAPAS-PLATA → 03-CARGA-CAPAS-ORO

Reglas:

- Repetir 01-* si cambió config, falta Bronce o error Copy/SHIR.
- Solo 02-* si Bronce OK pero falló validación.
- Solo 03-* si Plata OK pero falló Oro/aprobación geo.

12. Buenas prácticas de operación

1. **Verificar siempre el periodo** (p_periodo) antes de disparar un pipeline — un periodo incorrecto procesa datos del mes equivocado.
2. **Verificar siempre el correo de usuario** (p_correo_usuario) antes de disparar un pipeline — verificar que el correo sea correcto para evitar un fallo temprano.

3. **No modificar archivos en Bronce** manualmente; Bronce es copia inmutable — corregir en origen (landing zone o fuente geo).
4. **No borrar logs** (logs.json, logs-carga/) — son evidencia para diagnóstico y auditoría.
5. **No ejecutar procesos simultáneos** del mismo flujo y periodo — riesgo de locks inconsistentes.
6. **No forzar reprocesos con lock** — contactar soporte antes de re-ejecutar un stage con lock.json.
7. **Crear carpetas de periodo** para todas las dependencias en config.json **antes** de disparar carga de variables.
8. **Ejecutar preflight** (VALIDAR-CONEXION) antes de cargas largas cuando haya dudas sobre infraestructura.
9. **Adjuntar evidencia al escalar:** Run ID, p_periodo, p_env, export de logs, sub-pipeline fallido.

13. Glosario

#	Término	Definición
1	**ADF (Azure Data Factory)**	Motor de orquestación detrás de los pipelines en Synapse Integrate.
2	**ADLS (Azure Data Lake Storage)**	Almacenamiento del datalake donde residen contenedores Bronce y Plata.
3	**approval_status**	Estado de aprobación QA (Asegurador de calidad) para carga geo a Oro (`approved` / `pending`). Parámetro `p_approval_status`.
4	**Bronce**	Zona cruda del datalake (ADLS): archivos copiados desde landing zone, logs, locks.
5	**Capa geográfica**	Conjunto espacial extraído de PostgreSQL origen según query en `config.json`; procesado y publicado en Oro.
6	**`config.json`**	Archivo maestro en landing zone que declara dependencias Excel y capas geo.
7	**Copy Activity**	Actividad ADF que copia archivos entre origen (SHIR/ADLS) y destino (Bronce).
8	**Cuaderno (Notebook)**	Artefacto Synapse con código Python/Spark; ejecutable interactivo o invocado desde pipeline.
9	**Data flow**	Transformación declarativa ADF que lee `config.json` y genera manifiestos CSV (`VariablesDataFlow`, `CapasDataflow`).
10	**DDL**	Scripts SQL de definición de esquema (`pot.sql`, `vars.sql`) usados en setup de catálogo.
11	**Dependencia**	Organismo reportante; carpeta en landing zone y entrada en `archivos_variables`.
12	**Editor**	Rol que deposita insumos en landing zone y verifica configuración.
13	**Entra ID**	Servicio de identidad Microsoft; cuenta requerida para acceder a Synapse.
14	**ERR-CC**	Código de error de capas geográficas (ver Anexo 14.2).

#	Término	Definición
15	**ERR-CV**	Código de error de carga de variables (ver Anexo 14.2).
16	**ETL**	Extract, Transform, Load; proceso de carga de datos del POT.
17	**ForEach**	Actividad ADF que itera sobre filas del manifiesto CSV para copiar múltiples archivos/capas.
18	**GeoJSON**	Formato de texto para geometrías; las capas se proyectan como `ST_AsGeoJSON(geom)::text AS geom` en queries.
19	**Geoproceso**	Cadena de pasos espaciales/escalares sobre capas; un alias `Cap_*` por plan en catálogo POT.
20	**`HIN_*`**	Prefijo de columnas del histórico de indicadores en JSON/BD.
21	**Indicador**	Métrica del POT definida en catálogo; puede tener histórico en `TBL_HISTORICOINDICADOR`.
22	**Instrumento**	Excel de avances (layout filas 3/5/6/7) o catálogo maestro POT.
23	**Key Vault**	Almacén seguro de secretos; provee cadena de conexión PostgreSQL a pipelines.
24	**Landing zone**	Recurso compartido en servidor DAP (acceso SHIR): `config.json` en raíz y carpetas dependencias/.
25	**Lock (`lock.json`)**	Archivo que marca un stage cerrado; contactar soporte antes de reprocesar.
26	**Manifiesto**	CSV generado en Bronce a partir de `config.json` (`variables.csv` o `capas.csv`); alimenta ForEach de copia.
27	**Oro**	PostgreSQL (vars): hechos publicados (`TBL_VALORVARIABLE`, `TBL_HISTORICOINDICADOR`, `TBL_CAPARESULTADO`).
28	**Operador**	Rol que dispara y supervisa pipelines; no deposita insumos.
29	**Parquet**	Formato columnar de archivos procesados en Plata.
30	**Periodo**	Mes atómico de reporte (`2025_12`); parámetro `p_periodo` (vacío = mes anterior).

#	Término	Definición
31	**Pipeline**	Flujo ADF en Synapse Integrate; se dispara con `Trigger now` o programación.
32	**Plata**	Zona procesada del datalake: Parquet validados, logs de validación, salida intermedia antes de Oro.
33	**pot**	Esquema PostgreSQL de catálogo (variables, indicadores, fórmulas, usuarios).
34	**Preflight**	Comprobación opcional con `VALIDAR-CONEXION` antes de cargar.
35	**QA (Quality Assurance - Asegurador de calidad)**	Control de calidad que debe aprobar (`approved`) el periodo geo antes de carga a Oro.
36	**`p_correo_usuario`**	Correo del operador en actividades Notebook; debe existir en `pot.TBL_USUARIO`.
37	**`p_env`**	Ambiente: dev / test / prod en pipelines; dev / test / `prod` en cuadernos interactivos.
38	**`p_periodo`**	Periodo `YYYY_MM` del pipeline; vacío → mes calendario anterior.
39	**Rerun**	Re-ejecución de actividad o sub-pipeline fallido desde Monitor.
40	**`RN-CV-*`**	Reglas de negocio de validación de variables tabulares (layout Excel, tipos de dato, catálogo, vigencias).
41	**SCD2**	Slowly Changing Dimension tipo 2; estrategia de versionado de capas geo en Plata.
42	**SHIR (Self-Hosted Integration Runtime)**	Integration Runtime en servidor DAP; debe estar Running para leer landing zone y PostgreSQL origen.
43	**Slug**	Identificador corto de capa geográfica en diccionario POT
44	**Spark Pool**	Pool de Spark del workspace; requerido para ejecutar cuadernos Synapse.
45	**Sub-pipeline**	Pipeline hijo: `01-*` (Bronce), `02-*` (Plata), `03-*` (Oro).

#	Término	Definición
46	**Synapse Studio**	Interfaz web del workspace: Integrate (pipelines), Develop (cuadernos), Monitor (runs).
47	**`Trigger now`**	Disparo manual de pipeline desde Integrate con parámetros.
48	**vars**	Esquema PostgreSQL de hechos (valores, histórico, capas resultado).
49	**Visualizador**	Rol que consulta datos en Oro sin operar pipelines.
50	**`VVA_PERIODO`**	Columna de periodo en `TBL_VALORVARIABLE`.

14. Anexos

14.1 Tabla maestra Synapse ↔ Pipeline

Synapse (workspace)	Pipeline ADF	Sub-pipeline
RegistrarUsuario	— (Synapse Studio)	—
ETLCargaInstrumentoPOT	— (Synapse Studio)	—
ETLCargarHistoricoIndicadores	— (Synapse Studio)	—
VerificarConexiones	VALIDAR-CONEXION	—
ETLValidarArchivosVariables	ETL-CARGA-VARIABLES	02-VALIDAR-ARCHIVOS-PLATA
ETLCargaVariablesOro	ETL-CARGA-VARIABLES	03-CARGA-VARIABLES-ORO
—	ETL-CARGA-VARIABLES	01-CARGA-ARCHIVOS-BRONCE
ETLValidarCapasGeograficas	ETL-CARGA-CAPAS	02-VALIDAR-CAPAS-PLATA
ETLCargarGeodatosOro	ETL-CARGA-CAPAS	03-CARGA-CAPAS-ORO
—	ETL-CARGA-CAPAS	01-CARGA-CAPAS-BRONCE
ETLVisualizarLogsBronce	VISUALIZAR-LOGS	—
ETLVerDatosCargados	VISUALIZAR-DATOS	—

14.2 Catálogo completo de errores

Severidad Crítica y Alta. Referencia operativa para diagnóstico y recuperación.

Variables (ERR-CV)

ID	Síntoma en logs/pipeline	Causa probable	Acción operador	Re-ejecutar
ERR-CV-001	*"El periodo debe tener formato YYYY_MM"*	p_periodo mal formado	Corregir parámetro en Trigger now	Pipeline padre o sub-pipeline afectado

ID	Síntoma en logs/pipeline	Causa probable	Acción operador	Re-ejecutar
ERR-CV-002	*"El usuario debe estar registrado… Ejecute el cuaderno de registro"*	Correo no está en pot.TBL_USUARIO	Ejecutar RegistrarUsuario; verificar p_correo_usuario en ADF	02-* / 03-* tras registro
ERR-CV-003	Error en etapa pipeline_user (correo vacío)	Falta p_correo_usuario en actividad Notebook	Pedir al equipo técnico configurar propagación de p_correo_usuario desde pipeline padre	Tras corregir configuración
ERR-CV-004	*"Validación omitida: existe lock.json"*	Periodo ya validado/cerrado	Confirmar con soporte si corresponde reprocesar	Tras autorización de soporte
ERR-CV-005	JSON de indicadores inválido	Parámetro p_indicadores corrupto	Escalar a equipo técnico ETL	Tras corregir parámetro
ERR-CV-006	Catálogo PostgreSQL vacío o error de conexión	BD inaccesible o sin datos para el periodo	Verificar VALIDAR-CONEXION; escalar infra	Tras restaurar BD/catálogo
ERR-CV-007	Sin .xlsx en Bronce; aborted_early=true	Carpeta del periodo vacía o dependencia sin archivos; periodo incorrecto	Verificar landing zone y p_periodo; confirmar carpeta {share}/dependencias/{DEP}/{PER}/variables/ existe	01-CARGA-ARCHIVOS-BRONCE
ERR-CV-008	Error al abrir Excel	Archivo corrupto o no es .xlsx	Solicitar reenvío a la dependencia	02-* tras reemplazar archivo y repetir 01-* si hace falta

ID	Síntoma en logs/pipeline	Causa probable	Acción operador	Re-ejecutar
ERR-CV-009	Error al leer hoja	Hoja protegida/corrupta	Solicitar corrección del instrumento	02-*
ERR-CV-010	Layout inválido (filas 3/5/6/7)	Instrumento no usa plantilla oficial	Corregir Excel con plantilla POT	02-*
ERR-CV-011	Tipo de dato inválido en fila	Valor no numérico/fecha/texto según columna	Corregir celda en Excel	02-*
ERR-CV-012	*"Valor nulo en campo obligatorio"*	NA/ND en campo que no acepta nulos	Diligenciar dato obligatorio	02-*
ERR-CV-013	Fecha inicial no anterior al periodo	fecha_validez_inicial incorrecta	Corregir vigencia en Excel	02-*
ERR-CV-014	Fila excluida por fecha_validez_final	Vigencia terminó antes del periodo	Confirmar si corresponde otro periodo o actualizar vigencia	02-*
ERR-CV-015	Variable no existe en catálogo	Id variable / Id indicador incorrectos	Verificar contra catálogo POT; escalar alta de variable	Tras actualizar catálogo (ETLCargaInstrumentoPOT)
ERR-CV-016	Unidad desconocida	Texto de unidad no está en catálogo	Corregir unidad en Excel o catálogo	02-*
ERR-CV-017	`Unidad Excel ≠ catálogo`	Discrepancia unidad canónica	Ajustar Excel o solicitar cambio de catálogo	02-*
ERR-CV-021	Carga Oro no inserta	Validación sin filas válidas o gate explícito	Revisar logs.json Plata; corregir Excel	02-* luego 03-*

ID	Síntoma en logs/pipeline	Causa probable	Acción operador	Re-ejecutar
	(can_upload_variables=false)			
ERR-CV-022	Lock de carga ya existe	Periodo ya cargado a Oro	Confirmar con soporte antes de reprocesar	Solo 03-* si Plata OK y lock gestionado
ERR-CV-023	Filas omitidas en carga (skip_breakdown)	Parquet con datos inválidos post-validación	Escalar técnico; no editar parquet manualmente	02-* desde origen
ERR-CV-024	*"missing_fk_in_catalog"* — aborta carga	Variables en parquet ya no existen en catálogo	Sincronizar catálogo (ETLCargaInstrumentoPOT)	03-* tras catálogo

Capas (ERR-CC)

ID	Síntoma en logs/pipeline	Causa probable	Acción operador	Re-ejecutar
ERR-CC-001	Lock geo de periodo ya existe	Periodo geo ya procesado	Confirmar con soporte	Tras autorización de soporte
ERR-CC-002	Diccionario geográfico no encontrado	Ruta/config incorrecta en despliegue	Escalar equipo técnico ETL	Tras fix config
ERR-CC-003	Sin capas en Bronce para periodo	Capas no copiadas o config sin capas	Verificar config.json capas y 01-CARGA-CAPAS-BRONCE	01-*
ERR-CC-004	Capa no en diccionario geográfico	Nombre/slug incorrecto	Corregir capa o registrar en diccionario	02-VALIDAR-CAPAS-PLATA

ID	Síntoma en logs/pipeline	Causa probable	Acción operador	Re-ejecutar
ERR-CC-005	*"Columna objectId obligatoria ausente"*	Falta ObjectId en fuente	Incluir columna o alias "id AS objectid" en query	01-* / 02-*
ERR-CC-006	*"Columna obligatoria ausente: {nombre}"*	Atributo requerido falta	Completar capa según diccionario	02-*
ERR-CC-007	Columna de fecha ausente	Falta fecha de actualización definida	Incluir columna en fuente	02-*
ERR-CC-008	`CRS ausente o distinto (ej. ≠ EPSG:9377)`	Proyección incorrecta	Reproyectar capa antes de reenviar	01-* → 02-*
ERR-CC-009	Geometrías inválidas/nulas	Geometría corrupta en origen	Reparar en GIS y reexportar	01-* → 02-*
ERR-CC-010	*"Slug duplicado"*	Dos archivos misma capa/subsistema	Consolidar en un solo archivo	01-*
ERR-CC-011	Geoproceso sin o con múltiples alias Cap_*	Config geoproceso en catálogo incorrecta	Escalar administrador catálogo POT	02-* tras fix catálogo
ERR-CC-012	Paso geoproceso referencia alias inexistente	Plan mal encadenado	Escalar administrador catálogo	02-*
ERR-CC-014	Capas insumo faltantes para geoproceso	Entrada del plan no procesada	Verificar capas aprobadas en periodo	02-*

ID	Síntoma en logs/pipeline	Causa probable	Acción operador	Re-ejecutar
ERR-CC-015	var_id resultado no en catálogo	Variable escalar no registrada	Alta/sincronización catálogo	ETLCargaInstrumentoPOT + 02-*
ERR-CC-016	Excepción en geoproceso (geoprocess_traces)	Error de cálculo espacial	Escalar técnico con traza	02-*
ERR-CC-017	Payload VVA inválido	Campos obligatorios faltantes en escalar	Escalar técnico	02-*
ERR-CC-021	`p_approval_status ≠ approved o pending`	QA no aprobó periodo geo	Revisar bitácora; obtener aprobación QA	03-CARGA-CAPAS-ORO con approved
ERR-CC-022	Error INSERT en Oro (capas/VVA)	Constraint o conectividad BD	Escalar DBA/técnico; revisar manifiesto rollback	03-* tras fix BD

14.3 Checklists por escenario

Usar **solo el checklist que corresponda** a la situación.

Setup de ambiente (una vez)

- Insumos en ADLS (instrumento, histórico, DDL)
- ETLCargaInstrumentoPOT OK
- ETLCargarHistoricoIndicadores OK
- Histórico visible en VISUALIZAR-DATOS (indicadores)
- config.json en landing zone

Setup de usuario (una vez por operador)

- RegistrarUsuario ejecutado
- Usuario existe en pot.TBL_USUARIO

Carga de periodo — solo variables

Setup ambiente + usuario completos
(Opcional) VALIDAR-CONEXION
ETL-CARGA-VARIABLES con p_periodo correcto
VISUALIZAR-DATOS (variables)
Si falló: VISUALIZAR-LOGS + Sección 11

Carga de periodo — solo capas

Setup ambiente + usuario completos
(Opcional) VALIDAR-CONEXION
ETL-CARGA-CAPAS con p_periodo correcto
VISUALIZAR-DATOS (según necesidad)
Si falló: VISUALIZAR-LOGS + Sección 11

14.4 Escalamiento

Escalar a: equipo técnico ETL / soporte DAP.

Adjuntar:

1. Run ID del pipeline
2. p_periodo y p_env
3. Export de logs.json vía VISUALIZAR-LOGS
4. Sub-pipeline fallido
5. Log Copy en logs-carga/{PER}/ si aplica

Confirmar SHIR Running y RegistrarUsuario previo.

14.5 Matriz de decisiones

Si ocurre…	**Entonces…**
Error de Excel	Solicitar nuevo archivo a la dependencia
Error de catálogo	Actualizar catálogo (ver ETLCargaInstrumentoPOT en Sección 8.6)
Error de infraestructura (SHIR/conexión)	Escalar a soporte técnico
Error de parámetros (JSON/config)	Revisar config.json / parámetros del pipeline
Falla en cuaderno/pipeline	Reejecutar sub-pipeline afectado

14.6 Historial de cambios del documento

Fecha	Versión	Cambio
2026-07-20	v1.0	Manual operativo inicial

14.7 Convención de ambientes y parámetros técnicos

Existe un solo concepto de ambiente (desarrollo, pruebas o producción), pero la forma de indicarlo varía según dónde se ejecute:

- **Al disparar un proceso de carga (pipeline):** el parámetro de ambiente admite dev, test o **prod**.
- **Al ejecutar un cuaderno interactivo:** el mismo parámetro admite dev, test o **prod**.

IMPORTANTE: El proceso de carga valida el parámetro con una lista corta (prod); escribir production en ese contexto hace que falle antes de continuar. El código de los cuadernos acepta ambas formas (prod y production) y las normaliza internamente.

Reglas prácticas:

- Al disparar un **proceso de carga** → usar prod (no production).
- Al ejecutar un **cuaderno a mano** → usar prod para el ambiente oficial, o dev/vacío para pruebas.
- dev y test se escriben igual en ambos casos.

Periodo operativo:

- Formato: YYYY_MM (ejemplo: 2025_12).
- Si el parámetro de periodo (p_periodo) queda vacío al disparar un proceso, se asume el **mes anterior** al momento de ejecución.

Identidad del operador:

- En los **cuadernos interactivos** de setup (registro, catálogo, histórico), la identidad se toma automáticamente de la sesión y crea el registro del operador si no existe.
- En los **procesos de carga periódicos**, la identidad se pasa como parámetro (p_correo_usuario) y **debe existir previamente** en el registro de operadores — si no, el proceso falla y hay que completar el registro primero (ver Sección 8.6).

14.8 Configuración de dependencias y capas (config.json)

Ubicación: raíz del recurso compartido de landing zone.

Función: declara (1) qué dependencias aportan archivos de avances y (2) qué capas geográficas extraer de la base de datos de origen. Los procesos de carga leen este archivo para saber qué copiar o consultar en cada periodo.

Claves principales:

- Formato de periodo esperado en las carpetas (YYYY_MM).
- Lista de dependencias que reportan archivos de avances (archivos_variables).
- Lista de capas geográficas a extraer, con la consulta de origen de cada una (capas_estandarizadas).

Reglas para archivos de avances: toda dependencia listada debe tener la carpeta del periodo **creada antes** de disparar la carga, aunque esté vacía.

Reglas para capas geográficas:

1. El nombre de la capa debe estar alineado al diccionario de capas del catálogo.
2. La consulta de origen debe incluir un identificador único de objeto (objectid).
3. Los nombres de columnas/atributos deben respetar el límite de longitud del formato de destino (máximo 10 caracteres).
4. La proyección geográfica debe ser la estándar del sistema (EPSG:9377), salvo excepción documentada.

Las capas geográficas no se depositan como archivos en el servidor de origen del DAP: se extraen directamente de la base de datos de origen durante la carga a Bronce.

14.9 Setup técnico de ambiente (una vez por ambiente)

Antes de la operación recurrente, se completa una única vez por ambiente:

1. Cargar el instrumento de caracterización del POT (catálogo maestro de variables e indicadores) mediante el procedimiento de carga de catálogo.
2. Cargar el histórico de indicadores mediante su procedimiento correspondiente (ver Sección 8.6).
3. Dejar lista la configuración de dependencias y capas (config.json, Anexo 14.8) en el servidor de origen del DAP.
4. Registrar a cada operador que vaya a ejecutar procesos de carga.

El detalle operativo de estos pasos está en la Sección 8.6.

14.10 Estructura técnica del Data Lake y bases de datos

Cadena de datos:

Servidor de origen del DAP (landing zone) → Bronce (copia fiel) → Plata (validado y procesado) → Oro (base de datos de publicación)

Ambientes:

Ambiente	Base de datos Oro	Zonas Bronce / Plata
Desarrollo	indicadores-dev	bronce-dev, plata-dev
Pruebas	indicadores-test	bronce-test, plata-test
Producción	indicadores	bronce, plata (sin sufijo)

Bases de datos: la Zona Oro es una base de datos PostgreSQL con dos esquemas relacionados — uno de catálogo (pot: variables, indicadores, fórmulas, geoprocesos) y uno de hechos (vars: valores reportados, histórico de indicadores, capas resultado).

Organización en Bronce (referencia):

```
servidor/pot/
├── insumos/indicadores/          # catálogo: instrumento, histórico, scripts de esquema
├── dependencias/{DEP}/{PER}/
│   ├── variables/*.xlsx
│   └── capas/*.parquet
├── logs/
│   ├── archivos/{PER}/logs.json, lock.json
│   └── capas/{PER}/…/logs.json, lock.json
└── logs-carga/{PER}/             # registro técnico de copia de archivos
```

Plata almacena la información procesada bajo rutas equivalentes por periodo (variables_procesadas/, capas_procesadas/). Oro corresponde a las tablas de la base de datos de publicación (vars.*).