

Workflow Studio — Catálogo de Nodos (Resumen Rápido)

Minimánnual para intranet | Generado: 12/11/2025 18:38

Este documento resume qué hace cada tipo de nodo disponible en el editor, los parámetros típicos y consejos de uso. Pensado como referencia rápida para armar y mantener flujos de trabajo en la intranet.

Disparadores

Webhook ([trigger.webhook](#))

Inicia el flujo cuando llega una llamada HTTP a una ruta interna de la intranet.

- path (string): Ruta relativa (p. ej., /hook/orden)
- method (string): Verbo HTTP (GET/POST/...)
- secret/auth (bool/string): Validación opcional

Tip: Útil para integraciones con otros sistemas internos. Guardá logs y validá firmas si corresponde.

Programador (Cron) ([trigger.cron](#))

Dispara el flujo en un horario/periodo fijo.

- cron (string): Expresión cron (p. ej., */5 * * * *)
- zonaHoraria (string): IANA (America/Argentina/Buenos_Aires)

Tip: Ideal para tareas nocturnas y monitoreo periódico.

Disparador de Cola ([trigger.queue](#))

Consumo mensajes de una cola (ej. RabbitMQ) y dispara el flujo por mensaje.

- broker (string): Tipo/URI
- queue (string): Nombre de cola
- ack (bool): Confirmar recepción

Tip: Para desacoplar productores/consumidores. Usar deadletter si es posible.

Control

Condición (If) ([control.if](#))

Desvía el flujo según una expresión que evalúa a verdadero/falso.

- expression (string): Ej.: \${payload.status} == 200

Tip: Mantén expresiones simples y documentadas. Encadená con nodos Logger para depurar.

Selector (Switch) ([control.switch](#))

Deriva por caso (ramas con nombre).

- casos (map): Nombre → expresión

- default (string): Rama por defecto

Tip: Útil para manejar múltiples estados de negocio.

Paralelo (Split) ([control.parallel](#))

Lanza varias ramas a la vez.

- ramas (int): Cantidad de subramas

- maxConcurrencia (int): Límite

Tip: Usar con cuidado si hay recursos compartidos (BD/archivos).

Unir (Join) ([control.join](#))

Espera a que terminen N ramas.

- tipo (string): all/any

- timeoutMs (int): Máximo tiempo de espera

Tip: Complemento de Parallel; importante para evitar flujos colgados.

Bucle (Loop/ForEach) ([control.loop](#))

Itera sobre una colección.

- forEach (expr): Fuente

- itemVar (string): Variable del ítem

- max (int): Límite de iteraciones

Tip: Preferí colecciones acotadas para no saturar el servidor.

Demora (Delay) ([control.delay](#))

Pausa el flujo por un tiempo.

- ms (int): Milisegundos de espera

Tip: Útil para reintentos espaciados o ventanas de backoffice.

Reintentar (retry) ([control.retry](#))

Reintenta el bloque anterior en caso de error.

- reintentos (int)

- backoffMs (int)

Tip: Combiná con Logger para rastrear fallas recurrentes.

Límite de tasa (ratelimit) ([control.ratelimit](#))

Asegura que una sección del flujo no supere X ejecuciones por intervalo.

- porSegundos (int)

- max (int)

Tip: Evita saturar APIs o recursos críticos.

Datos e Integraciones

Solicitud HTTP ([http.request](#))

Llama a un endpoint HTTP interno/externo.

- url (string)
- method (string): GET/POST/...
- headers/query/body
- contentType (string)
- timeoutMs (int)

Tip: Cuidar timeouts; si es GET, el body se ignora en el motor.

Consulta SQL ([data.sql](#))

Ejecuta un SELECT/INSERT/UPDATE/DELETE contra SQL Server.

- connection (string) o connectionStringName (string)
- query/commandText (string)
- parameters (map)

Tip: Usá parámetros para evitar inyección; midas resultados antes de ramificar.

Cache GET (Redis) ([data.redis.get](#))

Obtiene un valor de cache por clave.

- connection (string)
- key (string)

Tip: Acelera flujos de alta frecuencia.

Cache SET (Redis) ([data.redis.set](#))

Guarda un valor con TTL opcional.

- connection (string)
- key (string)
- value (any)
- ttlSeconds (int)

Tip: Evita recomputar datos costosos.

Archivo: Leer ([file.read](#))

Lee un archivo del servidor de intranet.

- path (string)
- encoding (string)

Tip: Verificá permisos de lectura/escritura.

Archivo: Escribir ([file.write](#))

Escribe un archivo en disco.

- path (string)

■ encoding (string)

■ overwrite (bool)

Tip: No escribir en carpetas compartidas sin coordinación.

Correo: Enviar ([email.send](#))

Envia un email vía SMTP interno.

■ smtp (string)

■ from (string)

■ to (array)

■ subject (string)

■ html (string)

Tip: Para notificaciones operativas. Evitá datos sensibles en texto plano.

Chat/Slack ([chat.notify](#))

Notificación a canal de chat interno.

■ provider (string)

■ webhookUrl (string)

■ channel (string)

■ message (string)

Tip: Ideal para alertas y aprobaciones rápidas.

Almacenamiento Cloud ([cloud.storage](#))

Operaciones simples contra almacenamiento S3-like.

■ provider/bucket/key/region

Tip: Solo si la intranet tiene salida controlada. Caso contrario, preferir file.*

Cola: Publicar ([queue.publish](#))

Publica un mensaje en una cola.

■ broker (string)

■ queue (string)

■ payload (any)

Tip: Útil para orquestar procesos asíncronos.

Cola: Consumir ([queue.consume](#))

Consumir mensajes de una cola, dentro del flujo actual.

■ broker (string)

■ queue (string)

■ prefetch (int)

Tip: Usar con cuidado en WebForms; preferir servicios dedicados.

FTP: Descargar ([ftp.get](#))

Descarga un archivo por FTP.

- host/user/password
- remotePath
- localPath

Tip: Para legados. Preferir APIs modernas si existen.

FTP: Subir ([ftp.put](#))

Sube un archivo por FTP.

- host/user/password
- localPath
- remotePath

Tip: Configurar carpetas de salida con permisos mínimos.

Entrada de documento ([doc.entrada](#))

Incorpora un documento de trabajo al flujo (simulado/upload/carpeta/api).

- modo (string)
- extensiones (array)
- maxMB (int)
- salida (string)

Tip: Punto de entrada típico en procesos de suscripción/emisión.

Transformación y Lógica

Transformar (Mapeo) ([transform.map](#))

Mapea/transforma estructuras de datos.

- mapping (obj): destino → expresión

Tip: Centraliza transformaciones repetidas.

Función (C#) ([code.function](#))

Aplica una función rápida de servidor.

- code (string): C# inline

Tip: Para lógica pequeña. Si crece, migrar a handler dedicado.

Script (JS) ([code.script](#))

Ejecuta JavaScript simple en el servidor.

- language (js)
- code (string)

Tip: Perfecto para validaciones livianas.

Variables ([state.vars](#))

Escribe/lee variables en el contexto del flujo.

- set (map)
- get (array)

Tip: Evita dependencias escondidas; nombrá claro.

Secretos/Claves ([config.secrets](#))

Lee/define secretos en el contexto (no persistir en logs).

- set/get

Tip: Nunca hardcodear claves en el grafo final.

AI / LLM ([ai.call](#))

Invoca un modelo de lenguaje para clasificación, resumen, etc.

- provider/model/prompt/temperature/maxTokens

Tip: Para soporte operativo (no decisiones críticas sin revisión humana).

Utilidad / Operación

Inicio ([util.start](#))

Punto de arranque del grafo.

Tip: Debe existir exactamente uno.

Fin ([util.end](#))

Cierra una rama del flujo.

Tip: Debe existir ≥ 1 .

Logger ([util.logger](#))

Escribe un evento en el log del flujo.

- level (string)
- message (string)

Tip: Indispensable para auditar y depurar.

Notificar ([util.notify](#))

Notificación simple (email/chat/etc.).

- tipo (string)
- destino (string)
- mensaje (string)

Tip: Para avisos a operadores.

Manejador de Error ([util.error](#))

Bloque para capturar/gestionar errores.

- capturar (bool)
- volverAIintentar (bool)
- notificar (bool)

Tip: Colocar cerca de llamadas remotas/BD.

Subflujo ([util.subflow](#))

Invoca otro workflow por id.

- workflowId (string)

Tip: Reutilizá subprocessos comunes (p. ej., validar CUIT).

Buenas prácticas generales

- Definí siempre un único *util.start* y al menos un *util.end*.
- Usá *util.logger* en puntos clave (entrada/salida de integraciones).
- Preferí parámetros con nombres explícitos y validaciones iniciales.
- Evitá credenciales en claro dentro del grafo; centralizá en *config.secrets*.
- Para SQL, usá parámetros y registrá la cantidad de filas afectadas.
- Mantené los flujos idempotentes cuando sea posible (reintentos seguros).
- Documentá en el label de cada nodo el propósito funcional (no técnico).

Este resumen es complementario al inspector del editor. Para dudas puntuales, anotá ejemplos en cada nodo y guardá capturas de pantalla del grafo.