Automatizaciones programadas: cómo recuperar ejecuciones pendientes

Antes de reactivar, identifica las ejecuciones que debían comenzar y acuerda cuáles siguen vigentes, qué período procesan y cómo convivirán con el trabajo actual. Recupera solo lo autorizado y comprueba cada resultado. Conserva las omisiones y excepciones con su motivo; reactivar el calendario no acredita que se haya completado el trabajo pendiente.

Lista de chequeo antes de recuperar ejecuciones pendientes

[ ] La tarea tiene calendario, zona horaria y período de datos definidos.

[ ] La lista distingue horarios sin iniciar de acciones con resultado incierto.

[ ] Cada pendiente tiene una decisión: recuperar, omitir o revisar.

[ ] La recuperación histórica tiene alcance autorizado y límites de concurrencia.

[ ] La prueba contempla vencimientos, trabajo activo y resultados existentes.

[ ] El cierre identifica resultados confirmados y pendientes con responsable.

[ ] El siguiente horario normal mantiene el comportamiento acordado.

Próximas acciones

1. Elige una tarea recurrente y describe cuándo deja de ser útil ejecutarla.
2. Pide al proveedor una política de recuperación con alcance, responsables y evidencia.
3. Acuerda el ensayo con datos ficticios antes de aceptar la puesta en operación.

Notas para tu equipo

_____

_____

_____

_____

Artículo y fuentes

https://www.digital-dev.cl/blog/automatizacion-ejecuciones-pendientes-tras-interrupcion/

Kubernetes: CronJob: inicio tardío, concurrencia y limitaciones
https://kubernetes.io/docs/concepts/workloads/controllers/cron-jobs/

Apache Airflow: Dag Runs: intervalos, programación y catchup — Airflow 3.4.0
https://airflow.apache.org/docs/apache-airflow/stable/core-concepts/dag-run.html

Apache Airflow: Backfill: reprocesamiento y concurrencia — Airflow 3.4.0
https://airflow.apache.org/docs/apache-airflow/stable/core-concepts/backfill.html