

Definiciones Técnicas Migración de datos y lógica.

1. Definición del Esquema Destino (PostgreSQL)	2
1.1. Aislamiento por Sistema (Prefixing):	2
1.3. Regla de "Promoción de Columnas" (The 80/20 Rule)	2
1.4. Estrategia de Prefijado (Conditional Prefixing)	2
2. Estrategia de Integridad y Relaciones	3
2.1. Estándar de Nomenclatura (Exact Match)	3
2.2. Regla de "Redundancia Estratégica" (Copy vs Move)	4
2.3. Contenido y Vínculo (Transformación Espejo)	4
ESCENARIO 1: TABLA COMPARTIDA (Ejemplo Clientes)	4
ESCENARIO 2: TABLA REPETIDA (Con Colisión de IDs)	4
2.4. Integridad Referencial Lógica	5
3. Estrategia de Archivos (Hybrid Storage & Drive Indexing)	5
3.1. Arquitectura de Almacenamiento	5
3.2. Ejecución de Transferencia (Squad 3)	6
3.3. Lógica de Transformación en n8n	6
3.4. Manejo de "Links Rotos" (Drive)	6
4. Estrategia de Unificación y Deduplicación (Match & Map)	6
4.1. Carga de la Fuente Maestra (Golden Source)	7
4.3. Resolución de Expedientes (Smart Translation)	7
5. Estándar de Ejecución Técnica (N8N Blueprint)	8
5.1. Procesamiento por Lotes (Batching Obligatorio)	8
5.2. Manejo de Errores (Non-Blocking / Dead Letter Queue)	8
5.3. Idempotencia (Re-runnability)	9
5.4. Logs de Auditoría Técnica	9
6. Estrategia de QA y Aceptación (Definition Of Done)	9
6.1. La Ecuación de Reconciliación (The Golden Rule)	9
6.2. Validación de Integridad Referencial (SQL Checks)	9
6.3. Verificación de Archivos (Muestreo de Acceso)	10
6.4. Entregable de Cierre (Migration Sign-off)	10

1. Definición del Esquema Destino (PostgreSQL)

Contexto: Para acelerar el desarrollo y evitar "pérdida de datos" por cambios de requerimientos, utilizaremos un modelo híbrido: columnas relacionales para lo operativo y **JSONB** para el almacenamiento histórico completo.

1.1. Aislamiento por Sistema (Prefixing):

- Las tablas destino llevarán el prefijo de su fuente para evitar colisiones de IDs.
- Origen Sistema SRS → Destino **srs_[nombre_tabla]**.
- Origen Sistema ACH → Destino **ach_[nombre_tabla]**.
- **No homologar:** Si la tabla **usuarios** existe en ambos, se crean **srs_usuarios** y **ach_usuarios**.

1.2. Estructura Base Obligatoria (Technical Columns):

Toda tabla debe iniciar con estas columnas técnicas:

- **id (UUID/SERIAL):** PK nativa de PostgreSQL.
- **original_id (VARCHAR):** El ID tal cual viene del sistema origen (**INDEXADO**).
- **sys_migrated_at (TIMESTAMP):** Fecha de carga (**NOW()**).
- **attributes (JSONB):** El registro completo del origen en formato JSON.

1.3. Regla de "Promoción de Columnas" (The 80/20 Rule)

Se "promueven" a columnas explícitas (**VARCHAR, DATE, INT**) **SOLO** los datos que cumplen estos criterios:

- **A. Llaves Foráneas (FKs):** Necesarios para JOINs (ej. **cliente_id**, **municipio_id**).
- **B. Campos de Expediente o Formularios:** Datos de formulario o formulario en SDT (ej. **numero_expediente**, **dui**).
- **C. Campos de Ordenamiento:** Fechas críticas (ej. **fecha_creacion**).
- **D. Estados/Flags:** Datos de lógica de negocio (ej. **status**).
- *Todo lo demás se queda exclusivamente dentro de **attributes**.*

2. Estrategia de Integridad y Relaciones

Contexto:

Debido a la inconsistencia de nombres en las fuentes (mezcla de singular/plural), no impondremos reglas gramaticales. La nomenclatura de las relaciones debe coincidir

exactamente con el nombre técnico de la tabla destino para facilitar los JOINs automáticos.

2.1. Estándar de Nomenclatura (Exact Match)

Se prohíbe usar los nombres de columna del origen (`id_cte`, `cve_prov`).

- **Patrón Obligatorio:** `[Nombre_Tabla_Destino_Sin_Prefijo]_code`
- **Regla:** El nombre de la columna FK debe ser idéntico al nombre de la tabla a la que apunta (quitando el prefijo del sistema `srs_` o `ach_`) más el sufijo `_code`.
 - *Ejemplo 1:* Si la tabla padre es `srs_clientes` (plural), la columna en el hijo se llama `clientes_code`.
 - *Ejemplo 2:* Si la tabla padre es `ach_informacion_fiscal` (singular/compuesto), la columna se llama `informacion_fiscal_code`.

2.2. Regla de "Redundancia Estratégica" (Copy vs Move)

- **Preservación:** El campo original del legacy (ej. `id_cte`) **NO se elimina**. Se envía directo al campo `attributes` (JSONB) para auditoría.
- **Promoción:** El valor se **COPIA** a la nueva columna estandarizada (`clientes_code`).

2.3. Contenido y Vínculo (Transformación Espejo)

El valor que se guarda en la columna de relación (`..._code`) **NO** es el ID crudo del origen. Es el resultado de aplicar la **misma fórmula de transformación** que se usó para generar la llave del parent.

Fórmula General:

Depende de si la tabla padre cae en **Regla A (Compartida)** o **Regla B (Única)** según la Sección 1.4.

- **Regla A (Full):** '`[DB] - [ENT] - ' + Source.ID_Original`
- **Regla B (Short):** '`[ENT] - ' + Source.ID_Original`

ESCENARIO 1: TABLA COMPARTIDA (Ejemplo Clientes)

Supuesto: La tabla `srs_clientes` es crítica y existe en la Base Financiera (**FIN**). Aplica **Regla A** (Prefijo Completo).

1. **Carga del Padre (Clientes):**

- El Cliente con ID 100 viene de la base FIN.
- Se guarda en la tabla srs_clientes con código: FIN-CLI-100.

2.

3. Carga del Hijo (Facturas):

- La factura tiene id_cliente = 100 y viene de la base FIN.
- El Squad concatena el prefijo de sistema + entidad.
- Guarda en srs_facturas.clientes_code: FIN-CLI-100.
- **Resultado:** El sistema encuentra el match perfecto.

ESCENARIO 2: TABLA REPETIDA (Con Colisión de IDs)

Supuesto: La tabla srs_municipios existe en Base Argon (ARG) y Base Sisam (SIS). En ambas, el ID 1 es "San Salvador". Aplica Regla A.

1. Carga de Padres (Municipios):

- **Squad 1 (Base ARG):** Carga el ID 1 → Genera código ARG-MUN-1.
- **Squad 2 (Base SIS):** Carga el ID 1 → Genera código SIS-MUN-1.
- *Resultado en BD:* Existen dos registros únicos (ARG-MUN-1 y SIS-MUN-1).

2. Carga de Hijos (Expedientes):

- **Expediente A (Viene de Base ARG):**
 - Trae municipio_id = 1.
 - Como viene de Argon, el Squad transforma → ARG-MUN-1.
 - *Link:* Se conecta con el "San Salvador" de Argon.
- **Expediente B (Viene de Base SIS):**
 - Trae municipio_id = 1.
 - Como viene de Sisam, el Squad transforma → SIS-MUN-1.
 - *Link:* Se conecta con el "San Salvador" de Sisam.

2.4. Integridad Referencial Lógica

- **Orden de Carga:** Primero tablas Maestras/Padres, luego Transacciones/Hijos.
- **Manejo de Huérfanos:** Si el valor origen es nulo o no existe en el padre, y el campo es obligatorio, insertar el valor '**DEFAULT**'. Prohibido dejar NULLs en columnas de relación críticas.

3. Estrategia de Archivos (Hybrid Storage & Drive Indexing)

Contexto:

El sistema ACH almacena URLs de Google Drive (basadas en IDs), mientras que SRS usa

rutas físicas. GCS requiere rutas basadas en nombres. Para ACH, es obligatorio generar un índice intermedio que relacione `Drive_ID` VS `File_Path`.

3.1. Arquitectura de Almacenamiento

- **SRS (Filestore):** `gs://bucket/srs/[ruta_original]`.
- **ACH (G. Drive):** `gs://bucket/ach/[ruta_espejo_del_drive]`.
 - Nota: Storage Transfer Service (STS) replica la estructura de carpetas exacta del Drive en el Bucket.

3.2. Ejecución de Transferencia (Squad 3)

- **Paso 1: Generar el Índice (MANDATORIO para ACH):**
 - Antes (o durante) la transferencia, se debe correr un script de inventario (usando `rclone` o Google Drive API) sobre el Drive centralizado.
 - **Output CSV:** `Drive_ID | Full_Path_Name` (ej: `/Finanzas/2023/Factura.pdf`).
 - *Este CSV se carga en n8n como tabla de búsqueda.*
- **Paso 2: Transferencia Física:**
 - Ejecutar **STS** (Drive to GCS). Esto mueve los bytes y crea las carpetas en el bucket.

3.3. Lógica de Transformación en n8n

- **Caso A: SRS (Rutas Físicas):**
 1. Lógica simple de reemplazo de string (`c: → gs://`).
- **Caso B: ACH (URLs de Drive):**
 1. **Extracción:** El nodo Code extrae el ID de la URL almacenada en la BD.
 - `https://drive.../d/1A2B3C/view` → `1A2B3C`.
 2. **Lookup (El Cruce):**
 - Cruzar el ID extraído contra el **CSV de Índice** generado en el paso 3.2.
 - **Resultado:** Obtiene el nombre real del archivo:
`/Finanzas/2023/Factura.pdf`.
 3. **Construcción:**
 - Concatenar: `gs://bucket/ach + [Ruta_Del_CSV]`.

3.4. Manejo de "Links Rotos" (Drive)

- Si el ID de la base de datos no aparece en el CSV del Drive (el archivo fue borrado del Drive pero el link quedó en la BD):
- **Acción:** Guardar en BD la URL original de Drive en el campo `attributes` y dejar el campo `file_url` (GCS) en `NULL` o apuntando a un archivo de texto "MISSING_FILE.txt".

4. Estrategia de Unificación y Deduplicación (Match & Map)

Alcance:

Esta estrategia aplica **exclusivamente** a las tablas identificadas como "Compartidas/Riesgo Alto" en la Sección 1.4 (ej. Municipios, Clientes). Las tablas únicas se migran directo.

4.1. Carga de la Fuente Maestra (Golden Source)

- **Definición:** Se elige la base de datos principal (ej. **Argon**) como la dueña de la verdad.
- **Squad 1:** Carga Argon primero.
- **IDs Oficiales:** Se generan siguiendo la fórmula de prefijo completo: `ARG-MUN-01`.

4.2. Algoritmo de Coincidencia (Matching Logic)

Al procesar la Base Secundaria (ej. **Sisam**), el pipeline de n8n **NO inserta directo**.

- **Criterio de Match:** Comparar `TRIM(UPPER(Nombre_Origen))` vs `TRIM(UPPER(Nombre_Destino))`.
- **Escenario 1: Hay Coincidencia (Match Found)**
 - **Ejemplo:** "San Salvador" (ID 50 en Sisam) ya existe como "San Salvador" (ID 1 en Argon).
 - **Acción:** El registro de Sisam se **DESCARTA**.
 - **Mapeo en Memoria:** Se registra que el ID origen `50` de Sisam equivale al ID oficial `ARG-MUN-01`.
- **Escenario 2: No Hay Coincidencia (No Match)**
 - **Ejemplo:** "Villa Extinta" (ID 99 en Sisam) no existe en Argon.
 - **Acción:** Se inserta el registro nuevo aplicando el prefijo de su propio sistema.
 - **ID Generado:** `SIS-MUN-99`.

4.3. Resolución de Expedientes (Smart Translation)

Al migrar expedientes desde la Base Secundaria (Sisam):

- **Instrucción:** El campo `municipio_code` no puede llenarse ciegamente con `SIS-MUN-50`.
- **Proceso de Lookup:**
 1. Tomar el ID original (`50`).
 2. Buscar en la tabla de Mapeo (generada en 4.2).
 3. **Si hubo Match:** Escribir el ID del Maestro → `ARG-MUN-01`.
 4. **Si NO hubo Match:** Escribir el ID con prefijo propio → `SIS-MUN-50`.
- **Resultado:** El expediente queda vinculado al registro único, sin importar de qué base vino.

4.4. Manejo de Conflictos de Atributos

- **Regla:** Si los datos difieren (ej. Teléfono), prevalece el dato de la **Fuente Maestra (Argon)**.
- **Auditoría:** Los datos divergentes de la Base Secundaria no se pierden; se agregan al `attributes` (JSONB) del registro maestro, bajo la llave `legacy_conflict_sisam`.

5. Estándar de Ejecución Técnica (N8N Blueprint)

Contexto:

Para mitigar riesgos de timeout, consumo de memoria y bloqueos por errores de data sucia, todos los flujos de n8n deben seguir estrictamente este patrón de diseño.

Instrucción para Squads:

5.1. Procesamiento por Lotes (Batching Obligatorio)

- **Prohibido:** Hacer `SELECT *` de tablas con más de 2,000 registros.
- **Patrón:** Implementar lectura paginada.
 - *Source Node:* Query con `LIMIT 1000 OFFSET {{batch_index}}`.
 - *Loop:* El flujo debe iterar hasta que el Source devuelva 0 registros.
- **Justificación:** n8n maneja los datos en memoria JSON. Cargar 50k expedientes de golpe crasheará el proceso.

5.2. Manejo de Errores (Non-Blocking / Dead Letter Queue)

- **Configuración:** En el nodo de escritura (Postgres Insert), activar la opción "**Continue On Fail**".
- **Ruteo de Errores:**
 - El puerto de "Error" del nodo debe conectarse a un nodo secundario de inserción.
 - **Destino:** Tabla `sys_migration_errors`.
 - **Datos a guardar:** `table_name`, `batch_id`, `error_message` y el `json_payload` (el registro completo que falló).
- **Objetivo:** Si 5 registros de 1,000 están corruptos, se guardan en el log y los otros 995 se migran exitosamente.

5.3. Idempotencia (Re-runnability)

- **Regla:** Los pipelines deben poder ejecutarse N veces sin duplicar datos.
- **Método:** Usar siempre `UPSERT` (Insert or Update) en lugar de solo `INSERT`.
- **Llave de Conflicto:** Configurar el Upsert basado en la columna `original_id` (y el prefijo de sistema si aplica).
 - **Escenario:** Si el pipeline falló a la mitad y lo corres de nuevo, los registros ya migrados se actualizan (no pasa nada) y los faltantes se insertan.

5.4. Logs de Auditoría Técnica

- Cada registro insertado en las tablas destino debe llevar obligatoriamente:
 - `sys_migrated_at`: Timestamp del momento de inserción.
 - `sys_batch_id`: Identificador de la ejecución (para poder borrar un lote completo si algo sale mal).

6. Estrategia de QA y Aceptación (Definition Of Done)

Contexto:

El objetivo del QA no es revisar registro por registro, sino validar la **integridad estructural** y la **disponibilidad** de los datos para el SDT. Se utilizará un enfoque de "Validación por Excepción".

6.1. La Ecuación de Reconciliación (The Golden Rule)

Para dar por cerrado un bloque, se debe cumplir matemáticamente esta igualdad:

- **Total Registros Origen = Total Registros Destino + Total Registros en Tabla de Errores.**
- **Criterio de Aceptación:**
 - Discrepancia de conteo: **0%**.
 - Porcentaje de errores permitidos (en tabla `sys_migration_errors`): **< 1%** (Si es mayor, el bloque se rechaza y vuelve a Dev).

6.2. Validación de Integridad Referencial (SQL Checks)

El QA debe ejecutar queries automáticos para buscar "links rotos" en las columnas de relación estandarizadas.

- **Query de Validación:** `SELECT count(*) FROM srs_expedientes WHERE municipio_code IS NULL.`
- **Criterio:**
 - Campos Obligatorios: **0 NULLs**. (Deben apuntar a un ID válido o al registro 'DEFAULT').
 - Relaciones Prefijadas: Validar que los códigos sigan el formato `[DB] - [ENT] - ...` y no IDs crudos.

6.3. Verificación de Archivos (Muestreo de Acceso)

No basta con que el texto diga `gs://`. El archivo debe existir.

- **Método:** Muestreo aleatorio del 5% de los expedientes migrados.
- **Prueba:** Tomar la URL de la base de datos y verificar accesibilidad vía consola de GCS o navegador.
- **Criterio:** Si más del **2%** de la muestra arroja "File Not Found" (404), el bloque de archivos se rechaza.

6.4. Entregable de Cierre (Migration Sign-off)

El Squad debe entregar una "Ficha de Cierre" (Jira Ticket/Doc) que contenga:

1. Screenshot del Conteo (Origen vs Destino).
2. Screenshot del Log de Errores (vacío o con excepciones justificadas).
3. Confirmación de que los índices de búsqueda (PKs y FKs) fueron creados en Postgres.