REPORTE ESCALAS

Fuente: /var/www/html/qareportes/app/Http/Controllers/Cloud/EscalasController.php

1) Nombre del reporte

- Nombre: Reporte Escalas
- Código / Alias: report_escalas
- Responsable: Equipo de Reportes Cloud
- Propósito (1 línea):

2) Alcance temporal

- Campo(s) de fecha utilizados:
  - first_date
  - second_date

- Tipo de rango (diario / mensual / personalizado): Mensual
- Inclusión de horas (sí/no). Si sí: [00:00:00 – 23:59:59]
- Zona horaria: La del servidor

3) Origen de datos (tablas/APIs)

3.1 Conexión / Base: EMPRESARIAL

- Tabla / Recurso: emp_cargprogramado
- Alias (si aplica): no aplica
- Campos utilizados (clave → descripción):
  - cap_id → ID de carguero programado
  - cap_chofer1id →ID del primer chofer del carguero programado
  - cap_chofer2id →ID del segundo chofer del carguero programado
  - cap_rutacarguero → Ruta del carguero programado

    - cap_cargid → ID del carguero

- Condiciones (WHERE) aplicadas: no aplica
- Tipo de unión con otras fuentes (JOIN y llave): no aplica
- Observaciones (ej. particiones, índices):
    - Tablas particionadas por año y mes: report_monitoreo_YYYY_MM

3.2 Conexión / Base: EMPRESARIAL

- Tabla / Recurso: emp_reportes_validation
- Alias (si aplica): no aplica
- Campos utilizados (clave → descripción):
    - report → nombre de la tabla
    - first_date →Fecha inicio
    - second_date →Fecha Fin
    - created_date → Fecha y hora de creacion
    - status → Estado

- Condiciones (WHERE) aplicadas: no aplica
- Tipo de unión con otras fuentes (JOIN y llave): no aplica
- Observaciones (ej. particiones, índices):
    - Tablas particionadas por año y mes: report_monitoreo_YYYY_MM

4) Filtros globales del reporte

- Inclusiones (must-have):
    - En el método show() se construye obligatoriamente la data de programaciones, escalas, embarques, desembarques, retornos, etc.

- Los campos que se insertan en la tabla report_escalas_YYYY_MM son fijos y obligatorios (fecha, linea_tiempo, id_carguero, ruta, placa, etc.).

- Siempre se registra en la tabla de validación emp_reportes_validation con status = 1 (éxito) o status = 0 (falla).

- Exclusiones (reglas de descarte):
  - Registros de programación de cargueros que tengan el campo “eliminado” con valor “0”.

- Reglas por estado / habilitado:
- Parámetros externos (si el usuario puede filtrarlo): No aplica

5) Transformaciones y Reglas de negocio

- Derivaciones de campos (cómo se calculan):
  - km_acumulado → diferencia entre km_llegada y km_salida si es válido
  - dur_emb y dur_desemb → calculados con DateTime->diff() entre fechas de embarque y desembarque.
  - fecha, hora (salida/llegada, embarque, desembarque) → extraídos con explode sobre fh_*
  - cantPaquetes, cantGuias → forzados a 0
  - inicio_establecido, fin_establecido, dias_establecido → traídos de data_escalas

- Mapeos de estado:
  - 0 = “Pendiente”
  - 1 = “Proceso”
  - 2 = “Completado”

- Reglas de validación (p.ej., sólo registros validados):
  - La función programaciones() solo se traen registros que no hayan sido eliminados
    - ->where(“eliminados”, “0”)
  - Solo se incluyen programaciones que tengan escalas activas

    - estado != 0

- Reglas condicionales (si X entonces Y):
    - Si no existe tabla report_escalas_Y_m → se salta truncado.
    - Si valesc no existe en escalas → se construye un objeto vacío con valores básicos.
    - Si $escala["km_acumulado"] < 0 → se fuerza a vacío en lugar de un número negativo.
    - Si estadoEscalaDelete == false → la programación no se agrega al JSON final.
    - Si $valrel["estado"] cambia → el nombre del estado cambia (PENDIENTE, PROCESO, etc.).

6) Estructura de salida

- Tabla/archivo destino: report_escalas_AÑO_MES
- Particionado / sufijo: El particionado se logra mediante el sufijo año_mes (YYYY_MM) en el nombre de las tablas.
- Clave(s) primaria(s) o únicas: No especifica
- Columnas del reporte (nombre → tipo → descripción):
    - fecha → DATE → Fecha general del registro o salida
    - linea_tiempo → STRING → Línea de tiempo asociada
    - id_carguero → INT → Identificador del carguero
    - fecha_salida_origen → DATE → Fecha de salida desde el origen
    - hora_salida_origen → TIME → Hora de salida desde el origen
    - fecha_llegada → DATE → Fecha de llegada
    - hora_llegada → TIME → Hora de llegada
    - fecha_salida → DATE → Fecha de salida
    - hora_salida → TIME → Hora de salida
    - ruta → STRING → Ruta asignada
    - tipo_grupo → STRING → Tipo de grupo asociado
    - placa → STRING → Placa del vehículo
    - agencia → STRING → Nombre de la agencia
    - km_acumulado → INT → Kilometraje acumulado
    - km_llegada → INT → Kilometraje registrado a la llegada
    - km_salida → INT → Kilometraje registrado a la salida
    - fecha_inicio_emb → DATE → Fecha de inicio de embarque
    - hora_inicio_emb → TIME → Hora de inicio de embarque
    - fecha_fin_emb → DATE → Fecha de fin de embarque
    - hora_fin_emb → TIME → Hora de fin de embarque
    - duracion_emb → INT → Duración total del embarque (en horas/minutos)
    - fecha_inicio_desemb → DATE → Fecha de inicio de desembarque

    - hora_inicio_desemb → TIME → Hora de inicio de desembarque
    - fecha_fin_desemb → DATE → Fecha de fin de desembarque
    - hora_fin_desemb → TIME → Hora de fin de desembarque
    - duracion_desemb → INT → Duración total del desembarque (en horas/minutos)
    - estado → STRING → Estado actual de la escala
    - inicio_establecido → DATE → Fecha de inicio establecida
    - fin_establecido → DATE → Fecha de fin establecida
    - dias_establecido → INT → Cantidad de días establecidos

- Ordenamiento por defecto: no se aplica

7) Frecuencia y operación

- Frecuencia de actualización: PENDIENTE(Esperando info de eduardo)
- Ventana que recalcula (días/meses): Recalcula desde el primer día al último día del mes.
- Tamaño de lote / paginado (chunking): Los datos insertados se procesan en lotes de 900 registros.
- Tolerancia a fallos / reintentos: Se guarda un log con estado 0 si hay errores y se ejecuta un rollback.
- Tiempos de ejecución esperados: PENDIENTE(Esperando info de eduardo)

8) Calidad y controles

- Validaciones previas/post:
    - Previas:
        - Se valida si la tabla de destino existe con SHOW TABLES LIKE '{$table}'.
        - Si existe, se hace un TRUNCATE antes de insertar datos nuevos.
        - En programaciones() se filtra con whereBetween y eliminado = 0, y además se limpian los origenes y destinos dejando solo numéricos (array_filter + is_numeric).
    - Post:
        - En cada iteración se registra en la tabla emp_reportes_validation un log con status = 1 o 0, dependiendo de si el try/catch fue exitoso o falló.

    - Al final, se retorna un ["status" => true] si se completó el flujo, o false si no hubo datos procesados.

- Controles de duplicados:
    - Se usan array_unique y array_filter para evitar IDs repetidos en cargueros, retornos, escalas, etc.
    - Se usa $data_unique para evitar procesar dos veces el mismo idcarguero.
    - Se evita insertar orígenes/destinos repetidos con !in_array.

- Métricas de completitud (qué se monitorea):
    - cant_events inicializado en cada programación.
    - km_acumulado, dur_emb, dur_desemb como métricas operativas.
    - escalas_data y data_escalas para saber cuántos hitos de ruta se completaron.
    - Validación de estados: PENDIENTE, PROCESO, COMPLETADO, OMITIDO.
    - Fechas de embarque y desembarque (fh_emb_ini/fin, fh_desmb_ini/fin).
    - En emp_reportes_validation se guarda status (1 éxito, 0 fallo).

9) Dependencias externas

- APIs / servicios y endpoints: No aplica
- Autenticación / headers: No aplica
- Mapeos y contratos de datos:
    - report → nombre del reporte
    - first_date → fecha inicio del rango
    - second_date → fecha fin del rango
    - created_date→ timestamp del proceso
    - status → 0 o 1 (éxito o error)

10) Historial de cambios

2025-09-25 13:44 - Elian Franco Arroyo Rodas - Documentación inicial

---

Revisión #1
Creado 2025-10-24 15:29:01 -05 por Elian

Actualizado 2025-10-24 15:29:09 -05 por Elian