

# REPORTE MANTENIMIENTO TERCEROS

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

## 1) Nombre del reporte

- Nombre: Reporte Mantenimiento Tercero
- Código / Alias: report-mante-tercero
- Responsable: Equipo de Reportes Cloud
- Propósito (1 línea):

## 2) Alcance temporal

- Campo(s) de fecha utilizados:
  - fecha\_inicio
  - fecha\_fin
- 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)

- Conexión / Base: API CDTALLERES
- Tabla / Recurso: tabMantenimiento de Terceros
- Alias (si aplica): no aplica
- Campos utilizados (clave → descripción):
  - estado\_flujo → Estado del documento
  - name → Identificador del registro
  - docstatus → Estado interno de Frappe
  - cliente → Tipo de cliente (Registros de Costumer)

- numero\_placa → Placa relacionada al Registro de Vehículo
  - flota → Tipo dato, Dato relacionado al registro de Vehículo
  - nombre\_del\_usuario → Nombre del proveedor
  - costo\_final → Monto total de los productos o servicios del mantenimiento
  - fecha\_de\_registro → Fecha de creación del registro
  - creation → Fecha y hora de creación del registro
- Condiciones (WHERE) aplicadas: Filtra los registro dentro del rango establecido en fecha\_inicio y fecha\_fin
  - Tipo de unión con otras fuentes (JOIN y llave): no aplica
  - Observaciones (ej. particiones, índices): sin obs

## 4) Filtros globales del reporte

- Inclusiones (must-have):
  - Ingreso de fecha\_inicio y fecha\_fin
  - La consulta de los campos de la tabla tabMantenimiento de Terceros
- Exclusiones (reglas de descarte): Registros fuera del rango de fecha.
- Reglas por estado / habilitado: No aplica
- Parámetros externos (si el usuario puede filtrarlo): Aplica con fecha\_inicio y fecha\_fin

## 5) Transformaciones y Reglas de negocio

- Derivaciones de campos (cómo se calculan): No aplica
- Mapeos de estado: En baseReport se guarda logs con estados 0 = error y 1 = exitoso
- Reglas de validación (p.ej., sólo registros validados): Si no se especifica el rango de fecha no se ejecuta el servicio.
- Reglas condicionales (si X entonces Y): no aplica

## 6) Estructura de salida

- Tabla/archivo destino: report-mante-tercero
- Particionado / sufijo: no aplica
- Clave(s) primaria(s) o únicas: no aplica
- Columnas del reporte (nombre → tipo → descripción):
  - id → INT AUTO\_INCREMENT PRIMARY KEY → Identificador único de la fila
  - name → VARCHAR(50) → Nombre o identificador
  - estado\_flujo → VARCHAR(100) → Estado del flujo del documento
  - docstatus → VARCHAR(5) → Estado del documento (ej. 0 = Borrador, 1 = Enviado, 2 = Cancelado)
  - cliente → VARCHAR(100) → Nombre o código del cliente
  - numero\_placa → VARCHAR(50) → Número de placa del vehículo
  - flota → VARCHAR(50) → Flota a la que pertenece
  - nombre\_del\_usuario → VARCHAR(255) → Nombre completo del usuario
  - costo\_final → VARCHAR(50) → Costo final registrado
  - fecha\_de\_registro → VARCHAR(50) → Fecha de registro
  - creation → VARCHAR(50) → Fecha de creación del registro
  - created\_date → DATETIME DEFAULT CURRENT\_TIMESTAMP → Fecha de inserción en el datawarehouse
  - status → tinyint(4) DEFAULT '1' → Estado del registro (1 = activo, 0 = inactivo)
  
- Ordenamiento por defecto: no aplica

## 7) Frecuencia y operación

- Frecuencia de actualización: Cada que se ejecuta el servicio
- Ventana que recalcula (días/meses): no aplica
- Tamaño de lote / paginado (chunking): Los inserts se hacen en lotes de 900 registros máximo.
- Tolerancia a fallos / reintentos: try catch para la inserción de datos.
- Tiempos de ejecución esperados: Dependerá de la cantidad de datos.

## 8) Calidad y controles

- Validaciones previas/post: Se valida el ingreso de la fecha\_inicio y fecha\_fin
- Controles de duplicados: no aplica
- Métricas de completitud (qué se monitorea):
  - Se monitorea la inserción de datos en la tabla. En caso no se inserte correctamente, se guarda un log en la tabla error\_reports, emp\_reportes\_validation.
  - Se crea un log con estado (1 = exitoso ó 0 = error) Que confirma el estado de la creación de la tabla.

## 9) Dependencias externas

- APIs / servicios y endpoints: APICDTALLERES
  - API: APICDTALLERES = <https://erp-cd-talleres-qa-21-02-2025.shalom.com.pe/api/>
  - Servicio: Consulta de Campos(tabMantenimiento de Terceros)
  - endpoint: method/send-query-database
  
- Autenticación / headers: Se define el siguiente header:
  - \$options = [
    - 'headers' => [
      - "Accept" => "application/json",
      - "Content-Type" => "application/json"
    - ]
  - ];
  
- Mapeos y contratos de datos:
  - Mapeo: En el punto 6 se ve el mapeo de datos, como llega del API y como se guarda en la BD:
    - ejm: "name" => "VARCHAR(50) NULL, ",
  - Contrato de datos: El API debe entregar todos los campos indicados en el punto 6.

# 10) Historial de cambios

2025-09-23 11:55 - Elian Franco Arroyo Rodas - Documentación inicial

---

Revisión #1

Creado 2025-10-24 15:28:10 -05 por Elian

Actualizado 2025-10-24 15:28:17 -05 por Elian