

# REPORTE RAKKI

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

## 1) Nombre del reporte

- Nombre: Reporte Rakki
- Código / Alias: report\_vulnerabilidades
- Responsable: Equipo de Reportes Cloud
- Propósito (1 línea):

## 2) Alcance temporal

- Campo(s) de fecha utilizados:
  - ose\_fhpreferenciapartida (fecha preferente de partida de orden)
  - fecha\_registro (fecha de registro del proceso en histórico)
  - fecha y fecha\_llegada en salida del reporte
- Tipo de rango (diario / mensual / personalizado): Mensual (últimos 2 meses calculados dinámicamente)
- Inclusión de horas (sí/no). 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\_ordenservicio
- Alias (si aplica): os
- Campos utilizados (clave → descripción):
  - ose\_termorigenatencion → origen → Terminal origen de atención.

- ose\_id → idose → Identificador único de la orden de servicio.
  - ose\_termdestinoentrega → destino → Terminal de destino o punto de entrega.
  - ose\_montofinal → monto → Monto final asociado al servicio.
  - ose\_fhpreferenciapartida → fecha de partida o preferencia de salida del servicio.
- Condiciones (WHERE) aplicadas:
    - Se consideran únicamente las órdenes de servicio activas y no eliminadas.
    - Se descartan las órdenes anuladas o dadas de baja, y también aquellas que tengan algún tipo de anulación registrada.
    - Además, solo se incluyen los registros que tengan adicionales de tipo "E".  
Los registros deben pertenecer al rango de fechas definido para la ejecución del reporte (desde la fecha inicial hasta la fecha final del mes o periodo actual).
- Tipo de unión con otras fuentes (JOIN y llave):
    - LEFT JOIN emp\_adicionales as adic ON adic.adc\_ordenservicioid = os.ose\_id
    - Llave: ose\_id (orden de servicio).
- Observaciones (ej. particiones, índices):
    - Se aplica GROUP BY os.ose\_id.
    - Alias os para tabla principal y adic para tabla adicional.
    - Los resultados se agrupan y procesan para cálculo de tiempos de entrega y repartos.

### 3.2 Conexión / Base: EMPRESARIAL

- Tabla / Recurso: emp\_terminal
  - Alias (si aplica): No aplica
  - Campos utilizados (clave → descripción):
    - ter\_id → Identificador de terminal
    - ter\_nombre → Nombre de la terminal o agencia
- Condiciones (WHERE) aplicadas:
    - No se aplican filtros (SELECT \* FROM emp\_terminal).
- Tipo de unión con otras fuentes (JOIN y llave):
    - No aplica JOIN directo; se usa posteriormente para mapear id\_agencia → ter\_nombre.

- Observaciones (ej. particiones, índices):
  - Se utiliza para asociar los datos de reparto con el nombre de la agencia correspondiente.

### 3.3 Conexión / Base: CAPACITACIÓN

- Tabla / Recurso: emp\_procesos\_historial\_app
- Alias (si aplica): No aplica
- Campos utilizados (clave → descripción):
  - id → Identificador del proceso
  - idose → Identificador de la orden de servicio
  - proceso → Tipo de proceso (ruta, desembarque, devolución, reparto)
  - fecha\_registro → Fecha de registro del evento
  - terminal → Terminal asociada al proceso
  
- Condiciones (WHERE) aplicadas:
  - Se filtran únicamente los procesos que corresponden a eventos de desembarque en el primer bloque, siempre que el registro esté activo y asociado a alguna de las órdenes previamente identificadas.
  - En el segundo bloque, se filtran los procesos que correspondan a ruta, devolución o reparto, dentro del rango de fechas indicado para el reporte.
  - Solo se consideran registros válidos (con estado activo).
  
- Tipo de unión con otras fuentes (JOIN y llave):
  - No aplica JOIN directo; se usa posteriormente para mapear id\_agencia → ter\_nombre.
  
- Observaciones (ej. particiones, índices):
  - Se aplican filtros dinámicos por fechas.
  - Se agrupan los procesos por idose para calcular los días de entrega y número de intentos.
  - Se diferencia entre bases centos y centosold según el rango de fechas (antes o después de septiembre 2023).

## 4) Filtros globales del reporte

- Inclusiones (must-have):
  - Solo órdenes activas ( $ose\_estado = 1$ ).
  - Solo adicionales de tipo E.
  - Solo procesos con  $status = 1$ .
- Exclusiones (reglas de descarte):
  - Órdenes anuladas o eliminadas.
  - Pagos con estado "AN" o "DA".
- Reglas por estado / habilitado:
  - Registros con  $status = 1$  se consideran válidos;  $status = 0$  se registra en validación.
- Parámetros externos (si el usuario puede filtrarlo):
  - No expone parámetros de entrada externos (fechas calculadas internamente).

## 5) Transformaciones y Reglas de negocio

- Derivaciones de campos (cómo se calculan):
  - $pendientes\_entrega = cantidad\_reparto - (dia\_1 + dia\_2 + dia\_3 + otros\_entrega)$
  - Cálculo de días ( $diff.days$ ) entre desembarque y reparto → clasifica entregas por día ( $dia\_1, dia\_2, etc.$ )
- Mapeos de estado:
  - $qty\_reparto = 1 \rightarrow intento\_1$
  - $qty\_reparto = 2 \rightarrow intento\_2$
  - $qty\_reparto = 3 \rightarrow intento\_3$
  - $qty\_reparto > 3 \rightarrow otros\_intentos$

- Reglas de validación (p.ej., sólo registros validados):
  - Se omiten órdenes sin desembarque o sin reparto.
  - Se ignoran procesos con  $\text{fecha\_reparto} < \text{fecha\_desembarque}$ .
- Reglas condicionales (si X entonces Y):
  - Si año = 2023 y mes  $\leq$  9 → usar conexión centosold.
  - Si tabla no existe → crear con estructura predefinida.

## 6) Estructura de salida

- Tabla/archivo destino: report\_rakki\_reparto\_YYYY\_MM
- Particionado / sufijo: Mensual (YYYY\_MM)
- Clave(s) primaria(s) o únicas: id (auto\_increment)
- Columnas del reporte (nombre → tipo → descripción):
  - fecha → DATE → Fecha de la guía o pedido
  - id\_agencia → VARCHAR(20) → Identificador del terminal o agencia
  - terminal → VARCHAR(200) → Nombre del terminal
  - fecha\_llegada → DATE → Fecha de desembarque
  - cantidad\_reparto → VARCHAR(10) → Total de repartos registrados
  - dia\_1 → VARCHAR(10) → Entregas realizadas en el día 1
  - dia\_2 → VARCHAR(10) → Entregas realizadas en el día 2
  - dia\_3 → VARCHAR(10) → Entregas realizadas en el día 3
  - otros\_dias\_entrega → VARCHAR(10) → Entregas fuera del rango de 3 días
  - intento\_1 → VARCHAR(10) → Primer intento de entrega registrado
  - intento\_2 → VARCHAR(10) → Segundo intento de entrega registrado
  - intento\_3 → VARCHAR(10) → Tercer intento de entrega registrado
  - otros\_intentos → VARCHAR(10) → Intentos de entrega adicionales (más de 3)
  - devoluciones → VARCHAR(10) → Total de devoluciones registradas
  - pendientes\_entrega → VARCHAR(10) → Entregas no completadas o pendientes
  - created\_date → DATETIME → Fecha de creación del registro
  - status → TINYINT(1) → Estado de inserción (1 = activo)
- Ordenamiento por defecto: fecha ASC

## 7) Frecuencia y operación

- Frecuencia de actualización: PENDIENTE (Esperando info de Eduardo)
- Ventana que recalcula (días/meses): 2 meses hacia atrás
- Tamaño de lote / paginado (chunking): `array_chunk(..., 900)` → Inserta de 900 en 900 registros
- Tolerancia a fallos / reintentos:
  - Inserta en `emp_reportes_validation` con `status=0` si ocurre excepción.
  - Transacciones con `rollback` (`beginTransaction`, `commit`, `rollback`).
  
- Tiempos de ejecución esperados: PENDIENTE (Esperando info de Eduardo)

## 8) Calidad y controles

- Validaciones previas/post:
  - Verificación de existencia de tabla antes de insertar.
  - Truncado si ya existe tabla (`commandTruncate`).
  
- Controles de duplicados:
  - Tabla vaciada antes de nueva inserción.
  
- Métricas de completitud (qué se monitorea):
  - Validación de registros exitosos vs. fallidos (`status` en tabla `emp_reportes_validation`).

## 9) Dependencias externas

- APIs / servicios y endpoints: No aplica (solo conexiones BD internas)
- Autenticación / headers: No aplica
- Mapeos y contratos de datos: Tablas internas del entorno ERP logístico.

# 10) Historial de cambios

2025-10-17 11:24 - Elian Franco Arroyo Rodas - Documentación inicial

---

Revisión #1

Creado 2025-10-24 15:33:02 -05 por Elian

Actualizado 2025-10-24 15:33:08 -05 por Elian