

# REPORTE SHIPPING AMOUNT PARTTOW

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

## 1) Nombre del reporte

- Nombre: Reporte Shipping Amount PartTwo
- Código / Alias: report\_shipping\_amount\_part\_two\_
- Responsable: Equipo de Reportes Cloud
- Propósito (1 línea):

## 2) Alcance temporal

- Campo(s) de fecha utilizados:
  - fecha en emp\_cierre\_aereo
  - cop\_fechaemision en emp\_comppago\_dhl
  - ose\_fhpreferenciapartida, ose\_fhpreferencial en emp\_ordenservicio
- Tipo de rango (diario / mensual / personalizado):
  - Personalizado (desde \$first\_date hasta \$second\_date)
  - Se generan registros diarios (range con DatePeriod → iteración día a día).
- 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: CENTOS

- Tabla / Recurso: emp\_cierre\_aereo
- Alias (si aplica): No aplica
- Campos utilizados (clave → descripción):
  - id → Identificador del cierre aéreo
  - cap\_id → Código de capacidad o identificador del cierre
  - general → Kilaje general del cierre
  - peligrosa → Kilaje de carga peligrosa
  - valiosa → Kilaje de carga valiosa
  - detalle\_mercaderia → Detalle del contenido transportado (JSON)
  - fecha → Fecha del registro del cierre aéreo
  
- Condiciones (WHERE) aplicadas:
  - Se filtran los registros donde el campo estado sea igual a 1 (cerrados activos) y cuya fecha esté dentro del rango definido por las variables \$finit y \$ffend, es decir, entre la fecha de inicio y fin del reporte.
  
- Tipo de unión con otras fuentes (JOIN y llave): Relaciona mediante id con cierre\_id de la tabla emp\_embarque\_aereo.
- Observaciones (ej. particiones, índices):
  - Tabla utilizada para obtener cierres aéreos con detalle de mercadería, se decodifica su campo JSON para cálculo de kilaje.

### 3.2 Conexión / Base: CENTOS

- Tabla / Recurso: emp\_embarque\_aereo
- Alias (si aplica): No aplica
- Campos utilizados (clave → descripción):
  - id → Identificador del embarque aéreo
  - ose\_id → Identificador del orden de servicio relacionada
  - terminal → Terminal donde se realizó el embarque
  - fecha → Fecha del embarque
  - cierre\_id → Identificador del cierre aéreo asociado
  
- Condiciones (WHERE) aplicadas:
  - Se seleccionan los registros cuyo cierre\_id exista dentro del conjunto de cierres aéreos previamente filtrados (array\_keys(\$cierres\_aereos\_json)).

- Tipo de unión con otras fuentes (JOIN y llave): Une con emp\_cierre\_aereo (por cierre\_id) y con emp\_ordenservicio (por ose\_id).
- Observaciones (ej. particiones, índices):
  - El resultado se agrupa por ose\_id para obtener un resumen por orden de servicio.

### 3.3 Conexión / Base: EMPRESARIAL

- Tabla / Recurso: emp\_ordenservicio
- Alias (si aplica): No aplica
- Campos utilizados (clave → descripción):
  - ose\_id → Identificador de orden de servicio
  - ose\_fhpreferenciapartida → Fecha preferencial de partida
  - ose\_fhpreferencial → Fecha alternativa de programación
  - ose\_termorigenatencion → Terminal de origen
  - ose\_termdestinoentrega → Terminal de destino
  - ose\_estadoPago → Estado del pago de la orden
  - ose\_tipoanulado → Indicador de anulación
  - ose\_remiteempresa → Empresa remitente
  - ose\_destinaempresa → Empresa destinataria
  - ose\_montofinal → Monto total del servicio
  - ose\_estadoentregado → Estado de entrega del servicio
  - ose\_remitecontactofono → Contacto de remitente
  - ose\_remitecontactomail → Correo del remitente
  - ose\_observacion\_cambioprecio → Tipo de orden de servicio
- Condiciones (WHERE) aplicadas:
  - Filtra órdenes de servicio activas (ose\_estado = 1) y no eliminadas (eliminado = 0), excluye aquellas con estados de pago "AN" (Anulado) o "DA" (Dado de baja), y valida que no tengan tipo de anulación (ose\_tipoanulado vacío).
  - Además, restringe por rango de fecha (ose\_fhpreferenciapartida o ose\_fhpreferencial entre \$finit y \$ffend), y si la variable \$terminal está definida, se filtran registros cuyo origen o destino coincidan con esa terminal.
- Tipo de unión con otras fuentes (JOIN y llave): Relaciona con emp\_embarque\_aereo mediante ose\_id, y con emp\_envio\_aereo, emp\_incidencias, emp\_adicionales y emp\_empresa en distintas etapas del procesamiento.

- Observaciones (ej. particiones, índices):
  - Es la tabla base para la mayoría de cálculos de métricas (corporativo, comercial, reparto, etc.).

### 3.4 Conexión / Base: EMPRESARIAL

- Tabla / Recurso: emp\_comppago\_dhl
- Alias (si aplica): No aplica
- Campos utilizados (clave → descripción):
  - numero\_dhl → Número de comprobante DHL
  - cop\_id → Identificador del comprobante
  - cop\_terminal → Terminal del comprobante
  - cop\_montoTotal → Monto total del comprobante
  - cop\_fechaemision → Fecha de emisión del comprobante
- Condiciones (WHERE) aplicadas:
  - Solo se consideran los comprobantes con cop\_estado\_pago distinto de "AN" (no anulados), cop\_estado = 1 (activos) y eliminado = 0.
- Tipo de unión con otras fuentes (JOIN y llave): No aplica unión directa; los datos se agregan por terminal y fecha para análisis comparativo.
- Observaciones (ej. particiones, índices):
  - Los montos y cantidades se agrupan manualmente por terminal y día.

### 3.5 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

- Condiciones (WHERE) aplicadas:
  - Se obtienen únicamente terminales habilitadas (ter\_habilitado = 1).
- Tipo de unión con otras fuentes (JOIN y llave): Se usa como catálogo para recorrer todas las terminales y generar métricas por día.
- Observaciones (ej. particiones, índices):
  - Actúa como tabla maestra de referencia.

### 3.6 Conexión / Base: EMPRESARIAL

- Tabla / Recurso: emp\_incidencias (alias inc)
- Alias (si aplica): inc
- Campos utilizados (clave → descripción):
  - inc\_id → Identificador de incidencia
  - inc.usercreaid → Usuario creador
  - inc\_tipo → Tipo de incidencia
  - inc\_fhincidencia → Fecha de incidencia
  - inc\_idos → Orden de servicio afectada
  - inc\_responsable\_terminal → Terminal responsable
  - inc\_descripcion → Descripción de la incidencia
- Condiciones (WHERE) aplicadas:
  - Se incluyen incidencias que no estén eliminadas (eliminado nulo o 0), cuyo inc\_descripcion coincida con alguno de tres mensajes predefinidos de fallas o decomisos, y que su inc\_idos pertenezca al conjunto de órdenes de servicio analizadas.
- Tipo de unión con otras fuentes (JOIN y llave): Relaciona con emp\_ordenservicio mediante inc\_idos.

- Observaciones (ej. particiones, índices):
  - Se usa para excluir órdenes con incidencias registradas.

### 3.7 Conexión / Base: EMPRESARIAL

- Tabla / Recurso: emp\_adicionales (alias adc)
- Alias (si aplica): adc
- Campos utilizados (clave → descripción):
  - adc\_ordenservicioid → ID de la orden de servicio
  - adc\_tipo → Tipo de adicional (E = entrega)
  - adc\_costo → Costo del adicional
  - eliminado → Estado lógico del registro
- Condiciones (WHERE) aplicadas:
  - Se filtran registros con adc\_tipo = 'E' (entregas) y, según el bloque, algunos con eliminado = 1.
  - Además, se limita a los registros cuyo adc\_ordenservicioid esté dentro del conjunto de órdenes seleccionadas o devueltas por la API Rakky.
- Tipo de unión con otras fuentes (JOIN y llave): Relaciona con emp\_ordenservicio por adc\_ordenservicioid.
- Observaciones (ej. particiones, índices):
  - Utilizado para identificar entregas pendientes o completadas.

### 3.8 Conexión / Base: EMPRESARIAL

- Tabla / Recurso: emp\_empresa
- Alias (si aplica): No aplica
- Campos utilizados (clave → descripción):
  - ruc → Documento de la empresa
  - grupo → Tipo de grupo (CORPORATIVO / CREDITO / COMERCIAL)
- Condiciones (WHERE) aplicadas:
  - Filtrar registros donde grupo no sea nulo ni vacío, clasificando las empresas según su tipo de grupo.

- Tipo de unión con otras fuentes (JOIN y llave): Se cruza lógicamente con remitentes y destinatarios de emp\_ordenservicio.
- Observaciones (ej. particiones, índices):
  - Define categorías de empresa para clasificar los montos de servicio.

## 4) Filtros globales del reporte

- Inclusiones (must-have):
  - Órdenes activas (ose\_estado=1), no eliminadas, no anuladas.
  - Incidencias no críticas.
  - Terminales habilitados.
- Exclusiones (reglas de descarte):
  - tipo\_os = 'Retiro de mercadería'
  - ose\_estadoPago IN ('AN','DA')
  - ose\_tipoanulado != ''
- Reglas por estado / habilitado: Solo estado=1 para registros logísticos.
- Parámetros externos (si el usuario puede filtrarlo): \$first\_date, \$second\_date, \$terminal

## 5) Transformaciones y Reglas de negocio

- Derivaciones de campos (cómo se calculan):
  - kilaje\_aereo → suma de detalle\_mercaderia.general
  - corporativo\_\* / comercial\_\* → según grupo del RUC remitente/destinatario.
  - shalom\_pro / envia\_ya → según valor de rcontacto (WEB o REGISTRA).
  - cantidad\_reparto\_pendiente → diferencia entre entregas registradas vs confirmadas Rakky.

- Mapeos de estado:
  - Terminal (ter\_id → ter\_nombre)
  - Grupo empresa (grupo → tipo\_cliente)
- Reglas de validación (p.ej., sólo registros validados): Ignorar incidencias SUNAT o faltantes.
- Reglas condicionales (si X entonces Y):
  - Si \$terminal != "TODOS" → aplica filtro específico.
  - Si detalle\_mercaderia no es JSON → inicializa vacío.

## 6) Estructura de salida

- Tabla/archivo destino: report\_shipping\_amount\_part\_two\_YYYY\_MM
- Particionado / sufijo: mensual (date("Y\_m", \$fecha))
- Clave(s) primaria(s) o únicas: fecha, terminal\_id
- Columnas del reporte (nombre → tipo → descripción):
  - fecha → DATE → Día del registro
  - terminal\_id → INT → Identificador del terminal
  - terminal\_nombre → VARCHAR(150) → Nombre del terminal
  - envia\_ya → INT → Envíos registrados por el canal Envía Ya
  - shalom\_pro → INT → Envíos realizados vía portal Shalom Pro
  - contador\_rakki → INT → Cantidad de entregas confirmadas por Rakky
  - monto\_rakki → DOUBLE → Monto total de entregas registradas por Rakky
  - cantidad\_store → INT → Cantidad de ventas en Shalom Store
  - monto\_store → DOUBLE → Monto total de ventas en Shalom Store
  - corporativo\_cantidad → INT → Cantidad de envíos de clientes corporativos
  - corporativo\_monto → DOUBLE → Monto total de envíos corporativos
  - comercial\_cantidad → INT → Cantidad de envíos de clientes comerciales
  - comercial\_monto → DOUBLE → Monto total de envíos comerciales
  - cantidad\_enviado\_aereo → INT → Total de envíos aéreos emitidos
  - monto\_enviado\_aereo → DOUBLE → Monto total de envíos aéreos emitidos
  - cantidad\_recibido\_aereo → INT → Total de envíos aéreos recibidos
  - monto\_recibido\_aereo → DOUBLE → Monto total de envíos aéreos recibidos
  - aereo\_entregado\_no → INT → Cantidad de envíos aéreos no entregados
  - kilaje\_aereo → DOUBLE → Kilaje total de envíos aéreos
  - cantidad\_dhl → INT → Cantidad de comprobantes procesados por DHL
  - kilaje\_dhl → DOUBLE → Kilaje total asociado a DHL
  - amount\_dhl → DOUBLE → Monto total asociado a DHL
  - cantidad\_reparto\_total → INT → Total de entregas locales realizadas
  - cantidad\_reparto\_pendiente → INT → Total de entregas locales pendientes

- cantidad\_envios\_terrestres\_por\_aereo → INT → Total de envíos terrestres procesados vía aéreo
- monto\_envios\_terrestres\_por\_aereo → DOUBLE → Monto total de envíos terrestres procesados vía aéreo
- created\_date → DATETIME → Fecha de generación del reporte

- Ordenamiento por defecto: por fecha ascendente y terminal\_id

## 7) Frecuencia y operación

- Frecuencia de actualización: PENDIENTE (Esperando info de Eduardo)
- Ventana que recalcula (días/meses): rango dinámico (\$first\_date → \$second\_date)
- Tamaño de lote / paginado (chunking): 900 registros por inserción (comentado)
- Tolerancia a fallos / reintentos: manejo básico de cURL error
- Tiempos de ejecución esperados: PENDIENTE (Esperando info de Eduardo)

## 8) Calidad y controles

- Validaciones previas/post:
  - JSON válido en detalle\_mercaderia
  - Rango de fechas correcto
- Controles de duplicados: Conteos por terminal consistentes con datos API.
- Métricas de completitud (qué se monitorea): Llave terminal\_id + fecha asegura unicidad.
  - Cantidades y montos por tipo (aéreo, terrestre, corporativo, comercial).

## 9) Dependencias externas

- APIs / servicios y endpoints:
  - Rakky (servicio de entregas):

- Endpoint: DIRRAKKY/api/search\_deliveries\_by\_ose\_ids\_new
- Función: Verificación de entregas efectivas según lista de órdenes de servicio (OSE).
  
- Shalom Store (servicio de ventas físicas):
  - Endpoint: RECURSOS\_HUMANOS/api/detail-shalom-store-selling
  - Función: Consulta de ventas por terminal en el rango de fechas del reporte.
  
- Bases de datos internas:
  - centos (operativa / logística aérea)
  - empresarial (operativa ERP principal)
  
- Autenticación / headers: No explícitos. Uso directo de GuzzleHttp\Client->request() y curl\_init() sin encabezados personalizados o tokens.
- Mapeos y contratos de datos:
  - Rakky: espera un cuerpo JSON con la clave ose\_ids (array de identificadores de órdenes).
    - La respuesta esperada es JSON con campo message → lista de entregas confirmadas.
  
  - Shalom Store: recibe { fh\_init, fh\_end } como cuerpo del POST.
    - Devuelve JSON con message → array de objetos que contienen terminal, ventas\_cantidad, ventas\_monto.
  
- Bases internas: devuelven conjuntos de registros (arrays asociativos) para cada consulta SQL (DB::connection()->select()).

## 10) Historial de cambios

2025-10-17 13:52 - Elian Franco Arroyo Rodas - Documentación inicial

---

Revisión #1

Creado 2025-10-24 15:32:38 -05 por Elian

Actualizado 2025-10-24 15:32:43 -05 por Elian