

REPORTE CRM PERSONA

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

1) Nombre del reporte

- **Nombre:** Reporte CRM Persona
- **Código / Alias:** report_crm_person_
- **Responsable:** Equipo de Reportes Cloud
- **Propósito (1 línea):**

2) Alcance temporal

- **Campo(s) de fecha utilizados:**
 - `ose_fhpreferenciapartida` → Fecha de partida del servicio
 - `fecha_registro` → Fecha de entrega registrada
 - Parámetros externos: `$first_date`, `$second_date`
- **Tipo de rango (diario / mensual / personalizado):**
 - **Personalizado mensual**, subdividido en 4 semanas (`obtenerSemanasMes` genera semanas S1-S4)
- **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` (o `empresarial_old` para año 2023)

Tabla / Recurso: `emp_ordenservicio`

Alias (si aplica): `os`

Campos utilizados (clave → descripción):

- `ose_id` → Identificador único del servicio.
- `ose_termdestinoentrega` → Código de destino.
- `ose_fhpreferenciapartida` → Fecha preferencial de partida.
- `ose_remitecontactofono` → Teléfono de contacto del remitente.
- `ose_remitecontactomail` → Correo de confirmación.
- `ose_observacion_cambioprecio` → Observación (tipo de orden).
- `ose_montofinal` → Monto total del servicio.
- `ose_remiteempresa` → Documento del remitente.
- `ose_servpagotipo` → Tipo de servicio.
- `eliminado` → Flag lógico (0 = activo).
- `ose_estado` → Estado del registro.

Condiciones (WHERE) aplicadas:

- `ose_estadoPago NOT IN ('DA', 'AN')`
- `ose_tipoanulado = ''`
- `eliminado = 0`
- `ose_estado = 1`
- `ose_remiteempresa IN ($document)`
- `ose_fhpreferenciapartida BETWEEN $first_date 00:00:00 AND $second_date 23:59:59`

Tipo de unión con otras fuentes (JOIN y llave):

- `LEFT JOIN emp_persona AS per ON per.per_nrodocumento = os.ose_remiteempresa`

Observaciones (ej. particiones, índices):

- Conexión cambia dinámicamente según el año (`empresarial_old` para 2023, `empresarial` para años posteriores).
- Tablas grandes, por lo que se aplica paginación en bloques de 400 documentos (`array_chunk`).

3.2 Conexión / Base: `empresarial`

Tabla / Recurso: `emp_persona`

Alias (si aplica): `per`

Campos utilizados (clave → descripción):

- `per_nrodocumento` → Documento de identidad del cliente.
- `per_nombrerzsoc` → Nombres / razón social.
- `per_apellidopaterno` / `per_apellidomaterno` → Apellidos.
- `per_tipo` → Tipo de persona (N = Natural).
- `per_habilitado` → Estado (1 = habilitado).

Condiciones (WHERE) aplicadas:

- `per_tipo = 'N'`
- `per_habilitado = 1`

Tipo de unión con otras fuentes (JOIN y llave):

- `LEFT JOIN` con `emp_ordenservicio` por documento (`per_nrodocumento = ose_remiteempresa`)

Observaciones:

- Tabla maestra de personas utilizada para formar el campo calculado `remitentenombre`.
-

3.3 Conexión / Base: `capacitacion`

Tabla / Recurso: `tabPOS Invoice`

Alias (si aplica): No aplica

Campos utilizados (clave → descripción):

- `name` → Identificador de factura POS.
- `documento` → Documento del remitente vinculado.
- `grand_total` → Importe total de la venta.

Condiciones (WHERE) aplicadas:

- `creation <= %(second_date)s`
- `creation >= %(first_date)s`
- `docstatus = 1`
- `documento IN %(documents)s`

Tipo de unión con otras fuentes (JOIN y llave):

- No se realiza unión directa; los resultados se vinculan lógicamente con el documento del remitente.

Observaciones:

- Fuente obtenida a través de API interna `method/send-query-database` (Frappe ERPNext).
 - Se utiliza para calcular indicadores de “ventas store” (`qty_store`, `amount_store`).
-

3.4 Conexión / Base: `empresarial`

Tabla / Recurso: emp_envio_aereo

Alias (si aplica): No aplica

Campos utilizados (clave → descripción):

- id → Identificador interno.
- ose_id → Identificador de orden de servicio.
- contenido → Medio de envío (AÉREO o TERRESTRE).

Condiciones (WHERE) aplicadas:

- estado = 1
- contenido != 'TERRESTRE'
- ose_id IN (\$keysAereas)

Tipo de unión con otras fuentes (JOIN y llave):

- Vinculado lógicamente con emp_ordenservicio por ose_id.

Observaciones:

- Agrupa envíos aéreos para calcular totales (qty_aereo, amount_aereo).

3.5 Conexión / Base: centos (o centosold para año 2023 y mes ? 9)

Tabla / Recurso: emp_procesos_historial_app

Alias (si aplica): No aplica

Campos utilizados (clave → descripción):

- idose → Identificador de orden de servicio.
- terminal → Terminal origen.
- fecha_registro → Fecha de registro de la entrega.
- tipo_entrega → Canal de entrega (SISTEMA o APP).

Condiciones (WHERE) aplicadas:

- proceso = 'entrega'
- nombre_metodo = 'registrar_Entrega'
- status = 1
- idose IN (\$keysEntregas)

Tipo de unión con otras fuentes (JOIN y llave):

- Lógica: se enlaza por `idose` con registros de `emp_ordenservicio`.

Observaciones:

- Fuente utilizada para determinar si la entrega fue *en oficina* o *a domicilio*.
- Los registros se ordenan por `fecha_registro ASC` para tomar el primer evento válido.

4) Filtros globales del reporte

- **Inclusiones (must-have):**

- Personas naturales habilitadas (`per_tipo='N'`, `per_habilitado='1'`)
- Estados activos (`ose_estado='1'`, `status=1`)

- **Exclusiones:**

- Destino = 74
- Tipo de orden = “Retiro de mercadería”
- EstadoPago en (“DA”, “AN”)

- **Reglas por estado / habilitado:**

- Se excluyen anulados, eliminados y no habilitados.

- **Parámetros externos:**

- `$first_date`, `$second_date` (rango de fechas personalizable)

5) Transformaciones y Reglas de negocio

- **Derivaciones de campos:**

- `remitentename = CONCAT(nombres y apellidos)`
- Agrupación por remitente → totales de montos y cantidades por día/semana
- Identificación de tipo de envío (`WEB`, `REGISTRA`, `corporativo`, `aéreo`, `terrestre`)

- **Mapeos de estado:**

- Día de la semana (en inglés → español: lunes, martes, etc.)
- Tipo de entrega (`SISTEMA` → oficina, `APP` → domicilio)

- **Reglas de validación:**

- Solo órdenes con contacto válido o confirmado
- Validación de existencia de tabla de destino antes de insertar

- **Condicionales:**

- Si `year == 2023` usar conexiones `_old`
- Si `month <= 9` usar `centosold`

6) Estructura de salida

- **Tabla/archivo destino:** `report_crm_person_YYYY_MM`
- **Particionado / sufijo:** Por mes (`YYYY_MM`)
- **Claves primarias / únicas:** `id (AUTO_INCREMENT)`
- **Columnas del reporte(nombre → tipo → descripción):**
 - periodo → VARCHAR(20) → Periodo de análisis (ej. "ENERO 2025")
 - nombre → VARCHAR(255) → Nombre del cliente o punto de venta
 - tipo_cliente → VARCHAR(50) → Clasificación del cliente (comercial, corporativo, individual, etc.)
 - documento → VARCHAR(20) → Número de documento de identificación o RUC
 - cantidad_envios → INT → Total de envíos realizados en el periodo
 - monto → DOUBLE(12,2) → Monto total asociado a los envíos
 - envia_ya → INT → Total de envíos procesados por canal Envía Ya
 - shalom_pro → INT → Total de envíos procesados vía portal Shalom Pro
 - store → INT → Total de transacciones vía Shalom Store
 - corporativo → INT → Total de envíos corporativos
 - aereo → INT → Total de envíos transportados por vía aérea
 - terrestre → INT → Total de envíos transportados por vía terrestre
 - oficina → INT → Total de entregas en oficina
 - domicilio → INT → Total de entregas a domicilio
 - s1-s4 → INT → Métricas semanales (S1=Semana 1, S2=Semana 2, etc.)
 - lunes-domingo → INT → Métricas diarias por día de la semana
 - created_date → DATETIME → Fecha y hora de creación del registro
 - status → TINYINT(1) → Estado del registro (1=activo, 0=inactivo)
-
- **Ordenamiento por defecto:** no explícito (orden de inserción)

7) Frecuencia y operación

- **Frecuencia de actualización:** PENDIENTE (Esperando info de Eduardo)
- **Ventana de recalculo:** 1 mes dividido en 4 semanas
- **Tamaño de lote:** inserciones por chunks de 900 registros
- **Tolerancia a fallos:** verificación de tabla y recreación automática
- **Tiempos esperados:** PENDIENTE (Esperando info de Eduardo)

8) Calidad y controles

- **Validaciones previas/post:**
 - Verificación de tabla (`verifyExistTable`)
 - Creación de índices: `created_date`, `status`, `documento`
- **Controles de duplicados:**
 - Limpieza (truncate) de tabla existente antes de inserción
- **Métricas de completitud:**
 - Conteo de envíos y montos totales por remitente
 - Presencia de registros por día/semana

9) Dependencias externas

- **APIs / servicios y endpoints:**
 - `dbErp` → `method/send-query-database` (ERPNext)
- **Autenticación / headers:**
 - Utiliza configuración interna de `$this->capacitacion`
- **Mapeos y contratos de datos:**
 - Entrada: filtros JSON (`filters`, `where`, `sql_query`, `tables`)
 - Salida: `{ success: bool, data: [...] }`

10) Historial de cambios

2025-10-29 17:49 - Elian Franco Arroyo Rodas - Documentación inicial

Revisión #1

Creado 2025-10-29 17:31:56 -05 por Elian

Actualizado 2025-10-29 17:49:15 -05 por Elian