

REPORTE VULNERACIONES FINANZAS

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

1) Nombre del reporte

- Nombre: Reporte Vulneraciones Finanzas
- Código / Alias: report_vulneraciones_finanzas_YYYY_MM (Tablas dinámicas)
- Responsable: Equipo de Reportes Cloud
- Propósito (1 línea):

2) Alcance temporal

- Campo(s) de fecha utilizados:
 - cop_fechaemision
 - cop_fhanulado
 - re_datesys
 - created_date
 - list_fecha
 - fecha_crea
 - fecha
- Tipo de rango (diario / mensual / personalizado): Mensual (construido dinámicamente entre first_day y last_day por mes).
- 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_code_attemp
- Alias (si aplica): No aplica
- Campos utilizados (clave → descripción):
 - ter_id → Terminal
 - user_attempt → Usuario
 - fecha → Fecha (agrupada con DATE(fecha))
 - id → Identificador único del intento (para conteo COUNT)

- Condiciones (WHERE) aplicadas: Registros cuya fecha esté entre first_date y second_date.
- Tipo de unión con otras fuentes (JOIN y llave): No aplica (solo agrupación).
- Observaciones (ej. particiones, índices): Usa groupBy sobre ter_id, usuario y fecha.

3.2 Conexión / Base: CAPACITACION (API)

- Tabla / Recurso: Solicitud de Pagos (resource/Solicitud de Pagos)
- Alias (si aplica): sp
- Campos utilizados (clave → descripción):
 - name → Identificador de registro
 - id_de_sucursal → Sucursal asociada
 - usuario_que_registro_el_pago → Usuario que registró el pago
 - fecha → Fecha de creación del registro

- Condiciones (WHERE) aplicadas:
 - concepto = "Transacción / Compensaciones"
 - fecha entre first_date y second_date
 - docstatus != 2

- Tipo de unión con otras fuentes (JOIN y llave): No aplica (se integra por lógica en código).
- Observaciones: Se limpia el campo usuario_que_registro_el_pago para remover @dominio.

3.3 Conexión / Base: EMPRESARIAL

- Tabla / Recurso: emp_cambio_precio
- Alias (si aplica): No aplica
- Campos utilizados (clave → descripción):
 - usuariocambia → Usuario que realizó el cambio

- fecha → Fecha (agrupada con DATE(fecha))
- id → Identificador único (para conteo COUNT)

- Condiciones (WHERE) aplicadas: Registros cuya fecha esté entre first_date y second_date.
- Tipo de unión con otras fuentes (JOIN y llave): No aplica.
- Observaciones: Agrupado por usuario y fecha.

3.4 Conexión / Base: EMPRESARIAL

- Tabla / Recurso: emp_terminal
- Alias (si aplica): No aplica
- Campos utilizados (clave → descripción):
 - ter_id → Terminal
 - ter_nombre → Nombre de la terminal

- Condiciones (WHERE) aplicadas: Ninguna (trae todos).
- Tipo de unión con otras fuentes (JOIN y llave): Se cruza lógicamente con Branch para asignar tipo (agente/concesionario).
- Observaciones: Se usa como catálogo para complementar datos de sucursal.

3.5 Conexión / Base: CAPACITACION (API)

- Tabla / Recurso: Branch (resource/Branch)
- Alias (si aplica): No aplica
- Campos utilizados (clave → descripción):
 - ideentificador → Identificador de sucursal
 - name → Nombre de la sucursal
 - agente → Flag si es agente
 - concesionario → Flag si es concesionario

- Condiciones (WHERE) aplicadas: Ninguna (se trae todo).
- Tipo de unión con otras fuentes (JOIN y llave): Relación lógica con emp_terminal.
- Observaciones: Se arma diccionario para marcar sucursales agente/concesionario.

3.6 Conexión / Base: EMPRESARIAL

- Tabla / Recurso: emp_usuario
- Alias (si aplica): No aplica
- Campos utilizados (clave → descripción):
 - usr_id → Identificador de usuario
 - usr_terminalid → Terminal asignada al usuario
 - usr_rol → Rol del usuario
 - usr_alias → Alias o nombre corto

- Condiciones (WHERE) aplicadas: Usuarios incluidos en array_usuario_agencia.
- Tipo de unión con otras fuentes (JOIN y llave): Se cruza por usr_id con los usuarios recogidos en otras consultas.
- Observaciones: Completa información de detalle de usuario.

4) Filtros globales del reporte

- Inclusiones (must-have): Registros activos, notas de crédito, anulaciones, reimpresiones, intentos válidos.
- Exclusiones (reglas de descarte): status = 0 o eliminado = 1.
- Reglas por estado / habilitado: Considera cop_estado = 1, cop_estado_pago = 'AN', re_status = 0.
- Parámetros externos (si el usuario puede filtrarlo): Mes/año calculados dinámicamente.

5) Transformaciones y Reglas de negocio

- Derivaciones de campos (cómo se calculan):
 - Conteo de operaciones por día y terminal (COUNT).
 - Ajuste de reimpresiones (cantidad - 1).
 - Acumulación en \$nc_list[fecha][terminal][tipo].

- Mapeos de estado: Notas de crédito, reimpressiones QR, comprobantes anulados, caja eliminada, pinpad anulados, egresos anulados, intentos sesión, anulados store.
- Reglas condicionales (si X entonces Y):
 - Si la tabla no existe → CREATE TABLE.
 - Si es nueva → ALTER TABLE con índices (fecha, created_date, status).

6) Estructura de salida

- Tabla/archivo destino: report_vulneraciones_finanzas_YYYY_MM (Tablas dinámicas)
- Particionado / sufijo: Por mes (YYYY_MM)
- Clave(s) primaria(s) o únicas: id autoincremental
- Columnas del reporte (nombre → tipo → descripción):
 - fecha → DATE → fecha del registro o movimiento
 - terminal → VARCHAR → identificador del terminal o punto de venta
 - nota_credito → VARCHAR(20) → indicador si aplica nota de crédito
 - reimpression_qr → VARCHAR(20) → flag de reimpresión de código QR
 - comprobantes_anulados → VARCHAR(20) → cantidad de comprobantes anulados
 - eliminacion_caja_diaria → VARCHAR(20) → marca de eliminación en cierre de caja
 - comprobantes_anulados_pinpad → VARCHAR(20) → comprobantes anulados en Pinpad
 - intentos_session → VARCHAR(20) → número de intentos de sesión fallidos o registrados
 - egresos_anulados → VARCHAR(20) → número de egresos anulados
 - anulados_store → VARCHAR(20) → anulaciones registradas en la tienda
 - created_date → DATETIME → fecha y hora de creación del registro
 - status → TINYINT → estado del registro (activo/inactivo)
- Ordenamiento por defecto: no definido (se inserta en bulk).

7) Frecuencia y operación

- Frecuencia de actualización: PENDIENTE (Esperando info de Eduardo)
- Ventana que recalcula (días/meses): Últimos 2 meses + ajuste si enero/febrero.
- Tamaño de lote / paginado (chunking): Inserciones en lotes de 900 registros.

- Tolerancia a fallos / reintentos: Manejo con try/catch → inserta en emp_reportes_validation con status = 0 si falla.
- Tiempos de ejecución esperados: PENDIENTE (Esperando info de Eduardo)

8) Calidad y controles

- Validaciones previas/post:
 - Verifica existencia de tabla antes de insertar.
 - Aplica índices en creación.
- Controles de duplicados: Evita duplicados mediante groupBy y having.
- Métricas de completitud (qué se monitorea): Se monitorea en emp_reportes_validation.

9) Dependencias externas

- APIs / servicios y endpoints: ERPNext (APICAPACITACION).
 - Endpoints: /resource/POS Invoice, /resource/Branch
- Autenticación / headers: No especificada (se asume token en ServiceErp).
- Mapeos y contratos de datos: Respuestas JSON → campos name, sucursal, ideificador.

10) Historial de cambios

2025-09-30 17:33 - Elian Franco Arroyo Rodas - Documentación inicial

Revisión #1

Creado 2025-10-24 15:29:48 -05 por Elian

Actualizado 2025-10-24 15:29:59 -05 por Elian