

# REPORTE CIRCUITO COMPRAS

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

## 1) Nombre del reporte

- Nombre: Reporte Circuito de Compras
- Código / Alias: report\_circuito\_Compras
- Responsable: Equipo de Reportes Cloud
- Propósito (1 línea):

## 2) Alcance temporal

- Campo(s) de fecha utilizados:
  - first\_date → Primer día del mes
  - second\_date → Último día del mes
  - created\_date → Fecha y Hora de creación
- Tipo de rango (diario / mensual / personalizado): Personalizado (entre firstDate y secondDate)
- Inclusión de horas (sí/no). Si sí: Se incluye día completo con formato [00:00:00 - 23:59:59]
- Zona horaria: Hora del servidor

## 3) Origen de datos (tablas/APIs)

3.1 Conexión / Base: ERP\_CAPACITACION

- Tabla / Recurso: Solicitud de Compras
- Alias (si aplica): No aplica
- Campos utilizados (clave → descripción):
  - name → Identificador del registro
  - docstatus → Estado del documento( 0 = Borrador, 1 = Validado , 2 = Cancelado)
  - fecha\_creacion → Fecha de creación (Editable en estado Borrador)
  - fecha\_solicitud → Fecha y hora de la creación del registro
  - usuario\_solicitud → Usuario que creó el registro
  - tipo → Opciones (General/Computo)
  - supplier → Nombre Proveedor
  - document\_purchase\_order → Identificador de Orden de compra
  
- Condiciones (WHERE) aplicadas:
  - Filtra los registros que figuren del primer día del mes hasta el último día del mes(o fecha actual en ejecución)
  
- Tipo de unión con otras fuentes (JOIN y llave): No aplica
- Observaciones (ej. particiones, índices):

### 3.2 Conexión / Base: ERP\_CAPACITACION

- Tabla / Recurso: Purchase Order
- Alias (si aplica): No aplica
- Campos utilizados (clave → descripción):
  - name → Identificador del registro
  - creation → Fecha y hora de la creación del registro
  - owner → Usuario que creo el registro
  - per\_billed → Porcentaje Facturado
  - per\_received → Porcentaje Recibido
  - id\_factura → Identificador de Registro de Factura
  
- Condiciones (WHERE) aplicadas:
  - Filtra registro identificado de de Orden de compra, que estén vinculados a una Solicitud de Compras
  
- Tipo de unión con otras fuentes (JOIN y llave): No aplica
- Observaciones (ej. particiones, índices):

### 3.3 Conexión / Base: ERP\_CAPACITACION

- Tabla / Recurso: Purchase Invoice
- Alias (si aplica): No aplica
- Campos utilizados (clave → descripción):
  - name → Identificador del registro
  - solicitud\_de\_pago → Identificador de registro de Solicitud de Pago Vinculado
  
- Condiciones (WHERE) aplicadas:
  - Filtra los registros que estén vinculados a una orden de compra por el propio Identificador de Factura de compra
  
- Tipo de unión con otras fuentes (JOIN y llave): No aplica
- Observaciones (ej. particiones, índices):

### 3.4 Conexión / Base: ERP\_CAPACITACION

- Tabla / Recurso: Solicitud de Pagos
- Alias (si aplica): No aplica
- Campos utilizados (clave → descripción):
  - name → Identificador del registro
  - usuario\_que\_registro\_el\_pago → Usuario que realizó el pago
  - fecha\_de\_pago → Fecha de pago
  
- Condiciones (WHERE) aplicadas:
  - Filtra los registros de solicitud de pagos que se encuentren vinculados a una factura de compra.
  
- Tipo de unión con otras fuentes (JOIN y llave): No aplica
- Observaciones (ej. particiones, índices):

### 3.5 Conexión / Base: ERP\_CAPACITACION

- Tabla / Recurso: Recibo de Compra
- Alias (si aplica): rcp

- Campos utilizados (clave → descripción):
  - itm.purchase\_order → Identificador de recibo de compra
  - rcp.posting\_date → Fecha de registro Recibo de compra
  - rcp.owner → Usuario que creó el registro de Recibo de compra
- Condiciones (WHERE) aplicadas:
  - Descarta todos los registros de Orden de compra que tengan Docstatus == 2 (Estado Del documento -> cancelado)
- Tipo de unión con otras fuentes (JOIN y llave):
  - Une los datos del doctype con la tabla hija que tiene LEFT JOIN `tabPurchase Receipt Item` itm on (rcp.name = itm.parent)
- Observaciones (ej. particiones, índices):

## 4) Filtros globales del reporte

- Inclusiones (must-have):
  - Incluye solo los registros de Solicitud de Compra cuya fecha de creación este entre el 1ro de Enero hasta la fecha actual
  - Incluye Ordenes de compra cuyo identificador de registro(name) este en un registro de Solicitud de Compras
  - Incluye Facturas de compras cuyo identificador de registro(name) este en un registro de Orden de Compra
- Exclusiones (reglas de descarte):
  - Descarta todos los registros de Orden de compra que tengan Docstatus == 2 (Estado Del documento -> cancelado)
- Reglas por estado / habilitado:
  - Condición de filtros por docstatus (Estado del registro del Doctype )
- Parámetros externos (si el usuario puede filtrarlo): No aplica

# 5) Transformaciones y Reglas de negocio

- Derivaciones de campos (cómo se calculan):
  - fecha\_creacion\_orden\_compra → se toma de orden\_compra\_array[\$val['document\_purchase\_order']][0]['creation'].
  - usuario\_orden\_compra → se toma de owner en la orden de compra.
  - porcentaje\_facturado → se deriva de per\_billed de la orden de compra.
  - porcentaje\_recibido → se deriva de per\_received.
  - fecha\_pago y usuario\_pago → se obtienen cruzando factura → solicitud de pago.
  - fecha\_entrega y usuario\_entrega → se calculan desde tabla\_purchase\_list.
  
- Mapeos de estado:
  - Solicitud de Compras (search\_solicitud\_compra) se mapea con Purchase Order (orden\_compra).
  - Purchase Order se mapea con Purchase Invoice (factura\_compra).
  - Purchase Invoice se mapea con Solicitud de Pagos.
  - Purchase Receipt se mapea con Purchase Order para obtener datos de entrega.
  
- Reglas de validación (p.ej., sólo registros validados):
  - Solo se incluyen órdenes de compra con id\_factura existente (if (isset(\$val['id\_factura']))).
  - En recibos de compra (tabla\_purchase), se descartan registros con docstatus = 2 (rcp.docstatus != %(docstatus)s).
  
- Si una orden de compra no existe en orden\_compra\_array, se inicializan valores en 0 o vacío (validación de fallback).
  
- Reglas condicionales (si X entonces Y):
  - Si existe document\_purchase\_order válido en una solicitud de compra, entonces se enriquecen los campos con datos de la orden, factura y pagos.

- Si no existe → se rellenan campos con 0 o ''.
- Si la factura tiene solicitud de pago asociada → se obtienen fecha\_pago y usuario\_pago.
- Si no la tiene → se dejan vacíos.
- Si la orden tiene recibos de compra → se agregan fecha\_entrega y usuario\_entrega.
- Si no tiene → se dejan vacíos.

## 6) Estructura de salida

- Tabla/archivo destino: report\_circuito\_Compras
- Particionado / sufijo:
  - Sufijo: report\_circuito\_Compras + Y + M
  - Particion: Cada mes tiene su propia tabla con los registros del período de fechas procesado.
- Clave(s) primaria(s) o únicas: id autoincrementable
- Columnas del reporte (nombre → tipo → descripción):
  - id →INT→Identificador único autoincremental (PK).
  - name→VARCHAR(250)→Nombre del documento (Solicitud de compra / factura / etc.).
  - docstatus→VARCHAR(250)→Estado interno del documento en ERPNext (0 = Borrador, 1 = Confirmado, 2 = Cancelado).
  - estado →VARCHAR(250)→Estado de la solicitud de compras (campo del DocType).

- fecha\_cracion → VARCHAR(250)→Fecha de creación original del registro. ( tiene un typo, debería ser "creación").
  - fecha\_solicitud→VARCHAR(250)→Fecha de la solicitud (alias de creation).
  - usuario\_solicitud→VARCHAR(250)→Usuario que registró la solicitud.
  - tipo→VARCHAR(250)→Tipo de solicitud / compra.
  - supplier→VARCHAR(250)→Proveedor asociado.
  - document\_purchase\_order→VARCHAR(250)→ID/Name de la orden de compra vinculada.
  - fecha\_creacion\_orden\_compra→VARCHAR(250)→Fecha de creación de la orden de compra.
  - usuario\_orden\_compra→VARCHAR(250)→Usuario que generó la orden de compra.
  - porcentaje\_facturado→VARCHAR(250)→Avance de facturación de la orden de compra (en %).
  - porcentaje\_recibido→VARCHAR(250)→Avance de recepción de la orden de compra (en %).
  - fecha\_entrega →VARCHAR(250)→Fecha de entrega del recibo de compra.
  - usuario\_entrega→VARCHAR(250)→Usuario que registró la entrega.
  - fecha\_pago→VARCHAR(250)→Fecha en que se registró el pago.
  - usuario\_pago→VARCHAR(250)→Usuario que registró el pago.
  - created\_date→DATETIME→Fecha de inserción en la tabla de reporte (timestamp del ETL).
  - status→tinyint(4)→Estado interno de la fila en el reporte (por defecto 1 = habilitado).
  -
- Ordenamiento por defecto: No se especifica ORDER BY

## 7) Frecuencia y operación

- Frecuencia de actualización: No especifica( Posiblemente se ejecute por CRON)
- Ventana que recalcula (días/meses):
  - Si es Enero o Febrero toma los dos meses. En caso sea cualquier otro mes le resta 2 meses al mes actual para tomar la fecha de inicio.
    - `$first_month = in_array($last_month, [1, 2]) ? 1 : intval(date("m", strtotime(date("Y-m-d") . "-2 month")));`
- Tamaño de lote / paginado (chunking): Procesa en lotes de 900 registros.
- Tolerancia a fallos / reintentos:

- El proceso usa try-catch y maneja transacciones
- Si ocurre un error durante la creación/inserción → se hace rollback y no se guarda información corrupta.
- Hay un registro de errores en tabla error\_reports

- Tiempos de ejecución esperados: No especifica

## 8) Calidad y controles

- Validaciones previas/post:
  - Verifica que exista la tabla y si no existe la crea antes de insertar los datos
    - `$verify = $this->verifyExistTable($db_name, $sql);`
  - Tras insertar los datos, se confirma con `commit()`.
  - Si falla algo en medio → se hace `rollback()` para evitar datos inconsistentes.
- Controles de duplicados: Se escribe una nueva tabla nueva por cada ejecución: `$db = "report_circuito_Compras_" . date("Y_m", strtotime($first_date_sin_format));`
- Métricas de completitud (qué se monitorea): Guarda un registro de validacion en la tabla "emp\_reportes\_validation"
  - tabla: `$insert = $this->centos->table("emp_reportes_validation")->insert($body);`

## 9) Dependencias externas

- APIs / servicios y endpoints:
  - `resource/Solicitud de Compras` → para obtener las solicitudes de compra.
  - `resource/Purchase Order` → para obtener órdenes de compra asociadas.
  - `resource/Purchase Invoice` → para obtener facturas de compra.
  - `resource/Solicitud de Pagos` → para obtener solicitudes de pago.
  - `method/send-query-database` → para ejecutar consultas SQL personalizadas contra tablas internas (`tabPurchase Receipt`, `tabPurchase Receipt Item`).

- Autenticación / headers: Es probable que la autenticación se maneje dentro de ServiceErp o dbErp
- Mapeos y contratos de datos: JSON estructurado en filters, fields, response

# 10) Historial de cambios

2025-09-18 13:00 - Elian Franco Arroyo Rodas - Documentación inicial

---

Revisión #1

Creado 2025-10-24 15:26:05 -05 por Elian

Actualizado 2025-10-24 15:26:12 -05 por Elian