

REPORTE SOLICITUD DE MATERIALES

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

1) Nombre del reporte

- Nombre: Reporte de Solicitud de Materiales
- Código / Alias: report_solicitud_materiales
- Responsable: Equipo de Reportes Cloud
- Propósito (1 línea):

2) Alcance temporal

- Campo(s) de fecha utilizados:
 - fecha_inicio
 - fecha_fin
- Tipo de rango (diario / mensual / personalizado): Mensual
- Inclusión de horas (sí/no). Si sí: [00:00:00 - 23:59:59]
- Zona horaria: La del servidor

3) Origen de datos (tablas/APIs)

- Conexión / Base: ERP CAPACITACION
- Tabla / Recurso: tabMaterial Request
- Alias (si aplica): mtr
- Campos utilizados (clave → descripción):
 - material_request_type → Tipo de Proposito
 - propósito_de_transferencia → Tipo de Propósito de Transferencia
 - fecha_y_hora_de_transferencia → Fecha y hora de transferencia
 - status → Estado de Transferencia

- fecha_y_hora_de_transaccion_erp → Fecha de la Transacción
- confirmar_transferencia → Confirmación de transferencia
- numero_orden → Número de Guia asociada
- aprobador_por_ → Usuario que aprobó la transferencia
- aprobado_nombre_completo → Nombre completo del usuario que aprobó la transferencia
- aprobacion_de_solicitud → Tipo de Aprobación
- cancelado_por → Usuario que cancelo la solicitud
- cancelado_nombre_completo → Nombre completo del usuario que cancelo la solicitud
- detalle_cancelacion → Detalle de la cancelación
- id_sucursal → ID de la terminal que envía el producto (Origen)
- zona_nacional → Zona nacional a la que pertenece la terminal de Origen
- sucursal → Nombre de la terminal Origen
- division → Indica si la terminal Origen pertenece a Lima o Provincia
- departamento → Departamento(Area) de shalom
- per_ordered → Indica un porcentaje
- ultima_actualizacion_el → Fecha y hora en que se actualizó el registro por última vez
- ultima_actualizacion_por → Usuario que actualizó el registro por última vez
- owner → Usuario que creó el registro
- creation → Fecha de Creacion del registro
- item_code → Código del producto
- item_name → Nombre del producto
- stock_uom → Unidad de medida utilizada en el almacén
- qty → Cantidad
- puesto → Puesto del Empleado
- sexo → Sexo del Empleado
- nombre_completo_empleado → Nombre completo Empleado
- stock_qty → Cantidad existente
- actual_qty → Cantidad actual
- no_guia_as_estado_de_orden → Estado de la Guia

- Condiciones (WHERE) aplicadas:

- Filtra el registro, si la fecha de creación están entre los parámetros \$fecha_inicio y \$fecha_fin

- Tipo de unión con otras fuentes (JOIN y llave): Une la tabla tabMaterial Request y tabMaterial Request Item

- Observaciones (ej. particiones, índices): no se encontraron el valor exacto de actual_qty, stock_qty y per_ordered

4) Filtros globales del reporte

- Inclusiones (must-have):
 - Depende de lo que reciba dateUnlimited y el rango de fecha (first_day y last_day)
- Exclusiones (reglas de descarte):
 - Si numero_orden es null, se descarta del reporte
 - Si la API de estados devuelve un error o está vacía, el flujo corta y retorna vacío o la data sin procesar.
- Reglas por estado / habilitado:
 - mtr.status: Permite clasificar los registros en función de su estado.
 - obtain_estados: Sí retorna la api de NEWWEBSERVICES validado, se sobrescribe, si no queda vacío.
- Parámetros externos (si el usuario puede filtrarlo): Sí, el usuario puede incluir la fecha inicio y fecha fin en el rango de búsqueda.

5) Transformaciones y Reglas de negocio

- Derivaciones de campos (cómo se calculan):
 - usuario_creatransferencia se deriva de \$userTransfers y \$userGroups: Si existe el usuario dueño de la transferencia, se asigna su full_name, si no, queda vacío.
 - estado_de_orden Se deriva a partir de la consulta externa a otro servicio (buscar-estados-guia).
- Mapeos de estado:

- mtr.status as estado Se mapea directamente desde la tabla tabMaterial Request.
 - estado_de_orden Se mapea desde el servicio externo (buscar-estados-guia)
 - En la función baseReport, si la tabla es nueva se agrega un índice sobre status, reforzando que este campo es clave para consultas.
- Reglas de validación (p.ej., sólo registros validados):
 - Si no se ingresa un rango de fecha se retorna el siguiente mensaje: "Los campos fecha inicio y fecha fin son obligatorios"
 - Si la tabla destino no se creo correctamente se inserta un registro en error_reports
- Reglas condicionales (si X entonces Y):
 - Si no hay numero_orden, se salta ese registro
 - Si \$dateUnlimited == "TODOS", "12", "8" o cualquier otro valor → se construyen los rangos de fechas de forma distinta:
 - "TODOS" → desde el 2021 hasta el año actual.
 - "12" → desde un pivote hasta la fecha actual mes a mes.
 - "8" → últimos 8 meses.
 - Otro → mes actual y, en caso de enero/febrero, incluye diciembre del año anterior.
 - Si la tabla es nueva, se aplica ALTER TABLE para agregar índice sobre status.
 - Si ocurre algún error en la transacción se elimina la inserción de la tabla y se retorna un mensaje.

6) Estructura de salida

- Tabla/archivo destino: report_solicitud_materiales
- Particionado / sufijo: report_solicitud_materiales_YYYY_MM
- Clave(s) primaria(s) o únicas: name de la tabla "tabMaterial Request"
- Columnas del reporte (nombre → tipo → descripción):
 - id → INT AUTO_INCREMENT → Identificador único de la fila (PK interna)
 - name → VARCHAR(50) → ID del documento Material Request en ERPNext
 - material_request_type → VARCHAR(100) → Tipo de solicitud de material

- propósito_de_transferencia → VARCHAR(255) → Propósito de la transferencia
- fecha_y_hora_de_transferencia → VARCHAR(100) → Fecha/hora de la transferencia
- estado → VARCHAR(50) → Estado de la solicitud (ej. Draft, Submitted, Cancelled)
- fecha_y_hora_de_transaccion_erp → VARCHAR(100) → Fecha/hora de creación en ERP
- confirmar_transferencia → VARCHAR(50) → Indicador de confirmación
- numero_orden → VARCHAR(100) → Número de orden asociado
- aprobador_por → VARCHAR(100) → Usuario que aprobó
- aprobado_nombre_completo → VARCHAR(255) → Nombre completo del aprobador
- aprobacion_de_solicitud → VARCHAR(50) → Estado de la aprobación
- cancelado_por → VARCHAR(100) → Usuario que canceló
- cancelado_nombre_completo → VARCHAR(255) → Nombre completo del que canceló
- detalle_cancelacion → VARCHAR(500) → Motivo o detalle de cancelación
- id_sucursal → VARCHAR(20) → Código de la sucursal
- zona_nacional → VARCHAR(50) → Zona geográfica
- sucursal → VARCHAR(50) → Nombre de sucursal
- division → VARCHAR(50) → División interna
- departamento → VARCHAR(50) → Departamento
- per_ordered → INT → Cantidad pedida
- owner → VARCHAR(100) → Usuario creador del documento
- creation → VARCHAR(255) → Fecha de creación del documento
- item_code → VARCHAR(100) → Código del ítem
- item_name → VARCHAR(100) → Nombre del ítem
- stock_uom → VARCHAR(100) → Unidad de medida
- cantidad → INT → Cantidad solicitada
- puesto → VARCHAR(100) → Puesto del empleado asociado
- sexo → VARCHAR(50) → Sexo del empleado
- nombre_completo_empleado → VARCHAR(255) → Nombre completo del empleado
- stock_qty → INT → Cantidad en stock
- actual_qty → INT → Cantidad actual en almacén
- ultima_actualizacion_el → VARCHAR(255) → Fecha de última actualización
- ultima_actualizacion_por → VARCHAR(255) → Usuario que realizó la última actualización
- estado_de_orden → VARCHAR(255) → Estado de la orden (devuelto de API externa `buscar-estados-guia`)
- usuario_creatransferencia → VARCHAR(255) → Nombre completo del usuario que creó la transferencia
- created_date → DATETIME → Fecha de inserción en el datawarehouse
- status → tinyint(4) → Estado de la carga (1 = OK, 0 = error)

- Ordenamiento por defecto: No se especifica

7) Frecuencia y operación

- Frecuencia de actualización: Probablemente mensual(Por Cron)
- Ventana que recalcula (días/meses): Calcula en un rango de 8 meses consecutivos tomando el 1mer y ultimo dia de cada mes.
- Tamaño de lote / paginado (chunking): Los inserts se hacen en lotes de 900 registros máximo.
- Tolerancia a fallos / reintentos: try catch para la inserción de datos.
- Tiempos de ejecución esperados: Dependerá de la cantidad de datos.

8) Calidad y controles

- Validaciones previas/post:
 - Previas:
 - Se valida que \$fecha_inicio y \$fecha_fin existan, de lo contrario retorna un mensaje de error
 - Se controla la existencia de la tabla con verifyExistTable() antes de insertar los datos
 - Se encapsula la inserción de datos en transacciones con beginTransaction(), commit(), rollBack()
 - Post:
 - Si hay error en la consulta HTTP (\$guzzle->request), se atrapa con catch (\Exception \$e) y se retorna un mensaje claro.
 - En caso de error de base de datos (catch (\PDOException \$e)), se registra en la tabla emp_reportes_validation y se hace rollBack() para no dejar datos inconsistentes
- Controles de duplicados: Indirecta
 - Creación de Tablas por sufijo
 - Se usa CREATE TABLE {\$db_name} LIKE report_nota_entrega; → las tablas

nuevas se generan limpias

- Métricas de completitud (qué se monitorea):
 - error_reports → cuando falla la creación de tablas.
- emp_reportes_validation → cuando ocurre un error de inserción en la carga de datos.
- Cada registro de ejecución guarda report, first_date, second_date, created_date, y status, lo cual permite monitorear

9) Dependencias externas

ERPNext - CAPACITACION

- APIs / servicios y endpoints: method/send-query-database
- Autenticación / headers: Se envían headers básicos:
 - \$options = [
 - 'headers' => [
 - "Accept" => "application/json",
 - "Content-Type" => "application/json"
 -]
 -];
- No aparece un Authorization: Bearer <token> ni otra forma de autenticación explícita.
- Mapeos y contratos de datos:
 - Entrada: filtros por fecha_ini, fecha_fin.
 - Transformación: mapeo SQL → PHP arrays.
 - Persistencia: creación de tablas dinámicas en MySQL con tipos definidos.
 - Salida: array estructurado con campos enriquecidos (usuario_creatransferencia, estado_de_orden).

10) Historial de cambios

Revisión #1

Creado 2025-10-24 15:27:40 -05 por Elian

Actualizado 2025-10-24 15:27:45 -05 por Elian