

TABLAS EN LAS QUE CONSULTAS:

[USERLBL](#)

bs_mor_prco_pedidored_co

bs_cob_tick_coll_tickets

contract_all

contr_services

mdsrrtab

bs_mor_prco_pedidored_co

contract_all

curr_co_status

bs_cli_cust_customerprofile

bs_cli_prfl_profiles

[bs_mor_pdrd_pedidored](#)

FEES

TABLAS EN LAS QUE MODIFICA:

[bs_mor_rpdr_reentry_pedidored](#)

FEES (Cargos administrativos / financieros que deben ser cargados a un cliente)

[bs_mor_prco_pedidored_co](#)

[bs_mor_pdrd_pedidored](#)

sms_autogestion

[bs_mor_clia_accion](#)

1. Se obtiene la cantidad de hilos y de bloques, que se desea ejecutar el proceso configuran previamente configurados en la tabla userlbl:
 - a. Corre con 20 hilos
 - b. Bloques de a 500 registros.
2. Depura reconexiones reentrantes (DELETE en [bs_mor_rpdr_reentry_pedidored](#))
3. Si es HOT_DESACTIVACION Ejecuta CURSOR C_CONTRATOSEJECUTADOS
4. Si es HOT_BAJA (bs_mor_prco_pedidored_co, contract_all, curr_co_status)
5. Obtiene los recargos por conexión Busca los clientes a los que todavía no se les cobro el cargo de conexión filtra por estado (pdrd_cargo_conexion, 'S'). Si tiene X es por que ya se le cobró el cargo de conexión.

6. Obtiene el área a la que corresponde el cliente, dependiendo del área que sea el cliente es el monto que se estará cobrando por reconexión, Diferentes áreas: (NORTE, SUR, AMBA).
7. **PKG_MOROSIDAD.GenerarCargoConexion:**
 - a. *Selecciona el Maximo secuencial de Fees de el Responsable de Pago*
 - b. **INSERT en la FEES**
8. Actualiza y marca a los clientes que ya se les cobro el cargo con una 'X' (UPDATE [bs_mor_pdrd_pedidored](#)).
9. Actualiza los contratos ya ejecutados ([bs_mor_prco_pedidored_co](#)).
10. Si es una Desactivacion 'd' o sea una Rehabilitacion y si corresponde a que lo realice el DFR entonces ejecuta función **DCE_efnActualizarSMSAutogestion:**
 - a. Buscar el RP del Contrato
 - b. Buscar el tick_case_id del IPago
 - c. Si no tiene Tick_case_id busca el numero de línea del contrato.
11. Actualiza sms_autogestion (UPDATE) Si actualize ok ejecuta el PKG para actualizar el ticket en VAN: **PKG_INTEGRACION_VANTIVE.PMO_Cerrar_Caso_Van_Simple**
12. **PKG_INTEGRACION_VANTIVE.PMO_ACT_CERRADO_INF** (UPDATE)
13. Actualizo los pedidos con nueva cantidad de ejecutados en la tabla de pedidos (UPDATE [bs_mor_pdrd_pedidored](#)).
14. Pregunto si la cantidad de ejecutados es igual a la de enviados, de ser así ejecuta: **PKG_MOROSIDAD.PMO_PEDIDO_CONCLUIDO:**
 - a. Si es HOT 1 o 2 entonces actualiza el paso en 3 (ejecutado) en la [bs_more_clia_accion](#)
15. Verifico que el estado de la mora no este suspendido y si el usuario último que lo tocó es del DRZ
 - a. Si el estado de Mora NO está suspendido Cierro el paso actual en 4 ([bs_mor_clia_accion](#))
 - b. Actualizo la [bs_mor_clhp_hproc](#) al ultimo registro con estado 'n'
 - c. **INSERT en la [bs_mor_clhp_hproc](#) con el nuevo estado de mora y en el ultimo campo pongo en 's'**
 - d. **Obtengo el numero del cliente y el estado de cierre de la mora para actualizar los datos en la coll ticket indicando el motivo por el cual se esta cerrando por ejemplo: Monto de deuda pago:**
PKG_INTEGRACION_VANTIVE.PMO_INS_CASO_VANTIVE_SIMPLE(CIERRE_PROC ESO_MORA,Cliente,NULL,Comentario,Usuario);
16. El proceso actualiza la tabla indicando los clientes que fueron procesados ok [bs_mor_rpdr_reentry_pedidored](#)