-- Function: public.fc\_calcula\_pago\_credito\_abierto(bigint, double precision, character varying)
-- DROP FUNCTION public.fc\_calcula\_pago\_credito\_abierto(bigint, double precision, character varying);
CREATE OR REPLACE FUNCTION public.fc\_calcula\_pago\_credito\_abierto(
codigo\_prestamo bigint,
valor\_abono double precision,
tipo\_pago character varying)
RETURNS bigint AS
$BODY$
DECLARE
f\_prestamo\_id ALIAS FOR $1;
f\_valor\_abono ALIAS FOR $2;
f\_tipo\_pago ALIAS FOR $3;
s\_query VARCHAR;
result RECORD;
r\_record RECORD;
cteccodigo1 bigint;
cod\_prestamo integer;
-- f\_prestamo\_id integer;
f\_valor\_prestamo double precision;
f\_tasa double precision;
f\_tipo\_prestamo varchar;
f\_tiempo\_cobro varchar;
f\_cantidad\_cuotas\_pagar integer;
f\_valor\_cuota\_pagar double precision;
f\_fecha\_prestamo date;
f\_fecha\_inicio\_prestamo date;
f\_fecha\_proximo\_cobro date;
f\_valor\_total\_deuda double precision;
f\_valor\_abono\_deuda double precision;
f\_valor\_proximo\_pago\_deuda double precision;
f\_estado varchar;
f\_user\_id integer;
f\_cliente\_id integer;
f\_cuotas\_pagadas\_credito\_cerrado integer;
f\_cuotas\_pagadas\_credito\_abierto integer;
f\_cuotas\_por\_pagar integer;
f\_valor\_total\_deuda\_nuevo double precision;
f\_valor\_abono\_deuda\_nuevo double precision;
f\_proximovalor\_cancelar double precision;
f\_dias\_cobro integer;
f\_valor\_total\_cuotapagar\_abierto double precision;
f\_dias\_dividir\_interes integer;
f\_valor\_cuota\_actual double precision;
f\_prestamo\_interes\_abierto double precision;
f\_deuda\_real\_restante\_mes double precision;
f\_valor\_abono\_sin\_primera\_cuota double precision;
f\_cuotas\_pagadas\_en\_abono\_actual integer;
f\_valor\_restante\_pago double precision;
f\_prestamo\_estado\_pago varchar;
f\_prestamo\_estado varchar;
f\_prestamo\_abono\_bandera bigint;
f\_abono\_capital\_abonado\_abierto double precision;
BEGIN
f\_prestamo\_abono\_bandera = 0;
f\_abono\_capital\_abonado\_abierto = 0;
IF f\_prestamo\_id IS NOT NULL THEN
s\_query='SELECT id, prestamo\_valor, prestamo\_tasa, prestamo\_tipo, prestamo\_tiempo\_cobro, prestamo\_numero\_cuotas,
prestamo\_valor\_cuota, prestamo\_fecha, prestamo\_fecha\_inicial, prestamo\_fecha\_proximo\_cobro,
prestamo\_valor\_actual, prestamo\_valor\_abonado, prestamo\_valor\_proxima\_cuota,
prestamo\_estado, user\_id, cliente\_id, created\_at, updated\_at,
CASE WHEN prestamo\_cantidad\_cuota\_pagada\_creditocerrado IS NULL THEN 0
ELSE prestamo\_cantidad\_cuota\_pagada\_creditocerrado END AS prestamo\_cantidad\_cuota\_pagada\_creditocerrado ,
CASE WHEN prestamo\_cantidad\_cuota\_pagada\_creditoabierto IS NULL THEN 0
ELSE prestamo\_cantidad\_cuota\_pagada\_creditoabierto END AS prestamo\_cantidad\_cuota\_pagada\_creditoabierto,
((prestamo\_valor\_actual \* prestamo\_tasa)/100) as prestamo\_interes\_abierto,prestamo\_estado\_pago
FROM prestamos
WHERE id='||f\_prestamo\_id||' ';
--raise notice'cantidad%',s\_query;
FOR r\_record IN EXECUTE s\_query LOOP
--f\_prestamo\_id =r\_record.id;
f\_valor\_prestamo =r\_record.prestamo\_valor;
f\_tasa =r\_record.prestamo\_tasa;
f\_tipo\_prestamo =r\_record.prestamo\_tipo;
f\_tiempo\_cobro =r\_record.prestamo\_tiempo\_cobro;
f\_cantidad\_cuotas\_pagar =r\_record.prestamo\_numero\_cuotas;
f\_valor\_cuota\_pagar =r\_record.prestamo\_valor\_cuota;
f\_fecha\_prestamo =r\_record.prestamo\_fecha;
f\_fecha\_inicio\_prestamo =r\_record.prestamo\_fecha\_inicial;
f\_fecha\_proximo\_cobro =r\_record.prestamo\_fecha\_proximo\_cobro;
f\_valor\_total\_deuda =r\_record.prestamo\_valor\_actual;
f\_valor\_abono\_deuda =r\_record.prestamo\_valor\_abonado;
f\_valor\_proximo\_pago\_deuda =r\_record.prestamo\_valor\_proxima\_cuota;
f\_estado =r\_record.prestamo\_estado;
f\_user\_id =r\_record.user\_id;
f\_cliente\_id =r\_record.cliente\_id;
f\_cuotas\_pagadas\_credito\_cerrado =r\_record.prestamo\_cantidad\_cuota\_pagada\_creditocerrado;
f\_cuotas\_pagadas\_credito\_abierto =r\_record.prestamo\_cantidad\_cuota\_pagada\_creditoabierto;
f\_prestamo\_interes\_abierto =r\_record.prestamo\_interes\_abierto;
f\_prestamo\_estado\_pago =r\_record.prestamo\_estado\_pago;
/\* SE CALCULA EN NUMERO LA CANTIDAD DE DIAS A SUMAR A LA PROFIMA FECHA DE PAGO\*/
IF f\_tiempo\_cobro = 'SEMANAL' THEN
f\_dias\_cobro = 7;
f\_dias\_dividir\_interes = 4;
ELSIF f\_tiempo\_cobro = 'QUINCENAL' THEN
f\_dias\_cobro = 15;
f\_dias\_dividir\_interes = 2;
ELSIF f\_tiempo\_cobro = 'MENSUAL' THEN
f\_dias\_cobro = 30;
f\_dias\_dividir\_interes = 1;
END IF;
/\* INICIO PRESTAMO ABIERTO\*/
IF f\_tipo\_prestamo = 'ABIERTO' THEN
IF tipo\_pago ='PAGO CUOTA' THEN
f\_valor\_total\_deuda\_nuevo = f\_valor\_total\_deuda;
f\_valor\_abono\_deuda\_nuevo = f\_valor\_abono\_deuda;
-- SE CALCULA EL PAGO TOTAL MENSUAL
/\* esto se realiza de la siguiente manera, ya que la cuota a pagar siempre no va a ser del misma del pago generalmente,
entonces a las cuotas a pagar se le resta una,y las cuotas restantes se multiplican por el valor promedio a pagar,
y a este valor promedio se le suma la cuota a pagar, asi no habria diferencia de cuotas \*/
f\_valor\_cuota\_actual = round(f\_prestamo\_interes\_abierto / f\_dias\_dividir\_interes);
f\_valor\_total\_cuotapagar\_abierto = round( f\_valor\_proximo\_pago\_deuda + ( f\_valor\_cuota\_actual \* ( f\_cantidad\_cuotas\_pagar -1)));
-- SE CALCULA LAS CUOTAS RESTANTES PARA DIVIDIR
f\_cuotas\_pagadas\_credito\_abierto = f\_cuotas\_pagadas\_credito\_abierto + 1;
f\_cuotas\_por\_pagar = (f\_dias\_dividir\_interes - f\_cuotas\_pagadas\_credito\_abierto);
/\*se calcula la deuda real restante\*/
f\_deuda\_real\_restante\_mes = round(f\_valor\_proximo\_pago\_deuda + (f\_valor\_cuota\_actual \* f\_cuotas\_por\_pagar));
IF f\_valor\_abono <= f\_valor\_proximo\_pago\_deuda THEN
IF f\_valor\_abono < f\_valor\_proximo\_pago\_deuda THEN
f\_prestamo\_abono\_bandera = 1;
END IF;
--se resta el valor pagado contra el valor de la cuota a paga
--f\_valor\_restante\_pago = f\_valor\_proximo\_pago\_deuda - f\_valor\_abono;
--se determina el estado del pago
IF f\_valor\_restante\_pago = 0 THEN
f\_prestamo\_estado\_pago='AZUL';
ELSIF f\_valor\_restante\_pago > 0 THEN
f\_prestamo\_estado\_pago='MORADO';
END IF;
/\*se calcula la proxima cuota a pagar \*/
f\_proximovalor\_cancelar = round(f\_valor\_cuota\_actual + (f\_valor\_proximo\_pago\_deuda - f\_valor\_abono));
ELSIF f\_valor\_abono > f\_valor\_proximo\_pago\_deuda THEN
/\*raise notice'ingreseee%',f\_deuda\_real\_restante\_mes;
raise notice'deuda real restante mes%',f\_deuda\_real\_restante\_mes;\*/
f\_prestamo\_estado\_pago='AZUL';
/\* se calcula si la primera cuota pagada supera el valor total de la cuota mensual o no\*/
IF f\_valor\_abono > f\_deuda\_real\_restante\_mes THEN
/\* se calcula el nuevo valor total y valor abono de la deuda\*/
f\_valor\_total\_deuda\_nuevo = round(f\_valor\_total\_deuda - (f\_valor\_abono - f\_deuda\_real\_restante\_mes)) ;
f\_valor\_abono\_deuda\_nuevo = round(f\_valor\_abono\_deuda + (f\_valor\_abono - f\_deuda\_real\_restante\_mes));
/\* se guarda el valor que esta abonando a capital, para ser evidenciado en el visor utilidad\*/
f\_abono\_capital\_abonado\_abierto = round(f\_valor\_abono - f\_deuda\_real\_restante\_mes);
/\* se calcula el nuevo interes\*/
f\_proximovalor\_cancelar = round(((f\_valor\_total\_deuda\_nuevo \* f\_tasa)/100)/f\_dias\_dividir\_interes);
/\* aqui se determina los dias a sumar para la proxima fecha de pago \*/
f\_dias\_cobro = f\_dias\_cobro \* (f\_cuotas\_por\_pagar + 1);
/\* despues de realizar el calculo del valor apagar se actualiza, el numero de cuotas realmente pagadas,
ya que si ingresa por esta condicion estaria pagando el total de las cuotas del mes\*/
f\_cuotas\_pagadas\_credito\_abierto = f\_cantidad\_cuotas\_pagar;
ELSIF f\_valor\_abono < f\_deuda\_real\_restante\_mes THEN
/\* se calcula el valor de la segunda cuota \*/
f\_valor\_abono\_sin\_primera\_cuota = round(f\_valor\_abono - f\_valor\_proximo\_pago\_deuda);
/\*aqui se mira cuantas cuotas en total esta pagando en el abono \*/
f\_cuotas\_pagadas\_en\_abono\_actual = floor(f\_valor\_abono\_sin\_primera\_cuota / f\_valor\_cuota\_actual );
/\* aqui se calcula el proximo valor de la cuota\*/
f\_proximovalor\_cancelar = round((f\_valor\_cuota\_actual \* ((f\_cuotas\_pagadas\_en\_abono\_actual) + 1)) - f\_valor\_abono\_sin\_primera\_cuota );
/\* aqui se determina los dias a sumar para la proxima fecha de pago,mas la cuota que se esta pagando \*/
f\_dias\_cobro = f\_dias\_cobro \* (f\_cuotas\_pagadas\_en\_abono\_actual + 1);
/\* despues de realizar el calculo del valor apagar se actualiza, el numero de cuotas realmente pagadas,
ya que si ingresa por esta condicion estaria pagando el total de las cuotas del mes\*/
f\_cuotas\_pagadas\_credito\_abierto = f\_cuotas\_pagadas\_credito\_abierto + f\_cuotas\_pagadas\_en\_abono\_actual;
END IF;
END IF;
ELSIF tipo\_pago ='ABONO A CAPITAL' OR tipo\_pago ='PAGO TOTAL DEUDA' THEN
/\* se calcula el nuevo valor total y valor abono de la deuda\*/
f\_valor\_total\_deuda\_nuevo = round(f\_valor\_total\_deuda - f\_valor\_abono) ;
f\_valor\_abono\_deuda\_nuevo = round(f\_valor\_abono\_deuda + f\_valor\_abono);
/\* se calcula el nuevo interes\*/
f\_proximovalor\_cancelar = round(((f\_valor\_total\_deuda\_nuevo \* f\_tasa)/100)/f\_dias\_dividir\_interes);
END IF;
END IF;
/\* FIN PRESTAMO ABIERTO\*/
END LOOP;
/\* SI LA PROXIMA CUOTA A PAGAR ES 0 O MENOR SIGNIFICA QUE EL CREDITO YA SE TERMINO DE PAGAR\*/
IF f\_proximovalor\_cancelar <= 0 THEN
f\_prestamo\_estado = 'PAGADO';
f\_proximovalor\_cancelar = 0 ;
f\_valor\_total\_deuda\_nuevo = 0;
ELSE
f\_prestamo\_estado = 'ACTIVO';
END IF;
/\* se resetea la varieble f\_cuotas\_pagadas\_credito\_abierto a cero, cuando ha alcanzado el numero de cuotas por mes\*/
IF f\_tiempo\_cobro = 'SEMANAL' AND f\_cuotas\_pagadas\_credito\_abierto = 4 THEN
f\_cuotas\_pagadas\_credito\_abierto = 0;
ELSIF f\_tiempo\_cobro = 'QUINCENAL' AND f\_cuotas\_pagadas\_credito\_abierto = 2 THEN
f\_cuotas\_pagadas\_credito\_abierto = 0;
ELSIF f\_tiempo\_cobro = 'MENSUAL' AND f\_cuotas\_pagadas\_credito\_abierto = 1 THEN
f\_cuotas\_pagadas\_credito\_abierto = 0;
END IF;
/\* FIN RESETEO VARIABLE\*/
IF f\_prestamo\_abono\_bandera = 1 THEN
UPDATE prestamos SET prestamo\_valor\_actual = f\_valor\_total\_deuda\_nuevo, prestamo\_valor\_abonado=f\_valor\_abono\_deuda\_nuevo,
prestamo\_valor\_proxima\_cuota=f\_proximovalor\_cancelar,
prestamo\_estado\_pago = f\_prestamo\_estado\_pago, prestamo\_estado = f\_prestamo\_estado,
prestamo\_abono\_bandera=f\_prestamo\_abono\_bandera
WHERE id = f\_prestamo\_id;
ELSE
UPDATE prestamos SET prestamo\_valor\_actual = f\_valor\_total\_deuda\_nuevo, prestamo\_valor\_abonado=f\_valor\_abono\_deuda\_nuevo,
prestamo\_cantidad\_cuota\_pagada\_creditoabierto =f\_cuotas\_pagadas\_credito\_abierto,
prestamo\_valor\_proxima\_cuota=f\_proximovalor\_cancelar,
prestamo\_estado\_pago = f\_prestamo\_estado\_pago, prestamo\_estado = f\_prestamo\_estado,
prestamo\_fecha\_proximo\_cobro = (fc\_calcula\_fecha\_cobro(f\_prestamo\_id,f\_tiempo\_cobro, f\_fecha\_proximo\_cobro)),
prestamo\_abono\_bandera=f\_prestamo\_abono\_bandera
WHERE id = f\_prestamo\_id;
END IF;
UPDATE abonos SET abono\_capital\_abonado\_abierto = f\_abono\_capital\_abonado\_abierto
WHERE prestamo\_id = f\_prestamo\_id and id = (select id from abonos where prestamo\_id = prestamo\_id order by id desc limit 1);
END IF;
RETURN f\_valor\_total\_deuda\_nuevo;
END;
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;
ALTER FUNCTION public.fc\_calcula\_pago\_credito\_abierto(bigint, double precision, character varying)
OWNER TO postgres;