



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

“MEJORAS A LA PLATAFORMA DE CREDITO DISTRIBUIDOR”

EXAMEN DE GRADO (COMPLEXIVO)

Previa a la obtención del GRADO de:

**INGENIERO EN CIENCIAS COMPUTACIONALES
ESPECIALIZACIÓN SISTEMAS TECNOLÓGICOS**

LEONARDO VICENTE AYALA ROCAFUERTE

GUAYAQUIL – ECUADOR

AÑO: 2016

AGRADECIMIENTO

Mi agradecimiento primeramente a Dios porque él me dio la vida, a mis padres, quienes contribuyeron a mi formación, y por su esfuerzo incondicional. Agradezco también a mis familiares, porque tuve el apoyo moral que ellos me dieron, un agradecimiento especial al Dr. Anton Schiller y su esposa Annemarie quienes me apoyaron en muchos aspectos de mi educación desde la primaria hasta la Universidad.

También quiero agradecer a los amigos, por sus valiosos consejos y finalmente quiero agradecer de una forma muy especial a mis Docentes de la Universidad, a mis compañeros de estudio, de trabajo y a mis Jefes, que con su apoyo siempre me impulsaron a salir adelante.

DEDICATORIA

Este trabajo lo dedico a Dios y a mis Padres que fueron el pilar fundamental para lograr esta meta.

Quiero también dedicar este trabajo a todas las personas que influenciaron de forma positiva en mi vida y en mi formación académica

Finalmente, también lo dedico a las personas que formarán parte importante en mi vida ya que será un referente que podré compartir con ellos y que me permitirá de alguna manera influenciar de forma positiva en ellos.

TRIBUNAL DE SUSTENTACIÓN



M.Sc Cruz Maria Falcones

EVALUADOR



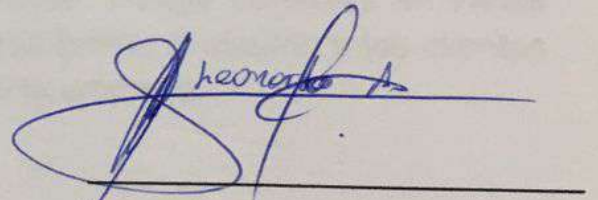
M.Sc Rafael Rivadeneira

EVALUADOR

DECLARACIÓN EXPRESA

"La responsabilidad por los hechos, ideas y doctrinas expuestas en este Informe me corresponde exclusivamente; y, el patrimonio intelectual de la misma, a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL"

(Reglamento de Graduación de la ESPOL).



Leonardo Ayala Rocafuerte

RESUMEN

El Proyecto PRY_MEJORAS CREDITO DISTRIBUIDOR, es un proyecto que parte de un sistema implementado con funcionalidades determinadas, pero que necesitaba estos cambios para adaptarse a las nuevas normativas de ley.

Este proyecto de Mejoras fue concebido por varias razones, una de las más importantes es el cambio en la malla contable. El cambio de la malla contable es un cambio que afecta a todo el sistema, ya que como veremos más adelante, existen un número considerable de procesos que utilizan dicha malla.

Otra de las razones también importantes, era la inclusión de funcionalidades nuevas, funcionalidades que también fueron solicitadas por temas que tienen que ver con normativas de ley y también por temas de mejoras en el negocio y el servicio que se brinda al cliente.

La funcionalidad implementada en el proyecto también incluye controles en varios procesos que permiten una mejora en la gestión que brinda el usuario a los clientes y también que el producto mejore la rentabilidad de la empresa.

ÍNDICE GENERAL

| | |
|--|-----------|
| AGRADECIMIENTO..... | ii |
| DEDICATORIA..... | iii |
| TRIBUNAL DE SUSTENTACIÓN..... | iv |
| DECLARACION EXPRESA..... | v |
| RESUMEN..... | vi |
| ÍNDICE GENERAL..... | vii |
| INTRODUCCIÓN..... | ix |
| CAPÍTULO 1..... | 1 |
| 1. CASOS GENERALES DEL PROYECTO MEJORAS CRÉDITO DISTRIBUIDOR Y SOLUCIÓN TECNOLÓGICA IMPLEMENTADA..... | 1 |
| 1.1 Cambio en la malla contable de Crédito Distribuidor Proveedor..... | 1 |
| 1.1.1 Eliminación de cuentas por cobrar por concepto de anticipos e inclusión de una operación de crédito para reemplazarlos..... | 3 |
| 1.1.2 Cambio de la cuenta de desembolso para los procesos de facturas proveedores..... | 7 |
| 1.1.3 Eliminación de cuentas por cobrar, se incluyen nuevos conceptos de GAF1 y GAF2 para los procesos de anticipos y cambio en el proceso de comisión para facturas anticipadas..... | 11 |
| 1.2 Inclusión de nuevas funcionalidades al sistema de Crédito Distribuidor Proveedor..... | 20 |
| 1.2.1 Sobre base imponible 0 (iva) a las comisiones para facturas anticipadas..... | 20 |
| 1.2.2 Inclusión de pago de facturas en período de gracia para facturas de tipo proveedor en el sistema de la Intranet de Crédito Distribuidor Proveedor..... | 24 |

| | | |
|-------|---|-----------|
| 1.2.3 | Control de sobregiros de cuentas para facturas..... | 25 |
| 1.2.4 | Funcionalidad de Carga Masiva de facturas..... | 29 |
| 1.3 | Inclusión de nuevos controles al sistema de Crédito Distribuidor Proveedor..... | 31 |
| 1.3.1 | Bloqueo, desbloqueo e inactivación manual de la línea de crédito..... | 31 |
| 1.3.2 | Control de recurso de la empresa ancla para facturas distribuidores..... | 34 |
| 1.3.3 | Réplica de montos desde el sistema de Crédito Distribuidor hacia el sistema Administrador de Líneas y viceversa..... | 37 |
| 1.3.4 | Nueva funcionalidad de control: Mantenimiento de especialistas de producto..... | 39 |
| 1.3.5 | Notificaciones de líneas de crédito próximas a su fecha de vencimiento..... | 41 |
| | CAPITULO 2..... | 44 |
| 2. | RESULTADOS OBTENIDOS..... | 44 |
| 2.1 | Inclusión de nuevas funcionalidades permiten una ampliación de servicios para las empresas ancla y clientes en general..... | 44 |
| 2.2 | Mejor control sobre las operaciones de crédito..... | 45 |
| 2.3 | Cumplimiento de las nuevas normativas necesarias en el giro del negocio gracias a la implementación del proyecto de Mejoras a la Plataforma de Crédito Distribuidor Proveedor..... | 45 |
| | CONCLUSIONES Y RECOMENDACIONES..... | 47 |
| | BIBLIOGRAFÍA..... | 49 |

INTRODUCCIÓN

El Proyecto de Mejoras Crédito Distribuidor nace por tres importantes necesidades, que luego se convertirían en funcionalidades que requería el sistema.

Estas funcionalidades son cambios en la malla contable, nuevos métodos para cumplir con las normativas de ley y para agregar controles al negocio.

El Cambio en la malla contable se lo puede dividir en tres casos generales, el primer caso se trata de la eliminación de cuentas por cobrar en los procesos de anticipos del sistema y la inclusión del sistema de préstamos **GAF** para reemplazarlas.

El segundo caso general es reemplazar la cuenta transitoria AMERAFIN que se utilizaba en los desembolsos para el caso de facturas de proveedores por la cuenta Transitoria de Crédito Distribuidor, esto por requerimiento interno del banco, éste cambio de la cuenta en el desembolso para una factura implica cambio en tres procesos del sistema que son efectivización de una factura normal proveedor, anticipo de una factura proveedor, efectivización de una factura anticipada proveedor. El tercer y último cambio general, se refiere a la introducción del concepto del **GAF1** y **GAF 2**, así como los conceptos de préstamo GAF con impuesto solca al desembolso y préstamo GAF con impuesto solca capitalizado, así como también éste cambio implicó también la inclusión del concepto traspaso de comisión por el cambio de la no utilización de las cuentas por cobrar en los procesos de anticipos.

El tema de la inclusión de las nuevas funcionalidades es diferente para cada caso, en el caso de la nueva funcionalidad de la base imponible, es que este servicio del anticipo no está sujeto a ser base de retención y por lo tanto la transacción de

comisión debía ser marcada como base imponible cero y ser considerada en este rubro en los estados de cuentas de los clientes.

Otra funcionalidad que se añade en este proyecto es el pago de facturas proveedores a través del Sistema Intranet de Crédito Distribuidor, en este sistema el funcionario del banco (ventanilla o vía telefónica) únicamente podía hacer los pagos de facturas de tipo distribuidor, mas no de facturas de tipo proveedor.

Otra funcionalidad que se incluye en el sistema es la funcionalidad de control de sobregiro para garantizar la correcta transaccionalidad en las facturas proveedores, o al menos que no existan problemas en las transacciones por temas de insuficiencia de cupos, éste control garantiza que las facturas proveedores se efectivizarán en el sistema únicamente si poseen los fondos suficientes en sus cuentas para la transacción.

Otras de las funcionalidades solicitadas fue la inclusión de las cargas masivas de facturas en el sistema, tanto para las facturas de Distribuidores, así como también para las facturas proveedores, el cliente del banco en ciertas ocasiones por ejemplo desea cargar 60 facturas y tenía que realizarlo de forma individual y esto le tomaba un día completo, ahora sólo arma un archivo tipo csv, puede ser utilizando Excel, y carga las 60 facturas al sistema con un solo clic.

Para la inclusión de nuevas funcionalidades de control, podemos mencionar, el tema de la asignación de un especialista de producto a cada empresa, esto nos ayuda en el caso de notificaciones de líneas próximas a vencer para que le llegue notificaciones al especialista y notifique a la empresa, la renovación de su línea de crédito, por otro

lado, las notificaciones por concepto de préstamos que entran en proceso de mora, el especialista podrá notificarles que una factura determinada ha entrado en proceso de mora.

Otras de las funcionalidades de control nuevas con que cuenta el sistema son el bloqueo, desbloqueo e inactivación manual de líneas a través del especialista del producto, en algunas ocasiones, una empresa por alguna razón debe ser bloqueada para no permitirle cargar facturas, hasta que cancele todas sus deudas con las otras facturas, también una vez que se ponga a cuentas, el funcionario del banco puede proceder a desbloquearlo para permitirle cargar facturas nuevamente, este proceso de bloqueo e inactivación de línea, no debe impedir los pagos de facturas en gracia ni pagos de préstamos de los clientes.

Una funcionalidad adicional de control que también se solicitó fue la de control del recurso, que consiste en controlar el cupo de una empresa ancla que no se sobrepase la suma de los cupos de sus distribuidores de los que la empresa ancla garantiza el recurso.

CAPÍTULO 1

1. CASOS GENERALES DEL PROYECTO MEJORAS CREDITO DISTRIBUIDOR Y SOLUCION TECNOLÓGICA IMPLEMENTADA

Se resume el proyecto en tres secciones implementadas, ya que cada una de ellas abarca soluciones a requerimientos que tienen un objetivo en común, y se las ha clasificado de la siguiente manera:

- a) Cambio en Malla Contable de Crédito Distribuidor
- b) Inclusión de nuevas funcionalidades al Sistema de Crédito Distribuidor
- c) Inclusión de nuevos controles al Sistema de Crédito Distribuidor.

1.1 Cambio en la Malla Contable de Crédito Distribuidor.

Crédito Distribuidor poseía una malla contable que había usado por un largo período. Esta malla contable para las personas que no conocen sobre este término, significa un conjunto de asientos contables que deben realizarse por cada proceso del sistema.

El cambio de malla contable aplicó en varios procesos como se detalla a continuación

- 1) Pago de factura sin anticipo (desembolso y acreditación) distribuidor cuando llega la fecha de efectivización. (no existe anticipo al mayorista) distribuidor préstamo
- 2) Pago de factura sin anticipo y con abono parcial (abono parcial antes f. efectivización factura) distribuidor.
- 3) Pago de factura sin anticipo y con precancelación distribuidor (sin considera f.anticipo o efectivización)
- 4) Pago de factura con anticipo (desembolso y acreditación) distribuidor (aplica descuento de compra de facturas)
- 5) Pago de factura con anticipo con abono parcial (abono parcial antes f. anticipo factura) distribuidor (aplica descuento de compra de facturas)
- 6) Pago de factura con anticipo con abono parcial (abono parcial después f. anticipo factura y antes f. efectivización) distribuidor
- 7) Abono o precancelación préstamo (funcionalidad de préstamo)
- 8) Pago de factura con anticipo (desembolso y acreditación) proveedor
- 9) Pago de factura sin anticipo (desembolso y acreditación) proveedor
- 10) Pago de factura con anticipo con pago total (abono total antes f. anticipo factura) proveedor (aplica descuento de compra de facturas)
- 11) Pago de factura con anticipo con abono total (abono total después f. anticipo factura y antes f. efectivización) proveedor

A continuación se muestra la tabla 1.1.1 que representa la nueva malla contable de crédito distribuidor proveedor.

| PAGO DE FACTURA CON ANTICIPO (DESEMBOLSO Y ACREDITACION) DISTRIBUIDOR | | | | | | | |
|---|--------------------|------------------------------|-----------------|------|-------|---|----------------------|
| NATURALEZA | NUMERO DE CUENTA | NOMBRE DE CUENTA CONTABLE | CENTRO DE COSTO | DEBE | HABER | CONCEPTO | APLICACION |
| D | 140105000023999000 | REDIST.CART.GAF.X.VENCER | AGENCIAS | XXX | | DESEMBOLSO/GAF | GAF |
| C | 259090337000000000 | TRANSITORIA CR. DISTRIBUIDOR | 8386 | | XXX | DESEMBOLSO/GAF | GAF |
| D | 160305001920000000 | INTERES CREDITOS COMERCIALE | AGENCIAS | XXX | | ACCRUAL/GAF | GAF |
| C | 510405047000000000 | INTERESES CREDITOS COMERCIAL | AGENCIAS | | XXX | ACCRUAL/GAF | GAF |
| C | 140105000023999000 | REDIST.CART.GAF.X.VENCER | AGENCIAS | | XXX | TMP_CTAS_CAP JGAF | GAF |
| D | 140105000023001000 | CR.COM.GAF.1-30 D.X VENCER | AGENCIAS | XXX | | TMP_CTAS_CAP JGAF | GAF |
| D | 259090337000000000 | TRANSITORIA CR.DISTRIBUID | 8386 | XXX | | E :1;R :YBAB;F :008-005-00004890 | CREDITO DISTRIBUIDOR |
| H | 299090025000000000 | OP.X LIQ.A MERAFIN DEB.AUT | 8331 | | XXX | E :1;R :YBAB;F :008-005-00004890 | CREDITO DISTRIBUIDOR |
| D | 299090025000000000 | OP.X LIQ.A MERAFIN DEB.AUT | AGENCIAS | XXX | | 0607-YGUE N/C COMP. INTERNET CREDITO DI | CREDITO DISTRIBUIDOR |
| H | 210105001899800000 | TRANS.DEP. MONETARIOS S21 | AGENCIAS | | XXX | 0607-YGUE N/C COMP. INTERNET CREDITO DI | CREDITO DISTRIBUIDOR |
| D | 210105001899800000 | TRANS.DEP. MONETARIOS S21 | AGENCIAS | XXX | | F :016-001-000017247;E :18 | CREDITO DISTRIBUIDOR |
| H | 519090007000000000 | PAG.ANTIC.FACT.PROD.DISTR | 8386 | | XXX | F :016-001-000017247;E :18 | CREDITO DISTRIBUIDOR |

Tabla 1.1.1: Nueva Malla contable para anticipos caso Distribuidor.

1.1.1 Eliminación de Cuentas por cobrar por concepto de anticipos e inclusión de una operación de Crédito para remplazarlos

Las Facturas anticipadas de crédito distribuidor se desembolsaban con cuentas por cobrar, la cuenta por cobrar para una factura distribuidor es 16XX90XXX000000XXX, para un desembolso de una factura proveedor era la cuenta 19XX90XXX000000XXX

Esta cuenta que realizaba el desembolso debía ser cambiada por otra cuenta que haga el desembolso pero desde cuenta especial llamada Cuenta Transitoria de Crédito Distribuidor, y para esto se debía utilizar un sistema especial de crédito implementado en AS/400 [1] llamado sistema GAF que lleva un control de crédito.

Este préstamo GAF no generaba intereses hasta antes de la fecha de efectivización de la factura, pasada la fecha de efectivización, si el cliente

no había pre cancelado, el sistema comenzaba a generar intereses para este préstamo.

Para realizar este cambio fue necesario identificar cómo se hace el desembolso de una cuenta por cobrar y cómo se hace el desembolso a través del sistema GAF.

La diferencia es que el desembolso realizado por la cuenta por cobrar es a través de una transacción middleware y se requiere una transferencia entre cuentas, como se muestra en la tabla 1.1.2

| Naturaleza | Cuenta | Valor |
|-------------------|--|--------------|
| D | 16XX90XXX000000XXX (CXC Distribuidor) | \$\$\$ |
| H | 2xx0900XX000000000 (Transitoria) | \$\$\$ |

Tabla 1.1.2: Transacción de desembolso con cuenta por cobrar.

Por otro lado, para hacer el desembolso de GAF, Crédito Distribuidor debe conectarse al sistema de GAF y enviar una trama, con un formato que el GAF comprende y procesa.

El desembolso en primera instancia quedaba como se muestra en la tabla 1.1.3

| Naturaleza | Cuenta | Valor |
|-------------------|-------------------------------------|--------------|
| D | 1XX1050000X3XXX000 (GAF) | \$\$\$ |
| H | 2xx0900XX000000000 (Transitoria) | \$\$\$ |

Tabla 1.1.3: Transacción de desembolso a través del sistema GAF.

La diferencia radica en que GAF calcula valores adicionales como son impuestos solca, cobro de intereses, entre otros.

Para el caso de desembolsos por préstamos no se aplica el tema de intereses pues se presume que la factura aún se encuentra en período de gracia.

En la capa del negocio del sistema crédito distribuidor nos ubicamos en el proceso de anticipos para facturas distribuidores, como lo indica la figura 1.1.1

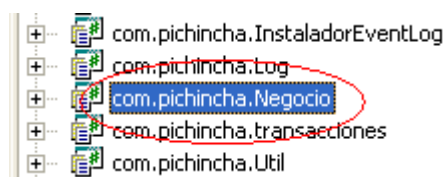


Figura 1.1.1: Capa del Negocio del Sistema Crédito Distribuidor.

En las clases Operaciones de distribuidor y operaciones de Proveedor se harán los cambios, estas clases se muestran en la figura 1.1.2

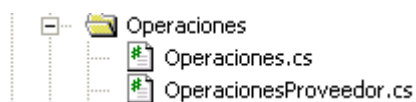


Figura 1.1.2: Clases que manejan los procesos de facturas anticipadas de Crédito Distribuidor.

Los métodos que se quitan son los siguientes:

- 1) **GeneracionPrecancelacionFacturasAEmpresa**
- 2) **GeneracionOperacionesGracia**

Para la clase Operaciones de facturas distribuidor, el método que se modifica es el que se muestra en la figura 1.1.3

```
//GENERACION PAGO DE FACTURAS (DISTRIBUIDOR) ANTICIPADAS-----
RegistraLog.EscribirLog(" >> INI (SUB PROCESO 3/4) - Generacion de Pago de Facturas (DISTRIBUIDORES) Anticipadas
error = objOperaciones.GeneracionPreCancelacionFacturasAEmpresa( dtEmpresas);
if ( error.Length > 0 )
{
    RegistraLog.EscribirLog(" << ERROR Generacion de Pago de Facturas (DISTRIBUIDORES) Anticipadas: " + DateTime
    RegistraLog.EscribirLog(" " +error + " ");
}
RegistraLog.EscribirLog(" >> FIN (SUB PROCESO 3/4) - Generacion de Pago de Facturas (DISTRIBUIDORES) Anticipadas
```

Figura 1.1.3: Método GeneracionPrecancelacionFacturasAEmpresa.

Éste es el proceso que se modifica internamente para que ya no trabaje con las cuentas por cobrar en el caso de facturas distribuidor específicamente.

Adicionalmente el método nuevo que se creó para que invoque al GAF por el tema del desembolso en el anticipo se muestra en la figura 1.1.4

```
//GENERA OPERACIONES CON GRACIA -----
RegistraLog.EscribirLog(" >> INI (SUB PROCESO 2/7) - GeneracionOperacionesGracia: " + DateTime.Now.ToString() + ".");
error = objOperaciones.GeneracionOperacionesGracia( dtEmpresas , entorno);
if ( error.Length > 0 )
{
    RegistraLog.EscribirLog(" << ERROR GeneracionOperacionesGracia: " + DateTime.Now.ToString() + ". >>");
    RegistraLog.EscribirLog(" " +error + " ");
}
RegistraLog.EscribirLog(" >> FIN (SUB PROCESO 2/7) - GeneracionOperacionesGracia: " + DateTime.Now.ToString() + ".");
```

Figura 1.1.4: Método que Genera el GAF en período de gracia para una factura anticipada de distribuidor en el sistema Crédito Distribuidor.

Éste es el método que invoca al sistema GAF y le envía la trama para que éste haga el desembolso desde la cuenta de préstamos y desembolse en la cuenta transitoria de crédito distribuidor.

Para el caso de facturas proveedores el método de la clase proveedor que se debía cambiar se muestra en la figura 1.1.5:

```
#region PreCancelacionFactura
public int PreCancelacionFactura(string cuentaAcreditar, string codigoOperacion, string concepto, decimal valorpagar, decimal d
{
    //Portal
}
```

Figura 1.1.5: Método PreCancelacionFactura utilizado para realizar los anticipos para facturas proveedores.

El proceso de anticipos para facturas distribuidores es diferente al método de anticipos para facturas proveedores, el método de anticipos para proveedores se realiza en línea, cuando un proveedor desea anticiparse el dinero correspondiente a una factura, en cambio para anticipar una factura distribuidor se debe esperar el proceso batch del sistema de crédito distribuidor y en función a la fecha de anticipo de la factura, ésta se anticipa a la empresa mayorista.

1.1.2 Cambio de la cuenta de desembolso para los procesos de facturas de Proveedores

Anteriormente, toda factura de proveedor que llegaba a su fecha de efectivización desembolsaba en una cuenta transitoria interna que

también es utilizada por otros aplicativos y posteriormente hacer un par de movimientos para terminar con la acreditación al respectivo proveedor.

Esta serie de movimientos se pueden visualizar en la tabla 1.1.4

| Naturaleza | Cuenta | Valor |
|-------------------|--|--------------|
| D | 1XX1050000X3XXX000 (GAF) | \$\$\$ |
| H | 2xx0900XX000000000 (TRANSITORIA) | \$\$\$ |
| D | 2xx0900XX000000000 (TRANSITORIA) | \$\$\$ |
| H | 210105001899800001 Cuenta Empresa ancla | \$\$\$ |
| D | 210105001899800001 Cuenta Empresa ancla | \$\$\$ |
| H | 210105001899800002 Cuenta Proveedor | \$\$\$ |

Tabla 1.1.4: Tabla con los movimientos de acreditación al proveedor con el esquema de desembolso anterior al proyecto.

En este proceso NO participaba la cuenta TRANSITORIA DE CREDITO DISTRIBUIDOR, por eso se solicitó como requerimiento funcional para éste proyecto.

Con respecto a la diferencia en la trama en GAF radica en un cambio en un parámetro de la trama GAF, en el campo 027 anteriormente se enviaba en la trama, para que el sistema GAF sepa que en esa cuenta hará el desembolso, utilizando como puente la cuenta Transitoria interna (llamada también AMERAFIN), pero actualmente NO se envía éste parámetro en la trama.

La clase donde se envía la trama se llama ProductosGaf.cs, podemos visualizarla en la figura 1.1.6



Figura 1.1.6: Clase que invoca a la mayoría de métodos del sistema GAF.

Esta clase ProductosGaf.cs, la clase Operaciones.cs y la clase OperacionesProveedor.cs, todas pertenecen a la capa del negocio de la solución.

```

if (empresaDistribuidor.Length > 0)
{
    if ((empresaDistribuidor == "E") || (empresaDistribuidor == "P"))
    {
        DatosDesembolso = "027N/C010 " + dr["NUMERO_CUENTA_DESEMBOLSO"].ToString().PadLeft(15, '0') + valordeser
    }
    else
    {
        if (ConstantesContabilidad.Equals("S"))
        {
            DatosDesembolso = "027N/C010 " + dr["NUMERO_CUENTA_EXPEDIENTE"].ToString().PadLeft(15, '0') + dr["MC
        }
        if (ConstantesContabilidad.Equals("N"))
        {
            DatosDesembolso = "";
        }
    }
}
}

```

Figura 1.1.6: Trama anterior para generación de préstamos en el GAF facturas proveedor.

Como vemos en la trama se incluye el campo 027 donde irá especificada la cuenta donde se hará el desembolso a la empresa ancla, y luego al proveedor.

Por otro lado en el cambio implementado, ésta línea de código fue modificada como lo podemos apreciar en la figura 1.1.7

```

if (empresaDistribuidor.Length > 0)
{
    if ((empresaDistribuidor == "E") || (empresaDistribuidor == "P"))
    {
        //Se omite valor para que se vaya a la 337 por default
        //DatosDesembolso="027N/C010 "+dr["NUMERO_CUENTA_DESEMBOLSO"].ToString().PadLeft(15,'0')+valordeseembo
        DatosDesembolso="";
    }
    else
    {
        if(ConstantsContabilidad.Equals("S"))
        {
            DatosDesembolso="027N/C010 "+dr["NUMERO_CUENTA_EXPEDIENTE"].ToString().PadLeft(15,'0')+dr["MONTO"
        }
        if(ConstantsContabilidad.Equals("N"))
        {
            DatosDesembolso = "";
        }
    }
}
}

```

Figura 1.1.7: Trama nueva para generación de préstamos en el GAF facturas proveedor.

Con esta omisión en la trama para procesos proveedores, el sistema GAF recibe la trama e interpreta directamente que el desembolso debe hacerlo a la cuenta TRANSITORIA DE CREDITO DISTRIBUIDOR.

1.1.3 Eliminación de cuentas por cobrar, se incluyen nuevos conceptos de GAF1 y GAF2 para los procesos de anticipo y cambio en el proceso de comisión de facturas anticipadas

Si bien el GAF reemplazaba la cuenta por cobrar, existía el inconveniente de la inclusión de una nueva normativa que decía que cualquiera que sea el préstamo generado, todo préstamo debe incluir impuesto SOLCA, de esta manera el GAF anterior tenía la característica de que llegaba la fecha de efectivización de la factura y allí recién comenzaba a generar intereses el sistema GAF.

Por esta razón nacieron dos conceptos:

Préstamo con SOLCA al desembolso y préstamo con SOLCA capitalizado.

El esquema que se planteó para estos dos GAF se lo muestra en la figura 1.1.8

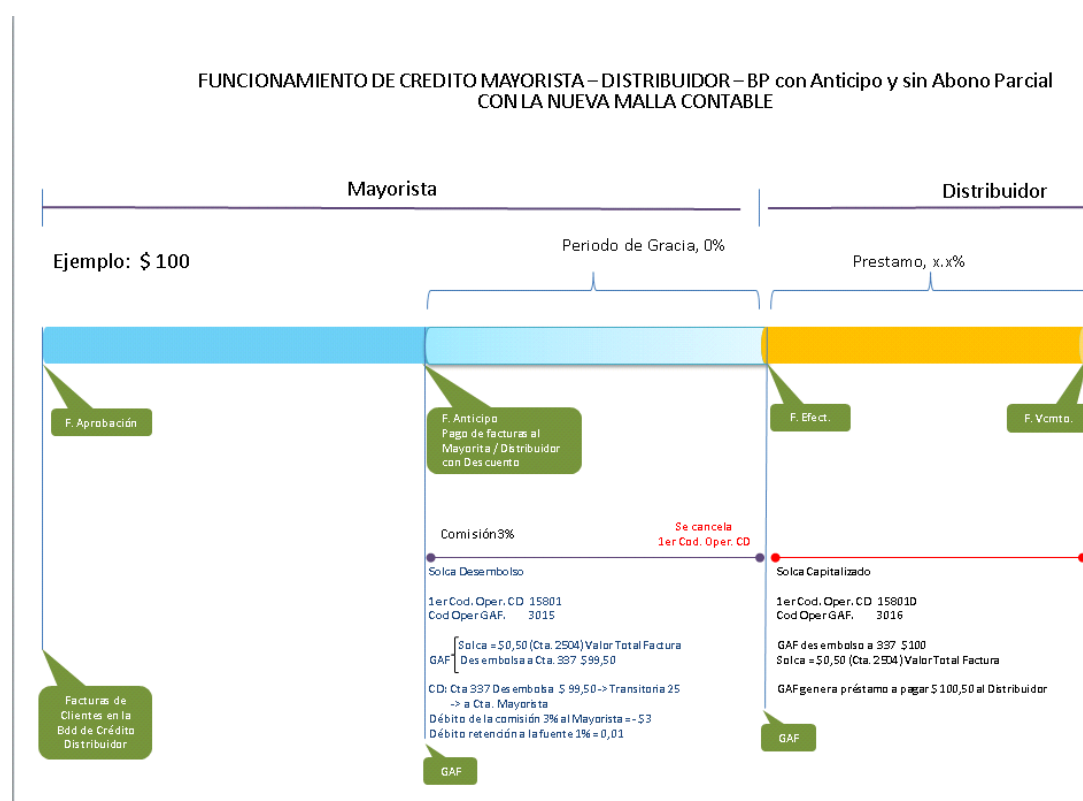


Figura 1.1.8: Esquema General para el funcionamiento de dos GAF en los procesos de facturas anticipadas del sistema Crédito Distribuidor.

Una problemática que surgió para introducir dos GAF en los procesos, fue que nunca antes una factura se podía relacionar con dos préstamos GAF, por lo que fue necesario aplicar un 'artificio' para que sistema pueda realizarlo, este mecanismo aplicado será detallado más adelante.

El préstamo GAF con SOLCA al desembolso deposita a la cuenta para el anticipo el valor de la factura, menos el valor de SOLCA, de esta manera, para una factura proveedor, el que paga el valor de SOLCA es el proveedor a quien se le hace el anticipo, pero en una factura distribuidor, el desembolso también va por el valor de la factura menos el SOLCA, pero quien paga el impuesto SOLCA de forma indirecta es la empresa ancla a quien se le hace la acreditación correspondiente.

Una característica adicional e importante de este tipo de préstamo, es que no genera intereses, mientras esté en su período de gracia, tiene dos períodos este tipo de préstamos, uno de gracia y pasado este periodo de vuelve como un préstamo normal.

Estos préstamos según la nueva malla, sólo se generan en las fechas de anticipos.

El préstamo GAF con impuesto SOLCA capitalizado sólo se genera cuando llega la fecha de efectivización de la factura, éste préstamo se considera el préstamo final y efectivo de toda factura que llega su fecha de efectivización y no ha sido pre cancelada.

Se llama préstamo con impuesto SOLCA capitalizado es porque cuando el GAF desembolsa el dinero, el desembolso lo hace por el monto exacto de la factura, pero adicionalmente genera el valor del impuesto SOLCA y lo carga al capital solicitado, es decir la deuda de préstamo será por el total de la factura más el impuesto SOLCA, pero la acreditación al proveedor/mayorista, será por el monto exacto de la factura.

A diferencia de lo expuesto en la sección 1.1.1 de que el GAF que actuaba en el desembolso de anticipo se esperaba que continúe pasada la fecha de efectivización, pero esta vez sí generando intereses, en ésta segunda y final implementación, el sistema no permitiría continuar al GAF1 pasada la fecha de efectivización, sino que el sistema procede a cancelarlo automáticamente con una cuenta interna del banco, y se lo cancela para que se genere un GAF2 que será el final y quedará como deuda para el cliente.

La comisión es un rubro que el banco gana por anticipar los dineros a la empresa, la comisión en la malla antigua se diferencia con la comisión en la nueva malla por dos puntos principales:

- 1) La cuenta de ingresos del banco recibía la comisión de la cuenta por cobrar
- 2) La fecha en que recibía el dinero la cuenta de ingresos era la fecha de anticipo.

Esto debía cambiar para que el proceso funcione acorde a la normativa vigente, para esto se tuvo que realizar las modificaciones en los procesos que se detallarán más adelante.

Los cambios que se realizaron para implementar esta funcionalidad fueron los siguientes:

Cambios en el método de anticipos proveedor, se incluye la generación del GAF1, como paso previo a la acreditación al proveedor, esta generación del GAF1 implica atar a la factura proveedor esta operación GAF, y cuando llegue la fecha de efectivización de ésta factura, se debe generar una segunda operación GAF y asociarla a la misma factura, para poder realizar esto se realizó el mecanismo ilustrado en la tabla 1.1.5

| Periodo | Numero Operación | Tipo Operación | Factura | Cód. Referencia Crédito Distribuidor |
|----------------|------------------|----------------|-------------|--------------------------------------|
| Anticipo | 02056877 | GAF1 | 00100125453 | 1987PAP |
| Efectivización | 02056882 | GAF2 | 00100125453 | 1987P |

Tabla 1.1.5: Ejemplo de dos préstamos GAF asociados a la misma factura.

Como podemos ver en el ejemplo anterior, la factura 00100125453 está relacionada con dos operaciones GAF en diferentes períodos, esto es posible porque la referencia enviada en la trama va concatenada con 'PAP' para los períodos de anticipo y con 'P' para los períodos de efectivización.

El primer método que fue cambiado para lograr este proceso lo podemos ver en la figura 1.1.9

```
public int PreCancelacionFactura(string cuentaAcreditar, string codigoOperacion,string concepto,decimal valorpagar,decimal descuento,string usuario, string canal,out string strMensaje)
```

Figura 1.1.9: Método para anticipos de facturas proveedor.

Este método pertenece a la clase **FPrecancelacionFacturas.cs** que es el que es utilizado para realizar los anticipos online para facturas proveedor.

En éste método se incluye la invocación al GAF como lo podemos observar en la figura 1.1.10

```
result = prodGAF.CrearOperacionGafGracia(dr, out SaldoOperacion,out NumeroOperacionGaf,out strMensaje,"P",out valorSolcaGaf);
```

Figura 1.1.10: Invocación al GAF en método de anticipo para facturas proveedor.

Adicionalmente, la clase **ProductosGaf.cs** fue modificada para realizar esta concatenación especial como se muestra en la figura 1.1.11

```

using System;
if (empresaDistribuidor == "P")
{
    RegistrarEjecucionProceso.EjecucionProcesoGAF("1", Dtain, dr["CODIGO_UNICO"].ToString(), "", dr["IDENTIFICACION"].ToString());
    CrearOperacion(CodPet, 0, dr["CODIGO_NEGOCIO_GAF"].ToString(), dr["CODIGO_UNICO"].ToString() + "AD", Convert.ToDecimal(AgeE
}
else
{
    RegistrarEjecucionProceso.EjecucionProcesoGAF("1", Dtain, dr["CODIGO_UNICO"].ToString(), "", dr["IDENTIFICACION"].ToString());
    CrearOperacion(CodPet, 0, dr["CODIGO_NEGOCIO_GAF"].ToString(), dr["CODIGO_UNICO"].ToString() + "AD", Convert.ToDecimal(AgeE
}
}

```

Figura 1.1.11: Modificación en la trama que va al GAF para la generación de un nuevo préstamo y atarla a la factura, aunque la factura ya se encuentre relacionada con un préstamo anterior.

De esta manera el desembolso queda realizado a la cuenta TRANSITORIA DE CREDITO DISTRIBUIDOR, pero con un impuesto SOLCA al desembolso, es decir que lo desembolsado en la cuenta, es el monto se la factura, menos el impuesto SOLCA, y eso va al proveedor / mayorista, según sea el caso.

Otro método que fue cambiado fue el método que realiza las acreditaciones anticipadas, para el caso de proveedor, el método lo observamos en la figura 1.1.12

```

retornoPreCancel=objOperacionProv.PreCancelacionFactura_Journals(cuentaProvee
dor,codigoOperacion,conceptoEnviar,conceptoDetalleEnviar,numeroFactura,valorpag
ar,descuento,Usuario,"2",dr["CODIGO_CLIENTE"].ToString(),codigoEmpresa,codigoLi
nea,valorSolcaGaf,cuentaEmpresa,out strMensaje,out numeroReferenciaBancs);

```

Figura 1.1.12: Invocación del método que realiza las acreditaciones anticipadas para facturas proveedor.

Este método fue cambiado para procesar las transferencias que se deben realizar en la acreditación al proveedor de acuerdo a la nueva malla contable.

En cuanto a los anticipos para las facturas distribuidor, ésta no se acreditan online, sino que pasan a través de un proceso batch, para éste proceso se utiliza un procedimiento almacenado de la base de datos que prepara la lista de datos a ser procesados por éste método, anteriormente las facturas que iban al proceso de acreditación anticipada, no poseían código de operación GAF, pero ahora en el nuevo procedimiento sí, se muestra un extracto de este cambio en el procedimiento almacenado de la base de datos.

```

SELECT AU.CODIGO_AUTORIZACION, AU.CODIGO_EMPRESA, AU.MONTO_PEDIDO,
DOP.CODIGO_OPERACION, DOP.CODIGO_ESTADO_OPERACION, DOP.CODIGO_PRODUCTO,
DOP.CODIGO_LINEA, DOP.MONTO_MONEDA_OPERACION, DOP.MONTO_MONEDA_LOCAL,
DOP.FECHA_EFECTIVIZACION, DOP.FECHA_VENCIMIENTO, DOP.CODIGO_OPERACION_GAF,
DOP.NUT, DOP.SALDO_OPERACION, DOP.NUMERO_BLOQUEO_LINEA,
DOP.REFERENCIA_SIGLO, DOP.SALDO_LINEA_CREDITO, DOP.FECHA_ANTICIPO,
DR.CODIGO_RETENCION, DR.CODIGO_CLIENTE, DR.CODIGO_ESTADO_RETENCION,
DR.VALOR_RETENCION, DR.NUMERO_AUTORIZACION_SRI, DR.COMPROBANTE_RETENCION,
DR.CODIGO_RETENCION_IMPUESTO, DR.NUMERO_CRUCE_CONTABILIDAD, DF.CODIGO_FACTURA,
DF.CODIGO_ESTADO_FACTURA, DF.NUMERO_FACTURA_SRI, DF.MONTO_FACTURA , AU.CODIGO_ORIGEN_AUTORIZACION, --Lib:12328
isnull(DOP.VALOR_SOLCA,0)
FROM CDP_AUTORIZACION AU INNER JOIN
CDP_DATOS_RETENCION DR ON AU.CODIGO_AUTORIZACION = DR.CODIGO_AUTORIZACION INNER JOIN
CDP_DATOS_OPERACION DOP ON AU.CODIGO_AUTORIZACION = DOP.CODIGO_AUTORIZACION INNER JOIN
CDP_DATOS_FACTURA DF ON DR.CODIGO_RETENCION = DF.CODIGO_RETENCION
WHERE CONVERT(VARCHAR, DOP.FECHA_ANTICIPO, 111) <= CONVERT(VARCHAR, @VarFechaActual , 111)
AND CONVERT(VARCHAR, DOP.FECHA_EFECTIVIZACION,111) > CONVERT(VARCHAR, @VarFechaActual , 111)
AND AU.CODIGO_EMPRESA = @inVarCodigoEmpresa --PARAMETRO
AND DF.CODIGO_ESTADO_FACTURA = @inVarEstFactura -- ACTIVO --parametro
AND ( DOP.PAGO_ANTICIPO IS NULL OR DOP.PAGO_ANTICIPO ='N' )
--20150829LVAVALAR
AND ((DOP.CODIGO_ESTADO_OPERACION = 2 and CODIGO_OPERACION GAF IS NOT NULL) or ( DOP.CODIGO_ESTADO_OPERACION = 4))
--AND DOP.CODIGO_OPERACION_GAF IS NOT NULL -- MODIFICACION MARZO 2015 PARA DESEMBOLSO SOLO PROCESE LOS QUE GENERARON GAF

```

Figura 1.1.13: Procedimiento almacenado de la base de datos que lista las facturas anticipadas que pasaran por el proceso de acreditación.

Un caso importante a tomar en cuenta es cuando una factura es cancelada antes que llegue la fecha de efectivización, en estos casos el proceso debe tomar en cuentas estas facturas, realizar la acreditación pero sin hacer desembolso GAF.

Algo similar debe ocurrir cuando una factura es pre cancelada antes que llegue la fecha de anticipo.

El proceso de anticipo, debe realizar la acreditación, pero sin generar GAF1, el sistema en la fecha de anticipo, debe colocarle el estado de CANCELADA.

Para realizar este cambio se realizaron modificaciones en la clase Operaciones.cs que corresponde a la clase de la regla del negocio que procesa facturas de tipo distribuidor.

```
#region Finalizacion de Precancelaciones Antes de fecha de anticipo

if(dr["CODIGO_ESTADO_OPERACION"].ToString()=="4")
{
    NOperaciones objOperaciones = new NOperaciones();
    eDatos.Opcion="ADCR";//Operacion para actualizar el estado debitado
    eDatos.Codigo      = codigoOperacion;
    eDatos.CodigoEstado = 5;
    eDatos.CodigoAutorizacion = codigoAutorizacion;
    eDatos.SaldoOperacion = Convert.ToDecimal(codigoEmpresa);
    eDatos.FechaVencimiento = objSalNumOperacionC.ToString() ;
    objOperaciones.Actualizar( eDatos );
}
#endregion
```

Figura 1.1.14: Sección de código para procesar facturas anticipadas que ya fueron pre canceladas.

Como podemos observar el sistema consulta si la factura ya sido pre cancelada para posteriormente ponerle estado "CANCELADO" (estado 5) y no tomarla en cuenta en los procesos siguientes en fecha de efectivización.

1.2 Inclusión de nuevas funcionalidades al sistema de Crédito Distribuidor Proveedor.

Las nuevas funcionalidades se introducen como un cambio necesario para cumplir con normativas y también como requerimiento necesario en el negocio.

1.2.1 Sobre Base Imponible 0 (IVA) a las comisiones para facturas anticipadas

El tema de comisiones que cobra el banco por concepto de anticipo nació con el negocio, este rubro es por un tema de servicio y según los expertos de la normativa no está sujeto a ser tomado como base IVA,

La transacción que realizaba la transferencia de la comisión era una transacción sencilla, la más básica, sin embargo el servicio que realizaba esta transacción pertenecía a un servidor middleware [2] llamado "Process" este servicio de Process recibía como entrada códigos siglo para interpretar a que cuenta debía realizar la transferencia, recibía el

código siglo de la cuenta a acreditar y el código siglo de la cuenta a debitar, según estaba parametrizada en el sistema.

El problema radica en que este servicio middleware no tenía forma de como marcar la transacción como una transacción de base imponible 0.

Las transacciones pueden realizarse a través de servicios middleware, en términos generales, todo aplicativo .net o java puede realizar transacciones en la plataforma **core del banco** a través de un servicio middleware.

La plataforma middleware tiene varios servidores, entre ellos están el servidor process y el servidor broker, para la transacción de comisión, el sistema utilizaba el servidor process que transacciona con códigos siglo y no con cuentas, este servicio se llama TLR3022 pero no marca la transacción con base imponible 0 (IVA).

En cambio el servidor broker, sí transacciona con cuentas directamente y además cuenta con un servicio llamado TRLDEBIT, éste sí marca las transacciones con base imponible cero.

Para poder realizar las transacciones el sistema crédito distribuidor maneja una tabla de interfaz, de esta manera se conoce que cuentas necesito enviar en una transacción determinada, así que el primer cambio que se realizó fue colocar en esta tabla las cuentas correspondientes a los códigos siglo y después, se reemplazó las llamadas a los servicios process para transacciones, por los servicios

broker y para la transacción de comisión utilizar el servicio de TLRDEBIT para marcarla con base imponible cero.

Para solventar este problema de marcar la transacción de comisión como transacción de base imponible 0 (IVA), primeramente se tuvo que identificar el proceso exacto donde se realiza la transacción de la comisión, para incluir al nuevo servicio, y también se realizó modificaciones en las clases de transferencias para direccionamiento al nuevo servicio

En la figura 1.2.1 se muestra la inclusión del nuevo servicio.

```

//lvayalar
case "TLRDEBIT": //CTA CLIENTE A BGL con IVA 0
//TRAMA.USUARIO = m_entorno.Usuario;// "USRCRDIS";
TRAMA.USUARIO = BuscarValor("TLR3022UsuarioSiglo");//DRPAREDA INICIO 22/
TRAMA.TIPO_TRANSFERENCIA = "1";
TRAMA.NUM_CTA_INT_D = Convert.ToInt64(cuenta);
TRAMA.NUM_CTA_INT_DSpecified = true;
TRAMA.NUM_CTA_INT_C = "";
TRAMA.CTA_CONTABLE_D = "";
TRAMA.CTA_CONTABLE_C = CtaContable_C;
TRAMA.IMPORTE_OPER = monto;
TRAMA.IMPORTE_OPERSpecified = true;
TRAMA.COD_MONEDA = "USD";
TRAMA.DEL_CONCEPTO_D = "; Descuento pago anticipado";
conceptoOperacion=conceptoOperacion.Trim();
TRAMA.DEL_CONCEPTO_D=conceptoOperacion+TRAMA.DEL_CONCEPTO_D; //20151027
TRAMA.VALOR_COBRO_COMISION = 0;
TRAMA.VALOR_COBRO_COMISIONSpecified = true;
TRAMA.CTA_CONTABLE_COMISION = "0";
TRAMA.COD_OPER_COMISION = "";
TRAMA.REFERENCIA = "";
TRAMA.TIPO_IVA = "NA";
TRAMA.BATCH_TANDEM = BuscarValor("CANAL");//"6";//CANAL
TRAMA.UUIDSOURCE = BuscarValor("APLICACION");//"CRD";//APLICACION
TRAMA.PROGRAMA = "TLRDEBIT";

```

Figura 1.2.1: Sección donde se implementa la funcionalidad de base imponible 0 (IVA) en el sistema de Crédito Distribuidor.

Re direccionamiento en las clases de transferencia para llamar al nuevo servicio.

Clase: **TranferenciaCuentas.cs**

```

if(operacion.Equals("DIVAO"))
    retorno=objTransferencia.EjecutarCreditoDebito(m_entorno,cuenta,monto,codigoreferencia,codigoContable,operacion)
else
    retorno=objTransferencia.EjecutarCreditoDebito(m_entorno,cuenta,monto,codigoreferencia,codigoContable,operacion)

//Seteo del numero de operacion devuelto desde siglo21

```

Figura 1.2.2: Re direccionamiento del nuevo servicio.

Clase: **DebitosCreditos.cs**

Apuntamiento final al servicio de middleware TLRDEBIT

```

try
{
    switch(tipooperacion)
    {
        case "D"://Debito a una cuenta
            TipoTransferencia=TipoTransferencia+ "-12";
            break;
        case "C":
            TipoTransferencia=TipoTransferencia+ "-21";
            break;
        case "DIVAO":
            TipoTransferencia="TLRDEBIT";
            break;
    }
    string descripcion = conceptooperacion;
    conceptooperacion= codrefer+"-"+descripcion;
    retorno = TransferenciaEntreCuentasTLR3022( TipoTransferencia, CtaCliente_D,
        CtaCliente_C, CtaContable_D, CtaContable_C, cuenta, monto, tipooperacion,
        codoperacion, conceptooperacion, CentroCostoD, CentroCostoC);
}
catch(ExcepcionSiglo21 exSiglo21)
{
    throw exSiglo21;
}

```

Figura 1.2.3: Elección de tipo de transferencia para la base imponible 0 (IVA)

1.2.2 Funcionalidad adicional: Inclusión de pago de facturas en período de gracia para facturas de tipo proveedor en el sistema de la Intranet de Crédito Distribuidor Proveedor.

El tema de pago de facturas proveedor (en período de gracia), se venía realizando a través del Portal de Crédito Distribuidor, éste es un sistema de acceso al público, claro con las respectivas credenciales otorgadas por el banco.

Sin embargo se suelen presentar casos de que un cliente desea que el banco a través de su plataforma realice los pagos, esta funcionalidad no existía.

Para poder implementar esta funcionalidad nos basamos en dos funcionalidades ya existentes:

- 1) Funcionalidad de Pagos Facturas proveedor en Portal
- 2) Funcionalidad de Pagos de Facturas distribuidor en Intranet.

Así que con estos referentes, se procedió a implementar ésta funcionalidad en la Intranet de Crédito Distribuidor.

Para los pagos de facturas en período de gracia tenemos dos procesos:

- 1) `PagodeOperacionesAnticipadasClienteProveedor`
- 2) `EjecutarLiberacionProveedor`

El proceso 1 aplica para las facturas proveedores en periodo de gracia que han sido anticipadas, el segundo proceso aplica a las facturas proveedores en periodo de gracia que no son sin anticipo.

El proceso 2 para facturas proveedores no anticipadas.

Cabe destacar que para pago de facturas proveedores, el pago siempre es por el monto total de la factura, no se realizan abonos parciales.

1.2.3 Funcionalidad adicional del control de sobregiros de cuenta para facturas

Los casos en los que se implementó esta funcionalidad de control de sobregiros en cuentas, son las siguientes:

- Control de sobregiro en generación de Prestamos GAF en fecha de efectivización para facturas proveedor anticipadas y no anticipadas.
- Control de sobregiro para anticipos.

Estos controles de sobregiros son para evitar que se realicen movimientos incompletos, todo tiene que quedar afectado de forma completa.

En estos casos en que por causa de un sobregiro, la factura no pasa por el proceso que necesita, el sistema envía una notificación al usuario

cliente del banco indicando que su factura no fue procesada por saldo insuficiente.

A continuación se presenta un extracto de la implementación de esta funcionalidad en la figura 1.2.4

```

DataTable dtESpecialistaMail= new DataTable ();
dtESpecialistaMail = com.pichincha.Negocio.CreditoDist.Administracion.EspecialistaProducto.Especial
string datosExtra = "<br> Empresa : "+ dr["DESCRIPCION"].ToString ();
datosExtra += "<br> Saldo Contable : $ "+ dr["SALDO_CONTABLE"].ToString ();
datosExtra += "<br> Saldo Disponible : $ "+ dr["SALDO_DISPONIBLE"].ToString ();
if(dtESpecialistaMail.Rows .Count >0 )
{
    string CorreoEspecialista= dtESpecialistaMail.Rows [0]["EMAIL"].ToString ();
    string NombreEspecialista= dtESpecialistaMail.Rows [0]["NOMBRE"].ToString ();
    com.pichincha.transacciones.Notificaciones notifi = new com.pichincha.transacciones.Notificac
notifi.EnviarNotificacionCuentaSobregirada (datosExtra,NombreEspecialista, CorreoEspecialista )
}
}
catch(Exception ex)
{
    Negocio.CreditoDist.Errores.LogError.RegistroError(ex.Message,"BatchOperacionesProveedorEnvioNotifi
RegistraLog.RegistrarLog (ex);
}

```

Figura 1.2.4: Notificaciones para control de cuentas sobregiradas.

Para el envío de la notificación se utiliza una clase. NET la cual mostramos en la siguiente figura:

```

-----
SmtpMail.SmtpServer = smtpServer; //ConfigurationSettings.AppSettings["strSmtpServer"];
try
{
    class System.Web.Mail.SmtpMail
    Proporciona propiedades y métodos para enviar mensajes mediante el componente de mensajes de CDOYS (Collaboration Data Objects para Windows 2000).
    {
        SmtpMail.Send(message);
        retorna = true;
    }
}
catch(Exception er)
{
    //ANTES
    //throw new Exception ("Existe un error con el servidor de correo, no pudo enviar el email : "+er.Message,e

```

Figura 1.2.5: Librería utilizada para envío de mails

La implementación de esta funcionalidad se la realizó en la clase OperacionesProveedor.cs y en la clase PrecancelacionFacturas.cs del Portal de Crédito Distribuidor.

Ya que en estas clases están los métodos de los procesos que se mencionó anteriormente que pasan por control de sobregiro.

A continuación en la tabla 1.2.1 se lista los métodos donde se implementó:

| Metodo | Clase |
|---|---------------------------|
| public string GeneracionOperaciones(DataTable dtEmpresas) | OperacionesProveedor.cs |
| public string GeneracionOperacionesAnticipadasFinal(DataTable dtEmpresas) | OperacionesProveedor.cs |
| public int PreCancelacionFactura(string cuentaAcreditar, string codigoOperacion, string concepto, decimal valorpagar, decimal descuento, string usuario, string canal, out string strMensaje) | PrecancelacionFacturas.cs |

Tabla 1.2.1: Lista de métodos donde se implementó los controles de sobregiros de cuentas.

A continuación en la figura 1.2.6 y figura 1.2.7 se muestra una porción de código de la implementación de control de sobregiro.

```

string identificacion = "";
cuentas.setEntorno("07", dominio,nodo,codEmpresa,codCentro,"USRCRDIS","PortalCreditoDistribuidorCom.Web",dir
identificacion= dtEmpresas.Rows [i] ["IDENTIFICACION"].ToString();
if (identificacion!= String.Empty )
{
    DataRow fila = dtEmpresasCuentas .NewRow ();
    fila["CODIGO"] = dtEmpresas.Rows [i] ["CODIGO"].ToString();
    fila["DESCRIPCION"] = dtEmpresas.Rows [i] ["DESCRIPCION"].ToString();
    fila["EMAIL"] = dtEmpresas.Rows [i] ["EMAIL"].ToString();
    fila["CUENTA"] = dtEmpresas.Rows [i] ["CUENTA"].ToString();
    fila["IDENTIFICACION"] = dtEmpresas.Rows [i] ["IDENTIFICACION"].ToString();
    fila["DEBITOS_POR_REGIONAL"] = dtEmpresas.Rows [i] ["DEBITOS_POR_REGIONAL"].ToString();
    fila["CODIGO_EMPRESA_XML"] = dtEmpresas.Rows [i] ["CODIGO_EMPRESA_XML"].ToString();
    string cuentaEmpresa = dtEmpresas.Rows [i] ["CUENTA"].ToString() ;
    try
    {
        DataTable tblCuentas = cuentas.ObtenerCuentasSIGLO21Empresa(identificacion, usuario,"");
        if (tblCuentas.Rows.Count > 0)
        {
            //20150730DT INICIO
            DataRow filaEncontrada = tblCuentas.Select("Cuenta = ' + cuentaEmpresa + '" ) [0];
            fila["SALDO_CONTABLE"] = filaEncontrada["Saldo Contable"].ToString ();
            fila["SALDO_DISPONIBLE"] = filaEncontrada["Saldo Disponible"].ToString ();
            fila["OBSERVACION"] = "OK";
            dtEmpresasCuentas.Rows.Add (fila);
            //20150730DT FIN
        }
    }
}
}

```

Figura 1.2.6: Control de Sobregiro de cuentas proceso de adquisición de datos.

```

//Generacion de Operacion de Cartera
foreach( DataRow dr in dtEmpresas.Rows )
{
    // cambio realizado por JJPLUAP
    double saldoDisponible = 0;
    if (dtEmpresas!= null )
    {
        //20150730DT INICIO
        saldoDisponible= Convert.ToDouble (dr["SALDO_DISPONIBLE"].ToString ());
        //20150730DT FIN
    }
    //LVAYALAR20151007
    if (saldoDisponible >= 0)
    {
        ..
    }
}

```

Figura 1.2.7: Control de sobregiro de cuentas, proceso de evaluación del saldo de la cuenta.

1.2.4 Funcionalidad de Carga Masiva de facturas

Las empresas que trabajan con el sistema crédito distribuidor necesitaban cargar varias facturas con hacer un solo clic.

Si bien el sistema contaba con una sección para el ingreso de facturas individual, se hacía necesario implementar la carga masiva de facturas para agilizar esta parte del proceso.

Esta implementación de carga masiva, se realizaría en el Portal de Crédito Distribuidor así como también en el Sistema Intranet de Crédito Distribuidor.

Para solucionar este requerimiento de diseño el siguiente proceso:

El usuario a través del navegador carga un archivo de texto con un formato específico sencillo, que cualquier sistema pequeño puede generar.

Este archivo es validado, y subido al servidor y una copia es alojada en un repositorio FTP, para luego mediante un DTS (ETL) es procesado para finalmente registrar las facturas en el sistema.

Esta implementación, se la realizó utilizando las siguientes clases:

La Clase FtpHelper para realizar la carga del archivo.


```

serverFTP = objdatoscd.ConsultarReintentos("SVFTP",out strMensaje);//
usuarioFTP = objdatoscd.ConsultarReintentos("USFTP",out strMensaje);//
passwordFTP = objdatoscd.ConsultarReintentos("PWFTP",out strMensaje);//
pathFTP = objdatoscd.ConsultarReintentos("PVFTP",out strMensaje);//

FtpHelper objFtp=new FtpHelper ();
//ftpDownload.DownloadFile(pathFTP,nombreArchivo,pathLocalArchivo, serverFTP, usuarioFTP, passwordFTP);
objFtp.Upload(pathLocalArchivo,nuevoNombreArchivo,pathFTP,serverFTP, usuarioFTP, passwordFTP);
retorno void FtpHelper.Upload(string pathFileSource, string dirNameUpload, string remoteHost, string remoteUser, string remotePass)(+ 1 sobrecargas)
Upload File desde el servidor de aplicacion al FTP

```

Figura 1.2.8: Invocación a la clase FtpHelper para la carga del archivo que el cliente sube al sistema.

Luego a través de un webservice, se realiza la invocación de un DTS que tomará la información del FTP para realizar la carga

```

public int CargaMasivaFacturaProveedor( string codigoEmpresa, string tipoRelacion, string codigoRegional, string estado,
string codigoCliente, string fechaTransicion, string tipoDocumento, string canalIngreso, string usuario, string usuarioCarga,
out string strMensaje, out DataSet dsRetorno )
{
Factura proxy = new Factura();
return proxy.CargaMasivaFacturas( codigoEmpresa, tipoRelacion, codigoRegional, estado,
codigoCliente, fechaTransicion, tipoDocumento, canalIngreso, usuario, usuarioCarga, out strMensaje, out dsRetorno);
}

```

Figura 1.2.9: Invocación vía webservice al Servidor que ejecuta el DTS.

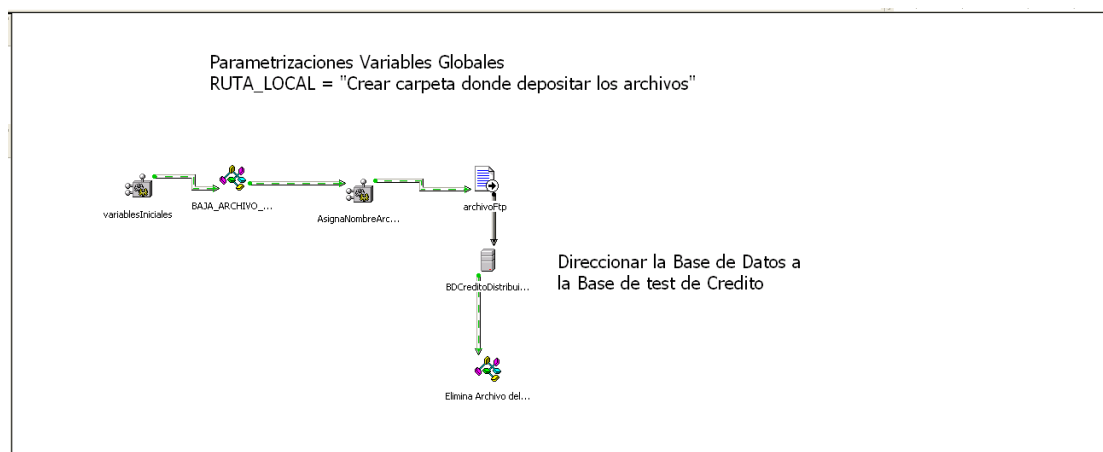


Figura 1.2.10: Diseño del DTS que realiza la inserción de las facturas.

1.3 Inclusión de nuevos Controles al sistema de Crédito Distribuidor Proveedor.

1.3.1 Bloqueo, desbloqueo e inactivación manual de la línea de crédito

Este requerimiento se refiere a la necesidad de que el funcionario del banco pueda ser capaz de bloquear una línea de crédito de forma manual de un cliente, ya que puede recibir una orden especial y realizarlo de forma inmediata, por otra parte todas las acciones de bloqueo, desbloqueo e inactivación permanente, son registradas en un log de sistema de esta forma queda la evidencia de si un funcionario bloqueo o desbloqueo una línea.

La solicitud de no afectación a los procesos de pagos, se refiere a que pese a que la línea haya sido bloqueada, el sistema le permita seguir pagando sus deuda mas no podrá ingresar más facturas mientras su línea permanezca con bloqueo.

Para desarrollar este requerimiento se tuvo que incrementar una nueva columna en la tabla de líneas de crédito en la siguiente figura podemos observar la nueva columna creada para este control.

| | A | ULTIMA MODIFICACION | MONTO AUTORIZADO | MONTO UTILIZADO | MONTO BLOQUEADO | NUMERO TARJETA | CODIGO FACILIDAD | CON RECURSO | ESTADO_CLIENTE |
|----|--------|---------------------|------------------|-----------------|-----------------|----------------|------------------|-------------|----------------|
| 1 | -01-11 | 18:09:37.370 | 10000.0000 | .0000 | .0000 | 00009 | .001 2 6 | S | I |
| 2 | -07-06 | 18:03:23.130 | 50000.0000 | .0000 | .0000 | 00009 | .001 2 6 | S | A |
| 3 | -07-06 | 18:03:39.457 | 90000.0000 | .0000 | .0000 | 09913 | .001 2 6 | S | A |
| 4 | -04-09 | 18:02:14.900 | 1.0000 | .0000 | .0000 | 00017 | .001 2 6 | S | I |
| 5 | -07-06 | 18:03:48.707 | 57000.0000 | .0000 | .0000 | 09913 | .001 2 6 | S | A |
| 6 | -07-06 | 18:03:06.143 | 96000.0000 | .0000 | .0000 | 00017 | .001 2 6 | S | A |
| 7 | -08-24 | 23:55:43.280 | 300000.0000 | 8402.1200 | .0000 | 00009 | .001 2 6 | S | I |
| 8 | -08-18 | 20:21:48.890 | 29000.0000 | .0000 | .0000 | 00017 | .001 2 6 | S | I |
| 9 | -06-30 | 18:04:24.263 | 16000.0000 | .0000 | .0000 | 00018 | .001 2 6 | S | A |
| 10 | -07-06 | 18:03:07.753 | 33000.0000 | .0000 | .0000 | 00001 | .001 2 6 | S | A |
| 11 | -07-06 | 18:03:28.487 | 30000.0000 | .0000 | .0000 | 00005 | .001 2 6 | S | A |

Figura 1.3.1 Nueva columna creada en la tabla de líneas de crédito para el control de bloqueo, desbloqueo e inactivación de líneas.

La columna ESTADO_CLIENTE es la que controla el proceso de carga de facturas, para saber si debe permitir o no una carga de facturas y esta funcionalidad no interviene en ningún sentido en el proceso de pagos.

Por otro lado se implementaron los controles en las secciones de carga de facturas, la siguiente figura muestra las secciones de carga de facturas donde se implementó el control:

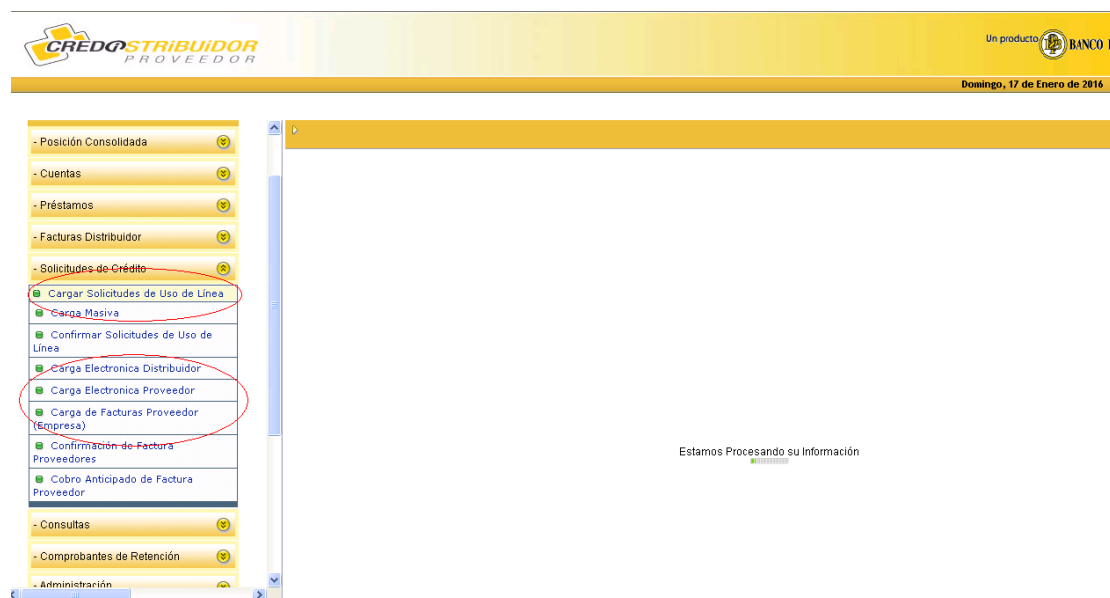


Figura 1.3.2 Sistema Portal donde se implementaron los controles para bloqueo de línea de crédito.

```

int estado_linea=0;

estado_linea= objConsultaLinea.ConsultaEstadoLineaCredito(0,Convert.ToInt32(codEmpre),"");

if(estado_linea==4)
{
    #region Informacion a ser enviada

    codigoEmpresa = txtCodigoEmpresa.Text;

    dsArchivo.Dispose();
}
else if(estado_linea==2)
    uscMensaje.MostrarMensaje("Error, La linea esta inactiva", TipoMensaje.ALERTA);
else if(estado_linea==1)
    uscMensaje.MostrarMensaje("Error, La linea esta bloqueada", TipoMensaje.ALERTA);
else if(estado_linea==3)
    uscMensaje.MostrarMensaje("Error, La Sobrepasado uso de cupo", TipoMensaje.ALERTA);
}

```

Figura 1.3.3 Validaciones en el portal implementadas a nivel de código.

Los códigos numéricos representan las diferentes acciones que se realizaron sobre la línea.

Este proceso se da cuando se decide inactivar una línea, cuando una línea se inactiva, esta acción es irreversible y para poder volverá utilizar la línea, el cliente deberá nuevamente pasar por todo el proceso de enrolamiento.

La acción de inactivación en esta funcionalidad implica conectarse el Administrador de Líneas de Crédito e inactivar la línea desde allí también, esto se lo implemento utilizando Web Service.

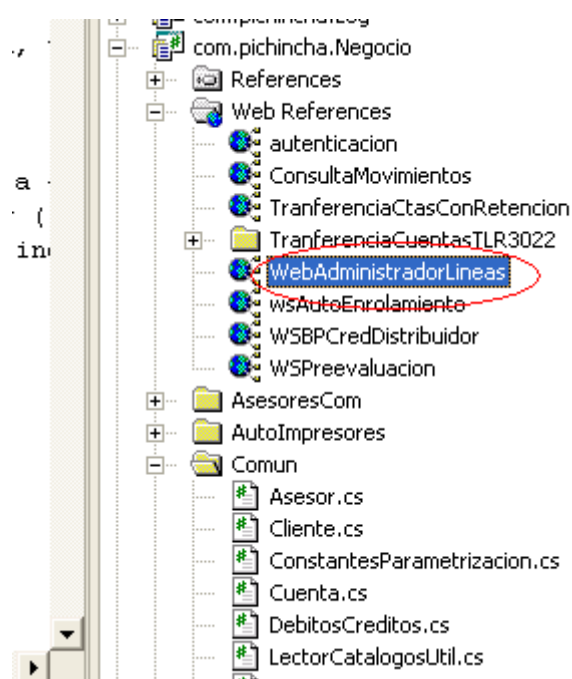


Figura 1.3.4: Web Service implementado para comunicación entre los sistemas de Crédito Distribuidor y el Administrador de Líneas de Crédito.

1.3.2 Control del Recurso de la Empresa Ancla para facturas distribuidores.

Este es un control que se solicitó para controlar que la Empresa ancla no cargue más distribuidores con recurso de los que permita su cupo.

Este control se implementó en la sección de carga de distribuidores.

En el momento en que este proceso recibe la lista de distribuidores con recurso que se desea cargar, el sistema se conecta con el administrador

de líneas de crédito y obtiene el monto del cupo de la empresa ancla, calcula la suma de los cupos solicitados de los distribuidores entrantes y compara que no exceda el cupo del administrador de líneas, de esta manera controla el sistema que no se sobrepase en la carga de distribuidores la empresa ancla.

Para implementar esta funcionalidad, se lo desarrolló en la ventana de carga de distribuidores y proveedores que se muestra en la siguiente figura:

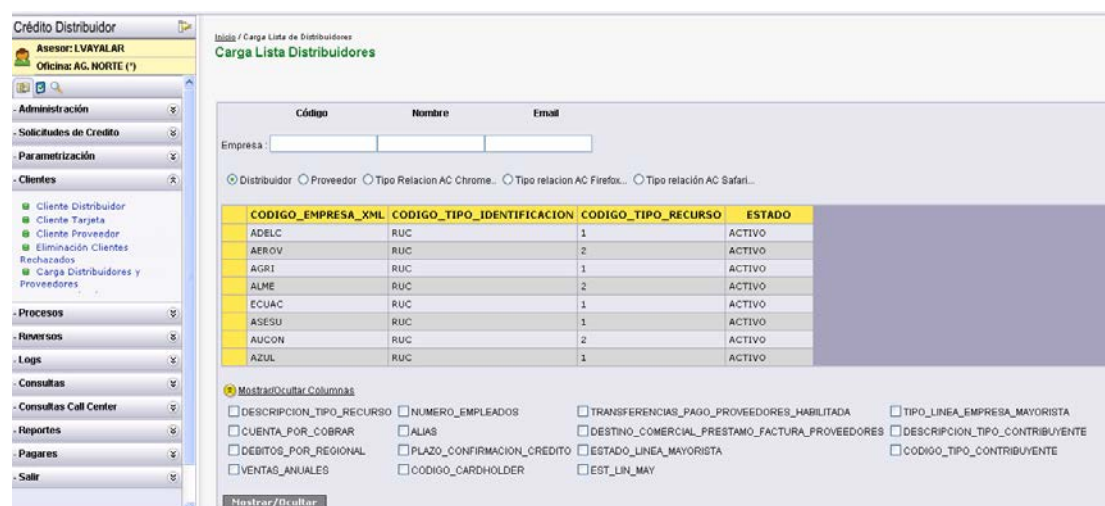


Figura 1.3.5: Ventana del sistema Crédito Distribuidor para la carga de distribuidores.



Figura 1.3.6: Control del recurso implementado a nivel de código.

La carga de distribuidores en esta sección funciona de la siguiente manera:

- 1) Se carga el archivo en el servidor
- 2) Se invoca a la clase cargar Excel que mostrara los resultados en el GRID
- 3) Se presiona el botón guardar para empezar el proceso de carga de los distribuidores con la validación del recurso

```

private void cmdCargaArchivo_ServerClick(object sender, EventArgs e)
{
    try
    {
        btninforma.Visible=false;
        wucMensaje.Visible=false;
        string retorno=SubirArchivoServidor();
        if (retorno.Length>0)
        {
            CargarExcel( retorno );
        }
    }
    catch(Exception ex)
    {
        wucMensaje.Visible=true;
    }
}

```

Figura 1.3.7: Porción de código que invoca el proceso de carga.

1.3.3 Réplicas de Montos desde el Sistema Crédito Distribuidor hacia el Administrador de Líneas de Crédito y viceversa

El sistema crédito distribuidor contaba con un batch que por algunos procesos del sistema, como por ejemplo, liberación de cupo, creación de operación, pago de préstamos, el sistema guardaba un registro especial en una tabla, luego en un proceso batch de la noche, el sistema procesaba esta tabla y en teoría actualizaba los saldos utilizados y bloqueados de las líneas al sistema Administrador de Líneas de Crédito.

Realmente este sistema no estaba funcionando de forma correcta por lo que nos solicitaron corregir el proceso o implementar uno mejor.

Otro tema importante sobre este requerimiento, es que también existe un monto que se replica, pero desde el Administrador de Líneas de Crédito hacia Crédito Distribuidor, y éste es el MONTO APROBADO, ya que

éste monto sólo se mueve en el Administrador de Línea de Crédito, ya que desde allí es donde se hacen las renovaciones de líneas.

Para solventar este requerimiento se definió replicar desde la tabla de líneas de Crédito Distribuidor y no desde la tabla anterior, ya que este proceso nuevo iba únicamente a recorrer las empresas y directamente de sus líneas, replicaría los montos.

Adicionalmente existe un proceso que actualiza los montos bloqueado y utilizado en Crédito Distribuidor, y el batch de la noche o la madrugada, actualizara los montos bloqueados y utilizados desde las líneas de Crédito Distribuidor hacia el Administrador de Líneas de Crédito.

El Web Service que realiza este trabajo se muestra en la siguiente figura:



Figura 1.3.8: método que invoca al Web Service de comunicación entre Crédito Distribuidor y el Administrador de líneas para la funcionalidad de réplica de montos.

El webservice se llama **WSCreditoDistribuidorALC.asmx**

Este Web Service trabaja en ambos sentidos, replica monto bloqueado y utilizado desde Crédito Distribuidor hacia Administrador de Líneas de Crédito, y monto autorizado desde Administrador de Líneas hacia Crédito Distribuidor, pero las invocaciones para estos procesos batch específicos, siempre inician desde Crédito Distribuidor.

1.3.4 Nueva Funcionalidad de Control: Mantenimiento de Especialistas de Producto

La necesidad de asociar un especialista de producto a una empresa siempre será una característica útil desde cualquier punto de vista que se lo mire.

Otro factor que influyó para la solicitud de este requerimiento es por el tema de las notificaciones para control, así fue como esta necesidad se incluyó como parte del proyecto de mejoras.

Esta sección se implementó desde cero, así como todas las secciones de esta funcionalidad nueva, por lo tanto se utilizó la misma tecnología en que está implementado Crédito Distribuidor, tecnología .NET con lenguaje de programación C#. [3]

La ventana que permite el ingreso del especialista se muestra en la siguiente figura:

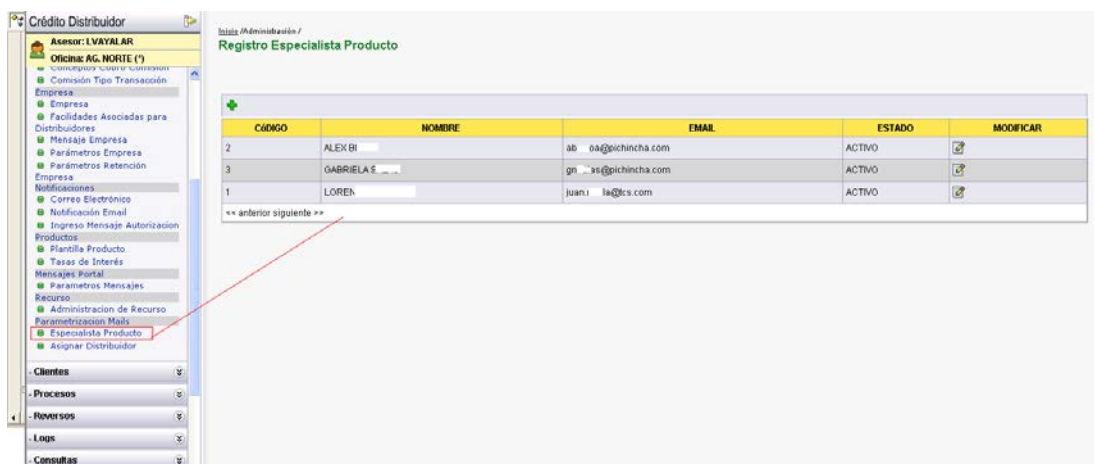


Figura 1.3.9: Ventana creada para mantenimiento de especialistas de producto.

Se implementó una nueva tabla en la BDD ya que si bien es cierto existía una tabla de especialista, no existía una tabla que relacione la empresa con dicho especialista, así que se procedió con la creación de una tabla intermedia que almacene esta relación, la tabla se muestra en la siguiente figura:

| | CODIGO_EMPRESA | CODIGO_USUARIO_EXPERTO |
|----|----------------|------------------------|
| 1 | 2 | 1 |
| 2 | 8 | 2 |
| 3 | 12 | 3 |
| 4 | 14 | 1 |
| 5 | 18 | 1 |
| 6 | 22 | 3 |
| 7 | 24 | 3 |
| 8 | 34 | 1 |
| 9 | 37 | 3 |
| 10 | 38 | 3 |

CDP_EMPRESA_USUARIO_EXPERTO
CDP_ESTABLECIMIENTO_CDI

Figura 1.3.10: Nueva tabla que relaciona al especialista de producto con una o varias empresas.



Figura 1.3.11: Asignación del especialista a una o varias empresas.

La opción de cambiar de especialista a una empresa o viceversa se realiza utilizando el botón de edición, de esta manera se realizó esta funcionalidad que permite la flexibilidad del aplicativo.

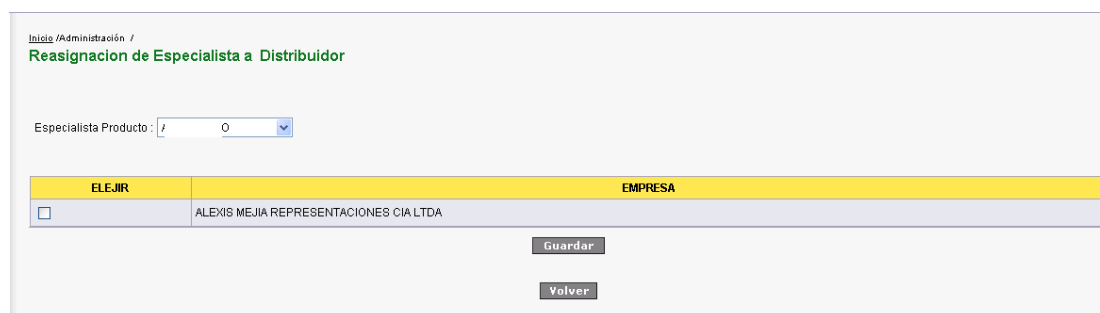


Figura 1.3.12: Ventana de reasignación de especialistas.

1.3.5 Notificaciones de líneas próximas a su fecha de vencimiento

Esta funcionalidad fue solicitada para el control de las vigencias de las líneas de crédito que estén enroladas en el sistema de Crédito Distribuidor.

Se coloca un parámetro actualizable de número de días de la alerta, antes de que venza la línea, por ejemplo:

Si el parámetro lo ponen en tres días y la fecha actual es 17/01/2016, el sistema emitirá la alerta al especialista de producto de todas las líneas que vencen el 20/01/2016.

Para implementar este requerimiento, la fecha de vencimiento a considerar es la fecha que tienen las líneas en el Administrador de Líneas de Crédito. Por lo tanto la solución a este requerimiento implica una comunicación entre los dos sistemas.

Y es así como se definió implementar un Servicio de Transformación de Datos [4] que realice el tema de la notificación y extraer la información necesaria del Administrador de Líneas para las notificaciones.

Se da solución al requerimiento, implementando un Servicio de transformación de datos (DTS) que se ejecute en una tarea automática (Job) de forma diaria y extraiga la información del Administrador de Líneas. Esta información para las notificaciones, es colocada en una tabla nueva que se creó para ese fin específico, la tabla es actualizada siempre que se ejecuta el Servicio de Transformación de Datos, pero cuando se envía la notificación por medio del proceso batch esta notificación es almacenada en el log general de notificaciones del sistema.

DTS PARA NOTIFICACIONES POR VENCIMIENTOS DE LINEAS DE CREDITO, CON DIAS DE ANTICIPACION.
Creado por: Ivayalar, 29 ENERO 2015
Actualizado por Ivayalar 13 Agosto 2015
Se añade funcionalidad de control del recurso

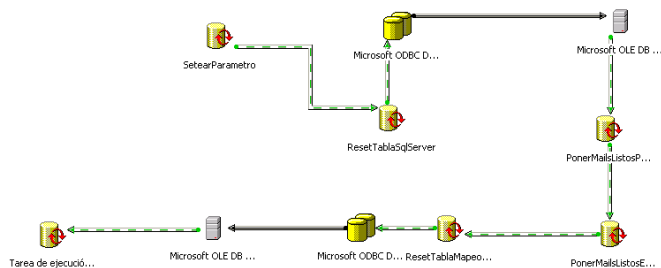


Figura 1.3.13: Breve descripción del Servicio de Transformación de Datos.

Aquí hay una interacción entre dos bases de datos, la base de datos de Crédito Distribuidor y la base de datos del Administrador de Líneas.

CAPÍTULO 2

2. SOLUCION DE LOS REQUERIMIENTOS Y RESULTADOS OBTENIDOS

El Sistema Crédito Distribuidor es un sistema que brinda un apoyo considerable a la gestión de crédito empresarial que da el banco a sus clientes. Entre los resultados más importantes que podemos destacar al implementar las mejoras de Crédito Distribuidor, podemos mencionar:

2.1 Inclusión de nuevas funcionalidades permiten una ampliación de servicios

para las empresas anclas, y clientes en general

Entre los resultados que podemos mencionar tenemos que las empresas anclas ya pueden liberar cupo de su crédito antes que que sus facturas se conviertan en operaciones de crédito, antes las empresas las empresas tenían dos opciones, la primera era solicitar al banco una liberación de cupo manual, o esperar que la factura adeudada se convierta en operación de crédito para que allí, el sistema les permita pagar la operación de crédito y recién allí incrementaban su cupo disponible.

Actualmente existe una nueva opción en el sistema, que les permite a las empresas ancla, cancelar las facturas antes que se conviertan en operaciones de crédito de forma automática.

Otra de las funcionalidades nuevas es la implementación de carga masiva de facturas, esta implementación, dio como resultado que muchas empresas utilicen esta nueva funcionalidad, y carguen facturas al sistema en el menor tiempo posible.

2.2 Mejor control sobre las operaciones de crédito.

La implementación de éstos nuevos controles, benefician a los funcionarios del banco, esto se puede comprobar en la funcionalidad de bloqueo, desbloqueo e inactivación de línea, permiten al funcionario del banco controlar a una empresa en caso de que ésta tenga obligaciones pendientes, y así no le permite cargar facturas nuevas hasta que arregle su situación crediticia.

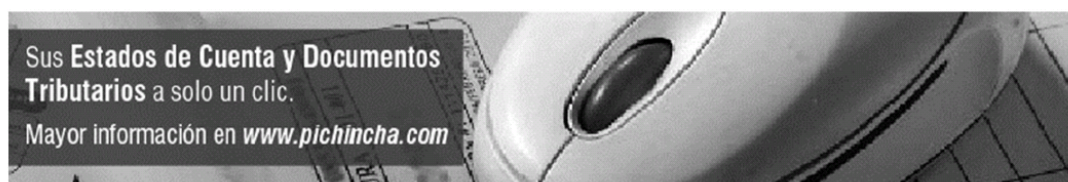
Por otra parte las nuevas notificaciones automáticas, ayudan a informar a los clientes sobre los créditos vigentes y aquellos créditos que están por vencer, de tal forma que simplifican el trabajo al ejecutivo de negocios.

2.3 Cumplimiento de las nuevas normativas necesarias en el giro del negocio gracias a la implementación del Proyecto de Mejoras a la Plataforma de Crédito Distribuidor.

La implementación de la nueva malla contable como mejoras al sistema de Crédito Distribuidor dio como resultado que todas las operaciones del sistema cumplan con las nuevas normativas, entre las cuales tenemos: El cobro de

impuesto solca a la generación de créditos, eliminación de uso de cuentas por cobrar y uso de créditos para las acreditaciones anticipadas que se realizan a las empresas.

Adicionalmente se implementó el tema de base imponible cero, en la comisión que se genera en las acreditaciones anticipadas, esta implementación dio el resultado esperado, y actualmente las comisiones son marcadas con esta particularidad, como lo podemos observar en la figura 2.3.1



| BANCO DEL PICHINCHA C.A. CONTRIBUYENTE ESPECIAL RESOLUCION SRI 5368 DEL 2-JUNIO-1995 FACTURA No: 022-051-000000001 | | CONCILIACIÓN | |
|---|-------------------------|-------------------------------|------------|
| OTORGADO EL : 09-06-2014 | REPRESENTACIÓN | FECHA ULTIMO CORTE (FACTURA) | 15-01-2015 |
| VALIDO HASTA : 09-06-2015 | GRÁFICA SIN VALOR | FECHA DE ESTE CORTE (FACTURA) | 23-01-2015 |
| TIPO DE EMISION: ORIGINAL | TRIBUTARIO | SALDO ANTERIOR | 981,812.56 |
| OFICINA: GUAYAQUIL TORRES PICHINCHA | | + DEPOSITOS / CREDITOS (2) | 706.73 |
| DIRECCION: AV. FRANCISCO DE ORELLANA Y JU GUAYAQUIL | | - CHEQUES / DEBITOS (7) | 508.80 |
| | | + INTERES PERIODO | 0.00 |
| * BASE IMPONIBLE 0 %: 8.78 | VALOR IVA COBRADO: 0.00 | SALDO ACTUAL | 982,019.03 |
| ** BASE IMPONIBLE 12 %: 0.00 | VALOR IVA COBRADO: 0.00 | SALDO PROMEDIO EFECTIVO | 979,237.31 |
| *** VALORES SUJETOS A RETENCION | | | |

Figura 2.3.1: Estado de cuenta donde se muestra la comisión sumada a la base imponible 0 (IVA)

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

1. La utilización de metodologías para toda la etapa de desarrollo como son la Generación de Matriz de Hitos y actividades, pruebas unitarias, por pares, pruebas internas, documentación, generación de librarians, control de cambios, paso a test y pruebas en test, en resumen, toda la metodología implementada, permite que todo el proceso de desarrollo e implementación sea más llevadero.
2. El desarrollo de un documentos de especificaciones técnicas reduce de manera significativa el tiempo de curva de aprendizaje, ya que se establece de forma puntual, específica y clara donde se debe hacer el desarrollo y establece lineamientos claros para el desarrollador permitiendo así y buen trabajo en términos generales.
3. El trabajo en conjunto con áreas y proveedores involucrados, permite el desarrollo de una funcionalidad determinada de manera más rápida y sin errores.

Recomendaciones

1. Para un desarrollo en tiempos aceptables es indispensable la elección correcta de la herramienta y tecnología con la que se va a desarrollar, en el caso de este

proyecto, como se parte de un sistema ya implementado, casi obligatoriamente se debe continuar con la misma tecnología.

2. Es muy importante hacer una revisión de las configuraciones de los servicios de los proveedores si tienen las mismas características en ambiente de test que en ambiente de producción, para que todos los procesos fluyan una vez de realizado el ultimo versionamiento al ambiente de producción.

BIBLIOGRAFÍA

[1] George Lin and Gayla Stewart , AS/400: System, Utilities, Database, and Programming (2nd Edition), 1998.

[2] Francesco Marchioni, Enterprise Application Servers CookBook, 2013

[3] Quick C#, <http://www.codeproject.com/Articles/4300/Quick-C>,
Fecha de consulta Agosto 2014.

[4] Albert Delgado, Servicios de Transformación de Datos en SQL Server 2000, Edición Especial, Madrid, Prentice Hall, 2001, pp. 784.