**Versión 1.6**

Compras, Pagos, ND,NC

**Versión 1.6.1**

Error en Frmpago y cobro, forma de pago. 04/02/2016

Se agregó grafico nuevo 04/02/2016

Se le agregó copiar y pegar correo electrónico en Frm clientes. 07/02/2016

El evento que llama myDataGrid1\_CellContentClick se quita del load, porque es el evento predeterminado o principal del objeto y por eso se ejecutaba dos veces. Pero era porque se llamaba en el load. Indicador ya no tiene uso. Esto es factura y compra, en Cobro ya se había quitado.07/02/2016

Se validó factura, compra, cobro y pago para que no permitan guardar documentos sin detalles.

Se permitió agregar anticipos por frm cobros y frm pagos, se habilitó la contabilización para estos documentos, ver empresa 994

En los formularios ajustes inventario, Ventas, Compras, pagos y cobros se muestran los registros ordenados por codcompracodigo, antes se ordenaban por fecha. Los registros no estabn ordenadose adecuadamente.Esto afectó las entidades de las mismas, en el método obtenertodos() y en la base de datos los metodods que tienen relación con paginación, se cambio fecha por codigo

Se modificaron los eventos de desplazamiento de paginas en ajustes inventario, Ventas, Compras, pagos y cobros, se leagregó dbBindingSource.MoveLast(); para que al cambiar d epagina se mueva al ultimo regiostro. Esto le da mejor percepción de las pagínas al usuario

**Version 1.6.2**

Se inició Modulo de caja y Bancos 15/02/2016

Se agregó Reinciar aplicación en el menú inicio 19/02/2016

Los formularios en el menú se desactivan de acuerdo a los permisos establecidos en el formulariode permisos. Se afectaron los formularios de permisos, acceso al sistema, la entidad permiso, formulario, y los módulos de MisRutinas. Tambien afecto la tabla de Formulario, se reclasificó. Seagregó un treeview en el formulario permiso.19/02/2016.

Se corrigió error al colocar un código de articulo inexsitente en los detalles de factura de venta y compra, se afectó el evento txt\_validating. 20/02/2016

Se le agregó un campo a la lista desplegable de cuenta bancaria, en la parte forma de pago, tanto para Formulario Cobro y Pago, se afectó la entidad de negocio Cuenta Bancaria, Método BuscarTodos, y el evento EditingControl del grid afectado 21/12/2016

Cuando no hay conexión se le agregó un mensaje de direccionamiento a la ventana de configuración de conexiones., es mas amigable que los mesnajes de errores. Se creó un método en la capa de datos, en la entidad datos de empresa y en el load del acceso para hacer la prueba de conexión y direccionar 21/02/2016

**1.6.3**

En el formulario Articulo, no se podrá actualizar el stock, en el formulario caja no se podrá actualizar el saldo, ni tampoco en cuenta bancaria, porque dependen de movimientos., si se dejase que el usuario modifique estos valores, podríamos tener valores no confiables, ya que si alguien abre el formulario de artículos, y en ese momento en esa u otra estación de trabajo se hacen movimientos como ventas, compras o ventas se actualizará en la base de datos en la tabla articuloel stock y será un valor diferente al que está formulario abierto, y si el usuario guarda en el formulario abierto se perdería el valor real por un valor ya obsoleto.

Corregido error en validación de factura de venta y compra. Donde se valida que la factura no quede sin registros.

**1.7**

En todos los procedimientos almacenados(BD) de paginado

Como por ejemplo dbo.spDocumccPaginado, se hizo la modificación de colocarle el where dentro de la consulta base.01/03/2016

.

.

.

SELECT tipodoc,numero,status,codcliente,fecemis,fecvenc,codvendedor,tipodocaso,numerodocaso,numerocontrol,numerocomp,descripcion,codimp,totalexento,totalimponible,porcimp,totalimpuesto,totalneto, saldo,numeroplanillaexp

FROM (SELECT Documcc.\*,

(row\_number()over(PARTITIONby codempresa orderby fecemis))as rownumber

FROM Documcc WHERE codempresa=@codempresa

.

.

.

En el proyecto Aplicacion Contable, en la carpeta Utilidades, Variables Globales, se cambió la constante local Glo\_CantidadRegistroPagina a 100, que alimenta la propiedad pública CantidadRegistroPagina que representa la cantidad de registros que se muestran por página, en los formularios de movimientos donde se usa paginación. Esto hace masligero el sistema con el uso de bases de datos remotas. 01/03/2016

Validación en guardar, formulario de Cobros y Pagos. Que el cliente coincida con el nro de movimiento y también que los movimientos coincidan con el cliente. 02/03/2016

Al formulario frmcobro se le agregó el campo codvendedor.02/03/2016

Se le agrega la validación en activar y anular cobros y pagos esto es en los sp de la BD.Validacion en caso de que sean solo facturas manuales

10/03/2016

-- y si la facura no es manual, si existe en factura

if((selectcount(\*)from Factura where codfactura=@numero and codempresa=@codempresa)>0)

Se le agrega a latabla comprobante el campo descripcióngeneral, y por ello a la entidad en al capa de negocios y al formulario 14/03/2016

Se Desactiva la validación en docum cxc y docum cxp “Documento no Existe”, ya que no está completa. Es temporal, 19/03/2016

Correción en el botón Pagar todos, en la línea

Documcc doc = newDocumcc(myDataGrid1.Rows[i].Cells["gstxttipodoc"].Value.ToString(), Int32.Parse(myDataGrid1.Rows[i].Cells["gstxtnumero"].Value.ToString()), VariablesGlobales.CodEmpresa);

Por

Documcp doc = newDocumcp(myDataGrid1.Rows[i].Cells["gstxttipodoc"].Value.ToString(), Int32.Parse(myDataGrid1.Rows[i].Cells["gstxtnumero"].Value.ToString()), VariablesGlobales.CodEmpresa);

Corrección en los procedimientos almacenados de estado de cuenta de clientes y proveedores. 20/03/2016

Se incorporó un menú estilo Modern UI. Finalizado el 20/03/2016

Validación en formulario de permisos, al guardar, se valida que el perfil se haya seleccionado, que no este en blanco

Finalizado el modulo de caja y bancos. Gracias a Dios 22/03/2016

**1.7.1**

Se agrega el proyecto Herramientas 1.0 para Contable.Net 1.7.1, donde se puede hacer mantenimientoa las tablas de países, estados ciudades, estructura contable, configuración de reportes, y otros. 10/04/2016.

En Documentos de cxc y cxp se agrega la validación para que no permita editar los registros provenientes de cobros, pagos, ni de factura de venta y compra.

Se corrige la validación al anular cobros y pagos que tienen un registro relacionado en pago de tipo Adelanto. Si ese documento(ADEL) que se pretende anular tiene saldo diferente del total neto, es decir, si ya fue cruzado, no se podrá anular. 10/04/2016.

Al anular o reactivar un cobro o pago, si tiene un registro relacionado en documentos (Adelanto), lo anula o reactiva también, 11/04/2016

Al formulario Documentos de cxc y cxp se le agregó anular y reactivar, solo documentos que no provengan de otros módulos( de cobro, pago, factura venta, y compra) 12/04/2016

Los frames de búsquedas de los formularios de movimientos ahora se pueden mover en la pantalla. 12/04/2016

Se cambió la forma de hacer las validaciones del campo Status o anulado, ya que antes se hacía consultando el objeto en el formulario(txtstatus) y ahora se valida haciendo solicitudes a la base de datos. Se hace para evitar errores 14/04/2016

Cuando el sistema Contable.Net se Conectaba a la base de datos en la nube, el procedimiento(PermisosMenu) que se encuentra en la Clase MisRutinas, llamado desde el formulario Acceso, hacía un recorrido por el menú y por cada opción hacía una solicitud en la BD, entonces se tornaba lento el acceso, por lo tanto se procedió a solicitar los datos requeridos(permisos y formularios) en datatables y así hacer validaciones de manera local, el rendimiento es notable.15/04/2016

**HERRAMIENTA para Contable 1.7.1 pasa de la V 1.0.0 a la V 1.0.1**

Se actualiza, ya que la versión 1.0.0 no permitía conexiones dinámicas a las Base de Datos 13/05/2016

**Contable 1.7.2**

Corrección en el procedimiento Contabilizar Documentos, cuando se contabiliza Notas de Créditos cxc con Detalles, se estaba seleccionado la variable @cuentaingreso, cuando sabemos que en la nota de crédito a clientes se usa una cuenta de egreso que justifica una devolución. Esta corrección también se hace sobre el asiento contable de ejemplo en la empresa 944 21/05/2016

Este error afecta la BD, por tanto revisar el Script de actualización 1.7.2, buscar como /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Actualización: 1\*/

De igual manera aplica para el procedimiento [spActualizarInventarioDeleted] Buscar como:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Actualización: 3\*/

En el procedimiento [dbo].[spActualizarInventarioInserted]

La longitud del tamaño del tipo de datos de la variable @codarticulo es de 10, cuando el tamaño máximo es de 50 según la tabla. Esto ocasionaba problemas a la hora de usar códigos de artículos con longitud mayor a 10. Entonces se debe modificar a 50

ALTERPROCEDURE [dbo].[spActualizarInventarioInserted]

@codarticulo varchar(50), 21/05/2016 Este error afecta la BD, por tanto revisar el Script de actualización 1.7.2, buscar como/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Actualización: 2\*/

En los formularios Empresa y Artículos se cambia el uso de la propiedad BackgroundImage por Image, que es la correcta. También se agrega un formulario para darle una vista previa a la imagen. No afecta a la base de datos. 22/05/2016

En los formularios de cobros y pagos se corrige el procedimiento validarsinregistros, el cual arrojaba el mensaje sobre la vaidación donde los detalles no podían quedar sin registros. 01/06/2016

Se Corrige en el botón buscar de los formularios Documentos de Cuentas por cobrar y Documentos de cuentas por Pagar, la validación del campo wtxttipodoc 04/06/2016

Tipodoccc tdo = newTipodoccc(this.wtxttipodoc.Text);

if (this.wtxttipodoc.Text.Trim() != ""&& tdo.ExisteReg(this.wtxttipodoc.Text) == false)

{

MessageBox.Show("El código seleccionado no existe en la base de datos en Tipo de Documentos", "Sistemas Administrativos", MessageBoxButtons.OK, MessageBoxIcon.Information);

wtxttipodoc.Focus();

SendKeys.Send("{HOME}+{END}");

return;

 }

En el procedimiento almacenado RptExistenciaAlmacenesHorizontal se le agrega la condición where codempresa=@codempresa a la consulta que selecciona los almacenes existentes.

setnocounton

SELECTROW\_NUMBER()OVER(ORDERBY codalmacen ASC)AS fila, codalmacen

into #temp1

FROM almacen where codempresa=@codempresa

13/06/2016

**En el Formulario de Comprobantes se agrega un botón en el detalle para explorar las cuentas contables, y asi el usuario pueda hacer una búsqueda de ellas. También se agrega un botón debajo de los detalles para que el usuario pueda actualizar las listas solamente de los detalles. 15/06/2016**

Corrección en el reporte [sprptLibroMayorDet] Libro mayor, en el where:

where c.numerocomprobante=dc.numerocomprobante and c.mes=dc.mes and c.año=dc.año and c.ejercicio=dc.ejercicio and c.codempresa=dc.codempresa and

 dc.codempresa=@codempresa and dc.ejercicio=@ejercicio and dc.año=@añoand dc.mes=@mes

18/06/2016

El tamaño de las variables @strsql y @ssss en el procedimiento spMovimientoBancoPaginadobusqueda fue cambiado a 2000. Cuando se hacen consultas largas usando estas variables se truncaba la sentencia. Esto también puede implicar en el resto de las sentencias de paginado. Se aplicó también a spMovimientoCajaPaginadobusqueda19/06/2016

**Se agregan 3 métodos a los formularios de mantenimiento de tablas maestros y movimientos. Búsqueda rápida por los campos principales, actualización de todos las listas y actualización de todos los datos incluyendo las listas.19/06/2016**

Se mejoró validación en el eventoSelectedIndexChanged del objeto cmbiva del formulario Orden de Pago

privatevoid cmbiva\_SelectedIndexChanged(object sender, System.EventArgs e)

 {

if (Equivalencias.IsNumeric(txtmontomenosretencion.Text) == true&& cmbiva.SelectedValue!=null)

{

Impuesto imp = newImpuesto();

double tasa = imp.ObtenerTasa(dtpfecha.Value, cmbiva.SelectedValue.ToString());

double total = tasa \* double.Parse(txtmontomenosretencion.Text);

 txtmontoiva.Text = String.Format("{0:#,0.00}", total);

}

 }

20/06/2016

**Contable VB.Net 1.7.2**

Se convierte el sistema a VB.NET

**Contable C# 1.8.0**

Se cambia validación en anular, cobrar, pagar y eliminar factura de compra y venta. Cuando se acude a el txtstatus para validar el status. Ahora se acude a la base de datos llamando la clase para hacer referencia al status. 15/07/2016

En las entidades de Compra y Factura, en los métodos Modificar se quitó el campo status. No debe estar allí, causará que se borre el status en las tablas cuando se modifique un registro. 16/07/2016

Se depuró una la trigger update en la tabla detallecompra, no coincidían los campos en el into cursor 16/07/2016

Se le agregó Pedidos al Módulode Cuentas por cobrar, con la funcionalidad que se puede importar desde la factura, maneja pendientes y actualiza stock comprometido.Se afectaron las bases de datos(Archivo Sql: Actualizaciones 1.8.0),también los formularios de factura, empresas. 20/07/2016

Se le agrega un botón en el menú principal para recargar el formulario actual. 23/07/2016

**ACTUALIZACION DE BASE DE DATOS 1.8.0 19/08/2016**

Se corrige el procedimiento SpCrearEmpresa en la instrucciones:

if (@copiarplan=1)

if (@copiarcentro=1)

if (@copiartipo=1)

Quedando de la siguiente manera

if(@copiarplan=1 and(selectcount(\*)from PlanContable where

codempresa=@codempresaorigen)>0 )

if(@copiarcentro=1 and(selectcount(\*)from CentroCostos where

codempresa=@codempresaorigen)>0

)

if(@copiartipo=1 and(selectcount(\*)from tipocomprobante where

codempresa=@codempresaorigen)>0)

Al crear una empresa por el asistente, al copiar estas tablas de una empresa a otra, y no tiene registros para copiar la triger de validación de la tabla enviará un mensaje de error, entonces evitamos este mensaje**19/08/2016 EJECUTAR SCRIPT DE ACTUALIZACION.**

**ACTUALIZACION DE BASE DE DATOS 1.8.0 19/08/2016**

Se le agregó a estos procedimientos la cláusula setdateformat dmy. El formato de fechas de servidores en la nube pueden variar de nuestros servidores locales, por lo que se le asigna el formato a las fechas cuando estamos uniendo cadenas para crear fechas.

RptFlujoDeCaja

spcrearempresa

RptChequeEstandar

 Para detectar los procedimiento se usó

select name,text

fromsysobjects c\_obj leftjoinsyscomments com on c\_obj.id = com.id

where(c\_obj.xtype ='P'or c\_obj.xtype ='TR')and(textlike'%convert(date%'ortextlike'%convert(smalldate%')

**19/08/2016 EJECUTAR SCRIPT DE ACTUALIZACION.**

**ACTUALIZACION DE BASE DE DATOS 1.8.1 22/08/2016**

Se quita el [Contable].[dbo] en los siguientes procedimientos de la base de datos.

Esto arrojará un error cuando la base de datos no se llame Contable

spActualizarInventarioInserted

spCierreContableEjercicio

spActualizarInventarioDeleted

spActualizarInventarioDeletedComprometido

spActualizarInventarioInsertedComprometido

Para localizarlos se usó:

SELECT

so.name,

sc.text

FROM

sysobjects so

INNERJOINsyscomments sc ON so.id = sc.id

WHERE

(so.type='P' so.type='FN' or so.type='TR')AND

sc.textLIKE'%![Contable!].![dbo!].%'ESCAPE'!'

23/08/2016 Se corrige una validación en el formulario Documentos por Pagar, en el botón Anular, evento clic. Se cambióDocumcc doc4 = newDocumcc porDocumcp doc4 = newDocumcp

Esta es bloque de código afectado:

if (txttipodoc.Text.Trim() != ""&&Int32.TryParse(this.txtnumero.Text.Trim(), out s) == true)

if (docc.ExisteReg(this.txttipodoc.Text, Int32.Parse(txtnumero.Text), VariablesGlobales.CodEmpresa) == true)

 {

Documcp doc4 = newDocumcp(this.txttipodoc.Text, Int32.Parse(txtnumero.Text), VariablesGlobales.CodEmpresa);

Compra fac = newCompra();

if (doc4.tipodoc == "FACT")

if (fac.ExisteReg(doc4.numero, VariablesGlobales.CodEmpresa) == true)

 {

MessageBox.Show("El documento no se puede anular, proviene de FACTURA de Compra", "Sistemas Administrativos", MessageBoxButtons.OK, MessageBoxIcon.Error);

txtnumerodocaso.Focus();

SendKeys.Send("{HOME}+{END}");

return;

 }

 }

**1.8.2c# Contable.Net**

Se Agrega cálculo de retenciones, métodos que se encuentran en Factura, Compra, Documentos cxc, Documentos cxp.

Se usan los campos Porcentaje de retención en Clientes y Proveedores.

Se agregan los procedimientos almacenadosspCrearRetencioncxc, spCrearRetencioncx, spcorreDOCCXCRETURN, spcorreDOCCXPRETURN 29/08/2016

**1.8.2c# Contable.Net**

Formulario Documentos cxp, en el boton guarder se corrige la validación siguiente:

Documcc doc4 = newDocumccpor Documcp doc4 = newDocumcp01/09/2016

if (txttipodoc.Text.Trim() != ""&&Int32.TryParse(this.txtnumero.Text.Trim(), out s) == true)

if (docc.ExisteReg(this.txttipodoc.Text, Int32.Parse(txtnumero.Text), VariablesGlobales.CodEmpresa) == true)

 {

Documcp doc4 = newDocumcp(this.txttipodoc.Text, Int32.Parse(txtnumero.Text), VariablesGlobales.CodEmpresa);

Compra fac = newCompra();

if (doc4.tipodoc == "FACT")

if (fac.ExisteReg(doc4.numero, VariablesGlobales.CodEmpresa) == true)

 {

MessageBox.Show("El documento no se puede modificar, proviene de FACTURA de Compra", "Sistemas Administrativos", MessageBoxButtons.OK, MessageBoxIcon.Error);

txtnumerodocaso.Focus();

SendKeys.Send("{HOME}+{END}");

return;

 }

 }

**1.8.2c# Contable.Net 05/09/2016**

En el formulario de Plan Contable, en tipos de Patrimonio, La cuarta opción está repetida, se cambió por “Otras Cuentas de Patrimonio”

**1.8.2c# Contable.Net 05/09/2016**

En el evento CellContentClick de todos los grid del formulario ParametrosContables agregar:

busc.Quien = "FrmParametrosContables";

antes de

busc.ShowDialog();

En el formulario BuscarCuentaContable en el botón y evento

privatevoid cmdseleccionar\_Click(object sender, EventArgs e)

Cambiar if (Quien == "") por

if (Quien == "FrmParametrosContables")

**1.8.2c# Contable.Net 09/09/2016**

Corrección procedimiento almacenado [RptEstadoCuentaProveedor]

Ver archivo de actualziaciones

FROM [dbo].Pago, DetallePagoForma d

where Pago.codproveedor=@codproveedor and(fecha between @fechaini and @fechafin)and pago.codempresa=@codempresa and d.codempresa=@codempresa

and anulado=0 and monto <>0 and pago.codpago=d.codpago and pago.codempresa **=d.codpago**

POR

FROM [dbo].Pago, DetallePagoForma d

where Pago.codproveedor=@codproveedor and(fecha between @fechaini and @fechafin)and pago.codempresa=@codempresa and d.codempresa=@codempresa

and anulado=0 and monto <>0 and pago.montototal<>0 and pago.codpago=d.codpago and pago.codempresa =

**=d.codempresa**

Y en el procedimiento almacenado [RptEstadoCuentaCliente]

Ver archivo de actualizaciones

FROM [dbo].Cobro , DetalleCobroForma d

where cobro.codcliente=@codcliente and(fecha between @fechaini and @fechafin)and cobro.codempresa=@codempresa

and anulado=0 and d.monto <>0 and cobro.codcobro=d.codcobro and cobro.codempresa=d.codempresa

Por

FROM [dbo].Cobro , DetalleCobroForma d

where cobro.codcliente=@codcliente and(fecha between @fechaini and @fechafin)and cobro.codempresa=@codempresa

and anulado=0 and d.monto <>0 and cobro.montototal<>0 and cobro.codcobro=d.codcobro and cobro.codempresa=d.codempresa

**1.8.2c# Contable.Net 10/09/2016**

Formulario Estado de Cuentas de Proveedores, Corregido el título del mismo.

**1.8.2c# Contable.Net 10/09/2016**

En el Formulario Factura de Compras, en el método aceptar, se usa una función llamada validarsinregistros, buscarla y corregir:

// valida monto abono

if (myDataGrid1.Rows[i].Cells[4].Value == null)

 cond2 = 0;

else

if (double.Parse(myDataGrid1.Rows[i].Cells[4].Value.ToString()) <= 0)

cond2 = 0;

Por

// valida monto abono

if (myDataGrid1.Rows[i].Cells[5].Value == null)

 cond2 = 0;

else

if (double.Parse(myDataGrid1.Rows[i].Cells[5].Value.ToString()) <= 0)

cond2 = 0;

**1.8.2c# Contable.Net 10/09/2016**

En la TRIGGER [dbo].[InsDetalleDocumcp] de la tabla DetalleDocumcp se corrije:

ifisnull((selectcount(\*)from inserted i,**documcc** t

where i.tipodoc= t.tipodoc and i.numero=t.numero and i.codempresa=t.codempresa),0)<>@totalregistrosafectados

begin

 RAISERROR

 (N'Nro de Documento no existe. Uno o mas registros no están relacionado con la tabla de Documento',10,1)

 if@@trancount>0

 begin

 rollbacktransaction;

 RETURN;

 end

end

Por

ifisnull((selectcount(\*)from inserted i,**documcp** t

where i.tipodoc= t.tipodoc and i.numero=t.numero and i.codempresa=t.codempresa),0)<>@totalregistrosafectados

begin

 RAISERROR

 (N'Nro de Documento no existe. Uno o mas registros no están relacionado con la tabla de Documento',10,1)

 if@@trancount>0

 begin

 rollbacktransaction;

 RETURN;

 end

end

**1.8.2 c# Contable.Net 10/09/2016**

En la TRIGGER [dbo].[UpdDetalleDocumcp] de la tabla DetalleDocumcp

Se corrije:

ifisnull((selectcount(\*)from inserted i,**Documcc** t

where i.tipodoc=t.tipodoc and i.numero=t.numero and i.codempresa=t.codempresa),0)<>@totalregistrosafectados

begin

 RAISERROR

 (N'Nro de Documento no existe. Uno o mas registros no están relacionado con la tabla de Documentos',10,1)

 if@@trancount>0

 begin

 rollbacktransaction;

 RETURN;

 end

end

Por

ifisnull((selectcount(\*)from inserted i,**Documcp** t

where i.tipodoc=t.tipodoc and i.numero=t.numero and i.codempresa=t.codempresa),0)<>@totalregistrosafectados

begin

 RAISERROR

 (N'Nro de Documento no existe. Uno o mas registros no están relacionado con la tabla de Documentos',10,1)

 if@@trancount>0

 begin

 rollbacktransaction;

 RETURN;

 end

end

**1.8.2 c# Contable.Net 11/09/2016**

Se creó una nueva tabla, llamada ImpuestoParametro, en el que se guardan los valores de impuesto general y exento. Estos valores pueden ser actualizados por el formulario Tipo de Impuestos.

Se hicieron modificaciones en:

Función FCodigoImpuestoExento en sql server

Función FCodigoImpuestoGEneral en sql server

Entidad TipoImpuesto en capa de negocio

Formulario FrmTipoImpuesto en capa de interfaces.

**1.8.2c# Contable.Net 15/09/2016**

El tipo de datos de los campos stockcomprometido de las tablas Articulo y Stalma deben ser cambiados de int a decimal(18,4). Ver script de actualización en la parte:

GO

--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*reparando stockcomprometido de int a decimal. No se pierden los valores

--elimina la restriccion

ALTERTABLE [dbo].[Articulo] DROPCONSTRAINT [DF\_\_Articulo\_\_stockc\_\_66C02818]

--cambia el tipo de datos

altertable articulo altercolumn stockcomprometido decimal(18,4)notnull

--agrega la restricccion con el nombre que tenia anteriormente

ALTERTABLE [dbo].[Articulo] ADDCONSTRAINT [DF\_\_Articulo\_\_stockc\_\_66C02818] DEFAULT ((0))FOR [stockcomprometido]

GO

--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*reparando stockcomprometido de int a decimal. No se pierden los valores

--elimina la restriccion

ALTERTABLE [dbo].STAlma DROPCONSTRAINT [DF\_\_STAlma\_\_stockcom\_\_65CC03DF]

--cambia el tipo de datos

altertable STAlma altercolumn stockcomprometido decimal(18,4)notnull

--agrega la restricccion con el nombre que tenia anteriormente

ALTERTABLE [dbo].STAlma ADDCONSTRAINT [DF\_\_STAlma\_\_stockcom\_\_65CC03DF] DEFAULT ((0))FOR [stockcomprometido]

GO

**1.8.2c# Contable.Net 21/09/2016**

En el Formulario de Facturade Venta, en el botón de “Cobrar”, se agregó y modificó el siguiente código, que busca el primer registro perteneciente a una caja, que luego se envía al formulario de Cobros en el grid de forma de pago. Y la instrucción fa.fecemis.ToString("dd/MM/yyyy") que envía la fecha del registro y evita un mensaje de error de tipo fecha incorrecto porque se enviaba vacío

Caja caj = newCaja(VariablesGlobales.CodEmpresa);

if (caj.CantReg(VariablesGlobales.CodEmpresa) == 0)

 {

MessageBox.Show("No existen cajas creadas. Cree al menos una caja para usar esta opción");

return;

 }

string[] filagrid2 = { "EFEC", "", "", fa.fecemis.ToString("dd/MM/yyyy"), "\*", "0", caj.codcaja, "N/A", String.Format("{0:#,0.00}", variabletotal) };

**1.8.2c# Contable.Net 21/09/2016**

En el Formulario de Compra, en el botón de “Pagar”, se agregó y modificó el siguiente código, que busca el primer registro perteneciente a una caja, que luego se envía al formulario de Pagos en el grid de forma de pago. Y la instrucción fa.fecemis.ToString("dd/MM/yyyy") que envía la fecha del registro y evita un mensaje de error de tipo fecha incorrecto porque se enviaba vacío

Caja caj = newCaja(VariablesGlobales.CodEmpresa);

if (caj.CantReg(VariablesGlobales.CodEmpresa) == 0)

 {

MessageBox.Show("No existen cajas creadas. Cree al menos una caja para usar esta opción");

return;

 }

string[] filagrid2 = { "EFEC", "", "", fa.fecemis.ToString("dd/MM/yyyy"), "\*", "0", caj.codcaja, "N/A", String.Format("{0:#,0.00}", variabletotal) };

**1.8.2c# Contable.Net 25/09/2016**

Se agrega Despacho en el módulo de Cuentas Por cobrar, y varios reportes.

**1.9.0c# Contable.Net 30/09/2016**

Se agregó un formulario emergente en el maestro de Plan Contable, para mostrar las cuentas en forma de árbol.

**1.9.0c# Contable.Net 30/09/2016**

En el Formulario Orden de Pago.

Se corrige el calculo de impuestos en los eventos de los siguientes objetos

cmdaceptar\_Click

cmbiva\_SelectedIndexChanged

txtmonto\_Validated

double total = tasa \* double.Parse(txtmontomenosretencion.Text);

**Por**

double total = (tasa \* double.Parse(txtmontomenosretencion.Text))/100;

**1.9.0c# Contable.Net 27/10/2016**

En el formulario de Factura de Ventas, en el grid detalles, se le agregó la columna de Item, al final.

**1.9.0c# Contable.Net 29/10/2016**

En el procedimiento almancenado [spAnularPedido]

Se corrige al sumar lo comprometido, cuando se debe restar al anular, la correcion esta en el archivo script de actualización a la versión 1.9.0

**1.9.0c# Contable.Net 06/11/2016**

En los métodos de los formularios FrmBuscarPedido y FrmBuscarOrden se gragegó al final de cada método “FormFactura.sumar();” y “FormCompra.sumar();”

**1.9.0c# Contable.Net 06/11/2016**

Se integra Orden de Compras y sus informes. Los cambios en las bases de datos se encuentran en el archivo de Actualización.

**1.9.0c# Contable.Net 06/11/2016**

Las compras no se despachan, por defecto estaba tomando el status PDE, por despachar. Se procedió a blanquear el campo donde estuviera dicho valor. Además en guardar ya no se usa PDE, sino se guarda con ‘’ sin valor. (FrmCompra)

También se quitó en el formulario cualquier rastro de ‘DES’ Despachado

Con esta sentencia, que se encuentra también en el archivo de actualizaciones 1.9.0, se puede corregir el valor en la tabla compra

/\*correccion las compras no se despachan\*/

update compra setstatus=''wherestatus='PDE'

**1.9.0c# VS2015 Contable.Net 01/12/2016**

Se migra a la version Visual Studio 2015

**2.0.0c# VS2015 Contable.Net 07/12/2016**

Se Corrige el reporte RptActivosDepreciacion.Rpt. Se verificó y corrigió el reporte.

**2.0.0c# VS2015 Contable.Net 07/12/2016**

Se agregó un reporte con máscara sobre el plan contable.

**2.0.0c# VS2015 Contable.Net 17/12/2016**

EL campo Correlativo PRES en El formulario Empresa pasa a ser Correlativo COTIZ, de Cotización, esto en la pestaña Correlativos de CXC.

**2.0.0c# VS2015 Contable.Net 21/01/2017**

En los formularios FrmBuscarPedido, FrmBuscarOrden,FrmBuscarFactura se corrigieron los dos métodos “Importar Item Seleccionados…”, los cuales tenían errores al multiplicar la cantidad seleccionada de cada ítem (escrito por el usuario) por el costo o precio**.**

**En los mismos formularios, en los dos métodos donde se importa hacia un documento nuevo, ya sea factura, pedido o orden de compra, se reemplaza**

FormFactura.dtpfecemis.Value = ped.fecemis;

Por:

FormFactura.dtpfecemis.Value = DateTime.Now;

Ya que se tiene que colocar la fecha actual en el documento(formulario) destino.

**2.0.0c# VS2015 Contable.Net 21/01/2017**

Se crearon trigger en la tabla detallepagodocumcp y detallecobrodocumcp que se encargan de incrementar o decrementar los saldos de los documentos asociados. Por esto se quitan los procedimientos en el método guardar de los formularios pagos y cobros y también de las entidades de los mismos en la capa de negocios.

**2.0.0c# VS2015 Contable.Net 03/02/2017**

En los frmbuscarcompra, frmbuscarorden,frmbuscarfactura y frmbuscarpedido se hacen los siguientes cambios: Value por FormattedValue.

Cuando se accede a una celda de un grid en la que el usuario haya modificado el valor a cero, este cero es interpretado como nada, vacío, entonces entraría en error. Ver ejemplo

...decimal.Parse(dataGridView2.Rows[i].Cells["gstxtseleccion"].Value.ToString…

…double.Parse(dataGridView2.Rows[i].Cells["gstxtseleccion"].Value.ToString…

Por lo que Cambia por:

...decimal.Parse(dataGridView2.Rows[i].Cells["gstxtseleccion"].FormattedValue.ToString…

…double.Parse(dataGridView2.Rows[i].Cells["gstxtseleccion"].FormattedValue.ToString…

FormattedValue desde un principio se usaba en los grid de los detalles de movimientos donde el usuario interactúa con las celdas.

**2.0.0c# VS2015 Contable.Net 04/02/2017**

Finaliza la inserción de Devoluciones de Compras y Devoluciones de Facturas de Ventas

**2.0.0c# VS2015 Contable.Net 05/02/2017**

Estaba ocurriendo un error con las instalaciones del sistema en sistemas operativos de 64 Bits, precisamente en la permisologia de las carpeta c:\Archivos de programas (x86)\RivasSistemas.

Mostraba un mensaje al tratar de crear los archivos para recordar datos de conexión, Hay tres formas para solucionar el problema:

La primera es agregar los archivos Recuerda.bin, Reporte.txt y RutaServidor.txt en el sistema de archivos del proyecto instalador, en la carpeta base de la aplicación, al instalarse serán copiados y tendrán permisos de edición y no será necesario crearlos. Actualmente no están incluidos.

La segunda forma, es instalar el sistema y darle todos los permisos a la carpeta RivasSistemas al usuario “usuario/……..” Esta es la manera de resolver hasta la versión 1.9.0. y lo resuelve el usuario.

La tercera, es redirigir la creación y ubicación de estos archivos a la carpeta Contable.Net en Mis documentos, esto se hizo en la versión 2.0.0. Lo mismo se hizo para el proyecto Herramientas de Contable.Net con el nombre de la carpeta “HerramientaContable” en mis documentos, en su versión para Visual Studio 2015

**2.0.0c# VS2015 Contable.Net 08/02/2017**

En los procedimientos almacenados de paginación donde se usaba una tabla temporal global fueron corregidos. Ya que el uso de tablas temporales globales no es recomendable en estos procedimientos. Ejemplo spAjustePaginadobusqueda

**2.0.0c# VS2015 Contable.Net 16/02/2017**

Se activaron los eventos siguientes, parece estaban comentados:

publicvoid txtcondpago\_Validating(object sender, System.ComponentModel.CancelEventArgs e)

{

int s;

 s = 0;

if (Int32.TryParse(txtcondpago.Text, out s) == true)

dtpfecvenc.Value = dtpfecemis.Value.AddDays(Int32.Parse(txtcondpago.Text));

}

privatevoid dtpfecemis\_Validating(object sender, System.ComponentModel.CancelEventArgs e)

{

int s;

 s = 0;

if (Int32.TryParse(txtcondpago.Text, out s) == true)

 dtpfecvenc.Value = dtpfecemis.Value.AddDays(Int32.Parse(txtcondpago.Text));

sumar();

 }

**2.0.0c# VS2015 Contable.Net 17/02/2017**

Se agregó Cotizaciones, que puede ser llamado desde Factura o desde Pedido, no tiene que ver con los stock comprometidos a diferencia de los pedidos.

**1.0.0 Xamarin Forms c# VS2015 Contable.Net App 24/02/2017**

Se crea la Aplicación para Móviles, Android, IOS y Windows Phone, aunque solo está probado para Android, se pueden hacer cambios y pruebas para que funcione para las otras plataformas. Esta aplicación en su primera Edici+on, permite ver gráficos de Compras, Ventas y Stock de Artículos con su imagen incluida.

Código Fuente disponible, y archivo distribuible en [www.rivasssitemas.com](http://www.rivasssitemas.com)

**2.0.0c# VS2015 Contable.Net 25/02/2017**

Visual Studio 2015 no viene con la herramienta para hacer proyectos de instalación, se descargó del link:

<https://marketplace.visualstudio.com/items?itemName=VisualStudioProductTeam.MicrosoftVisualStudio2015InstallerProjects>

Se hizo nuevamente el instalador, ahora el proyecto se llama instalador2, ya que le proyecto instalador hecho con Install Shields para VS2015 arroja problemas de consumo de memoria en tiempos de diseño. Es mas sencillo trabajar con el proyecto de Instalación de Visual Studio que con el Install Shield, aunque éste último tenga mas opciones.

**2.1.0c# VS2015 Contable.Net 18/03/2017**

Se agregaron los campos [titulotributario1] Y [titulotributario2] en la tabla y formulario Empresa. Estos serán los títulos de los campos rif y nit, de tal manera que en los formularios clientes, proveedores, beneficiarios, vendedores, Asistente de Empresas y empresas se usarán valores de etiqueta personalizados por el usuario. Esto aplica también para los reportes. Esta actualización nos evita tener que modificar los formularios y los reportes con respecto a los campos rif y nit, que no se aplican o tienen otros nombres en los distintos países.

Nota: el destino de los reporte en tiempos de desarrollo ya no será C:\contable14vs2010\Contable.Net version 1.4 modificaciones\AplicacionContable\CrystalReports

Estarán en la ruta:

C:\contable14vs2010\Contable.Net version 1.4 modificaciones\AplicacionContable\AplicacionContable\bin\Debug\Reportes

**2.1.0c# VS2015 Contable.Net 26/03/2017**

En el método aceptar de movimientos de bancos y cajas se corrije lo siguiente

En Movimientos de Bancos, esto

if (txtorigen.Text.Trim() != "")

 txtorigen.Text = "BAN";

Por

if (txtorigen.Text.Trim() == "")

 txtorigen.Text = "BAN

y en Movimiento de Cajas, esto

if (txtorigen.Text.Trim() != "")

 txtorigen.Text = "CAJ";

Por

if (txtorigen.Text.Trim() == "")

 txtorigen.Text = "CAJ";

No permitía crear movimientos nativos de ese módulo.

**2.1.0c# VS2015 Contable.Net 26/03/2017**

Se agregó eliminar grupos en el SpEliminarEmpresaCompleta

**2.1.0c# VS2015 Contable.Net 26/03/2017**

Se agregó una validación en el Frm Plan Contable, en el método “Ver Arbol”

ConfiguracionContable con = newConfiguracionContable(VariablesGlobales.CodEmpresa);

if (con.nivel1 == 0 || con.nivel2 == 0 || con.nivel3 == 0 || con.nivel4 == 0)

 {

MessageBox.Show("Precaución, no se han asignado los valores de niveles en Configuración Contable");

return;

 }

**2.1.0 c# VS2015 Contable.Net 26/03/2017**

Se agregó el método “Actualizar Listas” en Artículos.

**2.1.0c# VS2015 Contable.Net 26/03/2017**

Al contabilizar, en spcontabilizardocumentos en la contabilización de pagos, en la asignación de la descripción dice “Cobros”, se cambió por “Pagos”. Este procedimiento no se colocó en el script de actualización 2.1.0. Se debe cambiar manualmente.

**2.1.0c# VS2015 Contable.Net 01/04/2017**

Se creó la función dbo.FUltimoCosto para que actualice los costos según entrada x ajuste, compras, devoluciones en compras o anular compras. La actualización se hace sobre el campo Costo1 de Articulos.

La instancia se debe agregar manualmente en todas las trigger de renglones de compras y ajustes, tal como se muestra acá, debe ir en la última parte, antes de cerrar el cursor.

~~update articulo set costo1= dbo.FUltimoCosto(SYSDATETIME(),@codarticulo,@codempresa)~~

update articulo set costo1= dbo.FUltimoCosto(SYSDATETIME(),@codarticulo,@codempresa)where codarticulo=@codarticulo and codempresa=@codempresa

(corregido el 01/05/2017)

Nota: para entradas por ajustes y compras en un mismo dia, tienen prioridad los ajustes.

**2.1.0c# VS2015 Contable.Net 01/04/2017**

Se agregó: No se puede anular una factura de venta si el Usuario no tiene Perfil de “Administradores”

**2.1.0c# VS2015 Contable.Net 01/04/2017**

Se agregaron .ToUpper() en distintas partes del código fuente donde no estaba. Esto es para asegurar la integridad de los valores de códigos en las tablas.

**2.1.0c# VS2015 Contable.Net 15/04/2017**

Se corrigió esto en el siguiente método: DataGridView1\_SelectionChanged de los formularios de devoluciones:

if (DGV.Rows[DGV.CurrentRow.Index].Cells["column5"].Value.ToString() == "ANU")

 txtstatus.Text = "ANULADO";

else

 txtstatus.Text = "";

**2.1.0 c# VS2015 Contable.Net 16/04/2017**

En el procedimiento almacendado [spCierreContableMensual]

de la base de datos, en la parte final se agregó

--eliminar donde la cuenta movimiento tenga saldo antewrior, debe haber y saldo en cero

deletefrom Balance where codempresa=@codempresa and ejercicio=@ejercicio and mes=@mes and año=@año

and saldoanterior=0 and debe=0 and haber=0 and saldo=0

Si una cuenta queda con ceros en las cuatro columnas no vale la pena mostrarlo. Esta actualización se tiene que hacer sobre la base de datos directamente. No está en el archivo sql de actualizaciones

**2.1.0 c# VS2015 Contable.Net 20/04/2017**

En el método sumar() de los formularios Facturas, Compras,Devoluciones de compras y ventas, Cotizaciones, Pedidos, Despacho y Orden de Compras se agregó al inicio

TipoImpuesto tip = newTipoImpuesto();

String codimpexento = tip.CodigoImpuestoExento();

Para poder reemplazar mas adelante en el mismo método:

if (codimp == “6”)// aprovechamos para obtener lo exento,

Por

if (codimp == codimpexento)// aprovechamos para obtener lo exento,

**2.1.0 c# VS2015 Contable.Net 23/04/2017**

En el método Sumar() de los formularios, Factura, Compras, Orden de Compras, Devoluciones de Compras y Ventas, Cotizaciones, Pedidos, Despachos se hicieron correcciones en cuanto al cálculo del descuento general. A continuación una muestra de como quedó el procedmiento sumar() en el FrmFactura:

publicvoid sumar()

 {

try

 {

TipoImpuesto tip = newTipoImpuesto();

String codimpexento=tip.CodigoImpuestoExento();

double porcdescuento = 0;

double montodescuento = 0;

double varsubtotal = 0;

double varimpuesto = 0;

varexento = 0;//varible global de la clase

Impuesto imp = newImpuesto();

DataTable DTImpuesto2 = imp.ObtenerTodos(dtpfecemis.Value);

DataRow[] Filas= newDataRow[0];

DataRow fila=null;

int i = 0;

while (i < myDataGrid1.Rows.Count)

 {

int cond2 = 0;

if (myDataGrid1.Rows[i].Cells[10].Value == null)

 cond2 = 1;

else

if (myDataGrid1.Rows[i].Cells[10].Value.ToString() == "False")

 cond2 = 1;

else

if (myDataGrid1.Rows[i].Cells[10].Value.ToString() == "True")

 cond2 = 0;

if (myDataGrid1.Rows[i].Cells[9].Value != null&& cond2 == 1)

 {

double valor = Convert.ToDouble(myDataGrid1.Rows[i].Cells[9].Value.ToString());

//acà el descuento general para cada item 23042017 FJRC

 porcdescuento = 0;

 montodescuento = 0;

if (Equivalencias.IsNumeric(txtporcdesc.Text) == true)

{

porcdescuento = double.Parse(txtporcdesc.Text);

 montodescuento = (valor \* porcdescuento) / 100;

 }

 valor = valor - montodescuento;

 varsubtotal = varsubtotal + valor;

// hay que intenbtar usar el datatable para no concurrrir a la base de dfatos tantas veces.\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

String VARI="";

double tasa = 0;

if (myDataGrid1.Rows[i].Cells[4].Value != null)// esta linea es porque se validaba filas nulas

 {

 VARI = myDataGrid1.Rows[i].Cells[4].Value.ToString();//obtenemos el cod del imp

 Filas = (DataRow[])DTImpuesto2.Select("codimp= '" + (String)((VARI == null) ? "''" : VARI) + "'");//lo biscamos en el dataset

fila = (DataRow)Filas[0];//ahora lo pasamos a unsa fila. ytamiben su pudo tasa=double:Parse(Filas[0]["tasa"].ToString());

if (Filas.Length > 0)

 {

tasa = double.Parse(fila["tasa"].ToString());//obtenemos el vaor

String codimp;

codimp = fila["codimp"].ToString();//obtenemos el tipo de impuesto

if (codimp == codimpexento)// aprovechamos para obtener lo exento,

 varexento = varexento + valor;

 }

 }

 varimpuesto = varimpuesto + (valor \* tasa / 100);

//este precedimiento es efectivo, directo y mas claro, pero en este procedeimeitno hace mucha concurrencia a la BD, tantas filas tenga una factura, y tantas veces se valide.

//if (myDataGrid1.Rows[i].Cells[4].Value != null)// esta linea es porque se validaba filas nulas

//varimpuesto = varimpuesto +((valor\* imp.ObtenerTasa(dtpfecemis.Value, myDataGrid1.Rows[i].Cells[4].Value.ToString()))/100);

}

 i = i + 1;

 }

double vartotal = varsubtotal + varimpuesto;

 txtsubtotal.Text = String.Format("{0:#,0.00}", varsubtotal);

 txtdescuento.Text = String.Format("{0:#,0.00}", montodescuento);

 txtporcdesc.Text = String.Format("{0:#,0.00}", porcdescuento);

txtimpuesto.Text = String.Format("{0:#,0.00}", varimpuesto);

txttotal.Text = String.Format("{0:#,0.00}", vartotal);

return;

 }

catch (Exception ex)

 {

MessageBox.Show("Precaución19. " + ex.Message, "Sistemas Administrativos", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

 }

**2.1.0 c# VS2015 Contable.Net 23/04/2017**

En el procedimiento [spContabilizarDocumentos] de la base de datos se tomó en cuenta los descuentos generales de las facturas de compras y ventas, también se tomó en cuanta al hacer la contabilización de una nc/devolución para reversar el descuento de una factura de compra o venta. Si se desea actualizar se debe proceder manualmente copiando el script de la versión 2.1.0

**2.1.0 c# VS2015 Contable.Net 23/04/2017**

El libro de Compras y Ventaspredeterminado tenían errores al calcular el iva, también se le agregó la resta en caso de que la factura o compra tenga descuento general. Se modificaron los procedimientos [RptLibroCompras] y [RptLibroVentas] en la base de datos, no se tocó el rpt. Si se desea actualizar se debe proceder manualmente copiando el script de la versión 2.1.0

**2.1.1 c# VS2015 Contable.Net 29/04/2017**

En el formulario de “Buscar Pedidos” que se encuentra al importar pedidos desde “Facturas”.

Precisamente en los métodos:

cmdseleccionaritemactual\_Click

cmdseleccionaritemnueva\_Click

En donde se usaba gstxtcosto se cambió porgstxtprecio

**2.1.1 c# VS2015 Contable.Net 29/04/2017**

En los formularios de Documentos CxC y Documentos CxP al guardar datos, se corrige la validación de la siguiente manera:

if ((DateTime.Parse(dtpfecemis.Value.ToString("dd/MM/yyyy")) >DateTime.Parse(dtpfecvenc.Value.ToString("dd/MM/yyyy"))))

ya que la antigua

if (dtpfecemis.Value> dtpfecvenc.Value)

en algunas máquinas es inestable, por cuestión de las horas que recoge el tipo de datos DateTime.

**2.1.1 c# VS2015 Contable.Net 29/04/2017**

En el formulario de acceso FrmAcceso, se modificó la validación

FrmBuscarServidor bus = newFrmBuscarServidor();

if (Ubicacion.ExisteUbicacionServidor() == false)

bus.Show(this);

Por:

if (Ubicacion.ExisteUbicacionServidor() == false)

{

FrmBuscarServidor bus = newFrmBuscarServidor();

bus.acc = this;

bus.Show(this);

return;

 }

Esta validación funciona para que cuando los archivos txt que guardan la información de la conexión y otros datos no exista (la primera vez que se instala el sistema), entonces el Sistema no trate de conectarse a la nada, sino que pase directamente al formulario de conexiones(FrmBuscarServidor). Luego de que el usuario establece la conexión en el formulario, le corresponde presionar el botón “salir” o “Salir de esta Ventana. Entrar al Sistema”(FrmBuscarServidor), allí nos daba error con el antiguo código, con la corrección depuramos tal error.

**2.1.1 c# VS2015 Contable.Net 30/04/2017**

Error en el libro analítico al mostrar cuentas 3 ó 4 ó 5 ó 6

Se corrigió una conversión en los procedimientos:

[spRptLibroAnaliticoEnc]

[sprptLibroAnaliticoDet]

En la conversión

convert(int,left(@cuentas, @Posicion - 1))

se cambió por

convert(varchar(20),left(@cuentas, @Posicion - 1))

sucedía que iban cuentas contables en forma entera al reporte donde su tamaño como entero no era soportada.

Se procedió a actualizar el informe(rpt),

Se ordenaron los registros de detalle por fecha en el informe.

Hacer manualmente

**2.1.1 c# VS2015 Contable.Net 30/04/2017**

En el FrmBuscarFactura, que se usa para importar facturas de ventas en Devoluciones se le agregó:

FormDevolucionFactura.cmbvendedor.SelectedValue = ped.codvendedor;

Que va debajo de:

FormDevolucionFactura.cmbclientes.SelectedValue = ped.codcliente;

Esto en la parte de Devolución de Facturas, para que actualice el nombre del vendedor en el frmdevolucionfactura

**2.1.1 c# VS2015 Contable.Net 30/04/2017**

Muy Importante,

Se crearon las trigger:

[DelDetalleCobroDocumcc]

[InsDetalleCobroDocumcc]

[UpdDetalleCobroDocumcc]

Para la Tabla

DetalleCobroDocumcc

Las triggers se pueden encontrar en el archivo de actualización. Estas trigger se habían creado para DetallePagoDocumcp, pero olvidamos hacerlo para cobros, ella se encarga de los saldos en la hora de cruces de documentos.

**2.1.1 c# VS2015 Contable.Net 30/04/2017**

Desde devoluciones de ventas, cuando se agrega un registro cruzado entre una N/CR y una FACT en cobros no se estaba guardando el código del vendedor.

En agregar

cobro.codvendedor = mae.codvendedor;

En modificar

cobro.codvendedor = maeaux.codvendedor;

**2.1.1 c# VS2015 Contable.Net 30/04/2017**

Cuando se quiere imprimir un cheque desde el Frmpago

El mensaje:

MessageBox.Show("El número de pago no es numérico");

Se cambió por:

MessageBox.Show("El número de pago no es numérico. Guarde primero el Pago para imprimir el Cheque");

**2.1.1 c# VS2015 Contable.Net 01/05/2017**

En las trigger de DetalleAjustes y DetalleCompras se corrige la asignacion del ultimo costo al articulo. Esto se debe corregir manualmente en la Base de datos

Cambiar este:

update articulo set costo1= dbo.FUltimoCosto(SYSDATETIME(),@codarticulo,@codempresa)

Por este:

update articulo set costo1= dbo.FUltimoCosto(SYSDATETIME(),@codarticulo,@codempresa)where codarticulo=@codarticulo and codempresa=@codempresa

**2.1.1 c# VS2015 Contable.Net 01/05/2017**

Se agregó el campo ultimocosto ala tabla detallefactura, pero sin implementarse. Ver archivo de actualización(script sql).

**2.1.2 c# VS2015 Contable.Net 08/05/2017\***

En el evento txtcodarticulo\_KeyPress del formulario de “Articulo” se le agregó:

ptrpictureBox1.Image = null;

para limpiar la imagen al agregar un registro Nuevo, ya que solía quedar la imagen de otro artículo.

**2.1.2 c# VS2015 Contable.Net 08/05/2017**

En el evento txtcodempresa\_KeyPress del formulario de “Empresa” se le agregó:

ptrpictureBox1.Image = null;

para limpiar la imagen al agregar un registro Nuevo, ya que solía quedar la imagen de otra empresa.

**2.1.2 c# VS2015 Contable.Net 08/05/2017**

En el menú principal de contable.Net, en el módulo “Reportes”, “Cuentas por Pagar”, “Listado de Devoluciones…” fue corregido el Texto.

**2.121 c# VS2015 Contable.Net 08/05/2017**

En el evento myDataGrid1\_CellValidated de los formularios de Factura, Compras, Orden de Compras, Devoluciones de Compras y Ventas, Cotizaciones, Pedidos, Despachos se agregó la validación:

//agregado el 07052017, si el usuario salta el descuento(5), al validarse el campo 5,7,8 lo ponemos en 0 si esta null

if (myDataGrid1.Rows[e.RowIndex].Cells[5].Value == null&& (e.ColumnIndex == 5 | e.ColumnIndex == 7 | e.ColumnIndex == 8))

myDataGrid1.Rows[e.RowIndex].Cells[5].Value = String.Format("{0:#,0.00}", 0.0);

Pasaba que el usuario se saltaba el campo en el grid correspondiente a porcentaje de descuentos, y aunque mostraba el valor “0”, dentro de la `propiedad value era null, (porque no había sido seteado, null es el valor por defecto de las celdas de un grid) que generaba un mensaje de error al guardar el registro.

5 es el index usado para hacer referencia al campo de descuento en el grid (detalles del formulario). Cuando se editen los campos 5(desc),7(cant) y 8(precio) se validará que descuentos este en 0.00 en caso de que esté null

Nota: en el código debajo del agregado, también se hace una corrección.

**2.1.2 c# VS2015 Contable.Net 08/05/2017**

En el evento myDataGrid1\_KeyDownde los formularios de Factura, Compras, Orden de Compras, Devoluciones de Compras y Ventas, Cotizaciones, Pedidos, Despachos se encerró en un try catch, pero que en caso de haber error en esa rutina no nos va amostrarmensaje alguno.

Se hizo con la intención de silenciar un error que nos arrojaba el sistema Contable.Net ya que en caso de que el usuario dejase (como por ejemplo) la caja de almacén en blanco y trata de avanzar de ítem, el evento myDataGrid1\_RowValidatingmuestra un mensaje notificándolo, si el usuario elige NO, para mantenerse en el ítem y corregir envía el cursor a la caja de texto de almacén del ítem con el error, que en ese momento es el actual, y resulta que al mismo tiempo el evento myDataGrid1\_KeyDown está enviando el cursor a la caja 0 (código de articulo) de la siguiente fila, esto hace entrar en conflicto al sistema. Se silencia con el try catch. Se intentó corregir el error revisando la propiedad “texterror” de las columnas del grid pero no dio resultado.

**2.1.2 c# VS2015 Contable.Net 13/05/2017**

Los procedimientos de Libro de Ventas y Libro de Compras se le hicieron actualizaciones con respecto al número de comprobante. En los reportes se hicieron modificaciones en los tamaños de letras de los label y también en los campos de ambos libros. Se modificaron los procedimientos de los mismos. Procedimientos: spRptLibroCompras,spRptLibroventas. Reportes: RptLibroCompras.rpt, RptLibroVentas.rpt

**2.1.2 c# VS2015 Contable.Net 13/05/2017**

Se Agregó el botón con el método “Imprimir Formato” en el Formulario de Documentos de Cuentas por cobrar y pagar. Para imprimir reportes personalizados. Los campos de parámetros que envía son codempresa,numero, tipodoc.

**2.1.2 c# VS2015 Contable.Net 13/05/2017**

Se agregó el reporte Comprobante de Retención IVA, para Venezuela, se puede apreciar en reportes personalizados. Si no lo tiene copie el Rpt a la carpeta de Reportes y agréguelo usando la Herramienta de Contable .Net al sistema. Este es el nombre del formato RptRetencionIVA\_VEN.rpt. ProcedimientospRptRetencionIVA. Se modificaron los procedmientos [spCrearRetencioncxc],[spCrearRetencioncxp]

**2.1.2 c# VS2015 Contable.Net 14/05/2017**

En el formulario Devoluciones en Compras se hizo visible el campo Numero de documento del proveedor(txtnumdoc). Este valor representa el número de la nota de crédito que envía el proveedor.

También se le hizo visible el campo nro de control a Devoluciones en Ventas y Compras. Para que las notas de créditos copien el número de control

**2.1.2 c# VS2015 Contable.Net 14/05/2017**

En los formularios Documentos de Cuentas por Cobrar y Pagar (frmdocumcc y frmdocumcp) se cambió el orden en como muestra los registros, este cambio se hizo en las entidades Documcc y Documcp en el método ObtenerTodos, ahora se ordena “codtipo, numero”. También se afectaron los procedimientos siguientes:

[spDocumcpPaginado]

[spDocumcpPaginadobusqueda]

[spDocumcpUbicacionPaginado]

[spDocumccPaginado]

[spDocumccPaginadobusqueda]

[spDocumccUbicacionPaginado]

Para usar los formularios de esta versión es necesario actualizar los procedimientos nombrados.

**2.1.2 c# VS2015 Contable.Net 18/05/2017**

En todos procedimientos almacenados sp[NombreDeTabla]UbicacionPaginado se cambia la longitud de la variable @d.

cambiar

@d decimal(14,2),

Por

@d decimal(14,3),

Esto ocasionaba valores de ubicación incorrectos en el último registro de cada página.

Este cambio se debe hacer manualmente.

**2.2.0 c# VS2015 Contable.Net 11/06/2017**

Se corrrige el evento del boton imprimir en Formulario para imprimir Notas de Credito en Cuentas por Cobrar y de igual manera con Nota de Débito.

**2.2.0 c# VS2015 Contable.Net 11/06/2017**

Se corrige el procedimiento almacenado [dbo].[RptNotaCreditoDetalle]

Ver en Script de Actualización en C:\contable14vs2010\Contable.Net version 1.4 modificaciones\Actualizaciones\ActualizacionAContableSql2014\_2\_2\_0

**2.2.0 c# VS2015 Contable.Net 11/06/2017**

Se corrige evento del combo en el formulario para imprimir Balance General en Contabilidad.

**2.2.0 c# VS2015 Contable.Net 11/06/2017**

Se agrega un formulario para facturar moderno, para facturación rápida con scanner de código de barras y un menú para artículos mas vendidos, imprime a tickeras usando(opcionalmente) el programa de un tercero Super Spooler Fiscal.

**2.2.0 c# VS2015 Contable.Net 24/06/2017**

Se agrega la impresión a impresoras tickeras fiscales EPSON Solo se imprime la factura en POS.

Se usa el OCX del tercero: EPSON Argentina, “axPrinterFiscal”. Este OCX tiene sus propias reglas al enviar los comandos a las impresoras.

Las impresoras Soportadas son:

40 Columnas, Factura Ticket y Nota Credito

EPSON TM300AF

EPSON TMT285F

EPSON TMU950F

EPSON TM2000AF

EPSON TM2002AF+

EPSON TMU220AF/AF II

80 Columnas, Factura y Nota Debito,(no se ha probado NDetbito))

EPSON LX300F

EPSON LX300F+

EPSON FX880F

40 Columnas, Factura Ticket y Nota Credito

SAMSUNG BIXOLON (AR) SRP-250

SAMSUNG BIXOLON (AR) SRP-270

El OCX de Epson no se ejecuta en 64 bits. Las de 80 Pudieran aceptar Nota de Debito con el identificador D pero no se ha probado.

**2.2.0 c# VS2015 Contable.Net 24/06/2017**

Se corrige el calculo del descuento general en facturas de ventas y compras, cotizaciones, despachos, pedidos, devoluciones, compras, factura pos.

**2.2.0 c# VS2015 Contable.Net 28/06/2017**

Se Corrige a la hora de anular una factura, no lo estaba haciendo correctamente, se anulaba la factura en la tabla factura, pero no en la tabla de documentos.

En el procedimiento [spAnularFactura]

Se reemplaza

--actualizamos de una ves a documentos, lo de anulado

exec ('spActualizarFacturaDocumento ' + @codfactura+','+@codempresa )

Por

--actualizamos de una ves a documentos, lo de anulado

 UPDATE Documcc SET STATUS='ANU' WHERE tipodoc='FACT' and numero=@codfactura and codempresa=@codempresa

**Igual para Compras:**

En el procedimiento [spAnularCompra]

Se reemplaza

--actualizamos de una ves a documentos, lo de anulado

 exec ('spActualizarCompraDocumento ' + @codcompra+','+@codempresa )

Por

--actualizamos de una ves a documentos, lo de anulado

 --actualizamos de una ves a documentos, lo de anulado

 UPDATE Documcp SET STATUS='ANU' WHERE tipodoc='FACT' and numero=@codcompra and codempresa=@codempresa

**2.2.0 c# VS2015 Contable.Net 28/06/2017**

Se agregaron las impresoras fiscales a continuación, usando la OCX Fiscal150423.

40 columnas

“HASAR SMH/P-615F");

"HASAR SMH/P-PR4F"

"HASAR SMH/P-PR5F"

"HASAR SMH/P-930F"

"HASAR SMH/P-951F"

"HASAR SMH/P-715F"

"HASAR SMH/P-715F v2"

"HASAR SMH/P-441F"

"HASAR SMH/P-320F"

"HASAR SMH/P-321F"

"HASAR SMH/P-322F"

80columnas

"HASAR SMH/P-322F v2.01"

"HASAR SMH/P-330F"

"HASAR SMH/P-1120F"

"HASAR SMH/PL-8F"

"HASAR SMH/PL-8F v2.01"

"HASAR SMH/PL-23F"

40 columnas

"OLIVETTI PR4F"

80 columnas

"OLIVETTI 320F"

"OLIVETTI ARJET 20F"

80 columnas

"NCR 2008"

 40 columnas

"NCR3140 (En modo HASAR TM2000AF)"

"NCR3140 (En modo HASAR 615F)"

Para estas impresoras no se pudo imprimir Notas de Credito, no acepta los comandos en Enviar.

**2.3.0 c# VS2015 Contable.Net 15/07/2017**

se corrige el procedimiento almacenado [spAnularDocumentocp]

Cambiar la tabla

documcc

por

documcp

Se puede encontrar en el script de actualizaciones

**2.3.0 c# VS2015 Contable.Net 24/07/2017**

Se agrega el módulo de Impuestos sobre la Renta tanto a proveedores como de clientes.

**2.3.1 c# VS2015 Contable.Net 28/07/2017**

Se agrega el módulo de Retención al IVA tanto a proveedores como de clientes.

**2.3.2 c# VS2015 Contable.Net 13/08/2017**

Se agrega la función de imprimir en POS un formato de factura no fiscal tipo ticket. Se habilita el campo precio en la rejilla detalle de producto, el susuario puede modificarla, se agrega la opción reimprimir factura.

**2.3.3 c# VS2015 Contable.Net 26/08/2017**

Se agrega Libro Auxiliar de Inventario, Kardex. Se toma en cuenta los métodos: Ultimo Costo y Costo Promedio Ponderado.

**2.4.0 c# VS2015 Contable.Net 15/09/2017**

Se agrega módulo de Importación.

**2.5.0 c# VS2015 Contable.Net 03/11/2017**

Se agrega Notas de Entrega.

**2.6.0 c# VS2015 Contable.Net 05/12/2017**

Se le agrega Contabilización individual de documentos. Incluye asientos de ajustes de inventario y el incremento y disminución en los asiento de compras y ventas para los tipos de inventario permanente.

**2.7.0 c# VS2015 Contable.Net 06/12/2017**

Se le agrega qr code a una factura modelo.

Desde la consola de administración de paquete: pm> Install-Package QrCode.Net

Se le agregó Toma de Inventario. Hace un ajuste automático para cuadrar el inventario físico con la del sistema.

**2.8.0 c# VS2015 Contable.Net 18/01/2018**

Se cambia la coma por el punto y coma para separar listas en el formulario FrmPestañaObjetoPOS. Se usa para guardar el nombre de la fuente del objeto personalizado en facturación POS. Esto se hace para evitar problemas entre el separador de miles y decimal con el de listas.

**2.8.0 c# VS2015 Contable.Net 24/01/2018**

Se cambia el tipo de datos en los campos pendiente de los detalles. Ver Script:

alter table dbo.detallecompra drop constraint DF\_\_DetalleCo\_\_pendi\_\_30C33EC3

alter table dbo.detallecompra alter column pendiente decimal(18,4) not null

alter table dbo.detallecompra add default ((0)) for pendiente

alter table dbo.detallecompraimp drop constraint DF\_\_DetalleCo\_\_pendi\_\_367C1819

alter table dbo.detallecompraimp alter column pendiente decimal(18,4) not null

alter table dbo.detallecompraimp add default ((0)) for pendiente

alter table dbo.detallecompraprevia drop constraint DF\_\_DetalleCo\_\_pendi\_\_3E1D39E1

alter table dbo.detallecompraprevia alter column pendiente decimal(18,4) not null

alter table dbo.detallecompraprevia add default ((0)) for pendiente

alter table dbo.detallecotizacion drop constraint DF\_\_DetalleCo\_\_pendi\_\_47A6A41B

alter table dbo.detallecotizacion alter column pendiente decimal(18,4) not null

alter table dbo.detallecotizacion add default ((0)) for pendiente

alter table dbo.detalledespacho drop constraint DF\_\_DetalleDe\_\_pendi\_\_4D5F7D71

alter table dbo.detalledespacho alter column pendiente decimal(18,4) not null

alter table dbo.detalledespacho add default ((0)) for pendiente

alter table dbo.DetalleDevolucionCompra drop constraint DF\_\_DetalleDe\_\_pendi\_\_531856C7

alter table dbo.DetalleDevolucionCompra alter column pendiente decimal(18,4) not null

alter table dbo.DetalleDevolucionCompra add default ((0)) for pendiente

alter table dbo.DetalleDevolucionFactura drop constraint DF\_\_DetalleDe\_\_pendi\_\_58D1301D

alter table dbo.DetalleDevolucionFactura alter column pendiente decimal(18,4) not null

alter table dbo.DetalleDevolucionFactura add default ((0)) for pendiente

alter table dbo.DetalleEntrega drop constraint DF\_\_DetalleEn\_\_pendi\_\_6166761E

alter table dbo.DetalleEntrega alter column pendiente decimal(18,4) not null

alter table dbo.DetalleEntrega add default ((0)) for pendiente

alter table dbo.detallefactura drop constraint DF\_\_DetalleFa\_\_pendi\_\_1C0818FF

alter table dbo.detallefactura alter column pendiente decimal(18,4) not null

alter table dbo.detallefactura add default ((0)) for pendiente

alter table dbo.DetalleFacturaEspera drop constraint DF\_\_DetalleFa\_\_pendi\_\_70A8B9AE

alter table dbo.DetalleFacturaEspera alter column pendiente decimal(18,4) not null

alter table dbo.DetalleFacturaEspera add default ((0)) for pendiente

alter table dbo.DetalleOrden drop constraint DF\_\_DetalleOr\_\_pendi\_\_793DFFAF

alter table dbo.DetalleOrden alter column pendiente decimal(18,4) not null

alter table dbo.DetalleOrden add default ((0)) for pendiente

alter table dbo.DetallePedido drop constraint DF\_DetallePedido\_pendiente

alter table dbo.DetallePedido alter column pendiente decimal(18,4) not null

alter table dbo.DetallePedido add default ((0)) for pendiente

alter table dbo.DetalleRecepcion drop constraint DF\_\_DetalleRe\_\_pendi\_\_0C50D423

alter table dbo.DetalleRecepcion alter column pendiente decimal(18,4) not null

alter table dbo.DetalleRecepcion add default ((0)) for pendiente

**2.8.1 c# VS2015 Contable.Net 10/02/2018**

Se le agrega Arqueo de caja.

Se corrige la forma en que POS guarda la fuente de los objetos.

**2.8.2 c# VS2015 Contable.Net 14/02/2018**

Se le otorga al usuario la posibilidad de elegir la configuración regional

Se le agrega Requisiciones a Cuentas por pagar

Se corrige errores en Orden de Compras.

Al sistema de Herramientas Contable se le agrega el campo Solo Lectura y Valor por defecto en los detalles de los reportes personalizados. Se crea la versión 1.0.2

**2.9.0 c# VS2015 Contable.Net 20/02/2018**

Se le agrega un Formulario para Facturar artículos con la utilidad Multidemedidas.

Si quiere actualizar la BD ejecute los siguientes querys

CREATE TABLE [dbo].[ArticuloMedida](

 [codarticulo] [varchar](50) NOT NULL,

 [codmedida] [varchar](5) NOT NULL,

 [codempresa] [int] NOT NULL,

 [contenido] [decimal](18, 4) NOT NULL,

 [valor] [decimal](18, 6) NOT NULL,

 [precio] [decimal](18, 4) NOT NULL,

 [esbase] [bit] NOT NULL,

 CONSTRAINT [PK\_ArticuloMedida] PRIMARY KEY CLUSTERED

(

 [codarticulo] ASC,

 [codmedida] ASC,

 [codempresa] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

SET ANSI\_PADDING OFF

GO

ALTER TABLE [dbo].[ArticuloMedida] ADD CONSTRAINT [DF\_ArticuloMedida\_codempresa] DEFAULT ((0)) FOR [codempresa]

GO

ALTER TABLE [dbo].[ArticuloMedida] ADD CONSTRAINT [DF\_ArticuloMedida\_contenido] DEFAULT ((0)) FOR [contenido]

GO

ALTER TABLE [dbo].[ArticuloMedida] ADD CONSTRAINT [DF\_ArticuloMedida\_valor] DEFAULT ((0)) FOR [valor]

GO

ALTER TABLE [dbo].[ArticuloMedida] ADD CONSTRAINT [DF\_ArticuloMedida\_precio] DEFAULT ((0)) FOR [precio]

GO

ALTER TABLE [dbo].[ArticuloMedida] ADD CONSTRAINT [DF\_ArticuloMedida\_esbase] DEFAULT ((0)) FOR [esbase]

GO

alter table detallefactura add valorsec decimal(18, 4) not null default (0)

GO

alter table detallefactura add contenidosec decimal(18, 4) not null default (0)

GO

alter table detallefactura add codmedidasec varchar (5) not null default ('')

GO

alter table detallefactura add cantidadsec decimal(18, 4) not null default (0)

GO

alter table detallefactura add preciosec decimal(18, 4) not null default (0)

GO

alter table detallefactura add totalsec decimal(18, 2) not null default (0)

GO

update detallefactura set valorsec=1, contenidosec=1,codmedidasec=codmedida,cantidadsec=cantidad,preciosec=precio,totalsec=total

**2.9.2 c# VS2015 Contable.Net del 23/02/2018 al 08/03/2018**

Se le agregó la opción MANUFACTURA en tipo de artículo en el formulario Articulos.

Nos permite facturar por encima del inventario. FrmFactura

Nos permite facturar por encima del inventario. FrmFacturaM (multimedida)

Nos permite entrega por encima del inventario. FrmEntrega

Nos permite entrega por encima del inventario. FrmFactura2 (POS)

No reserva por encima del inventario. FrmPedido

Respeta el inventario en las salidas. FrmAjuste

Se corrigen y/o agregan los procedimientos validarExcesoImportadoXXXX, en Frm Factura, Factura2,Pedido, Entrega,Despacho, Compras, Orden de Compra. Por un algoritmo mejorado, eficiente.

Se respeta lo Comprometido, que es marcado por Pedidos.

Se corrigió método eliminar Usuario, en la capa de negocio.

Se corrige una validación en Devolucion de compras y facturas de ventas. Para lo de anular NC sin registro asociado directamente.

Se corrige la forma de validfaciones en POS,en frm cobro POS, a la hora de crear la factura. El procediemitno crerar registros se transforma en una función.

Se agrega validación de numéricos en los grids editables por el usuario, en frm Factura, factura2,facturaM,pedido, cotización, entrega, despacho, cobros, detalledocumcc,compras, orden,requisicion, detalle, pagos,documcp, ajuste, toma, comprobantes. Ya quer colocar . ó , sin números arrojaba error, ejemplo “,” “.” “,,” “..” “,.” etc

Error corregigo en el método mostrarclientes del frmclientes, selecteditem por selectedvalue.

El frmbuscarcliente se habilita para el formulario FrmFacturaM.

En POS, en frmconfipos, al abrir caja se corrigió lo de actualizar combo, de hecho se comentó, ya no va acá, sino que se llama desde el frmfactura2(pos),donde se hicieron algunas modificaciones.

En frmconfigpos, en abrircaja había un error. Cliente xxx=new(txtvendedor

Se le agrega las validaciones de existencia a los formularios Compras, también en anular y eliminar. Igualmente se modifica el algoritmo en entrada y salida de articulos, en ajsutes, se le agrega validación al eliminar ajustes.

Se le agrega validfacione4s de esxistencia en Formulario Toma, al acpetar y eliminar.

Se le agregó Listas de Precios

**2.9.3 c# VS2015 Contable.Net 20/03/2018**

Se resuelve un pequeño bug para la importación de facturas de compras y ventas a las devoluciones. En el método guardar del formulario Devoluciones de ventas y compras, hay un procedimiento que valida que no haya registros importados de diferentes facturas llamado: validardiferentefactura

Acá if (i>0 && (var\_tipoorigen != myDataGrid1.Rows[i].Cells["gstxttipoorigen"].Value.ToString() || var\_nroorigen != myDataGrid1.Rows[i].Cells["gstxtnroorigen"].Value.ToString()))

**Se modifica por** if (i>0 && var\_tipoorigen != ""&& (var\_tipoorigen != myDataGrid1.Rows[i].Cells["gstxttipoorigen"].Value.ToString() || var\_nroorigen != myDataGrid1.Rows[i].Cells["gstxtnroorigen"].Value.ToString()))

En el procedimiento global se había extraviado el AJNM para clientes, fue reincorporado

Había un error a la hora de guardar detalles de documentos en el frmdocumcp. Se cambió DetalleDocumcc por detalleDocumcp

Se corrigió en el método: Importar selección en el documento x actual.

Se hizo en Frm BuscarORden, Frm Buscar Pedido, Frm Buscar Requiscion, FrmBUscar Entrega, Frm BuscarCotizacion, Frm Buscar Compra,Frm Buscar Factura

El cambio que se hizo fue el value por FormattedValue, que daba problemas cuando el susario coloca el cero en el grid. Al seleccionar valores

Uno d elos métodos por ejemplo, del buscar compra es: “Importar Items Seleccionados en Devolución Actual”

if (decimal.Parse(dataGridView2.Rows[i].Cells["gstxtseleccion"].FormattedValue.ToString().Trim()) <= decimal.Parse(dataGridView2.Rows[i].Cells["gstxtpendiente"].FormattedValue.ToString().Trim()))

Se le agrega Auxiliar o terceros a la contabilidad, de igual manera en los procedimientos individuales y globales para contabilizar

**2.9.4 c# VS2015 Contable.Net desde 02/04/2018**

Se le agrega el botón de busca en cuentas de ingreso y egreso en el formulario emergente “FrmDetalleOrdenPAgo”

Corregido error en los FrmFactutra, FrmFacturaM y FrmFactura2(POS)

Método ActualizaCombo()

 if (Frmfact.cmblista.SelectedValue == null)

 Frmfact.cmblista.SelectedIndex = 0;

Por

 if (Frmfact.cmblista.SelectedValue == null)

 if (Frmfact.cmblista.SelectedIndex != -1)

 Frmfact.cmblista.SelectedIndex = 0;

Cuando no hay elementos en la tabla lista emanaba error.

Se agregar al botón nuevo de frmfactura y frmfacturam instrucciones para limpiar el arreglo de desxricpion de ítem.

//con este bucle lo dejo todo a 0

 //se limpia el arreglo

 for (int j = 0; j <500; j++)

 {

 descripciondetalle[j] = "";

 }

Se le agrega la validación para que el usuario no agregue cantidades negativas ni en cero en los siguientes formularios:

FrmFactura, FrmFacturaM, FrmFactura2(POS), FrmDevulcionCompra, FrmDevolucionVenta, FrmDespacho, FrmToma, FrmCompra, FrmCotizacion, FrmORden, FrmPEdido, FrmEntrega, FrmAjuste

Se le agrega Centro de Costos al detalle de FrmComprobantes, también dos resportes. Aun no se le agerega centro de costos desde los procedimientos automaticos porque no han sido agregados ha compras, requisicion, orden pago, orden compra…

Se le agregó crear cotixaciones por reposición, para esto se le agregó el campo sotckreposicion a la tabla detallecomprobante.

En el método ValidarValores del formulario FrmReportesParametros se agregó la validación en caso de que el usuario coloque valores que no estén dentro de la lista de un combobox. OCacionó cambios en los métodos ObtenerValor y ObtenerValor2 del mismo formulario. Además se hicieron cambios en los métodos crearComboBoxxxxx del mismo formulario, de autocompletación de datos tipeados por los usuarios en los combobox.

En los formularios donde se genera un documento a partir de otro, como por ejemplo, en FrmFactura hay un método para generar un despacho de esa factura, se cambia txtcondPago\_Validating(null,null) por sForm2.txtcondPago\_Validating(null,null). Esto también en Genera Factura y pedido de requisicon y otros.

En el procedimiento almacenado de cierre mensual contable se cambió:

delete from Balance where codempresa=@codempresa and ejercicio=@ejercicio and mes=@mes and año=@año

and saldoanterior=0 and debe=0 and haber=0 and saldo=0

por

delete from Balance where codempresa=@codempresa and ejercicio=@ejercicio and mes=@mes and año=@año

and saldoanterior=0 and debe=0 and haber=0 and saldo=0 and codcuentamov<>@CodCuentaResultado

**2.9.5 c# VS2015 Contable.Net 24/04/2018**

-Se le agregó “Envios por correos” a varios documentos

-Se le agregó una instrucción que faltaba en la trigger: UpdDetalleFactura] ON [dbo].[DetalleFactura]

Cambió

if (@codarticulo is not null)

begin

 exec spActualizarInventarioInserted @codarticulo,@codempresa,@cantidad,@codalmacen,'S'

por

if (@codarticulo is not null)

begin

if(@tipoorigen<>'E')

 exec spActualizarInventarioInserted @codarticulo,@codempresa,@cantidad,@codalmacen,'S'

-En la TRIGGER [dbo].[UpdDetalleCompra] ON [dbo].[DetalleCompra] faltaban unas instrucciones con respecto a las requisiciones “Q” en la parte de “insert” porque en el sector “delete” ya estaba hecho OK.

 if (@tipoorigen='Q')

 if ((select count(\*) from DetalleRequisicion d where d.codrequisicion= @nroorigen and d.registro = @renglonorigen and d.codempresa=@codempresa)>0)

 begin

 update detallerequisicion set pendiente=pendiente-@cantidad where codrequisicion=@nroorigen and registro=@renglonorigen and codempresa=@codempresa

 end

y

if (@tipoorigen='Q')

 if ((select count(\*) from detallerequisicion d where d.codrequisicion= @nroorigen and d.registro = @renglonorigen and d.codempresa=@codempresa)>0)

 begin

 --parcialmente procesado

 if ((select count(\*) from requisicion p, detallerequisicion d where (d.pendiente<d.cantidad and d.pendiente>0) and d.codrequisicion=@nroorigen and d.codempresa=p.codempresa and p.codrequisicion=@nroorigen and p.codempresa=@codempresa)>0)

 update requisicion set status='PPR' where codrequisicion=@nroorigen and codempresa=@codempresa

 --procesado completo

 if ((select count(\*) from requisicion p, detallerequisicion d where (d.pendiente>0) and d.codrequisicion=@nroorigen and d.codempresa=p.codempresa and p.codrequisicion=@nroorigen and p.codempresa=@codempresa)=0)

 update requisicion set status='PRO' where codrequisicion=@nroorigen and codempresa=@codempresa

 --sin procesar

 if ((select count(\*) from requisicion p, detallerequisicion d where (d.pendiente<d.cantidad) and d.codrequisicion=@nroorigen and d.codempresa=p.codempresa and p.codrequisicion=@nroorigen and p.codempresa=@codempresa)=0)

 update requisicion set status='SPR' where codrequisicion=@nroorigen and codempresa=@codempresa

end

* En el botón cmdeliminar del frmcompra, se cambió:

 MessageBox.Show("La Devolución de la Factura se encuentra ANULADA, No se puede Eliminar");

Por:

 MessageBox.Show("La Factura se encuentra ANULADA, No se puede Eliminar");

-En el método validarExcesoImportadoORDEN del frmcompra

Cambiar

DetallePedido detped = new DetallePedido(Int32.Parse(myDataGrid1.Rows[i].Cells["gstxtrenglonorigen"].Value.ToString()), Int32.Parse(myDataGrid1.Rows[i].Cells["gstxtnroorigen"].Value.ToString()), VariablesGlobales.CodEmpresa);

Por

DetalleOrden detped = new DetalleOrden(Int32.Parse(myDataGrid1.Rows[i].Cells["gstxtrenglonorigen"].Value.ToString()), Int32.Parse(myDataGrid1.Rows[i].Cells["gstxtnroorigen"].Value.ToString()), VariablesGlobales.CodEmpresa);

En FrmCompra, en el método validarExcesoImportadoREQUISICION cambiar

MessageBox.Show("El registro número: " + (i + 1).ToString() + " de artículo: " + myDataGrid1.Rows[i].Cells["gstxtcodarticulo"].Value.ToString() + " Rebasa lo pendiente en la Cotización");

Por:

MessageBox.Show("El registro número: " + (i + 1).ToString() + " de artículo: " + myDataGrid1.Rows[i].Cells["gstxtcodarticulo"].Value.ToString() + " Rebasa lo pendiente en la Requisición");

**3.0.0 c# VS2015 Contable.Net 27/04/2018**

-Se corrigieron errores en el FRMBuscarRequision, se sustituyeron declaraciones de la Clase DetalleOrden por DetalleRequisicion en varias partes

-Se le agrega al proyecto Recepcion en CXP

-En las compras no aumenta el inventario(activo) en el asiento contable individual cuando esté originada por una recepción, esta ultima se encargará de generar ese asiento de ajuste del inventario.

-En las ventas no disminuye el inventario(activo) en el asiento contable individual cuando la venta esté originada por una nota de entrega, esta ultima se encargará de generar ese asiento de ajuste del inventario.

-En el procedimiento almacenado spAnularCompra, no había sido tomado en cuenta la requisición. Ya fue agregada. Se reconoce por la “Q”

**3.0.1 c# VS2015 Contable.Net 04/05/2018**

Error depurado al modificar una factura de compra importada desde una recepción, no se había tomado en cuenta.

Depurado en el spcontabilizarRECECXP, no se hacía la búsqueda del nombre del proveedor, por tal razón no nos mostraba la descripción en el asiento contable.

Botones en el formulario de búsqueda de recpeciones, el formulario que se muestra cuando deseas importar una recpeción. Se cambió la propiedad text que hacia referencia a REquisisciones, se llevò a Recepciones.

En la trigger de detallecompras y detallerecepcion de eliminación se cambió el cursordelete por cursorcompradelete y cursorrecepciondelete respectivamente.

**3.1.0 c# VS2015 Contable.Net 08/05/2018**

Se le agregó al formulario de respaldos la utilidad de subir a la nube el backup.

En el segundo constructor de la clase Caja, no estaba agregado el campo status. Se agregó emanaba error.

 public Caja(int codempresa)

 {

 DBDatos d = new DBDatos();

 DataRow drSQL = d.RegistroTabla("SELECT top 1 codcaja,nombre,saldo,usuario,status FROM caja where codempresa=" + codempresa);

Se le agregó importar precios de listas dese Excel.

**3.1.1 c# VS2015 Contable.Net 17/05/2018**

-Se le agregó animación al frm acceso

-Cuando el usuario le da clic a la lista de artículos en facturación, muestra el precio si tiene una lista seleccionada, anteriormente no lo hacía, solo con enter.

-En el formulario emergente buscar articulo, se le agregó un botón para que el usuario pueda escoger sus Articulos sin salir del buscador.

**3.2.0 c# VS2015 Contable.Net 23/05/2018**

En la entidad TOMA en la capa de negocio, se cambió

pcodtipoajusteentrada = drSQL["costipoajusteentrada"].ToString();

 pcodtipoajustesalida= drSQL["costipoajustesalida"].ToString();

por

pcodtipoajusteentrada = drSQL["codtipoajusteentrada"].ToString();

 pcodtipoajustesalida= drSQL["codtipoajustesalida"].ToString();

(hay un error en costipoajuste…..)

**25/05/2018**

En el formulario FRmToma se le agregó el botón cancelar con el siguiente codigo

InicializarControles();

 cmdnuevo.Focus();

En el formulario FRmToma en el método de guardar se le agregó a los campos cod de tipo de ajustes:

.ToUpper()

**28/05/2018**

Se valida que no quede stock negativo en frmentrega.

**28/05/2018**

Se agrega Traslado entre almacences de inventario

Se actualizó el informe “Movimientos de un artículo”

**04/06/2018**

En el frmFacturaM (multimedidaas) en el evento row validating del objeto mydatagrid1 se comentó un bloque de código que no va:

/\* Este parte tal parece no va, pendiente por borrar.

 //validando CODmedida asignada en el maestro

 fila = Filas2[0]; //si no exciste en articulo entonces chequeamos en articulomedida

 if (fila["codmedida"].ToString() != myDataGrid1.Rows[myDataGrid1.CurrentCell.RowIndex].Cells["gscmbmedida"].Value.ToString())

 {

 DialogResult result = MessageBox.Show(this, "La Unidad de Medida no está asignada en el maestro de articulos. Presione Si para ignorar y continuar (el registro se marcará como eliminado), Presione NO para Corregir", "Aplicación Contable", MessageBoxButtons.YesNo, MessageBoxIcon.Question, MessageBoxDefaultButton.Button1);

 if (result == DialogResult.No)

 {

 e.Cancel = true;

 myDataGrid1.CurrentCell = myDataGrid1.Rows[e.RowIndex].Cells[6];

 myDataGrid1.Rows[e.RowIndex].Cells[6].ErrorText = "La Unidad de Medida no está asignada en el maestro de articulos";

 return;

 }

 else

 {

 e.Cancel = false;// true;

 //ojovalidar = true;

 myDataGrid1.Rows[e.RowIndex].Cells["gschkelimin"].Value = true;

 myDataGrid1.Rows[e.RowIndex].Cells[6].ErrorText = "La Unidad de Medida no está asignada en el maestro de articulos";

 colorear(myDataGrid1.Rows[e.RowIndex]);

 // myDataGrid1.Rows.RemoveAt(e.RowIndex); //InicializarControles(); //este es lo que hace cancelar

 return;

 }

 }\*/

**04/06/2018**

Se agregó al FrmTipoAjuste en el evento load:

cmbtipo.SelectedIndex = 0;

Esto es para que el combo siempre muestre el primer registro, cuando es la primera vez que se agrega un tipo de ajuste y si el usuario no selecciona si es entrada o salida, el sistema arroja un error.

**04/06/2018**

Se agregó en el FrmTabuladorRislr en el evento cell validated del objeto mydatagrid1.

if (myDataGrid1.Rows[myDataGrid1.CurrentCell.RowIndex].Cells["gstxtsustraendo"].Value == null)

 myDataGrid1.Rows[myDataGrid1.CurrentCell.RowIndex].Cells["gstxtsustraendo"].Value = 0;

**04/06/2018**

Los valores de los registros de la tabla FrmTabuladorRislr tenían un carácter sobrante en la descripción, no era el espacio en blanco sinó el char 160

**04/06/2018**

Se agregó Retención de Impuesto Municipal a Proveedores con su contabilización.

**05/06/2018**

Se le agregó CCO a envíos de correos

**08/06/2018**

Se agregó Retención de Impuesto Municipal de clientes con su contabilización.

**21/06/2018**

En los procedimientos [RptEstadoCuentaCaja]

Y [RptEstadoCuentaBanco] Se le agregó al final de la consulta ORDER BY Fecha

En los reportes RptEstadoCuentaCaja.rpt Y RptEstadoCuentaBanco.rpt se movió la formula @inicial hacia el área del encabezado.

**26/06/2018**

Se corrigió en FrmFactura2, evento txtcodarticulo\_KeyPress del objeto txtcodarticulo

DataRow[] Filas2 = (DataRow[])DTArticuloListaPrecio.Select("codlista='" + (String)((cmblista.SelectedValue.ToString()== null)

Por

DataRow[] Filas2 = (DataRow[])DTArticuloListaPrecio.Select("codlista='" + (String)((cmblista.SelectedValue== null)

En el procedimiento private void Poscmdart001001\_Click(object sender, EventArgs e) también se hizo algo similar (02/07/2018) línea 655

**26/06/2018**

Se le agrega

Correlativo De Otras Tablas,

Correlativos de Factura Clasica

Correlativos de Factura Moderna (por cajas)

**3.2.1 c# VS2015 Contable.Net**

**04/07/2018**

En spanularrecepcion y en spanularordencompra no se había tomado en cuenta las reqwuisiones(Q)

**04/07/2018**

En la Entidad Articulo, método agregar, stock actual se agrega 0 de forma arbitraria.

**04/07/2018**

En el Formulario FrmBuscarFactura, evento cmdseleccionaritemnueva\_Click, cambiar

Pedido ped = new Pedido (Int32.Parse(wtxtcodfactura.Text), VariablesGlobales.CodEmpresa);

por

Factura ped = new Factura(Int32.Parse(wtxtcodfactura.Text), VariablesGlobales.CodEmpresa);

Se le agregó gstxtitem a FrmDespacho, en el grid de detalles y en mostrardatosdetalles(método)

En [UpdDetalleFactura] trigger no se tomaba en cuenta Entrega, al decreemntar

En [spAnularFactura] no se había tomado en cuenta Entrega

En frmfactura y frmfacturaM(multimedida) validar si tiene despacho activo, no anular no aliminar no actualizar

 Despacho des = new Despacho();

 if (des.ExisteDespachoRelacionada(Int32.Parse(txtcodfactura.Text), VariablesGlobales.CodEmpresa) == true)

 {

 MessageBox.Show("La factura está relacionada con una o más despachos, No se puede Modificar");

 return;

 }

**Contable Version 3.2.2 13/07/2018**

**En el evento** txtcodbarra\_TextChanged del FrmFactura2 se agregó el if antes de instanciar ArtuculoListaPrecio: (en dos bloques de código)

 if (cmblista.SelectedValue != null)

 {

 ArticuloListaPrecio artlistaprecio = new ArticuloListaPrecio();

 DataRow[] Filas2 = (DataRow[])DTArticuloListaPrecio.Select("codlista='" + (String)((cmblista.SelectedValue.ToString() == null) ? "''" : cmblista.SelectedValue.ToString()) + "' and codarticulo= '" + (String)((fila["codarticulo"].ToString() == null) ? "''" : fila["codarticulo"].ToString()) + "'");

**13/07/2018**

En el constructor de la entidad ARTICULO:

 public Articulo(String NoAsignarNada, String codbarra, int codempresa)

se agrega Stockreposicion en el select from

**19/07/2018**

Esto en el load de frmfactura y framfacturaM

 txtcondpago.Text = "0";

 txtporcdesc.Text = "0";

**19/07/2018**

Esto en botón guardar del frmbuscararticuloPOS

 if (Quien == "FrmFactura2")

 {

**25/07/2018**

En la entidad Factura, el método ObtenerTodosCriterio2

Faltaba una coma entre:

nombrecliente,codlista

**30/07/2018**

En los formularios de frmcobrto y frmpago, se suavisaron las validaciones, ya que eran rústicas y engorrosas. Por lo tanto se suprimieron algunos mernsajes en el evento rowvalidation del grid forma de pago/cobro, y se llevaron a una función llamada validarVariosForma que se usa en el evento click del botón guardar.

**02/08/2018**

En FrmOrdenPago en lblultimo\_Click

cambiar

MovimientoBanco mae = new MovimientoBanco();

Por

OrdenPago mae = new OrdenPago();