

```

<row>
    <CodigoNet>
using AhoraAsistentes;
using AhoraCore;
using AhoraOCX;
using AhoraPartes;
using AhoraSistema;
using static AhoraCore.VBA.Constants;
using static AhoraCore.VBA.Conversion;
using static AhoraCore.VBA.DateTime;
using static AhoraCore.VBA.Strings;
using static AhoraCore.VbMsgBoxResult;
using static AhoraCore.VbMsgBoxStyle;
using System;

namespace AhoraScriptsPantalla
{
    /** FORMULARIO PEDIDOS **/
    public class AhoraCl_frmPedidos : AhoraOCX.AhoraBaseScript
    {
        const int ANCHO_CAPTION = 4; //67;
        const int ANCHO_FECHAS = 153; //2300;

        const int ALTO_BUSCADORES = 20; //300;
        const int DESPLAZA_V = 3; //50;
        const int DESPLAZA_H = 3; //50;
        const int ANCHO_BOTONERTEXTOSIMP = 227; //3400;
        const int ALTO_BOTONERATEXTOSIMP = 27; //550;

        public void pAnadirControlDebajo(Valor vsControl, Valor vsControlRef,
Valor viAncho, Valor viAlto, Valor viDesplazaV, Valor viDesplazaH)
        {
            Valor lTop = ValorEmpty.Value;
            Valor lIzq = ValorEmpty.Value;
            try
            {
                lIzq = 0;
                if (!System.String.Equals("" + (string)viDesplazaH, "",
StringComparison.Ordinal))
                {
                    lIzq = (int)viDesplazaH;
                }
                lTop = viDesplazaV;
                if ("" + vsControl == "")
                {
                    return;
                }
                if ("" + vsControlRef != "")
                {
                    lIzq = gForm.Controls[(string)vsControlRef].Left + lIzq;
                    lTop = gForm.Controls[(string)vsControlRef].Top +
gForm.Controls[(string)vsControlRef].Height + viDesplazaV;
                }
                gForm.Controls[(string)vsControl].SetBounds(lIzq, lTop, viAncho,
viAlto);
            }
            catch (Exception)
            {
                throw;
            }
        }

        public void pAnadirControlDerecha(Valor vsControl, Valor vsControlRef,

```

```

Valor viAncho, Valor viAlto, Valor viDesplazaH, Valor viDesplazaV)
{
    Valor lTop = ValorEmpty.Value;
    Valor lIzq = ValorEmpty.Value;
    try
    {
        lIzq = viDesplazaH;
        lTop = 0;
        if (!System.String.Equals("" + (string)viDesplazaV, "",
StringComparison.Ordinal))
        {
            lTop = (int)viDesplazaV;
        }
        if ("" + vsControl == "")
        {
            return;
        }
        if ("" + vsControlRef != "")
        {
            lIzq = gForm.Controls[(string)vsControlRef].Left +
gForm.Controls[(string)vsControlRef].Width + viDesplazaH;
            lTop = gForm.Controls[(string)vsControlRef].Top + lTop;
        }
        gForm.Controls[(string)vsControl].SetBounds(lIzq, lTop, viAncho,
viAlto);
    }
    catch (Exception)
    {
        throw;
    }
}

public Valor fvLeerParametroEmpresa(string vsParam, string vsTabla)
{
    Valor fvLeerParametroEmpresa = ValorEmpty.Value;
    try
    {
        fvLeerParametroEmpresa = gCn.DameValorCampo(CStr("SELECT " +
vsParam + " FROM " + vsTabla + " WHERE IdEmpresa = " + CStr(gCn.IdEmpresa)));
    }
    catch (Exception)
    {
        throw;
    }
    return fvLeerParametroEmpresa;
}

public override void Initialize()
{
    try
    {
        ((ICntBotonera)gForm.FindControl("Botonera")).ActivarScripts =
true;
        ((ICntBotonera)gForm.FindControl("Botonera")).BotonAdd("Últimas
ventas", "pers_botUltVent", "", (ETipo_Menu)0, true,
gCn.Obj.Objetos["Articulos"].Icono1);
        //V1.1
        ((ICntBotonera)gForm.FindControl("Botonera")).BotonAdd("Generar
Partes", "pers_botGenerarPartes", "", (ETipo_Menu)0, true,
gCn.Obj.Objetos["Pedidos"].Icono1);
        ((ICntBotonera)gForm.FindControl("Botonera")).HabilitaBotones();
    }
}

```

```

        ((IControlScriptable)gForm.Controls["grdLineas"]).ActivarScripts
= true;

        ((IControlScriptable)gForm.Controls["mnuMain"]).ActivarScripts =
true;
        ((IControlScriptable)gForm.Controls["IdCliente" + "_" +
CStr(0)]).ActivarScripts = true;
        ((IControlScriptable)gForm.Controls["Cliente" + "_" +
CStr(0)]).ActivarScripts = true;
        ((IControlScriptable)gForm.Controls["IdLista"]).ActivarScripts =
true;

        pConfigurarPanelImpresos();
        pAnadirCamposEconomicos();
        pAjustarCamposWidthEconomicos();

        gForm.Width = 1100;
    }
    catch (Exception)
    {
        throw;
    }
}

public override void Show()
{
    try
    {

        pConfiguraComboListaPreciosDetalle();
        //pConfigurarColumnaIconosPartesDetalle();
        pConfigurarValueItemsColIVA();

    }
    catch (Exception)
    {
        throw;
    }
}

public override void Grid_DblClick(ICntGridUsuario aGrid)
{
    try
    {
        if (System.String.Equals(aGrid.Name, "grdLineas",
StringComparison.Ordinal))
        {
            if (System.String.Equals(aGrid.CampoActual, "Descrip2",
StringComparison.Ordinal) | System.String.Equals(aGrid.CampoActual,
"pers_DescripExtendida", StringComparison.Ordinal))
            {
                pAsignarDescAmpliada(aGrid);
            }

            //If aGrid.CampoActual = "@pers_IdParte" Then ''Or
aGrid.CampoActual = "IdParte" Then
            // pMostrarParte aGrid.GetValue("IdParte")
            //End If

            if (System.String.Equals(aGrid.CampoActual, "@NumParte",
StringComparison.Ordinal))
            {

```

```

        pMostrarParte(aGrid.GetValue("@NumParte"));
    }

}

}
catch (Exception)
{
    throw;
}
}

public void pAsignarDescAmpliada(ICntGridUsuario aGrid)
{
    Valor lTexto = ValorEmpty.Value;
    IFrmZoom lzFrm = null;
    try
    {
        // lDescAmp = gform.Controls("grdLineas").Getvalue("Descrip2")
        // Set lFrmZ = gCn.AhoraProceso("DameZoom", False, gCn)
        // lFrmZ.Ver2 CStr(lDescAmp), 2000, False, gform
        // If lFrmZ.Texto <> "" Then
        //     agrid.SetValue "Descrip2", lFrmZ.Texto
        // End If

        lzFrm = (IFrmZoom)gCn.AhoraProceso("DameZoom", out _, gCn);
        ((ITextoUsuario)lzFrm.FindControl("txtZoom")).ReadOnly =
aGrid.Columns[aGrid.Col].Locked;
        lTexto = lzFrm.Carga(aGrid.Columns[aGrid.Col].Caption,
aGrid.Columns[aGrid.Col].Text);
        if ((int)lzFrm.BotonPulsado != 2)
        {
            aGrid.SetValue(aGrid.Columns[aGrid.Col].DataField,
(string)lTexto);
        }
    }
    catch (Exception)
    {
        throw;
    }
}

public override void Botonera_AfterExecute(ICntBotonera aBotonera,
IAhoraTool aBoton)
{
    try
    {
        if (System.String.Equals(aBoton.Name, "pers_botUltVent",
StringComparison.Ordinal))
        {
            pMostrarUltimasVentas();
        }
        //V1.1
        if (System.String.Equals(aBoton.Name, "pers_botGenerarPartes",
StringComparison.Ordinal))
        {
            pGenerarPartesTodasLineasSinPartes();
        }

        if (System.String.Equals(aBotonera.Name,
"pers_btnra_TextosCabYPieImpreso", StringComparison.Ordinal))
        {
            if (System.String.Equals(aBoton.Name, "pers_botTxtCabecera",

```

```

StringComparison.Ordinal))
    {

pMostrarTextoCabeceraYPie(gForm.Controls["pers_edTxtCabecera"]);
    }
    if (System.String.Equals(aBoton.Name, "pers_botTxtPie",
StringComparison.Ordinal))
    {

pMostrarTextoCabeceraYPie(gForm.Controls["pers_edTxtPie"]);
    }
    }
    catch (Exception)
    {
        throw;
    }
}

//V1.1
//NOTA: Se puede capturar el nº de partes generados a través de la
variable Retorno
//Una vez generados los partes, habrá que crear una nueva condición para
mostrar la información adicional
public void pGenerarPartesTodasLineasSinPartes()
{
    Valor lIdPedido = ValorEmpty.Value;
    Valor Retorno = ValorEmpty.Value;
    IVB6Collection<Valor> lCol = null;
    try
    {

        lCol = gCn.DameNewCollection();

        lIdPedido =
((IEnlaceObjetos)gForm.Controls["EObjeto"]).ObjGlobal["IdPedido"];
        Retorno = 99;

        lCol.Add(lIdPedido);
        lCol.Add(Retorno);

        if (!gCn.EjecutaStoreCol("pPers_GenerarPartesPedido", lCol))
        {
            gCn.Obj.ShowMsgBox("Error generando partes", vbInformation,
"Generar Partes", gForm);
            return;
        }
        else
        {
            pRefrescarObj();
            gCn.Obj.ShowMsgBox("Partes generados correctamente",
vbInformation, "Generar Partes", gForm);
        }
    }
    catch (Exception)
    {
        throw;
    }
}

```

```

//V1.1
public void pGenerarParteLineaEspecifica(Valor lIdLinea)
{
    Valor lIdPedido = ValorEmpty.Value;
    IVB6Collection<Valor> lCol = null;
    try
    {
        //VALIDAR QUE LA LINEA NO TENGA VALOR EN IDPARTE o
pers_HayPartes
        lCol = gCn.DameNewCollection();

        lIdPedido =
((IEnlaceObjetos)gForm.Controls["EObjeto"]).ObjGlobal["IdPedido"];

        lCol.Add(lIdPedido);
        lCol.Add(lIdLinea);

        if (!gCn.EjecutaStoreCol("pPers_GenerarParteLineaPedido", lCol))
        {
            gCn.Obj.ShowMsgBox("Error generando parte", vbInformation,
"Generar Parte Línea", gForm);
            return;
        }
        else
        {
            pRefrescarObj();
            gCn.Obj.ShowMsgBox("Parte de la línea generado
correctamente", vbInformation, "Generar Parte Línea", gForm);
        }
    }
    catch (Exception)
    {
        throw;
    }
}

public void pMostrarUltimasVentas()
{
    IMantenimiento lFrm = null;
    Valor lIdArticulo = ValorEmpty.Value;
    Valor lIdCliente = ValorEmpty.Value;
    try
    {
        lIdCliente = ((IControlEnlazable)gForm.Controls["IdCliente" +
"_" + CStr(0)]).Text;
        lIdArticulo =
((ICntGridUsuario)gForm.Controls["grdLineas"]).GetValue("IdArticulo");

        lFrm = (IMantenimiento)gCn.AhoraProceso("NewFrmMantenimiento",
out _, gCn);
        lFrm.Form.NombreForm = "Pers_frmMant_UltimasVentasCliente";
        lFrm["UltimasVentasCliente"].Agregar = true;
        lFrm["UltimasVentasCliente"].Editar = true;
        lFrm["UltimasVentasCliente"].Eliminar = true;
        lFrm["UltimasVentasCliente"].CargaObjetos = false;
        lFrm["UltimasVentasCliente"].EditarPorObjeto = false;
        lFrm["UltimasVentasCliente"].GridHeadLines = 2;
        lFrm["UltimasVentasCliente"].AgregaColumna("@Marca", 27, "M");
        lFrm["UltimasVentasCliente"].AgregaColumna("FechaAlbaran", 87,
"Fecha Albarán", true);
    }
}

```

```

        lFrm["UltimasVentasCliente"].AgregaColumna("NumAlbaran", 50,
"Albarán", true);
        lFrm["UltimasVentasCliente"].AgregaColumna("IdArticulo", 93, "Id
Artículo", true);
        lFrm["UltimasVentasCliente"].AgregaColumna("Descrip", 167,
"Descripción", true);
        lFrm["UltimasVentasCliente"].AgregaColumna("Descrip2", 167,
"Descripción 2", true);
        lFrm["UltimasVentasCliente"].AgregaColumna("Cantidad", 57,
"Cantidad", true, "", false, "#,##0.00");
        lFrm["UltimasVentasCliente"].AgregaColumna("Precio_Euro", 83,
"Precio (€)", true, "", false, "#,##0.0000");
        lFrm["UltimasVentasCliente"].AgregaColumna("DtoLP1", 43, "Dto
LP1", true, "", false, "#,##0.00");
        lFrm["UltimasVentasCliente"].AgregaColumna("IdIva", 50, "IVA",
true, "Select IdIva, Descrip, DescCompleta from vPers_Combo_IVAS", false, "",
false, "Select IdIva, Descrip from vPers_Combo_IVAS");
        lFrm["UltimasVentasCliente"].AgregaColumna("Descuento", 43, "%
Dto", true, "", false, "#,##0.00");
        lFrm["UltimasVentasCliente"].AgregaColumna("pers_PrecioCoste",
80, "Coste €", true, "", false, "#,##0.0000");
        lFrm["UltimasVentasCliente"].AgregaColumna("pers_Margen", 57,
"%Margen", true, "", false, "#,##0.00");
        lFrm["UltimasVentasCliente"].AgregaColumna("Total_Euros", 80,
"Total €", true, "", false, "#,##0.0000");
        lFrm["UltimasVentasCliente"].AgregaColumna("IdPedido", 0,
"Pedido", true, "", false, "#");
        lFrm["UltimasVentasCliente"].AgregaColumna("IdLinea", 0,
"Linea", true, "", false, "#");
        lFrm["UltimasVentasCliente"].From = "vPers_UltimasVentas";
        lFrm["UltimasVentasCliente"].Where = "WHERE IdCliente = '" +
(string)lIdCliente + "'";
        lFrm["UltimasVentasCliente"].Campo("@Marca").Booleano = true;
        lFrm["UltimasVentasCliente"].ColumnaEscalada = "Descrip";
        lFrm["UltimasVentasCliente"].OrdenMultiple = "FechaAlbaran Desc,
NumAlbaran Desc, IdArticulo Asc";
        lFrm["UltimasVentasCliente"].Tag = (string)
((IEnlaceObjetos)gForm.Controls["EObjeto"]).ObjGlobal["SeriePedido"] + ";" +
(string)((IEnlaceObjetos)gForm.Controls["EObjeto"]).ObjGlobal["IdPedido"] + ";" +
(string)((IEnlaceObjetos)gForm.Controls["EObjeto"]).ObjGlobal["IdCliente"] +
";" + (string)((IEnlaceObjetos)
(IEnlaceObjetos)gForm.Controls["EObjeto"]).ObjGlobal["IdLista"] + ";" +
((IEnlaceObjetos)
(IEnlaceObjetos)gForm.Controls["EObjeto"]).ObjGlobal["IdMoneda"] + ";" +
((IEnlaceObjetos)(IEnlaceObjetos)gForm.Controls["EObjeto"]).ObjGlobal["Fecha"];
        lFrm["UltimasVentasCliente"].SoloArray = false;
        lFrm["UltimasVentasCliente"].Refresca = true;
        lFrm["UltimasVentasCliente"].SoloArray = true;
        lFrm["UltimasVentasCliente"].Refresca = false;
        lFrm.SetMuestraFiltro("UltimasVentasCliente", true);
        lFrm.Form.Width = 1137;
        lFrm.Form.Height = 533;
        lFrm.MultiSeleccion = true;

        lFrm.Form.Text = "Últimas Ventas del Cliente";
        lFrm.Botonera.BotonAdd("Incluir seleccionados",
"pers_botAsignar", "", ETipo_Menu.eTipo_Menu_Boton, false, 415);
        lFrm.Botonera.BotonAdd("Desmarcar todo", "pers_botDesmarcar",
"", ETipo_Menu.eTipo_Menu_Boton, false, 105);

        lFrm.Carga(null, true, (EBotonesMant)4, null, false);

((ICntBotonera)gForm.Controls["Botonera"]).BotonExecute("botGuardar");

```

```

        ((ICntGridUsuario)gForm.Controls["grdLineas"]).Refrescar();
        ((IEnlaceObjetos)gForm.FindControl("EObjeto")).Refresh();
    }
    catch (Exception)
    {
        throw;
    }
}

public override void Grid_AfterColUpdate(ICntGridUsuario aGrid, Valor
aColIndex)
{
    Valor lDtoLP5 = ValorEmpty.Value;
    Valor lDtoLP4 = ValorEmpty.Value;
    Valor lDtoLP3 = ValorEmpty.Value;
    Valor lCoste = ValorEmpty.Value;
    Valor lDtoGD = ValorEmpty.Value;
    Valor lDtoLP2 = ValorEmpty.Value;
    Valor lDtoLP1 = ValorEmpty.Value;
    Valor lDescuento = ValorEmpty.Value;
    Valor lPrecio = ValorEmpty.Value;
    Valor lNULL = ValorEmpty.Value;
    IVB6Collection<Valor> lColArt = null;
    Valor lObjeto = ValorEmpty.Value;
    Valor lTipoUnidad = ValorEmpty.Value;
    Valor lCampoPrecioCoste = ValorEmpty.Value;
    Valor lPCoste = ValorEmpty.Value;
    Valor lIdAlmacen = ValorEmpty.Value;
    Valor lCantidad = ValorEmpty.Value;
    Valor lMoneda = ValorEmpty.Value;
    Valor lListaCte = ValorEmpty.Value;
    Valor lCliente = ValorEmpty.Value;
    Valor lFecha = ValorEmpty.Value;
    Valor lArticulo = ValorEmpty.Value;
    Valor lDescExt = ValorEmpty.Value;
    Valor lIdArticulo = ValorEmpty.Value;
    try
    {
        if (System.String.Equals(aGrid.Name,
gForm.Controls["grdLineas"].Name, StringComparison.Ordinal))
        {
            if (aGrid.ColIndex("IdArticulo") == aColIndex)
            {
                lIdArticulo = aGrid.GetValue("IdArticulo");
                lDescExt = gCn.DameValorCampo("SELECT pers_DescripVentas
FROM Conf_Articulos WHERE IdArticulo = '" + lIdArticulo + "'");
                if ("" + lDescExt != "")
                {
                    ((ICntGridUsuario)gForm.Controls["grdLineas"]).SetValue("pers_DescripExtendida",
lDescExt);
                }
            }

            //Cambio de precio
            if (aGrid.ColIndex("pers_IdLista") == aColIndex)
            {
                //Aunque está oculto, mantengo la restricción por si
                acaso se realiza alguna asignación por debajo
                if (!fbTrabajaConListaPreciosEnDetalle())
                {
                    return;
                }
            }
        }
    }
    catch (Exception)
    {
        throw;
    }
}

```



```

    }

    lArticulo = aGrid.GetValue("IdArticulo");
    lFecha = (DateTime)((IEnlaceObjetos)
(IEnlaceObjetos)gForm.Controls["EObjeto"]).ObjGlobal["Fecha"];
    lCliente = ((IEnlaceObjetos)
(IEnlaceObjetos)gForm.Controls["EObjeto"]).ObjGlobal["IdCliente"];
    lListaCte = aGrid.GetValue("pers_IdLista");
    lMoneda = ((IEnlaceObjetos)
(IEnlaceObjetos)gForm.Controls["EObjeto"]).ObjGlobal["IdMoneda"];
    lCantidad = aGrid.GetValue("Cantidad");
    lIdAlmacen = aGrid.GetValue("IdAlmacen");

    if ((decimal)lCantidad == 0)
    {
        lCantidad = 1;
    }

    lPCoste = (decimal)0;
    lCampoPrecioCoste = "";
    lTipoUnidad = "";
    lObjeto = "Pedido_Lineas";

    lColArt = gCn.DameNewCollection();
    lColArt.Add(lArticulo);
    lColArt.Add(lCliente);
    lColArt.Add(lListaCte);
    lColArt.Add(lCantidad);
    lColArt.Add(lMoneda);
    lColArt.Add(0);
    lColArt.Add(0);
    lColArt.Add(0);
    lColArt.Add(0);
    lColArt.Add(0);
    lColArt.Add(0);
    lColArt.Add(0);
    lColArt.Add(0);
    lColArt.Add(lIdAlmacen);
    lColArt.Add(lPCoste);
    lColArt.Add(0);
    lColArt.Add(lCampoPrecioCoste);
    lColArt.Add(lTipoUnidad);
    lColArt.Add(lFecha);
    lColArt.Add(0);
    lColArt.Add(0);
    lColArt.Add(0);
    lColArt.Add(0);
    lColArt.Add(lObjeto);
    lColArt.Add(lNULL);
    lColArt.Add(lNULL);
    lColArt.Add(lNULL);

    if (!gCn.EjecutaStoreCol("PDamePrecio_Articulo",
lColArt))
    {
        gCn.Obj.ShowMsgBox("Error obteniendo precios de la
línea", vbExclamation, "Atención", gForm);
        return;
    }

    lPrecio = lColArt[6];
    lDescuento = lColArt[8];
    lDtoLP1 = lColArt[11];
    lDtoLP2 = lColArt[12];

```

```

        lDtoGD = lColArt[13];
        lCoste = lColArt[15];
        lDtoLP3 = lColArt[21];
        lDtoLP4 = lColArt[22];
        lDtoLP5 = lColArt[23];

        //aGrid.Setvalue "Precio", lPrecio
        aGrid.SetValue("Precio_EURO", lPrecio);
        aGrid.SetValue("Descuento", lDescuento);
        aGrid.SetValue("DtoLP1", lDtoLP1);
        aGrid.SetValue("DtoLP2", lDtoLP2);
        aGrid.SetValue("DtoGD", lDtoGD);
        //aGrid.Setvalue "PrecioCoste", lCoste
        aGrid.SetValue("DtoLP3", lDtoLP3);
        aGrid.SetValue("DtoLP4", lDtoLP4);
        aGrid.SetValue("DtoLP5", lDtoLP5);
    }

    }
}
catch (Exception)
{
    throw;
}
}

    public override void Grid_MenuBeforeExecute(ICntGridUsuario aGrid,
IAhoraTool aMenuItem, ref Valor aCancel)
    {
        try
        {
            if (System.String.Equals(aGrid.Name, "grdLineas",
StringComparison.Ordinal))
            {
                if (System.String.Equals(aMenuItem.Name, "mnuParte_Nuevo",
StringComparison.Ordinal))
                {
                    if ((bool)fbLineaAsociadaAParte(aGrid))
                    {
                        aCancel = true;
                        return;
                        //DEBUG - NUEVO - INTERCEPTAMOS LA CREACION DE PARTE
NUEVO x LINEA
                    }
                    else
                    {
                        if (!fbLineaAsociadaAParte(aGrid))
                        {
                            pGenerarParteLineaEspecifica(aGrid.GetValue("IdLinea"));
                            pRefrescarObj();
                            aCancel = true;
                            return;
                        }
                    }
                    else
                    {
                        gCn.Obj.ShowMsgBox("Ya hay un parte generado
para esta línea.", vbInformation, "Parte Vinculados", gForm);
                        aCancel = true;
                        return;
                    }
                }
            }
        }
    }
}

```

```

    }
}
catch (Exception)
{
    throw;
}
}

```

```

public Valor fbLineaAsociadaAParte(ICntGridUsuario aGrid)
{
    Valor fbLineaAsociadaAParte = ValorEmpty.Value;
    Valor lIdParte = ValorEmpty.Value;
    Valor lIdLinea = ValorEmpty.Value;
    Valor lIdPedido = ValorEmpty.Value;
    Valor lResult = ValorEmpty.Value;
    try
    {
        fbLineaAsociadaAParte = null;
        lResult = false;
        lIdPedido = aGrid.GetValue("IdPedido");
        lIdLinea = aGrid.GetValue("IdLinea");
        lIdParte = gCn.DameValorCampo("SELECT ISNULL(IdParte, 0) FROM
Pedidos_Cli_Lineas WHERE IdPedido = '" + (string)lIdPedido + "' AND IdLinea =
'" + (string)lIdLinea + "'");
        if ("'" + lIdParte != "" & (int)lIdParte > 0)
        {
            lResult = true;
            gCn.Obj.ShowMsgBox("No se puede crear un parte nuevo" +
vbCrLf + "La línea ya está asociada a un parte", vbInformation, "Parte nuevo",
gForm);
        }
        fbLineaAsociadaAParte = lResult;
    }
    catch (Exception)
    {
        throw;
    }
    return fbLineaAsociadaAParte;
}

```

```

public override void Menu_BeforeExecute(ICntMenuFormulario aMenu,
IAhoraTool aMenuItem, ref Valor aCancel)
{
    Valor lNPartes = ValorEmpty.Value;
    Valor lSQL = ValorEmpty.Value;
    try
    {
        if (System.String.Equals(aMenuItem.Name, "mnuParteSub(0)",
StringComparison.Ordinal))
        {
            lSQL = "SELECT ISNULL(COUNT(*), 0) FROM Pedidos_Cli_Lineas
WHERE IdParte IS NOT NULL AND IdPedido = " + ((IEnlaceObjetos)
(IEnlaceObjetos)gForm.Controls["EObjeto"]).ObjGlobal["IdPedido"];
            lNPartes = gCn.DameValorCampo((string)lSQL);
            if ((int)lNPartes > 0)
            {
                aCancel = true;
                gCn.Obj.ShowMsgBox("No se puede crear un parte nuevo" +
vbCrLf + "Hay líneas asociadas a partes", vbInformation, "Parte nuevo", gForm);
                return;
            }
        }
    }
}

```

```

        }

        if (System.String.Equals(aMenu.Name,
gForm.Controls["mnuMain"].Name, StringComparison.Ordinal))
        {
            if (System.String.Equals(aMenuItem.Name, "mnuVer_Partес",
StringComparison.Ordinal))
            {
                aCancel = true;
                pMostrarPartes();
                return;
            }
        }
    }
    catch (Exception)
    {
        throw;
    }
}

public void pMostrarParte(Valor viIdParte)
{
    IColecciones lCol = null;
    Valor lIdParte = ValorEmpty.Value;
    try
    {
        if ("" + viIdParte == "")
        {
            return;
        }

        lIdParte = viIdParte;

        lCol = gCn.Obj.DameColeccion("Partes", "WHERE NumParte = '" +
(string)lIdParte + "'");

        if (lCol is null)
        {
            gCn.Obj.ShowMsgBox("No se puede abrir el parte",
vbInformation, "Mostrar parte", gForm);
        }
        else
        {
            lCol.Show();
        }
    }
    catch (Exception)
    {
        throw;
    }
}

public void pOcultarPestanaConf()
{
    try
    {
        ((ICntTab)gForm.Controls["TabDatos"]).Item(3).Visible = false;
    }
    catch (Exception)
    {
        throw;
    }
}

```

```
    }  
}
```

```
public override void CargaObjeto()  
{  
    try  
    {  
        pOcultarPestanaConf();  
        pConfiguraComboListaPreciosDetalle();  
    }  
    catch (Exception)  
    {  
        throw;  
    }  
}
```

```
public override void ActualizaObjeto()  
{  
    try  
    {  
        pOcultarPestanaConf();  
    }  
    catch (Exception)  
    {  
        throw;  
    }  
}
```

```
public override void EliminaObjeto()  
{  
    try  
    {  
        pOcultarPestanaConf();  
    }  
    catch (Exception)  
    {  
        throw;  
    }  
}
```

```
public void pConfigurarPanelImpresos()  
{  
    int CAPTION_CAMPO_CALIDAD = 0;  
    Valor ANCHO_CAMPO_CALIDAD = ValorEmpty.Value;  
    Valor ALTO_CAMPO_CALIDAD = ValorEmpty.Value;  
    ICntBotonera lBtnra = null;  
    ICheckBoxUser lChk = null;  
    ITextoUsuario lTxt = null;  
    ICntPanel lPnlP = null;  
    ICntTab lTab = null;  
    try  
    {  
        lTab = (ICntTab)gForm.Controls["TabDatos"];  
        lPnlP = (ICntPanel)gForm.Controls.Add("Threed.SSPanel",  
"pers_pnlImpresos");  
        lPnlP.Visible = true;  
        lPnlP.ResizeInterior = true;  
        lPnlP.Text = "";
```

```

        //RAFA: es necesario añadir un campo de tipo texto que permanece
oculto para evitar el redimensionamiento del check
        // si sólo se crea el campo check, al ser SSPanel con
autosize, el check se muestra estirado de lado a lado del panel
        // añadiendo el TextBox oculto, parece que toma la referencia
del éste y no se redimensiona
        lTxt =
(ITextoUsuario)gForm.Controls.Add("Ahora0cx.Textousuario", "pers_edFecha0culto",
lPnlP);

        lTxt.ActivarScripts = true;
        lTxt.Formato = EFormatoCampo.FechaAnyoCompleto;
        lTxt.TipoDato = ETipoDato.Date;
        //.ObjOrigen = "EObjeto"
        //.ObjPOrigen = "pers_FechaInicioCertificacion"
        lTxt.Visible = false;
        //.Move lControl.Left + lControl.Width + 50, lControl.Top, 2200,
300

        lTxt.CaptionVisible = true;
        lTxt.CaptionLink = true;
        lTxt.CaptionControl = "Fecha Impreso";
        lTxt.CaptionWidth = ANCHO_CAPTION;
        lTxt.TabStop = true;
        lTxt.Enabled = true;
        //.BackColor = vbButtonFace
        //.ForeColor = vbBlue
        lTxt.TabIndex = 6;
        lTxt.Text = "";
        //lTxt.AplicaEstilo();
        pAnadirControlDebajo("pers_edFecha0culto", "", ANCHO_FECHAS,
ALTO_BUSCADORES, 0, DESPLAZA_H);

        lChk =
(ICheckBoxUser)gForm.Controls.Add("Ahora0cx.CheckBoxUser", "pers_chImpValorado",
lPnlP);

        lChk.ActivarScripts = true;
        lChk.ObjOrigen = "EObjeto";
        lChk.ObjPOrigen = "pers_ImpValorado";
        lChk.Visible = true;
        //.Move lControl.Left + lControl.Width + 50, lControl.Top, 2200,
300

        lChk.CaptionVisible = true;
        lChk.CaptionControl = "Imprimir valorado";
        lChk.CaptionWidth = ANCHO_CAPTION;
        lChk.TabStop = true;
        lChk.Enabled = true;
        //.BackColor = vbButtonFace
        //.ForeColor = vbBlue
        lChk.TabIndex = 8;
        //.AplicaEstilo
        pAnadirControlDebajo("pers_chImpValorado", "pers_edFecha0culto",
120, ALTO_BUSCADORES, -ALTO_BUSCADORES, DESPLAZA_H);

        //'---
---

        //'--- CAMPOS PARA LOS TEXTOS DE CABECERA Y PIE (OCULTOS)
        //'---
---

        //'---Botonera para textos de cabecera y pie
        lBtnra =
(ICntBotonera)gForm.Controls.Add("Ahora0CX.cntBotonera",
"pers_btnra_TextosCabYPieImpreso", lPnlP);
        lBtnra.ActivarScripts = true;

```

```

lBtnra.SeguridadObjeto = (EMenu)0;
lBtnra.BotonesMantenimiento = (EBotonesMant)0;
lBtnra.BotAsociados = false;
lBtnra.Visible = true;
lBtnra.BotonAdd("Texto cabecera", "pers_botTxtCabecera", "",
ETipo_Menu.eTipo_Menu_Boton, false, 275);
lBtnra.BotonAdd("Texto pie", "pers_botTxtPie", "",
ETipo_Menu.eTipo_Menu_Boton, false, 331);
lBtnra.HabilitaBotones();
lBtnra.Tag = "TextosCabYPieImpreso";
pAnadirControlDebajo("pers_btnra_TextosCabYPieImpreso",
"pers_chImpValorado", ANCHO_BOTONERTEXTOSIMP, ALTO_BOTONERATEXTOSIMP,
DESPLAZA_V, 0);

lTxt =
(ITextoUsuario)gForm.Controls.Add("AhoraOCX.TextoUsuario", "pers_edTxtCabecera",
lPnlP);

lTxt.ActivarScripts = true;
lTxt.Formato = EFormatoCampo.MayusculasyMinusculas;
lTxt.TipoDato = ETipoDato.String;
lTxt.ObjOrigen = "EObjeto";
lTxt.ObjPOrigen = "pers_TxtCabecera";
lTxt.Visible = false;
lTxt.CaptionVisible = true;
lTxt.CaptionControl = "Texto cabecera";
lTxt.CaptionWidth = 0;
lTxt.CaptionLink = false;
lTxt.CaptionPosition = (ECaptionPosition)1;
//lTxt.AplicaEstilo();
lTxt.TabStop = false;
lTxt.Enabled = false;
pAnadirControlDebajo("pers_edTxtCabecera",
"pers_btnra_TextosCabYPieImpreso", 50, 1, 1, 1);

lTxt =
(ITextoUsuario)gForm.Controls.Add("AhoraOCX.TextoUsuario", "pers_edTxtPie",
lPnlP);

lTxt.ActivarScripts = true;
lTxt.Formato = EFormatoCampo.MayusculasyMinusculas;
lTxt.TipoDato = ETipoDato.String;
lTxt.ObjOrigen = "EObjeto";
lTxt.ObjPOrigen = "pers_TxtPie";
lTxt.Visible = false;
lTxt.CaptionVisible = true;
lTxt.CaptionControl = "Texto pie";
lTxt.CaptionWidth = 0;
lTxt.CaptionLink = false;
lTxt.CaptionPosition = (ECaptionPosition)1;
//lTxt.AplicaEstilo();
lTxt.TabStop = false;
lTxt.Enabled = false;
pAnadirControlDerecha("pers_edTxtPie", "pers_edTxtCabecera", 50,
1, 1, 1);

ALTO_CAMPO_CALIDAD = 20;
ANCHO_CAMPO_CALIDAD = 267;
CAPTION_CAMPO_CALIDAD = 120;
lTxt =
(ITextoUsuario)gForm.Controls.Add("AhoraOCX.TextoUsuario",
"pers_edIdControlCalidad", lPnlP);
lTxt.ActivarScripts = true;
lTxt.Formato = EFormatoCampo.MayusculasyMinusculas;
lTxt.TipoDato = ETipoDato.String;
lTxt.ObjOrigen = "EObjeto";

```

```

lTxt.ObjPOrigen = "pers_IdControlCalidad";
lTxt.Visible = true;
//.Move LEFT, TOP, WIDTH, HEIGHT
lTxt.CaptionVisible = true;
lTxt.CaptionControl = "Identificador calidad";
lTxt.CaptionWidth = CAPTION_CAMPO_CALIDAD;
lTxt.CaptionLink = false;
lTxt.CaptionPosition = (ECaptionPosition)0;
//lTxt.AplicaEstilo();
lTxt.TabStop = true;
lTxt.Enabled = true;
pAnadirControlDebajo("pers_edIdControlCalidad",
"pers_btnra_TextosCabYPieImpreso", ANCHO_CAMPO_CALIDAD, ALTO_CAMPO_CALIDAD,
DESPLAZA_V, 0);

```

```

lTab.InsertItem(1000, "Impresos", lPnlP, 0);

```

```

lTab = null;
lPnlP = null;

```

```

}
catch (Exception)
{
    throw;
}
}

```

```

public void pAsignarValoresPorDefecto()
{

```

```

    Valor lIdentificador = ValorEmpty.Value;
    Valor lTxtPie = ValorEmpty.Value;
    Valor lTxtCabecera = ValorEmpty.Value;
    Valor lIdDomEmp = ValorEmpty.Value;
    Valor lIdEmpresa = ValorEmpty.Value;
    Valor lImpAlb = ValorEmpty.Value;
    Valor lSQL = ValorEmpty.Value;
    Valor lIdCliente = ValorEmpty.Value;

```

```

    try
    {

```

```

        //Imprimir albaranes por defecto
        lIdCliente = ((IControlEnlazable)gForm.Controls["IdCliente" +
"_" + CStr(0)]).Text;

```

```

        lSQL = "SELECT ISNULL(AlbValorado, 0) FROM
Clientes_Datos_Economicos WHERE IdCliente = '" + (string)lIdCliente + "'";
        lImpAlb = gCn.DameValorCampo((string)lSQL);
        ((IControlEnlazable)gForm.Controls["pers_chImpValorado"]).Value
= false;

```

```

        if (" " + lImpAlb != " " & (int)lImpAlb != 0)
        {

```

```

            ((IControlEnlazable)gForm.Controls["pers_chImpValorado"]).Value = true;
        }

```

```

        //Banco de la empresa

```

```

        lIdEmpresa = gCn.IdEmpresa;
        lIdCliente = ((IControlEnlazable)gForm.Controls["IdCliente" +
"_" + CStr(0)]).Text;
        lSQL = "SELECT dbo.funPers_Dame_Domiciliacion_Empresa ('" +
(string)lIdEmpresa + "'", "'" + lIdCliente + "'", NULL)";
        lIdDomEmp = gCn.DameValorCampo((string)lSQL);

```

```

        ((IComboUsuario)gForm.Controls["pers_cbIdDomiciliacionEmp"]).Text = "";
        if (" " + lIdDomEmp != " ")
        {

```



```

((IComboUsuario)gForm.Controls["pers_cbIdDomiciliacionEmp"]).Text = lIdDomEmp;
    }

    //Textos adicionales de cabecera y pie del documento
    lSQL = "SELECT pers_TxtCabeceraPDV FROM Conf_Empresas WHERE
IdEmpresa = '" + (string)lIdEmpresa + "'";
    lTxtCabecera = gCn.DameValorCampo((string)lSQL);

    lSQL = "SELECT pers_TxtPiePDV FROM Conf_Empresas WHERE IdEmpresa
= '" + (string)lIdEmpresa + "'";
    lTxtPie = gCn.DameValorCampo((string)lSQL);

    ((IEnlaceObjetos)
(IEnlaceObjetos)gForm.Controls["EObjeto"]).ObjGlobal["pers_TxtCabecera"] =
lTxtCabecera;
    ((ITextoUsuario)gForm.Controls["pers_edTxtCabecera"]).Text =
lTxtCabecera;
    ((IEnlaceObjetos)
(IEnlaceObjetos)gForm.Controls["EObjeto"]).ObjGlobal["pers_TxtPie"] = lTxtPie;
    ((ITextoUsuario)gForm.Controls["pers_edTxtPie"]).Text = lTxtPie;

    lSQL = "SELECT pers_IdControlCalidadPDV FROM Conf_Empresas WHERE
IdEmpresa = '" + (string)lIdEmpresa + "'";
    lIdentificador = gCn.DameValorCampo((string)lSQL);

    ((IEnlaceObjetos)
(IEnlaceObjetos)gForm.Controls["EObjeto"]).ObjGlobal["pers_IdControlCalidad"] =
lIdentificador;
    ((ITextoUsuario)gForm.Controls["pers_edIdControlCalidad"]).Text
= lIdentificador;

    }
    catch (Exception)
    {
        throw;
    }
}

//Sub Botonera_BeforeExecute(aBotonera, aBoton, aCancel)
// If aBotonera.Name = gForm.Controls("Botonera").Name Then
// If aBoton.Name = "botGuardar" Then
// If gForm.Eobjeto.ObjGlobal.Nuevo Then
// pAsignarValoresPorDefecto
// End If
// End If
// End If
//End Sub

public override void Combo_AfterUpdate(IComboUsuario aCombo)
{
    try
    {
        if (System.String.Equals(aCombo.Name,
((IControl)gForm.Controls["IdCliente" + "_" + CStr(0)]).Name,
StringComparison.Ordinal) | System.String.Equals(aCombo.Name,
((IControl)gForm.Controls["Cliente" + "_" + CStr(0)]).Name,
StringComparison.Ordinal))
        {
            if
(((IEnlaceObjetos)gForm.FindControl("EObjeto")).ObjGlobal.Nuevo)
            {
                pAsignarValoresPorDefecto();
            }
        }
    }
}

```

```

        }
        pConfiguraComboListaPreciosDetalle();
    }
    if (System.String.Equals(aCombo.Name,
gForm.Controls["IdLista"].Name, StringComparison.Ordinal))
    {
        pConfiguraComboListaPreciosDetalle();
    }
}
catch (Exception)
{
    throw;
}
}

public override void Grid_MenuAfterExecute(ICntGridUsuario aGrid,
IAhoraTool aMenuItem)
{
    bool aCancel = false;
    try
    {
        if (System.String.Equals(aGrid.Name, "grdLineas",
StringComparison.Ordinal))
        {

            // == M E T O D O (1) ==
            if (System.String.Equals(aMenuItem.Name,
"pers_mnuAsociarParte", StringComparison.Ordinal))
            {

                if ((bool)fbLineaAsociadaAParte(aGrid))
                {
                    aCancel = true;
                    return;
                }
                pAsignarLineaAParte(aGrid.GetValue("IdParte"),
aGrid.GetValue("IdPedido"), aGrid.GetValue("IdLinea"));
                pRefrescarObj();

            }

            // == M E T O D O (2) ==
            if (System.String.Equals(aMenuItem.Name,
"pers_mnuNuevoParteLinea", StringComparison.Ordinal))
            {

                if (!fbLineaAsociadaAParte(aGrid))
                {

                    pGenerarParteLineaEspecifica(aGrid.GetValue("IdLinea"));
                    pRefrescarObj();
                }
                else
                {
                    gCn.Obj.ShowMsgBox("Ya hay un parte generado para
esta línea.", vbInformation, "Parte Vinculados", gForm);
                    aCancel = true;
                    return;
                }

                //lIdParte = fiObtenerIdParte(lIdDoc)

```

```

        //lIdPedido = aGrid.GetValue("IdPedido")
        //lIdLinea = aGrid.GetValue("IdLinea")

        //      If Not fbRelacionarPedidoYParteNuevo(lIdPedido,
lIdLinea, lIdParte) Then
            //          pBorrarNuevoParte lIdDoc
            //          Exit Sub
            //      End If

            //COMO PODEMOS MOSTRAR EL NUEVO PARTE CREADO, SIN EL
IDDOC ¿?¿?¿
            //pMostrarNuevoParte lIdDoc
        }

        if (System.String.Equals(aMenuItem.Name,
"pers_mnuVerPartesLinea", StringComparison.Ordinal))
        {
            pMostrarPartesLinea(aGrid);
        }
    }
}
catch (Exception)
{
    throw;
}
}

public void pAsignarLineaAParte(Valor viIdParte, Valor viIdPedido, Valor
viIdLinea)
{
    Valor lParteDes = ValorEmpty.Value;
    Valor lParteOri = ValorEmpty.Value;
    bool lAsignar = false;
    Valor lLista = ValorEmpty.Value;
    IAsistente lAsist = null;
    try
    {
        lAsist = (IAsistente)gCn.AhoraProceso("DameAsistente", out _,
gCn);
        lAsist.Text = "Indique el parte a asignar";
        lAsist.AddField("DescripDocs", "Parte", ETipoDato.String,
ValorEmpty.Value, true, false, true);
        lAsist.AddField("IdParte", "Id", ETipoDato.String, 0, false,
true);
        lAsist.AddField("Descrip", "Descripción", ETipoDato.String,
ValorEmpty.Value, true, false, true);
        lLista = lAsist.Carga("vpers_Partес_Lineas_Pedido", " IdPedido =
" + viIdPedido + " ORDER BY DescripDocs", "IdParte", "IdParte", viIdParte, 0, 0,
gCn.Sesion.MainForm);
        if (Len(lLista) > 0)
        {
            lAsignar = true;

            if ((" " + viIdParte != " ") & (" " + lLista != " " +
viIdParte))
            {
                lParteOri = gCn.DameValorCampo("SELECT DescripDocs FROM
vpers_VDLG_Partес WHERE IdParte = " + viIdParte);
                lParteDes = gCn.DameValorCampo("SELECT DescripDocs FROM
vpers_VDLG_Partес WHERE IdParte = " + lLista);
                if (gCn.Obj.ShowMsgBox("Va a reasignar la línea del
parte " + (string)lParteOri + " al parte " + (string)lParteDes + vbCrLf + "¿Está

```

```

seguro de que desea continuar?", vbYesNo | vbQuestion | vbDefaultButton2,
"Asignar partes", gForm) == vbNo)
    {
        lAsignar = true;
    }
}

if (lAsignar)
{
    if (!gCn.ExecuteSql("UPDATE Pedidos_Cli_Lineas SET
IdParte = '" + (string)lLista + "' WHERE IdPedido = '" + (string)viIdPedido +
"' AND IdLinea = '" + (string)viIdLinea + "'"))
    {
        gCn.Obj.ShowMsgBox("No se ha podido asignar la línea
al parte", vbExclamation, "Atención", gForm);
    }
    else
    {
        //gForm.Controls("grdLineas").Refresh

        ((IEnlaceObjetos)gForm.FindControl("EObjeto")).Refresh();
        pEstableceFocoAGrid(gForm.Controls["grdLineas"]);
        //Redimensionamos el formulario
        gCn.Obj.Form_AutoResize(gForm);
        gCn.Obj.ShowMsgBox("Linea asignada al parte
correctamente", vbInformation, "Atención", gForm);
    }
}

}
else
{
    gCn.Obj.ShowMsgBox("No se ha seleccionado ningún parte",
vbExclamation, "Atención", gForm);
}
}
catch (Exception)
{
    throw;
}
}

public void pEstableceFocoAGrid(IControl aGrid)
{
    try
    {
        if (((ICntGridUsuario)aGrid).RowCount == 0)
        {
            aGrid.Select();
            ((ICntGridUsuario)aGrid).Bookmark = 1;
            ((ICntGridUsuario)aGrid).Col = 1;
            ((ICntGridUsuario)aGrid).Row = 1;
        }
    }
    catch (Exception)
    {
        throw;
    }
}

public void pRefrescarObj()
{

```

```

        try
        {
            gForm.Controls["EObjeto"].Refresh();
            CargaObjeto();
            ((ICntGridUsuario)gForm.Controls["grdLineas"]).Refrescar();
        }
        catch (Exception)
        {
            throw;
        }
    }

    public void pAnadirCamposEconomicos()
    {
        IComboUsuario lCombo = null;
        Valor lSQL = ValorEmpty.Value;
        ICntPanel lPnlP = null;
        try
        {
            lPnlP = (ICntPanel)gForm.Controls["FrameEconomicos"];

            lSQL = "SELECT Descrip, IdDomiciliacion FROM
VCombo_Empresa_Domiciliaciones WHERE Activa=1 ORDER BY IdDomiciliacion";

            lCombo =
(IComboUsuario)gForm.Controls.Add("AhoraOCX.ComboUsuario",
"pers_cbIdDomiciliacionEmp", lPnlP);
            lCombo.ActivarScripts = true;
            lCombo.C1Nombre = "Descrip";
            lCombo.C1TipoDato = (ETipoDatoColumnaCombo)8;
            lCombo.C1Anchura = 200;
            lCombo.C2Nombre = "IdDomiciliacion";
            lCombo.C2TipoDato = (ETipoDatoColumnaCombo)3;
            lCombo.C2Anchura = 100;
            lCombo.CActiva = 2;
            lCombo.NColumnas = 2;
            lCombo.Descripcion = (string)lSQL;
            lCombo.Formato = EFormatoCampo.SinDecimales;
            lCombo.TipoDato = ETipoDato.Numeric;
            lCombo.ObjOrigen = "EObjeto";
            lCombo.ObjPOrigen = "pers_IdDomiciliacionEmp";
            lCombo.Visible = true;
            lCombo.CaptionVisible = true;
            lCombo.CaptionControl = "Domi.Emp:";
            lCombo.CaptionWidth = 67;
            lCombo.Enabled = true;
            lCombo.CaptionLink = false;
            lCombo.TabStop = true;
            lCombo.TabIndex = gForm.Controls["Fianzas"].TabIndex + 1;
            //lCombo.AplicaEstilo();
            lCombo.Refresh();
            pAnadirControlDebajo("pers_cbIdDomiciliacionEmp", "Fianzas",
gForm.Controls["Fianzas"].Width, gForm.Controls["Fianzas"].Height, 0, 0);
            pAnadirControlDerecha("IdTipoCli", "CambioBloqueado",
gForm.Controls["Comision"].Width + 27, gForm.Controls["IdTipoCli"].Height,
DESPLAZA_H, 0);

        }
        catch (Exception)
        {
            throw;
        }
    }
}

```

```

public void pMostrarTextoCabeceraYPie(IControl aTextoUsuario)
{
    Valor lTexto = ValorEmpty.Value;
    IFrmZoom lzFrm = null;
    try
    {
        ///' Set lObsZ = gCn.AhoraProceso("DameZoom", False, gCn)
        ///' lObsZ.Caption = aTextoUsuario.CaptionControl
        ///' lObsZ.Ver aTextoUsuario, False, Len(aTextoUsuario)

        lzFrm = (IFrmZoom)gCn.AhoraProceso("DameZoom", out _, gCn);
        ((ITextoUsuario)lzFrm.FindControl("txtZoom")).ReadOnly = false;
        lTexto =
lzFrm.Carga(((IControlEnlazable)aTextoUsuario).CaptionControl,
((IControlEnlazable)aTextoUsuario).Text);
        if ((int)lzFrm.BotonPulsado != 2)
        {
            ((IControlEnlazable)aTextoUsuario).Text = lTexto;
            var lObject =
((IEnlaceObjetos)gForm.Controls["EObjeto"]).ObjGlobal(((IControlEnlazable)aTexto
Usuario).ObjPOrigen);
            ((IEnlaceObjetos)gForm.Controls["EObjeto"]).Modificado =
true;
        }
    }
    catch (Exception)
    {
        throw;
    }
}

//public void pConfigurarColumnaIconosPartesDetalle()
//{
//    ICntGridUsuario lGrid = null;
//    try
//    {
//        /// Debemos hacerlo así, ya que la columna está añadida a la
grid por configuración y no por código
//        lGrid = (ICntGridUsuario)gForm.Controls["grdLineas"];
//        if ((lGrid.ExisteColumna("pers_HayPartes") == true))
//        {
//            lGrid.ValueItems("pers_HayPartes", "SELECT Id, Icono FROM
vPers_ICOS_Pedidos_Cli_Lineas_Part", true);
//        }
//    }
//    catch (Exception)
//    {
//        throw;
//    }
//}

public void pConfigurarValueItemsColIVA()
{
    ICntGridUsuario lGrid = null;
    try
    {
        /// Debemos hacerlo así, ya que la columna está añadida a la grid
por configuración y no por código
        lGrid = (ICntGridUsuario)gForm.Controls["grdLineas"];

```

```

        if ((lGrid.ExisteColumna("IdIVA") == true))
        {
            lGrid.ValueItems("IdIva", "SELECT IdIva, Descrip FROM
VCOMBO_IVAS", false);
        }
    }
    catch (Exception)
    {
        throw;
    }
}

public void pConfiguraComboListaPreciosDetalle()
{
    Valor lSQL1 = ValorEmpty.Value;
    Valor lSQL = ValorEmpty.Value;
    ICntGridUsuario lGrid = null;
    Valor lIdLista = ValorEmpty.Value;
    Valor lIdDelegacion = ValorEmpty.Value;
    Valor lIdCliente = ValorEmpty.Value;
    try
    {
        if (!fbTrabajaConListaPreciosEnDetalle())
        {
            return;
        }

        lIdCliente = ((IControlEnlazable)gForm.Controls["IdCliente" +
"_" + CStr(0)]).Text;
        lIdDelegacion = gCn.IdDelegacion;
        lIdLista = ((IControlEnlazable)gForm.Controls["IdLista"]).Text;

        lGrid = (ICntGridUsuario)gForm.Controls["grdLineas"];
        if ((lGrid.ExisteColumna("pers_IdLista") == true))
        {
            lSQL = "SELECT IdLista, IdLista Lista, Descrip FROM
dbo.FunListas_Precios_Cli('"' + (string)lIdCliente + "''," + lIdDelegacion +
"')";
            lSQL1 = "SELECT IdLista, Descrip FROM
dbo.FunListas_Precios_Cli('"' + (string)lIdCliente + "''," + lIdDelegacion +
"')";
            lSQL = lSQL + " ORDER BY IdLista";
            lSQL1 = lSQL1 + " ORDER BY IdLista";

            lGrid.Campo("pers_IdLista").Combo = (string)lSQL;
            lGrid.Campo("pers_IdLista").ValueItems = (string)lSQL1;
            lGrid.Campo("pers_IdLista").RowRefreshCombo = true;
            lGrid.Combo("pers_IdLista", (string)lSQL);
            lGrid.ValueItems("pers_IdLista", (string)lSQL1, false);
            lGrid.Campo("pers_IdLista").RowRefreshCombo = false;
            //.Combo("pers_IdLista").Columns(0).Width = 0;
            lGrid.Combo("pers_IdLista").Columns[1].Width = 7;
            lGrid.Combo("pers_IdLista").Columns[2].Width = 53;

            lGrid.Campo("pers_IdLista").Default = lIdLista;
        }
    }
    catch (Exception)
    {
        throw;
    }
}

```

```

    }

    public override void Grid_BeforeLoad(ICntGridUsuario aGrid)
    {
        try
        {
            // Aunque este código se ejecuta, se machaca esta configuración
            // porque la grid tiene personalización de columnas propia
            // SE REPITE EN EL AFTERLOAD DE LA GRID
            if (System.String.Equals(aGrid.Name,
                gForm.Controls["grdLineas"].Name, StringComparison.Ordinal))
            {
                // == M E T O D O (1) ==
                if ((aGrid.ExisteColumna("@pers_IdParte") == true))
                {
                    aGrid.Campo("@pers_IdParte").Column.Visible =
                    fiModoGeneracionPartes() == 1;
                    if (!aGrid.Campo("@pers_IdParte").Column.Visible)
                    {
                        aGrid.Campo("@pers_IdParte").Column.Width = 0;
                    }
                }

                // == M E T O D O (2) ==
                if ((aGrid.ExisteColumna("pers_HayPartes") == true))
                {
                    aGrid.Campo("pers_HayPartes").Column.Visible =
                    fiModoGeneracionPartes() == 2;
                    if (!aGrid.Campo("pers_HayPartes").Column.Visible)
                    {
                        aGrid.Campo("pers_HayPartes").Column.Width = 0;
                    }
                }

                if ((aGrid.ExisteColumna("pers_IdLista") == true))
                {
                    if (!fbTrabajaConListaPreciosEnDetalle())
                    {
                        aGrid.Campo("pers_IdLista").Column.Visible = false;
                        aGrid.Campo("pers_IdLista").Column.Width = 0;
                        return;
                    }

                    aGrid.Campo("pers_IdLista").Default = ((IEnlaceObjetos)
                    (IEnlaceObjetos)gForm.Controls["EObjeto"]).ObjGlobal["IdLista"];
                }
            }
        }
        catch (Exception)
        {
            throw;
        }
    }

    public override void Grid_AfterLoad(ICntGridUsuario aGrid)
    {
        try
        {
            if (System.String.Equals(aGrid.Name,

```



```

gForm.Controls["grdLineas"].Name, StringComparison.Ordinal))
{
    // == M E T O D O (1) ==
    if ((aGrid.ExisteColumna("@pers_IdParte") == true))
    {
        aGrid.Campo("@pers_IdParte").Column.Visible =
fiModoGeneracionPartes() == 1;
        if (!aGrid.Campo("@pers_IdParte").Column.Visible)
        {
            aGrid.Campo("@pers_IdParte").Column.Width = 0;
        }
    }

    // == M E T O D O (2) ==
    if ((aGrid.ExisteColumna("pers_HayPartes") == true))
    {
        aGrid.Campo("pers_HayPartes").Column.Visible =
fiModoGeneracionPartes() == 2;
        if (!aGrid.Campo("pers_HayPartes").Column.Visible)
        {
            aGrid.Campo("pers_HayPartes").Column.Width = 0;
        }
    }

    if ((aGrid.ExisteColumna("pers_IdLista") == true))
    {
        if (!fbTrabajaConListaPreciosEnDetalle())
        {
            aGrid.Campo("pers_IdLista").Column.Visible = false;
            aGrid.Campo("pers_IdLista").Column.Width = 0;
            return;
        }

        aGrid.Campo("pers_IdLista").Default = ((IEnlaceObjetos)
(IEnlaceObjetos)gForm.Controls["EObjeto"]).ObjGlobal["IdLista"];
    }
}
catch (Exception)
{
    throw;
}
}

public override void Grid_BeforeColUpdate(ICntGridUsuario aGrid, int
aColIndex, Valor aOldValue, ref Valor aCancel)
{
    Valor lIdListaLin = ValorEmpty.Value;
    Valor lIdListaCab = ValorEmpty.Value;
    try
    {
        if (System.String.Equals(aGrid.Name,
gForm.Controls["grdLineas"].Name, StringComparison.Ordinal))
        {
            if (aGrid.ColIndex("IdArticulo") == aColIndex)
            {
                if (!fbTrabajaConListaPreciosEnDetalle())
                {
                    return;
                }
            }
        }
    }
}

```

```

        lIdListaCab =
((IControlEnlazable)gForm.Controls["IdLista"]).Text;
        lIdListaLin = aGrid.GetValue("pers_IdLista");
        if (" " + lIdListaLin == "")
        {
            lIdListaLin = (int)(-1);
        }
        if ((int)lIdListaLin != (int)lIdListaCab)
        {
            aGrid.SetValue("pers_IdLista", lIdListaCab);
        }
    }
}
catch (Exception)
{
    throw;
}
}

public bool fbTrabajaConListaPreciosEnDetalle()
{
    bool fbTrabajaConListaPreciosEnDetalle = false;
    try
    {
        fbTrabajaConListaPreciosEnDetalle =
(bool)fvLeerParametroEmpresa("pers_ListasPreciosDetalleVentas",
"Conf_Empresas");
    }
    catch (Exception)
    {
        throw;
    }
    return fbTrabajaConListaPreciosEnDetalle;
}

public void pMostrarPartes()
{
    IColecciones lCol = null;
    Valor lWhere = ValorEmpty.Value;
    string lIdPedido = "";
    try
    {
        lIdPedido = ((IEnlaceObjetos)
(IEnlaceObjetos)gForm.Controls["EObjeto"]).ObjGlobal["IdPedido"];
        lWhere = "WHERE IdParte IN (SELECT IdParte FROM
vpers_Part es_Origen WHERE IdPedido = " + lIdPedido + " GROUP BY IdParte)";

        lCol = gCn.Obj.DameColeccion("Partes", (string)lWhere);
        if (lCol is null)
        {
            gCn.Obj.ShowMsgBox("No hay partes asociados al pedido.",
vbInformation, "Partes", gForm);
            lCol = null;
            return;
        }
        lCol.Show();
    }
    catch (Exception)
    {

```

```

        throw;
    }
}

public int fiModoGeneracionPartes()
{
    int fiModoGeneracionPartes = 0;
    try
    {
        //V1.1
        //lSQL = "SELECT pers_ModoPartesDePedidos FROM Conf_Empresas
WHERE IdEmpresa = " & gCn.IdEmpresa
        //fiModoGeneracionPartes = CInt(gCn.DameValorCampo(CStr(lSQL)))
        //MENU PERSONALIZADO DE PARTES
        fiModoGeneracionPartes = 0;
    }
    catch (Exception)
    {
        throw;
    }
    return fiModoGeneracionPartes;
}

public Valor fsPartesHeredaSeriePedido()
{
    Valor fsPartesHeredaSeriePedido = ValorEmpty.Value;
    Valor lSQL = ValorEmpty.Value;
    try
    {
        lSQL = "SELECT UPPER(dbo.Valor_Parametro
('PARTE_HEREDA_SERIEPEDIDO',USER))";
        fsPartesHeredaSeriePedido = gCn.DameValorCampo((string)lSQL);
    }
    catch (Exception)
    {
        throw;
    }
    return fsPartesHeredaSeriePedido;
}

public Valor fiPartesTaller()
{
    Valor fiPartesTaller = ValorEmpty.Value;
    Valor lSQL = ValorEmpty.Value;
    try
    {
        lSQL = "SELECT UPPER(dbo.Valor_Parametro
('PARTE_TALLER',USER))";
        fiPartesTaller = gCn.DameValorCampo((string)lSQL);
    }
    catch (Exception)
    {
        throw;
    }
    return fiPartesTaller;
}

public Valor fiTipoGarantia()
{
    Valor fiTipoGarantia = ValorEmpty.Value;

```

```

        Valor lSQL = ValorEmpty.Value;
        try
        {
            lSQL = "SELECT UPPER(dbo.Valor_Parametro
(''PARTE_TIPOGARANTIA'',USER))";
            fiTipoGarantia = gCn.DameValorCampo((string)lSQL);
        }
        catch (Exception)
        {
            throw;
        }
        return fiTipoGarantia;
    }

```

```

    public Valor fiUnidadesGarantia()
    {
        Valor fiUnidadesGarantia = ValorEmpty.Value;
        Valor lSQL = ValorEmpty.Value;
        try
        {
            lSQL = "SELECT UPPER(dbo.Valor_Parametro
(''PARTE_UNIDADES GARANTIA'',USER))";
            fiUnidadesGarantia = gCn.DameValorCampo((string)lSQL);
        }
        catch (Exception)
        {
            throw;
        }
        return fiUnidadesGarantia;
    }

```

```

    public Valor fsListasPrecios()
    {
        Valor fsListasPrecios = ValorEmpty.Value;
        Valor lSQL = ValorEmpty.Value;
        try
        {
            lSQL = "SELECT UPPER(dbo.Valor_Parametro
(''PARTES_LISTAS PRECIOS'',USER))";
            fsListasPrecios = gCn.DameValorCampo((string)lSQL);
        }
        catch (Exception)
        {
            throw;
        }
        return fsListasPrecios;
    }

```

```

    public Valor fsObtenerSerieFacturacion()
    {
        Valor fsObtenerSerieFacturacion = ValorEmpty.Value;
        Valor lSQL = ValorEmpty.Value;
        Valor lIdTaller = ValorEmpty.Value;
        Valor lResult = ValorEmpty.Value;
        try
        {
            fsObtenerSerieFacturacion = null;
            lResult = ((IEnlaceObjetos)
(IEnlaceObjetos)gForm.Controls["EObjeto"]).ObjGlobal["SeriePedido"];
            if (fsPartesHeredaSeriePedido() == "OFF")
            {

```

```

        lIdTaller = fiPartesTaller();
        lSQL = "SELECT SerieFacturacion FROM PartesTalleres WHERE
IdTaller = " + lIdTaller;
        lResult = gCn.DameValorCampo((string)lSQL);
    }
    fsObtenerSerieFacturacion = lResult;
}
catch (Exception)
{
    throw;
}
return fsObtenerSerieFacturacion;
}

```

```

public Valor fiObtenerIdParte(string viIdDoc)
{
    Valor fiObtenerIdParte = ValorEmpty.Value;
    Valor lSQL = ValorEmpty.Value;
    try
    {
        lSQL = "SELECT IdParte FROM Partes WHERE IdDoc = " + viIdDoc;
        fiObtenerIdParte = gCn.DameValorCampo((string)lSQL);
    }
    catch (Exception)
    {
        throw;
    }
    return fiObtenerIdParte;
}

```

```

public Valor fiObtenerListaPrecios()
{
    Valor fiObtenerListaPrecios = ValorEmpty.Value;
    Valor lResult = ValorEmpty.Value;
    try
    {
        fiObtenerListaPrecios = null;
        lResult = ((IEnlaceObjetos)
(IEnlaceObjetos)gForm.Controls["EObjeto"]).ObjGlobal["IdLista"];
        if (fsListasPrecios() == "OFF")
        {
            lResult = 0;
        }
        fiObtenerListaPrecios = lResult;
    }
    catch (Exception)
    {
        throw;
    }
    return fiObtenerListaPrecios;
}

```

```

public Valor fiNuevoParte()
{
    Valor fiNuevoParte = ValorEmpty.Value;
    IParte lParte = null;
    Valor lIdSubtipo = ValorEmpty.Value;
    Valor lIdTipoParte = ValorEmpty.Value;
    Valor lIdContactoF = ValorEmpty.Value;
    Valor lClientePadre = ValorEmpty.Value;
    Valor lIdContactoA = ValorEmpty.Value;
}

```

```

Valor lIdContacto = ValorEmpty.Value;
Valor lFormaPago = ValorEmpty.Value;
Valor lIdTaller = ValorEmpty.Value;
Valor lSQL = ValorEmpty.Value;
int lOrigenGarantia = 0;
int lEnGarantia = 0;
Valor lUnidadesGarantia = ValorEmpty.Value;
int lGarantia = 0;
int lReparacionEnCli = 0;
int lIdAveria = 0;
int lIdAnomalia = 0;
Valor lNULL = ValorEmpty.Value;
Valor lModeloMaquina = ValorEmpty.Value;
Valor lIdMarcaMaquina = ValorEmpty.Value;
Valor lIdTipoMaquina = ValorEmpty.Value;
Valor lIdArticulo = ValorEmpty.Value;
Valor lIdPedido = ValorEmpty.Value;
Valor lIVA = ValorEmpty.Value;
Valor lDescuento = ValorEmpty.Value;
Valor lIdLista = ValorEmpty.Value;
Valor lIdMoneda = ValorEmpty.Value;
Valor lIdEmpleado = ValorEmpty.Value;
Valor lIdCentroCoste = ValorEmpty.Value;
Valor lDescrip = ValorEmpty.Value;
Valor lIdCliente = ValorEmpty.Value;
Valor lIdProyecto = ValorEmpty.Value;
Valor lSerieFacturacion = ValorEmpty.Value;
int lResult = 0;
try
{
    lResult = 0;

    lSerieFacturacion = fsObtenerSerieFacturacion();
    lIdProyecto = ((IEnlaceObjetos)
(IEnlaceObjetos)gForm.Controls["EObjeto"]).ObjGlobal["IdProyecto"];
    lIdCliente = ((IEnlaceObjetos)
(IEnlaceObjetos)gForm.Controls["EObjeto"]).ObjGlobal["IdCliente"];
    lDescrip = ((IEnlaceObjetos)
(IEnlaceObjetos)gForm.Controls["EObjeto"]).ObjGlobal["DescripcionPed"];
    lIdCentroCoste = ((IEnlaceObjetos)
(IEnlaceObjetos)gForm.Controls["EObjeto"]).ObjGlobal["IdCentroCoste"];
    lIdEmpleado = gCn.IdEmpleado;
    lIdMoneda = ((IEnlaceObjetos)
(IEnlaceObjetos)gForm.Controls["EObjeto"]).ObjGlobal["IdMoneda"];
    lIdLista = fiObtenerListaPrecios();
    lDescuento = 0;
    lIVA = ((IEnlaceObjetos)
(IEnlaceObjetos)gForm.Controls["EObjeto"]).ObjGlobal["IdIva"];
    lIdPedido = ((IEnlaceObjetos)
(IEnlaceObjetos)gForm.Controls["EObjeto"]).ObjGlobal["IdPedido"];

    //Datos de la máquina
    lIdArticulo = "GEN. PARTE";
    lIdTipoMaquina = 0;
    lIdMarcaMaquina = 0;
    lModeloMaquina = lNULL;
    lIdAnomalia = 0;
    lIdAveria = 0;
    lReparacionEnCli = 1;
    lGarantia = 0;
    lUnidadesGarantia = fiUnidadesGarantia();
    lEnGarantia = 0;
    lOrigenGarantia = 0;

```

```

        lSQL = "SELECT IdTaller FROM PartesTalleres WHERE
SerieFacturacion = " + lSerieFacturacion;
        lIdTaller = gCn.DameValorCampo((string)lSQL);
        if ("" + lIdTaller == "")
        {
            lIdTaller = fiPartesTaller();
        }

        lSQL = "SELECT IdLista FROM Clientes_Datos_Economicos WHERE
IdCliente = '" + (string)lIdCliente + "'";
        lIdLista = gCn.DameValorCampo((string)lSQL);

        lSQL = "SELECT FormaPago FROM Clientes_Datos_Economicos WHERE
IdCliente = '" + (string)lIdCliente + "'";
        lFormaPago = gCn.DameValorCampo((string)lSQL);

        lSQL = "SELECT IdContacto FROM Clientes_Datos WHERE IdCliente =
'" + (string)lIdCliente + "'";
        lIdContacto = gCn.DameValorCampo((string)lSQL);

        lSQL = "SELECT IdContactoA FROM Clientes_Datos WHERE IdCliente =
'" + (string)lIdCliente + "'";
        lIdContactoA = gCn.DameValorCampo((string)lSQL);

        lSQL = "SELECT Padre FROM Clientes_Datos WHERE IdCliente = '" +
(string)lIdCliente + "'";
        lClientePadre = gCn.DameValorCampo((string)lSQL);

        //'Si es una planta, el contacto de facturación es el del
cliente
        if ("" + lClientePadre != "")
        {
            lSQL = "SELECT IdContactoF FROM Clientes_Datos WHERE
IdCliente = '" + (string)lClientePadre + "'";
        }
        else
        {
            lSQL = "SELECT IdContactoF FROM Clientes_Datos WHERE
IdCliente = '" + (string)lIdCliente + "'";
        }
        lIdContactoF = gCn.DameValorCampo((string)lSQL);

        lIdTipoParte = 0;
        lIdSubtipo = 0;

        lParte = (IParte)gCn.Obj.DameObjStr("Parte");
        lParte["IdTaller"] = lIdTaller;
        lParte["SerieFacturacion"] = lSerieFacturacion;
        lParte["IdTipoParte"] = lIdTipoParte;
        lParte["IdSubtipo"] = lIdSubtipo;
        lParte["IdPedido"] = lIdPedido;
        lParte["IdCliente"] = lIdCliente;
        lParte["IdContacto"] = lIdContacto;
        lParte["IdArticulo"] = lIdArticulo;
        lParte["IdTipoMaquina"] = lIdTipoMaquina;
        lParte["IdMarcaMaquina"] = lIdMarcaMaquina;
        lParte["ModeloMaquina"] = lModeloMaquina;
        lParte["IdAnomalia"] = 0;
        lParte["IdAveria"] = 0;
        lParte["Descrip"] = lDescrip;
        lParte["ReparacionEnCli"] = 1;
        lParte["Garantia"] = 0;
        lParte["UnidadesGarantia"] = 1;
        lParte["EnGarantia"] = 0;

```

```

        lParte["OrigenGarantia"] = 0;
        lParte["FechaParte"] = Date();
        lParte["FechaEntrada"] = Date();
        lParte["FechaInicio"] = lNULL;
        lParte["FechaFin"] = lNULL;
        lParte["FechaFinEstimada"] = lNULL;
        lParte["FechaSalida"] = lNULL;
        lParte["IdEstado"] = 1;
        lParte["Facturable"] = 1;
        lParte["ManoObraDetall"] = 1;
        lParte["DesplDetall"] = 1;
        lParte["Descuento"] = lDescuento;
        lParte["Iva"] = lIVA;
        lParte["Observaciones"] = lNULL;
        lParte["IdEmpleado"] = lIdEmpleado;
        lParte["FormaPago"] = lFormaPago;
        lParte["IdEstadoReparacion"] = 0;
        lParte["IdCentroCoste"] = lIdCentroCoste;
        lParte["IdProyecto"] = lIdProyecto;
        lParte["IdMoneda"] = lIdMoneda;
        lParte["Cantidad"] = 1;
        lParte["Presupuesto"] = 0;
        lParte["ContadorDesde"] = 0;
        lParte["ContadorHasta"] = 0;
        lParte["IdLista"] = lIdLista;
        lParte["Bloqueado"] = 0;
        lParte.Update();

        fiNuevoParte = lParte["IdDoc"];
    }
    catch (Exception)
    {
        throw;
    }
    return fiNuevoParte;
}

```

```

public Valor fbRelacionarPedidoYParteNuevo(Valor viIdPedido, Valor
viIdLinea, Valor viIdParte)
{
    Valor fbRelacionarPedidoYParteNuevo = ValorEmpty.Value;
    IVB6Collection<Valor> lCol = null;
    Valor lResult = ValorEmpty.Value;
    try
    {
        fbRelacionarPedidoYParteNuevo = null;
        lResult = true;

        lCol = gCn.DameNewCollection();
        lCol.Add(viIdPedido);
        lCol.Add(viIdLinea);
        lCol.Add(viIdParte);

        if (!
gCn.EjecutaStoreCol("pPers_GenerarGrupoParte_DeLineaPedido", lCol))
        {
            lResult = false;
            gCn.Obj.ShowMsgBox("Error vinculando el parte con la línea
de pedido", vbExclamation, "Atención", gForm);
        }

        fbRelacionarPedidoYParteNuevo = lResult;
    }
    catch (Exception)
    {
        throw;
    }
}

```



```

    }
    catch (Exception)
    {
        throw;
    }
    return fbRelacionarPedidoYParteNuevo;
}

```

```

public void pBorrarNuevoParte(string viIdDoc)
{
    Valor lSQL = ValorEmpty.Value;
    try
    {
        lSQL = "DELETE FROM Partes WHERE IdDoc = " + viIdDoc;
        if (!gCn.ExecuteSql((string)lSQL))
        {
            gCn.Obj.ShowMsgBox("Error deshaciendo el parte nuevo. Debe borrarlo manualmente desde la colección de partes", vbExclamation, "Atención", gForm);
        }
    }
    catch (Exception)
    {
        throw;
    }
}

```

```

public void pMostrarNuevoParte(Valor viIdDoc)
{
    IParte lObj = null;
    IColecciones lCol = null;
    try
    {
        lCol = gCn.Obj.DameColeccion("Partes", "WHERE IdDoc = " + (string)viIdDoc);
        if (lCol is null)
        {
            gCn.Obj.ShowMsgBox("No se encuentra el parte " + viIdDoc, (VbMsgBoxStyle)(int)vbYes, "Error", gForm);
            lCol = null;
            return;
        }
        lObj = (IParte)gCn.Obj.DameObjeto("Partes", "WHERE IdDoc = " + (string)viIdDoc);
        if (gCn.Obj.ShowMsgBox("Parte generado correctamente. Número: " + (string)lObj["NumParte"] + vbCrLf + "¿Desea ver el parte generado?", vbYesNo | vbQuestion | vbDefaultButton1, "Parte generado correctamente", gForm) == vbYes)
        {
            lCol.Show(gForm, true, false);
        }
        lCol = null;
    }
    catch (Exception)
    {
        throw;
    }
}

```

```

public void pMostrarPartesLinea(ICntGridUsuario aGrid)
{
    IColecciones lCol = null;

```

```

        Valor lWhere = ValorEmpty.Value;
        Valor lIdLinea = ValorEmpty.Value;
        Valor lIdPedido = ValorEmpty.Value;
        try
        {

            lIdPedido = aGrid.GetValue("IdPedido");
            lIdLinea = aGrid.GetValue("IdLinea");

            //lWHERE = "WHERE IdParte IN (SELECT IdParte FROM
Ofertas_Partес_Groups_Details WHERE IdPedido = " & lIdPedido & " AND IdLinea =
" & lIdLinea & ")"
            lWhere = "WHERE IdParte IN (SELECT IdParte FROM
Pedidos_Cli_Lineas WHERE IdPedido = " + lIdPedido + " AND IdLinea = " + lIdLinea
+ ")";

            lCol = gCn.Obj.DameColeccion("Partes", (string)lWhere);
            if (lCol is null)
            {
                gCn.Obj.ShowMsgBox("La línea no tiene partes asociados",
vbInformation, "Partes", gForm);
                lCol = null;
                return;
            }

            lCol.Show(gForm, true, false);

            lCol = null;
        }
        catch (Exception)
        {
            throw;
        }
    }

    public void pAjustarCamposWidthEconomicos()
    {
        ICntPanel lPnlP = null;
        try
        {
            lPnlP = (ICntPanel)gForm.Controls["FrameEconomicos"];
            gForm.Controls["IdListaRevision"].Width =
gForm.Controls["IdListaRevision"].Width + 23;
            gForm.Controls["ProntoPago"].Width = 143;
            gForm.Controls["Descuento"].Width = 143;
            gForm.Controls["Comision"].Width = 143;

        }
        catch (Exception)
        {
            throw;
        }
    }

}

}
</CodigoNet>
</row>

<row>
    <CodigoNet>
using AhoraAsistentes;
using AhoraCore;

```

```

using AhoraOCX;
using AhoraSistema;
using static AhoraCore.VBA.Conversion;
using static AhoraCore.VBA.Interaction;
using static AhoraCore.VBA.Strings;
using static AhoraCore.VbMsgBoxResult;
using static AhoraCore.VbMsgBoxStyle;
using System;

namespace AhoraScriptsPantalla
{
    /** FORMULARIO ALBARANES**/

    //Para la botonera de textos cabecera y pie

    //-----
    // CDS-SOFT
    // Fecha:      20250128
    // Versión:    1.1
    // Autor:      Fernando
    // Descripción: 1.1 - Ajustes para asignar referencia cliente a pedido
origen de línea
    //          1.2 - Ajuste para consultar descripción de la línea
    //-----
    -----
    public class AhoraCl_frmAlbaranes : AhoraOCX.AhoraBaseScript
    {
        const int ANCHO_CAPTION = 5; //67;
        const int ANCHO_FECHAS = 153; //2300;
        const int ALTO_BUSCADORES = 20; //300;
        const int DESPLAZA_V = 3; //50;
        const int DESPLAZA_H = 3; //50;
        const int ANCHO_BOTONERTEXTOSIMP = 227; // 3400;
        const int ALTO_BOTONERATEXTOSIMP = 37; //550;

        public void pAnadirControlDebajo(Valor vsControl, Valor vsControlRef,
Valor viAncho, Valor viAlto, Valor viDesplazaV, Valor viDesplazaH)
        {
            Valor lTop = ValorEmpty.Value;
            Valor lIzq = ValorEmpty.Value;
            try
            {
                lIzq = 0;
                if (!System.String.Equals("" + (string)viDesplazaH, "",
StringComparison.Ordinal))
                {
                    lIzq = (int)viDesplazaH;
                }
                lTop = viDesplazaV;
                if ("" + vsControl == "")
                {
                    return;
                }
                if ("" + vsControlRef != "")
                {
                    lIzq = gForm.Controls[(string)vsControlRef].Left + lIzq;
                    lTop = gForm.Controls[(string)vsControlRef].Top +
gForm.Controls[(string)vsControlRef].Height + viDesplazaV;
                }
                gForm.Controls[(string)vsControl].SetBounds(lIzq, lTop, viAncho,
viAlto);
            }
            catch (Exception)

```

```

        {
            throw;
        }
    }
}

```

```

        public void pAnadirControlDerecha(Valor vsControl, Valor vsControlRef,
Valor viAncho, Valor viAlto, Valor viDesplazaH, Valor viDesplazaV)
        {
            Valor lTop = ValorEmpty.Value;
            Valor lIzq = ValorEmpty.Value;
            try
            {
                lIzq = viDesplazaH;
                lTop = 0;
                if (!System.String.Equals("" + (string)viDesplazaV, "",
StringComparison.Ordinal))
                {
                    lTop = (int)viDesplazaV;
                }
                if ("" + vsControl == "")
                {
                    return;
                }
                if ("" + vsControlRef != "")
                {
                    lIzq = gForm.Controls[(string)vsControlRef].Left +
gForm.Controls[(string)vsControlRef].Width + viDesplazaH;
                    lTop = gForm.Controls[(string)vsControlRef].Top + lTop;
                }
                gForm.Controls[(string)vsControl].SetBounds(lIzq, lTop, viAncho,
viAlto);
            }
            catch (Exception)
            {
                throw;
            }
        }
    }
}

```

```

//''-----
//'' CDS-SOFT
//''-----

```

```

public override void Initialize()
{
    ICntGridUsuario lGrid = null;
    try
    {
        //Redimensionamos el formulario
        gForm.Width = gForm.Width + 200;
        gForm.Left = gForm.Left - 100;

        lGrid = (ICntGridUsuario)gForm.Controls["grdLineas"];
        lGrid.ActivarScripts = true;
        lGrid = null;
        pConfigurarPanelImpresos();
        pConfigurarCamposValidacion();
        pOcultarPestanaConf();
    }
    catch (Exception)
    {
        throw;
    }
}

```

```

    }

    public override void Show()
    {
        try
        {
            pDesplazarCamposValidacionIzq();
            //Al mostrar forzamos para que se redimensione
            gCn.Obj.Form_AutoResize(gForm);

            //V1.1
            pAnhadirMenuAsignarPedidoCliente();

        }
        catch (Exception)
        {
            throw;
        }
    }

    public override void Grid_DblClick(ICntGridUsuario aGrid)
    {
        try
        {
            if (System.String.Equals(aGrid.Name, "grdLineas",
StringComparison.Ordinal))
            {
                if (System.String.Equals(aGrid.CampoActual, "Descrip2",
StringComparison.Ordinal) | System.String.Equals(aGrid.CampoActual,
"pers_DescripExtendida", StringComparison.Ordinal))
                {
                    pAsignarDescAmpliada(aGrid);
                }
                //V1.2
                if (System.String.Equals(aGrid.CampoActual, "Descrip",
StringComparison.Ordinal))
                {
                    pMostrarDescripcion(aGrid);
                }
            }
        }
        catch (Exception)
        {
            throw;
        }
    }

    //''V1.1
    public override void Grid_MenuAfterExecute(ICntGridUsuario aGrid,
IAhoraTool aMenuItem)
    {
        Valor respuesta = ValorEmpty.Value;
        Valor lMensaje = ValorEmpty.Value;
        Valor lValorPedCli = ValorEmpty.Value;
        Valor lSQLValorPedCli = ValorEmpty.Value;
        Valor lValorPedido = ValorEmpty.Value;
        Valor lSQLPedido = ValorEmpty.Value;
        Valor lNuevoValorPedCli = ValorEmpty.Value;
        try
        {

```

```

        if (System.String.Equals(aGrid.Name, "grdLineas",
StringComparison.Ordinal))
        {
            if (System.String.Equals(aMenuItem.Name,
"pers_mnuAsignarPedCli", StringComparison.Ordinal))
            {
                lNuevoValorPedCli = InputBox("Indique el nuevo valor del
Pedido Cliente:", "Pedido Cliente");

                //'Obtener composicion de numpedido y ejercicio para
aplicar.
                lSQLPedido = "SELECT CAST(AñoNum AS VARCHAR) + '/' +
CAST(NumPedido AS VARCHAR) FROM Pedidos_Cli_Cabecera WHERE IdPedido = " +
aGrid.GetValue("IdPedido");
                lValorPedido = gCn.DameValorCampo((string)lSQLPedido,
"");
                lSQLValorPedCli = "SELECT IdPedidoCli FROM
Pedidos_Cli_Cabecera WHERE IdPedido = " + aGrid.GetValue("IdPedido");
                lValorPedCli =
gCn.DameValorCampo((string)lSQLValorPedCli, "");

                if ((lSQLPedido == "") | (lSQLValorPedCli == "") |
(aGrid.GetValue("IdPedido") == "") | (lNuevoValorPedCli == ""))
                {
                    return;
                }

                if (!System.String.Equals(Trim(lValorPedCli), "",
StringComparison.Ordinal))
                {
                    lMensaje = "¿Asignar nuevo valor de Pedido Cliente
al pedido vinculado a la línea (Pedido: " + (string)lValorPedido + " -
actualmente con valor PedidoCli: " + lValorPedCli + ") ?";
                }
                else
                {
                    lMensaje = "¿Asignar nuevo valor de Pedido Cliente
al pedido vinculado a la línea (Pedido: " + (string)lValorPedido + ") ?";
                }

                respuesta = (int)MsgBox(lMensaje, vbYesNo | vbQuestion,
(string)lValorPedido);

                if (respuesta == (int)vbYes)
                {
                    gCn.Obj.HourglassAll(true);

                    if (!gCn.ExecuteSql("UPDATE Pedidos_Cli_Cabecera SET
IdPedidoCli = '" + lNuevoValorPedCli + "' WHERE IdPedido = " +
aGrid.GetValue("IdPedido")))
                    {
                        gCn.Obj.ShowMsgBox("No se ha podido actualizar
el valor en el pedido de origen.", vbCritical, "Actualizando Pedido", gForm);
                    }
                    else
                    {
                        gCn.Obj.ShowMsgBox("Referencia de Pedido Cliente
corregida correctamente.", vbOKOnly, "Actualizando Pedido", gForm);
                    }

                    gCn.Obj.HourglassAll(false);
                }
            }
        }

```

```

        }
    }
}
catch (Exception)
{
    throw;
}
}

public void pAsignarDescAmpliada(ICntGridUsuario aGrid)
{
    Valor lTexto = ValorEmpty.Value;
    IFrmZoom lzFrm = null;
    try
    {
        // lDescAmp = gform.Controls("grdLineas").Getvalue("Descrip2")
        // Set lFrmZ = gCn.AhoraProceso("DameZoom", False, gCn)
        // lFrmZ.Ver2 CStr(lDescAmp), 2000, False, gform
        // If lFrmZ.Texto <> "" Then
        //     agrid.SetValue "Descrip2", lFrmZ.Texto
        // End If

        lzFrm = (IFrmZoom)gCn.AhoraProceso("DameZoom", out _, gCn);
        ((ITextoUsuario)lzFrm.FindControl("txtZoom")).ReadOnly =
aGrid.Columns[aGrid.Col].Locked;
        lTexto = lzFrm.Carga(aGrid.Columns[aGrid.Col].Caption,
aGrid.Columns[aGrid.Col].Text);
        if ((int)lzFrm.BotonPulsado != 2)
        {
            aGrid.SetValue(aGrid.Columns[aGrid.Col].DataField,
(string)lTexto);
        }
    }
    catch (Exception)
    {
        throw;
    }
}

//V1.2
public void pMostrarDescripcion(ICntGridUsuario aGrid)
{
    Valor lTexto = ValorEmpty.Value;
    IFrmZoom lzFrm = null;
    try
    {
        lzFrm = (IFrmZoom)gCn.AhoraProceso("DameZoom", out _, gCn);
        ((ITextoUsuario)lzFrm.FindControl("txtZoom")).ReadOnly =
aGrid.Columns[aGrid.Col].Locked;
        lTexto = lzFrm.Carga(aGrid.Columns[aGrid.Col].Caption,
aGrid.Columns[aGrid.Col].Text);

    }
    catch (Exception)
    {
        throw;
    }
}

```

```

public void pConfigurarPanelImpresos()
{
    int CAPTION_CAMPO_CALIDAD = 0;
    Valor ANCHO_CAMPO_CALIDAD = ValorEmpty.Value;
    Valor ALTO_CAMPO_CALIDAD = ValorEmpty.Value;
    ICntBotonera lBtnra = null;
    ICheckBoxUser lChk = null;
    ITextoUsuario lTxt = null;
    ICntPanel lPnlP = null;
    ICntTab lTab = null;
    try
    {
        lTab = (ICntTab)gForm.Controls["TabDatos"];

        lPnlP = (ICntPanel)gForm.Controls.Add("Threed.SSPanel",
"pers_pnlImpresos");
        lPnlP.Visible = true;
        lPnlP.ResizeInterior = true;
        lPnlP.Text = "";

        //RAFA: es necesario añadir un campo de tipo texto que permanece
oculto para evitar el redimensionamiento del check
        //    si sólo se crea el campo check, al ser SSPanel con
autosize, el check se muestra estirado de lado a lado del panel
        //    añadiendo el TextBox oculto, parece que toma la referencia
del éste y no se redimensiona
        lTxt =
(ITextoUsuario)gForm.Controls.Add("Ahora0cx.Textusuario", "pers_edFecha0culto",
lPnlP);

        lTxt.ActivarScripts = true;
        lTxt.Formato = EFormatoCampo.FechaAnyoCompleto;
        lTxt.TipoDato = ETipoDato.Date;
        //ObjOrigen = "EObjeto"
        //ObjPOrigen = "pers_FechaInicioCertificacion"
        lTxt.Visible = false;
        //Move lControl.Left + lControl.Width + 50, lControl.Top, 2200,
300

        lTxt.CaptionVisible = true;
        lTxt.CaptionLink = true;
        lTxt.CaptionControl = "Fecha Impreso";
        lTxt.CaptionWidth = ANCHO_CAPTION;
        lTxt.TabStop = true;
        lTxt.Enabled = true;
        //.BackColor = vbButtonFace
        //.ForeColor = vbBlue
        lTxt.TabIndex = 6;
        lTxt.Text = "";
        //lTxt.AplicaEstilo();
        pAnadirControlDebajo("pers_edFecha0culto", "", ANCHO_FECHAS,
ALTO_BUSCADORES, 0, DESPLAZA_H);

        lChk =
(ICheckBoxUser)gForm.Controls.Add("Ahora0cx.CheckBoxUser", "pers_chImpValorado",
lPnlP);

        lChk.ActivarScripts = true;
        lChk.ObjOrigen = "EObjeto";
        lChk.ObjPOrigen = "pers_ImpValorado";
        lChk.Visible = true;
        //Move lControl.Left + lControl.Width + 50, lControl.Top, 2200,
300

        lChk.CaptionVisible = true;

```



```

lChk.CaptionControl = "Imprimir valorado";
lChk.CaptionWidth = ANCHO_CAPTION;
lChk.TabStop = true;
lChk.Enabled = true;
//.BackColor = vbButtonFace
//.ForeColor = vbBlue
lChk.TabIndex = 8;
//.AplicaEstilo
//pAnadirControlDebajo("pers_chImpValorado",
"pers_edFechaOculto", 1800, ALTO_BUSCADORES, - ALTO_BUSCADORES, DESPLAZA_H);
pAnadirControlDebajo("pers_chImpValorado", "pers_edFechaOculto",
120, ALTO_BUSCADORES, -ALTO_BUSCADORES, DESPLAZA_H);

//'-----
-

//' CAMPOS PARA LOS TEXTOS DE CABECERA Y PIE (OCULTOS)
//'-----
-

//'Botonera para textos de cabecera y pie
lBtnra =
(ICntBotonera)gForm.Controls.Add("AhoraOCX.cntBotonera",
"pers_btnra_TextosCabYPieImpreso", lPnlP);
lBtnra.ActivarScripts = true;
lBtnra.SeguridadObjeto = (EMenu)0;
lBtnra.BotonesMantenimiento = (EBotonesMant)0;
lBtnra.Visible = true;
lBtnra.BotonAdd("Texto cabecera", "pers_botTxtCabecera", "",
ETipo_Menu.eTipo_Menu_Boton, false, 275);
lBtnra.BotonAdd("Texto pie", "pers_botTxtPie", "",
ETipo_Menu.eTipo_Menu_Boton, false, 331);
lBtnra.BotonesMantenimiento = (EBotonesMant)0;
lBtnra.BotAsociados = false;
lBtnra.HabilitaBotones();
lBtnra.Tag = "TextosCabYPieImpreso";
pAnadirControlDebajo("pers_btnra_TextosCabYPieImpreso",
"pers_chImpValorado", ANCHO_BOTONERTEXTOSIMP, ALTO_BOTONERATEXTOSIMP,
DESPLAZA_V, 0);

lTxt =
(ITextoUsuario)gForm.Controls.Add("AhoraOCX.TextoUsuario", "pers_edTxtCabecera",
lPnlP);
lTxt.ActivarScripts = true;
lTxt.Formato = EFormatoCampo.MayusculasyMinusculas;
lTxt.TipoDato = ETipoDato.String;
lTxt.ObjOrigen = "EObjeto";
lTxt.ObjPOrigen = "pers_TxtCabecera";
lTxt.Visible = false;
lTxt.CaptionVisible = true;
lTxt.CaptionControl = "Texto cabecera";
lTxt.CaptionWidth = 0;
lTxt.CaptionLink = false;
lTxt.CaptionPosition = (ECaptionPosition)1;
//lTxt.AplicaEstilo();
lTxt.TabStop = false;
lTxt.Enabled = false;
// pAnadirControlDebajo("pers_edTxtCabecera",
"pers_btnra_TextosCabYPieImpreso", 50, 1, 1, 1);
pAnadirControlDebajo("pers_edTxtCabecera",
"pers_btnra_TextosCabYPieImpreso", 3, 1, 1, 1);

lTxt =
(ITextoUsuario)gForm.Controls.Add("AhoraOCX.TextoUsuario", "pers_edTxtPie",
lPnlP);

```

```

lTxt.ActivarScripts = true;
lTxt.Formato = EFormatoCampo.MayusculasyMinusculas;
lTxt.TipoDato = ETipoDato.String;
lTxt.ObjOrigen = "EObjeto";
lTxt.ObjPOrigen = "pers_TxtPie";
lTxt.Visible = false;
lTxt.CaptionVisible = true;
lTxt.CaptionControl = "Texto pie";
lTxt.CaptionWidth = 0;
lTxt.CaptionLink = false;
lTxt.CaptionPosition = (ECaptionPosition)1;
//lTxt.AplicaEstilo();
lTxt.TabStop = false;
lTxt.Enabled = false;
//pAnadirControlDerecha("pers_edTxtPie", "pers_edTxtCabecera",
50, 1, 1, 1);
pAnadirControlDerecha("pers_edTxtPie", "pers_edTxtCabecera", 3,
1, 1, 1);

ALTO_CAMPO_CALIDAD = 20; //300;
ANCHO_CAMPO_CALIDAD = 267; //4000;
CAPTION_CAMPO_CALIDAD = 120; //120;
lTxt =
(ITextoUsuario)gForm.Controls.Add("AhoraOCX.TextoUsuario",
"pers_edIdControlCalidad", lPnlP);
lTxt.ActivarScripts = true;
lTxt.Formato = EFormatoCampo.MayusculasyMinusculas;
lTxt.TipoDato = ETipoDato.String;
lTxt.ObjOrigen = "EObjeto";
lTxt.ObjPOrigen = "pers_IdControlCalidad";
lTxt.Visible = true;
//.Move LEFT, TOP, WIDTH, HEIGHT
lTxt.CaptionVisible = true;
lTxt.CaptionControl = "Identificador calidad";
lTxt.CaptionWidth = CAPTION_CAMPO_CALIDAD;
lTxt.CaptionLink = false;
lTxt.CaptionPosition = (ECaptionPosition)0;
//lTxt.AplicaEstilo();
lTxt.TabStop = true;
lTxt.Enabled = true;
pAnadirControlDebajo("pers_edIdControlCalidad",
"pers_btnra_TextosCabYPieImpreso", ANCHO_CAMPO_CALIDAD, ALTO_CAMPO_CALIDAD,
DESPLAZA_V, DESPLAZA_H);

lTab.InsertItem(1000, "Impresos", lPnlP, 0);

lTab = null;
lPnlP = null;
}
catch (Exception)
{
    throw;
}
}

public void pOcultarPestanaConf()
{
    try
    {
        ((ICntTab)gForm.Controls["TabDatos"]).Item(3).Visible = false;
    }
    catch (Exception)
    {

```

```

        throw;
    }
}

public override void CargaObjeto()
{
    try
    {
        pOcultarPestanaConf();
        pDesplazarCamposValidacionIzq();
    }
    catch (Exception)
    {
        throw;
    }
}

public override void ActualizaObjeto()
{
    try
    {
        pOcultarPestanaConf();
    }
    catch (Exception)
    {
        throw;
    }
}

public override void EliminaObjeto()
{
    try
    {
        pOcultarPestanaConf();
        pDesplazarCamposValidacionIzq();
    }
    catch (Exception)
    {
        throw;
    }
}

public void pConfigurarCamposValidacion()
{
    IComboUsuario lCombo = null;
    Valor lSql = ValorEmpty.Value;
    ICheckBoxUser lChk = null;
    ICntPanel lPnlP = null;
    try
    {
        lPnlP = (ICntPanel)gForm.Controls["FrameOtros"];

        lChk =
(ICheckBoxUser)gForm.Controls.Add("AhoraOcx.CheckBoxUser", "pers_chValidado",
lPnlP);

        lChk.ActivarScripts = true;
        lChk.ObjOrigen = "EObjeto";
        lChk.ObjPOrigen = "pers_Validado";
        lChk.Visible = true;
        //.Move lControl.Left + lControl.Width + 50, lControl.Top, 2200,

```

300

```
        lChk.CaptionVisible = true;
        lChk.CaptionControl = "Validado";
        lChk.CaptionWidth =
((IControlEnlazable)gForm.Controls["Deposito"]).CaptionWidth;
        lChk.TabStop = false;
        lChk.Enabled = false;
        //.BackColor = vbButtonFace
        //.ForeColor = vbBlue
        lChk.TabIndex = 8;
        //.AplicaEstilo
        pAnadirControlDerecha("pers_chValidado", "Deposito",
gForm.Controls["Deposito"].Width, gForm.Controls["Moneda"].Height, DESPLAZA_H,
0);

        lSql = "Select Nombre, IdEmpleado from VCombo_Empleados";

        lCombo =
(IComboUsuario)gForm.Controls.Add("AhoraOCX.ComboUsuario",
"pers_cbIdEmpleadoVal", lPnlP);
        lCombo.ActivarScripts = true;
        lCombo.C1Nombre = "Nombre";
        lCombo.C1TipoDato = (ETipoDatoColumnaCombo)8;
        lCombo.C1Anchura = 200;
        lCombo.C2Nombre = "IdEmpleado";
        lCombo.C2TipoDato = (ETipoDatoColumnaCombo)3;
        lCombo.C2Anchura = 53;
        lCombo.CActiva = 2;
        lCombo.NColumnas = 2;
        lCombo.Descripcion = (string)lSql;
        lCombo.Formato = EFormatoCampo.MayusculasyMinusculas;
        lCombo.TipoDato = ETipoDato.String;
        lCombo.ObjOrigen = "EObjeto";
        lCombo.ObjPOrigen = "pers_IdEmpleadoVal";
        lCombo.Visible = true;
        lCombo.CaptionVisible = false;
        lCombo.CaptionControl = "Empleado";
        lCombo.CaptionWidth = 80;
        lCombo.Enabled = false;
        lCombo.CaptionLink = false;
        //.move 0,0,1800,300
        lCombo.TabStop = false;
        lCombo.TabIndex = 0;
        //.lCombo.AplicaEstilo();
        lCombo.Refresh();

        //.pAnadirControlDerecha("pers_cbIdEmpleadoVal",
"pers_chValidado", 3000, gForm.Controls["Moneda"].Height, DESPLAZA_H, 0);
        pAnadirControlDerecha("pers_cbIdEmpleadoVal", "pers_chValidado",
200, gForm.Controls["Moneda"].Height, DESPLAZA_H, 0);
    }
    catch (Exception)
    {
        throw;
    }
}

public void pDesplazarCamposValidacionIzq()
{
    try
    {
        pAnadirControlDerecha("pers_chValidado", "Deposito",
gForm.Controls["Deposito"].Width, gForm.Controls["Moneda"].Height, DESPLAZA_H,
```

```

0);
        //pAnadirControlDerecha("pers_cbIdEmpleadoVal",
"pers_chValidado", 3000, gForm.Controls["Moneda"].Height, DESPLAZA_H, 0);
        pAnadirControlDerecha("pers_cbIdEmpleadoVal", "pers_chValidado",
200, gForm.Controls["Moneda"].Height, DESPLAZA_H, 0);

        if (gForm.Controls["Deposito"].Visible)
        {
            return;
        }

        pAnadirControlDebajo("pers_chValidado", "IdMoneda",
gForm.Controls["Deposito"].Width, gForm.Controls["Moneda"].Height, DESPLAZA_H,
0);
        // pAnadirControlDerecha("pers_cbIdEmpleadoVal",
"pers_chValidado", 4000, gForm.Controls["Moneda"].Height, DESPLAZA_H, 0);
        pAnadirControlDerecha("pers_cbIdEmpleadoVal", "pers_chValidado",
267, gForm.Controls["Moneda"].Height, DESPLAZA_H, 0);
    }
    catch (Exception)
    {
        throw;
    }
}

public override void Botonera_AfterExecute(ICntBotonera aBotonera,
IAhoraTool aBoton)
{
    try
    {
        if (System.String.Equals(aBotonera.Name,
"pers_btnra_TextosCabYPieImpreso", StringComparison.Ordinal))
        {
            if (System.String.Equals(aBoton.Name, "pers_botTxtCabecera",
StringComparison.Ordinal))
            {
                pMostrarTextoCabeceraYPie(gForm.Controls["pers_edTxtCabecera"]);
            }
            if (System.String.Equals(aBoton.Name, "pers_botTxtPie",
StringComparison.Ordinal))
            {
                pMostrarTextoCabeceraYPie(gForm.Controls["pers_edTxtPie"]);
            }
        }
    }
    catch (Exception)
    {
        throw;
    }
}

public void pMostrarTextoCabeceraYPie(IControl aTextoUsuario)
{
    Valor lTexto = ValorEmpty.Value;
    IFrmZoom lzFrm = null;
    try
    {
        ///'' Set lObsZ = gCn.AhoraProceso("DameZoom", False, gCn)
        ///'' lObsZ.Caption = aTextoUsuario.CaptionControl

```

```

        //' ' lObsZ.Ver aTextoUsuario, False, Len(aTextoUsuario)

        lzFrm = (IFrmZoom)gCn.AhoraProceso("DameZoom", out _, gCn);
        ((ITextoUsuario)lzFrm.FindControl("txtZoom")).ReadOnly = false;
        lTexto =
lzFrm.Carga(((IControlEnlazable)aTextoUsuario).CaptionControl,
((IControlEnlazable)aTextoUsuario).Text);
        if ((int)lzFrm.BotonPulsado != 2)
        {
            ((IControlEnlazable)aTextoUsuario).Text = lTexto;
            var lObject =
((IEnlaceObjetos)gForm.Controls["EObjeto"]).ObjGlobal[(((IControlEnlazable)aTexto
Usuario).ObjPOrigen)];
            ((IEnlaceObjetos)gForm.Controls["EObjeto"]).Modificado =
true;
        }
    }
    catch (Exception)
    {
        throw;
    }
}

public void pAnhadirMenuAsignarPedidoCliente()
{
    ICntGridUsuario lGrid = null;
    try
    {
        //V1.1
        lGrid = (ICntGridUsuario)gForm.Controls["grdLineas"];
        lGrid.MenuItemAdd("Asignar Pedido Cliente",
"pers_mnuAsignarPedCli");
        lGrid.ActivarScripts = true;
    }
    catch (Exception)
    {
        throw;
    }
}

}

}
</CodigoNet>
</row>

```

```

<row>
    <CodigoNet>
using AhoraAsistentes;
using AhoraCl;
using AhoraCore;
using AhoraOCX;
using AhoraSistema;
using static AhoraCore.VBA.Conversion;
using System;

namespace AhoraScriptsPantalla
{
    /** FORMULARIO FACTURA **/

    public class AhoraCl_frmFacturas : AhoraOCX.AhoraBaseScript
    {

```

```

const int DESPLAZA_V = 3; //50;
const int ANCHO_BOTONERTEXTOSIMP = 227; //3400;
const int ALTO_BOTONERTEXTOSIMP = 37; //550;
const int ANCHO_CAPTION = 4; //67;
const int ALTO_BUSCADORES = 20; //300;
const int DESPLAZA_H = 3; //50;
public void pAnadirControlDebajo(Valor vsControl, Valor vsControlRef,
Valor viAncho, Valor viAlto, Valor viDesplazaV, Valor viDesplazaH)
{
    Valor lTop = ValorEmpty.Value;
    Valor lIzq = ValorEmpty.Value;
    try
    {
        lIzq = 0;
        if (!System.String.Equals("" + (string)viDesplazaH, "",
StringComparison.Ordinal))
        {
            lIzq = (int)viDesplazaH;
        }
        lTop = viDesplazaV;
        if ("" + vsControl == "")
        {
            return;
        }
        if ("" + vsControlRef != "")
        {
            lIzq = gForm.Controls[(string)vsControlRef].Left + lIzq;
            lTop = gForm.Controls[(string)vsControlRef].Top +
gForm.Controls[(string)vsControlRef].Height + viDesplazaV;
        }
        gForm.Controls[(string)vsControl].SetBounds(lIzq, lTop, viAncho,
viAlto);
    }
    catch (Exception)
    {
        throw;
    }
}

```

```

public void pAnadirControlDerecha(Valor vsControl, Valor vsControlRef,
Valor viAncho, Valor viAlto, Valor viDesplazaH, Valor viDesplazaV)
{
    Valor lTop = ValorEmpty.Value;
    Valor lIzq = ValorEmpty.Value;
    try
    {
        lIzq = viDesplazaH;
        lTop = 0;
        if (!System.String.Equals("" + (string)viDesplazaV, "",
StringComparison.Ordinal))
        {
            lTop = (int)viDesplazaV;
        }
        if ("" + vsControl == "")
        {
            return;
        }
        if ("" + vsControlRef != "")
        {
            lIzq = gForm.Controls[(string)vsControlRef].Left +
gForm.Controls[(string)vsControlRef].Width + viDesplazaH;
            lTop = gForm.Controls[(string)vsControlRef].Top + lTop;
        }
    }
}

```

```

        gForm.Controls[(string)vsControl].SetBounds(lIzq, lTop, viAncho,
viAlto);
    }
    catch (Exception)
    {
        throw;
    }
}

//'''-----
//''' CDS-SOFT
//'''-----

public override void Initialize()
{
    IControl lGrid = null;
    try
    {
        //Redimensionamos el formulario
        // gForm.Width = gForm.Width + 3000
        // gForm.Left = gForm.Left - 1500

        pConfigurarPanelImpresos();

        lGrid = gForm.Controls["grdLineas"];
        ((IControlScriptable)lGrid).ActivarScripts = true;
        lGrid = null;
    }
    catch (Exception)
    {
        throw;
    }
}

public override void Show()
{
    try
    {
        //Al mostrar forzamos para que se redimensione
        // gcn.obj.Form_AutoResize gForm
    }
    catch (Exception)
    {
        throw;
    }
}

public override void Grid_DblClick(ICntGridUsuario aGrid)
{
    try
    {
        if (System.String.Equals(aGrid.Name, "grdLineas",
StringComparison.Ordinal))
        {
            if (System.String.Equals(aGrid.CampoActual, "Descrip2",
StringComparison.Ordinal) | System.String.Equals(aGrid.CampoActual,
"pers_DescripExtendida", StringComparison.Ordinal))
            {
                pAsignarDescAmpliada(aGrid);
            }
        }
    }
}

```



```

        catch (Exception)
        {
            throw;
        }
    }

    public void pAsignarDescAmpliada(ICntGridUsuario aGrid)
    {
        Valor lTexto = ValorEmpty.Value;
        IFrmZoom lzFrm = null;
        try
        {
            // lDescAmp = gform.Controls("grdLineas").Getvalue("Descrip2")
            // Set lFrmZ = gCn.AhoraProceso("DameZoom", False, gCn)
            // lFrmZ.Ver2 CStr(lDescAmp), 2000, False, gform
            // If lFrmZ.Texto <> "" Then
            //     agrid.SetValue "Descrip2", lFrmZ.Texto
            // End If

            lzFrm = (IFrmZoom)gCn.AhoraProceso("DameZoom", out _, gCn);
            ((ITextoUsuario)lzFrm.FindControl("txtZoom")).ReadOnly =
aGrid.Columns[aGrid.Col].Locked;
            lTexto = lzFrm.Carga(aGrid.Columns[aGrid.Col].Caption,
aGrid.Columns[aGrid.Col].Text);
            if ((int)lzFrm.BotonPulsado != 2)
            {
                aGrid.SetValue(aGrid.Columns[aGrid.Col].DataField,
(string)lTexto);
            }
        }
        catch (Exception)
        {
            throw;
        }
    }

    public void pConfigurarPanelImpresos()
    {
        Valor _refpar_pAnadirControlDebajo_viAlto = ValorEmpty.Value;
        Valor _refpar_pAnadirControlDebajo_viAncho = ValorEmpty.Value;
        int CAPTION_CAMPO_CALIDAD = 0;
        Valor ANCHO_CAMPO_CALIDAD = ValorEmpty.Value;
        Valor ALTO_CAMPO_CALIDAD = ValorEmpty.Value;
        ICntBotonera lBtnra = null;
        ICheckBoxUser lChk = null;
        ITextoUsuario lTxt = null;
        ICntPanel lPnlP = null;
        ICntTab lTab = null;
        try
        {
            lTab = (ICntTab)gForm.Controls["TabDatos"];

            lPnlP = (ICntPanel)gForm.Controls.Add("Threed.SSPanel",
"pers_pnlImpresos");
            lPnlP.Visible = true;
            lPnlP.ResizeInterior = true;
            lPnlP.Text = "";

            //-----

```

```

// CAMPOS PARA LOS TEXTOS DE CABECERA Y PIE (OCULTOS)
//-----
//'Botonera para textos de cabecera y pie
lBtnra =
(ICntBotonera)gForm.Controls.Add("AhoraOCX.cntBotonera",
"pers_btnra_TextosCabYPieImpreso", lPnlP);
    lBtnra.ActivarScripts = true;
    lBtnra.SeguridadObjeto = (EMenu)0;
    lBtnra.BotonesMantenimiento = (EBotonesMant)0;
    lBtnra.BotAsociados = false;
    lBtnra.Visible = true;
    lBtnra.BotonAdd("Texto cabecera", "pers_botTxtCabecera", "",
ETipo_Menu.eTipo_Menu_Boton, false, 275);
    lBtnra.BotonAdd("Texto pie", "pers_botTxtPie", "",
ETipo_Menu.eTipo_Menu_Boton, false, 331);
    lBtnra.HabilitaBotones();
    lBtnra.Tag = "TextosCabYPieImpreso";
    pAnadirControlDebajo("pers_btnra_TextosCabYPieImpreso", "",
ANCHO_BOTONERTEXTOSIMP, ALTO_BOTONERTEXTOSIMP, DESPLAZA_V, 0);

    lTxt =
(ITextoUsuario)gForm.Controls.Add("AhoraOCX.TextoUsuario", "pers_edTxtCabecera",
lPnlP);
    lTxt.ActivarScripts = true;
    lTxt.Formato = EFormatoCampo.MayusculasyMinusculas;
    lTxt.TipoDato = ETipoDato.String;
    lTxt.ObjOrigen = "EObjeto";
    lTxt.ObjPOrigen = "pers_TxtCabecera";
    lTxt.Visible = false;
    lTxt.CaptionVisible = true;
    lTxt.CaptionControl = "Texto cabecera";
    lTxt.CaptionWidth = 0;
    lTxt.CaptionLink = false;
    lTxt.CaptionPosition = (ECaptionPosition)1;
    //lTxt.AplicaEstilo();
    lTxt.TabStop = false;
    lTxt.Enabled = false;
    pAnadirControlDebajo("pers_edTxtCabecera",
"pers_btnra_TextosCabYPieImpreso", 3, 1, 1, 1);

    lTxt =
(ITextoUsuario)gForm.Controls.Add("AhoraOCX.TextoUsuario", "pers_edTxtPie",
lPnlP);
    lTxt.ActivarScripts = true;
    lTxt.Formato = EFormatoCampo.MayusculasyMinusculas;
    lTxt.TipoDato = ETipoDato.String;
    lTxt.ObjOrigen = "EObjeto";
    lTxt.ObjPOrigen = "pers_TxtPie";
    lTxt.Visible = false;
    lTxt.CaptionVisible = true;
    lTxt.CaptionControl = "Texto pie";
    lTxt.CaptionWidth = 0;
    lTxt.CaptionLink = false;
    lTxt.CaptionPosition = (ECaptionPosition)1;
    //lTxt.AplicaEstilo();
    lTxt.TabStop = false;
    lTxt.Enabled = false;
    pAnadirControlDerecha("pers_edTxtPie", "pers_edTxtCabecera", 3,
1, 1, 1);

    ALTO_CAMPO_CALIDAD = 20;
    ANCHO_CAMPO_CALIDAD = 267;
    CAPTION_CAMPO_CALIDAD = 120;
    lTxt =

```

```

(ITextoUsuario)gForm.Controls.Add("AhoraOCX.TextoUsuario",
"pers_edIdControlCalidad", lPnlP);
    lTxt.ActivarScripts = true;
    lTxt.Formato = EFormatoCampo.MayusculasyMinusculas;
    lTxt.TipoDato = ETipoDato.String;
    lTxt.ObjOrigen = "EObjeto";
    lTxt.ObjPOrigen = "pers_IdControlCalidad";
    lTxt.Visible = true;
    //.Move LEFT, TOP, WIDTH, HEIGHT
    lTxt.CaptionVisible = true;
    lTxt.CaptionControl = "Identificador calidad";
    lTxt.CaptionWidth = CAPTION_CAMPO_CALIDAD;
    lTxt.CaptionLink = false;
    lTxt.CaptionPosition = (ECaptionPosition)0;
    //lTxt.AplicaEstilo();
    lTxt.TabStop = true;
    lTxt.Enabled = true;
    pAnadirControlDebajo("pers_edIdControlCalidad",
"pers_btnra_TextosCabYPieImpreso", ANCHO_CAMPO_CALIDAD, ALTO_CAMPO_CALIDAD,
DESPLAZA_V, 0);

```

```

        lTab.InsertItem(1000, "Impresos", lPnlP, 0);

```

```

        lTab = null;
        lPnlP = null;

```

```

    }
    catch (Exception)
    {
        throw;
    }
}

```

```

    public override void Botonera_AfterExecute(ICntBotonera aBotonera,
IAhoraTool aBoton)
    {
        try
        {
            if (System.String.Equals(aBotonera.Name,
"pers_btnra_TextosCabYPieImpreso", StringComparison.Ordinal))
            {
                if (System.String.Equals(aBoton.Name, "pers_botTxtCabecera",
StringComparison.Ordinal))
                {
                    pMostrarTextoCabeceraYPie(gForm.Controls["pers_edTxtCabecera"]);
                }
                if (System.String.Equals(aBoton.Name, "pers_botTxtPie",
StringComparison.Ordinal))
                {
                    pMostrarTextoCabeceraYPie(gForm.Controls["pers_edTxtPie"]);
                }
            }
        }
        catch (Exception)
        {
            throw;
        }
    }

```

```

    public void pMostrarTextoCabeceraYPie(IControl aTextoUsuario)
    {

```

```

        Valor lTexto = ValorEmpty.Value;
        IFrmZoom lzFrm = null;
        try
        {

            //' ' Set lObsZ = gCn.AhoraProceso("DameZoom", False, gCn)
            //' ' lObsZ.Caption = aTextoUsuario.CaptionControl
            //' ' lObsZ.Ver aTextoUsuario, False, Len(aTextoUsuario)

            lzFrm = (IFrmZoom)gCn.AhoraProceso("DameZoom", out _, gCn);
            ((ITextoUsuario)lzFrm.FindControl("txtZoom")).ReadOnly = false;
            lTexto =

            lzFrm.Carga(((IControlEnlazable)aTextoUsuario).CaptionControl,
            ((IControlEnlazable)aTextoUsuario).Text);
            if ((int)lzFrm.BotonPulsado != 2)
            {
                ((IControlEnlazable)aTextoUsuario).Text = lTexto;
                var lObject =
                ((IEnlaceObjetos)gForm.Controls["EObjeto"]).ObjGlobal[(((IControlEnlazable)aTexto
                Usuario).ObjPOrigen)];
                ((IEnlaceObjetos)gForm.Controls["EObjeto"]).Modificado =
                true;
            }
        }
        catch (Exception)
        {
            throw;
        }
    }
}

```

</CodigoNet>

</row>