

```
1 using System;
2 using System.Linq;
3 using TecserEF.Entity;
4
5 namespace Tecser.Business.Transactional.CO.AsientoContable
6 {
7     /// <summary>
8     /// 2018.03.11 Funciones especificas para asientos de Vendors/Proveedores
9     /// Se inicia siempre creando el header
10    /// </summary>
11    public class AsientoVendorSpecific : AsientoBase
12    {
13        public AsientoVendorSpecific(string tcode) : base(tcode)
14        {
15        }
16
17        public IdentificacionAsiento AsientoFromVendorDocument(int idFacturaT0403)
18        {
19            var resultado = new AsientoBase.IdentificacionAsiento
20            {
21                IdDocu = 0,
22                NasX1 = 0,
23                NasX2 = 0
24            };
25
26            using (var db = new TecserData())
27            {
28                var data = db.T0403_VENDOR_FACT_H.SingleOrDefault(c => c.IDINT == idFacturaT0403);
29                if (data == null)
30                    return resultado; //(0,0,0)
31
32                var descripcionHeader = string.IsNullOrEmpty(data.OBSERVACION)
33                    ? "Contabilizacion de Gastos"
34                    : data.OBSERVACION;
35
36                CreacionHeaderAsiento(data.TIPO, data.FECHA.Value, data.TFACTURA, data.NFACTURA, descripcionHeader, data.MON, data.NETO.Value, data.TC.Value);
37
38                if (AddSegmentoGastoFromVendorDocument(idFacturaT0403) == 0)
39                    return resultado; //(0,0,0)
40
41                if (data.DTO != 0)
42                {
43                    AddSegment("DESC", Header.REFE, Header.TIPO,
44                        GLAccountManagement.GetGLAccount
45                        (GLAccountManagement.GLAccount.DescuentoCompras),
46                        GLAccountManagement.GetGLDescriptionFromT135(
47                            GLAccountManagement.GetGLAccount
48                            (GLAccountManagement.GLAccount.DescuentoCompras)), "",
```

```
48         data.MON, DebeHaber.Debe, data.DTO.Value, "CONTAR", 0,
49         data.IDPROV, "T0403_VENDOR_FACT_H",
50         idFacturaT0403);
51     }
52     if (data.IVA10 > 0)
53     AddSegment("IMP", Header.REFE, Header.TIPO,
54         GLAccountManagement.GetGLAccount
55         (GLAccountManagement.GLAccount.IvaCompra10),
56         GLAccountManagement.GetGLDescriptionFromT135(
57         GLAccountManagement.GetGLAccount
58         (GLAccountManagement.GLAccount.IvaCompra10)), "",
59         data.MON, DebeHaber.Debe, data.IVA10.Value, "CONTAR", 0,
60         data.IDPROV, "T0403_VENDOR_FACT_H",
61         idFacturaT0403);
62
63     if (data.IVA21 > 0)
64     AddSegment("IMP", Header.REFE, Header.TIPO,
65         GLAccountManagement.GetGLAccount
66         (GLAccountManagement.GLAccount.IvaCompra21),
67         GLAccountManagement.GetGLDescriptionFromT135(
68         GLAccountManagement.GetGLAccount
69         (GLAccountManagement.GLAccount.IvaCompra21)), "",
70         data.MON, DebeHaber.Debe, data.IVA21.Value, "CONTAR", 0,
71         data.IDPROV, "T0403_VENDOR_FACT_H",
72         idFacturaT0403);
73
74     if (data.IVA27 > 0)
75     {
76     AddSegment("IMP", Header.REFE, Header.TIPO,
77         GLAccountManagement.GetGLAccount
78         (GLAccountManagement.GLAccount.IvaCompra27),
79         GLAccountManagement.GetGLDescriptionFromT135(
80         GLAccountManagement.GetGLAccount
81         (GLAccountManagement.GLAccount.IvaCompra27)), "",
82         data.MON, DebeHaber.Debe, data.IVA27.Value, "CONTAR", 0,
83         data.IDPROV, "T0403_VENDOR_FACT_H",
84         idFacturaT0403);
85     }
86
87     if (data.PerIVA > 0)
88     {
89     AddSegment("IMP", Header.REFE, Header.TIPO,
90         GLAccountManagement.GetGLAccount
91         (GLAccountManagement.GLAccount.PercepcionIVA),
92         GLAccountManagement.GetGLDescriptionFromT135(
93         GLAccountManagement.GetGLAccount
94         (GLAccountManagement.GLAccount.PercepcionIVA)), "",
95         data.MON, DebeHaber.Debe, data.PerIVA.Value, "CONTAR", 0,
96         data.IDPROV, "T0403_VENDOR_FACT_H",
97         idFacturaT0403);
98     }
```

```
87
88     decimal alicuota = 0;
89     if (data.PerIIBB > 0)
90     {
91         alicuota = data.PerIIBB ?? 0;
92
93         AddSegmentoImpuesto
94             (GLAccountManagement.GLAccount.PercepcionIIBB, data.IDPROV,
95              data.PerIIBB.Value,
96              data.MON, alicuota, data.PerIIBB_TXT,
97              "T0403_VENDOR_FACT_H",
98              idFacturaT0403);
99     }
100
101     if (data.ImpuestoMunicipal > 0)
102     {
103         alicuota = 0;
104
105         AddSegment("IMP", Header.REFE, Header.TIPO,
106                    GLAccountManagement.GetGLAccount
107                    (GLAccountManagement.GLAccount.CompraImpuestoMunicipal),
108                    GLAccountManagement.GetGLDescriptionFromT135(
109                    GLAccountManagement.GetGLAccount
110                    (GLAccountManagement.GLAccount.CompraImpuestoMunicipal))), "",
111                    data.MON, DebeHaber.Debe, data.ImpuestoMunicipal.Value,
112                    "CONTAR", 0, data.IDPROV,
113                    "T0403_VENDOR_FACT_H",
114                    idFacturaT0403);
115     }
116
117     if (data.ImpuestoProvincial > 0)
118     {
119         AddSegment("IMP", Header.REFE, Header.TIPO,
120                    GLAccountManagement.GetGLAccount
121                    (GLAccountManagement.GLAccount.CompraImpuestoProvincial),
122                    GLAccountManagement.GetGLDescriptionFromT135(
123                    GLAccountManagement.GetGLAccount
124                    (GLAccountManagement.GLAccount.CompraImpuestoProvincial))),
125                    "",
126                    data.MON, DebeHaber.Debe, data.ImpuestoProvincial.Value,
127                    "CONTAR", 0, data.IDPROV,
128                    "T0403_VENDOR_FACT_H",
129                    idFacturaT0403);
130     }
131
132     if (data.IMPINT > 0)
133     {
134         AddSegment("IMP", Header.REFE, Header.TIPO,
135                    GLAccountManagement.GetGLAccount
136                    (GLAccountManagement.GLAccount.CompraImpuestoInterno),
137                    GLAccountManagement.GetGLDescriptionFromT135(
138                    GLAccountManagement.GetGLAccount
```

```

129         (GLAccountManagement.GLAccount.CompraImpuestoInterno)), "",
130         data.MON, DebeHaber.Debe, data.IMPINT.Value, "CONTAR", 0,
131         data.IDPROV, "T0403_VENDOR_FACT_H",
132         idFacturaT0403);
133     }
134     if (data.IMPOTR > 0)
135     {
136         AddSegment("IMP", Header.REFE, Header.TIPO,
137             GLAccountManagement.GetGLAccount
138             (GLAccountManagement.GLAccount.CompraImpuestoOtros),
139             GLAccountManagement.GetGLDescriptionFromT135(
140             GLAccountManagement.GetGLAccount
141             (GLAccountManagement.GLAccount.CompraImpuestoOtros)), "",
142             data.MON, DebeHaber.Debe, data.IMPOTR.Value, "CONTAR", 0,
143             data.IDPROV, "T0403_VENDOR_FACT_H",
144             idFacturaT0403);
145     }
146     if (data.ConceptosNoGravados > 0)
147     {
148         AddSegment("IMP", Header.REFE, Header.TIPO,
149             GLAccountManagement.GetGLAccount
150             (GLAccountManagement.GLAccount.CompraImpuestoNoGravado),
151             GLAccountManagement.GetGLDescriptionFromT135(
152             GLAccountManagement.GetGLAccount
153             (GLAccountManagement.GLAccount.CompraImpuestoNoGravado)), "",
154             data.MON, DebeHaber.Debe, data.ConceptosNoGravados.Value,
155             "CONTAR", 0, data.IDPROV,
156             "T0403_VENDOR_FACT_H",
157             idFacturaT0403);
158     }
159     }
160     //Asiento AP
161     AddSegmentoAp(data.IDPROV, data.NETO.Value, data.MON,
162         "T0403_VENDOR_FACT_H", idFacturaT0403);
163     return base.GraabaAsiento();
164 }
165 }
166
167 /// <summary>
168 /// Agrega segmentos de Gastos desde T0403->T0404
169 /// Retorna numero de items gastos agregados que debe ser >0
170 /// </summary>
171 private int AddSegmentoGastoFromVendorDocument(int idFacturaT0403)
172 {
173     using (var db0 = new TecserData())
174     {
175         var it = db0.T0404_VENDOR_FACT_I.Where(c => c.IDINT ==
176             idFacturaT0403).ToList();
177         if (it.Count == 0)

```

```
171         return 0;
172
173         var h = db0.T0403_VENDOR_FACT_H.SingleOrDefault(c => c.IDINT == idFacturaT0403);
174
175         foreach (var i in it)
176         {
177             AddSegment("G", Header.REFE, Header.TIPO, i.GL, i.ITEM_DESC,
178                 null, Header.MONE_ORI, DebeHaber.Debe,
179                 i.PUNIT.Value*i.CANT.Value, base.Tcode, 0, h.IDPROV,
180                 "T0403_VENDOR_FACT_H", idFacturaT0403,
181                 kgMaterial: i.CANT);
182         }
183         return base.Items.Count;
184     }
185
186     private void AddSegmentoImpuesto(GLAccountManagement.GLAccount
187         tipoImpuesto, int idVendor,
188         decimal importeImpuesto, string moneda = "ARS", decimal? alicuota =
189         0, string distrito = null,
190         string nombreTablaReferencia = "T0403_VENDOR_FACT_H", int
191         idTablaReferencia = 0)
192     {
193         var glAccount = GLAccountManagement.GetGLAccount(tipoImpuesto);
194         var glAccountDescription =
195             GLAccountManagement.GetGLDescriptionFromT135(glAccount);
196
197         string descripcion1;
198         string descripcion2 = null;
199
200         if (alicuota > 0)
201         {
202             descripcion1 = glAccountDescription + "Alic: " + alicuota;
203         }
204         else
205         {
206             descripcion1 = glAccountDescription;
207         }
208
209         if (string.IsNullOrEmpty(distrito) == false)
210         {
211             descripcion2 = "(Alic: " + alicuota + ")" + distrito;
212         }
213
214         AddSegment("IMP", Header.REFE, Header.TIPO, glAccount, descripcion1,
215             descripcion2, moneda, DebeHaber.Debe,
216             importeImpuesto, "CONTAR", 0, idVendor, idNumerico:
217             idTablaReferencia);
218     }
219
220     private void AddSegmentoAp(int idVendor, decimal importeNetoTotal, string
221         moneda = "ARS",
```

```

...Transactional\CO\AsientoContable\AsientoVendorSpecific.cs 6
213     string nombreTablaReferencia = "T0403_VENDOR_FACT_H", int ➤
        idTablaReferencia = 0)
214 {
215     var glap = new GLAccountManagement().GetApVendorGl(idVendor);
216     AddSegment("AP", Header.REFE, Header.TIPO, glap,
217         "A Proveedores " + GLAccountManagement.GetGLDescriptionFromT135 ➤
            (glap)
218         , "", moneda, DebeHaber.Haber, importeNetoTotal, "CONTAR", 0, ➤
            idVendor, nombreTablaReferencia,
219         idTablaReferencia);
220 }
221
222 public IdentificacionAsiento GeneraAsientoPagoAutomaticoContar(int ➤
    asientoGasto, int idFactura403,
223     string cuentaPago)
224 {
225     var resultado = new IdentificacionAsiento()
226     {
227         IdDocu = 0,
228         NasX1 = 0,
229         NasX2 = 0,
230     };
231
232     using (var db = new TecserData())
233     {
234         var asGasto = db.T0602_DOCU_S.SingleOrDefault(c => c.IDDOCU == ➤
            asientoGasto && c.DOCUTIPO == "AP");
235         if (asGasto == null)
236             return resultado;
237
238         var asAuxHeader = db.T0601_DOCU_H.SingleOrDefault(c => c.IDDOCU ➤
            == asientoGasto);
239
240         var factu = db.T0403_VENDOR_FACT_H.SingleOrDefault(c => c.IDINT ➤
            == idFactura403);
241         if (factu == null)
242             return resultado;
243
244
245         var glAp = asGasto.GL;
246         var importeACancelar = asGasto.HABER;
247
248         if (factu.SALDOIMPAGO != importeACancelar)
249             return resultado;
250
251         var datosCuentaPago = db.T0150_CUENTAS.SingleOrDefault(c => ➤
            c.ID_CUENTA == cuentaPago);
252
253
254         CreacionHeaderAsiento(asGasto.TIPO, asAuxHeader.FECHA.Value, ➤
            "PD", factu.NFACTURA,
255         "Pago Directo " + cuentaPago, asAuxHeader.MONE_ORI, ➤

```

```

        asGasto.HABER.Value, asGasto.TC.Value);
256
257     base.AddSegment("PD", factu.NFACTURA, asGasto.TIPO,
        datosCuentaPago.GLMAP,
258     "Pago desde " + datosCuentaPago.CUENTA_DESC, null,
        asGasto.MONE_ORI, DebeHaber.Haber,
259     importeACancelar.Value, "CONTAR", idProveedor: factu.IDPROV);
260
261     base.AddSegment("PD", factu.NFACTURA, asGasto.TIPO, glAp,
262     "A/P - Pago", null, asGasto.MONE_ORI, DebeHaber.Debe,
263     importeACancelar.Value, "CONTAR", idProveedor: factu.IDPROV);
264
265     return base.GrabaAsiento();
266     return resultado;
267 }
268
269 //
    '-----
    -----
270 //'Funcion que hace el PAGO Rapido * Contabiliza Pago directamente
271 //'-----
    -----
272 //Public Sub ASIENTO_PAGO(NAS As Double)
273 //Dim db As DAO.Database
274 //Dim rs As DAO.Recordset
275 //Dim ra As DAO.Recordset
276 //Dim rx As DAO.Recordset
277 //Dim I As Integer
278
279 //Set db = CurrentDb
280 //Set ra = db.OpenRecordset("SELECT * FROM T0601_DOCU_H WHERE IDDOCU=" & NAS)
281 //I = DMax("IDSEG", "T0602_DOCU_S", "IDDOCU=" & NAS) + 1
282 //'ASIENTO CAJA Y BANCO
283 //Set rx = db.OpenRecordset("SELECT * FROM T0602_DOCU_S WHERE IDDOCU=-1")
284 //    rx.AddNew
285 //    rx!IDDOCU = NAS
286 //    rx!IDSEG = I
287 //    rx!MES = ra!MES
288 //    rx!YEAR = ra!YEAR
289 //    rx!CUENTA = Me.CCUENTA_PAGO
290 //    rx!MONE_ORI = "ARS"
291 //    rx!TC = ra!TC
292 //    rx!IMPORTE_ORI = Round(ra!TOT_ARS, 2)
293 //    rx!IMPORTE_ARS = Round(ra!TOT_ARS, 2)
294 //    rx!SEGTEXT = "Pago desde " & Me.TCUENTAO_DESC
295 //    rx!DOCUTIPO = "P"
296 //    rx!REFE = Me.NFACTURA
297 //    rx!Log_User = Environ("USERNAME")
298 //    rx!FECHA_HOY = Date
299 //    If Me.CKGN = True Then
300 //        rx!DEBE = 0
301 //        rx!HABER = Round((Me.TIVA10 + Me.TIVA21 + Me.TRETIIBB + Me.TRETIVA), 2)

```

```
302 //      rx!PROV = Me.TVENDOREMPL '*Aca va el vendor del empleado
303 //      Else
304 //      rx!DEBE = 0
305 //      rx!HABER = Round(ra!TOT_ARS, 2)
306 //      rx!PROV = Me.CPROVEEDOR
307 //      End If
308 //      rx!GL = Me.CCUENTA_PAGO
309 //      'rx!cc = rs!cc
310 //      rx!PERIODO = rx!YEAR & rx!MES
311 //      rx!TIPO = Me.TTIPO
312 //      rx!TCODE = "CONTAR"
313 //      rx.Update
314 //      '
315 //      I = I + 1
316 //      'Asiento A/P
317 //      If Me.CKGN = True Then
318 //          XGL = Me.TGLEMPL
319 //          PRO = Me.TVENDOREMPL
320 //      Else
321 //          XGL = Me.TGL
322 //          PRO = Me.CPROVEEDOR
323 //      End If
324 //      rx.AddNew
325 //      rx!IDDOCU = NAS
326 //      rx!IDSEG = I
327 //      rx!MES = ra!MES
328 //      rx!YEAR = ra!YEAR
329 //      rx!CUENTA = XGL
330 //      rx!MONE_ORI = "ARS"
331 //      rx!TC = ra!TC
332 //      rx!IMPORTE_ORI = Round(ra!TOT_ARS, 2)
333 //      rx!IMPORTE_ARS = Round(ra!TOT_ARS, 2)
334 //      rx!SEGTEXT = "A/P - Pago"
335 //      rx!DOCUTIPO = "P"
336 //      rx!REFE = Me.NFACTURA
337 //      rx!PROV = PRO
338 //      rx!Log_User = Environ("USERNAME")
339 //      rx!FECHA_HOY = Date
340 //      If Me.CKGN = True Then
341 //          rx!DEBE = Round((Me.TIVA10 + Me.TIVA21 + Me.TRETIIBB + Me.TRETIVA), 2)
342 //          rx!HABER = 0
343 //      Else
344 //          rx!DEBE = Round(ra!TOT_ARS, 2)
345 //          rx!HABER = 0
346 //      End If
347 //      rx!GL = XGL
348 //      'rx!cc = rs!cc
349 //      rx!PERIODO = rx!YEAR & rx!MES
350 //      rx!TIPO = Me.TTIPO
351 //      rx!TCODE = "CONTAR"
352 //rx.Update
353 //End Sub
```



```
354     }  
355 }  
356 }  
357
```