

75 FO-295 - Muestra datos diferente de tarjeta y falta de información al reimprimir un ticket de venta y 76 FO-299-Reimpresión de ticket de venta con donativo con inconsistencias

		Documento de causa raíz sobre incidentes N3	
Autor	Versión	Fecha de elaboración	Fecha autorización
@Sergio Calalpa Torres	1.0	24 jun 2026	Teclear "/" y seleccionar el elemento fecha - Solo el Líder Técnico

1.- Control de Versiones

Versión	Fecha	Autor	Descripción
1.0	24 jun 2026	@Sergio Calalpa Torres	Se realiza análisis del problema en reimpresión de tickets, información diferente.

2.- Descripción general

En este documento se consideran dos casos reportados como Jiras por parte de QA, los cuales son: **75 - FO-295 Muestra datos diferentes de tarjeta y falta de información al reimprimir un ticket de venta** y **76 - FO-299 Reimpresión de ticket de venta con donativo con inconsistencias**.

Se está reportando que, al momento de realizar una reimpresión de tickets de alguna venta realizada con tarjeta, esta muestra información diferente al ticket original. A continuación, se listan las diferencias detectadas al momento de la reimpresión.

- Información de la tarjeta con la que se realizó el pago

- Ticket original: ejemplo XXXXXXXXXXXXX0745.

```
      I.V.A. %:      $0.00
      FORMA DE PAGO:
      TARJETA DE CREDITO ACI (AC):      $340.40
      Autorización : 60879U
      Tarjeta      : XXXXXXXXXXXXX0745
      No. Monedero: *****4571 Tran: 1629017844
      Record Promoción: $15.93 Te faltan $84.07
      Saldo total del monedero: $50.88
      Esta compra acumuló: $2.52
```

- Reimpresión del ticket: ejemplo XXXXXXXXXXXXX45|0.

```
      FORMA DE PAGO:
      TARJETA DE CREDITO ACI (AC):      $340.40
      Autorización : 60879U
      Tarjeta      : XXXXXXXXXXXXX45|0
      No. Monedero: *****4571 Tran: 1629017844
      Record Promoción: $15.93 Te faltan $84.07
      Saldo total del monedero: $50.88
      Esta compra acumuló: $2.52
      !!! TU AHORRO $22.07 !!!
```

- Información de la compra con la leyenda “Su compra ha sido acumulada a su historial de promoción”

- El ticket original contiene la leyenda “Su compra ha sido acumulada a su historial de promoción” debajo de la leyenda “Esta compra acumuló: \$2.52”.

```
      TARJETA DE CREDITO ACI (AC):      $340.40
      Autorización : 60879U
      Tarjeta      : XXXXXXXXXXXXX0745
      No. Monedero: *****4571 Tran: 1629017844
      Record Promoción: $15.93 Te faltan $84.07
      Saldo total del monedero: $50.88
      Esta compra acumuló: $2.52
      Su compra ha sido acumulada a su historial de
      promoción.
      !!! TU AHORRO $22.07 !!!
      Autorización: 14458243357
      Beneficios:
      Monedero
      15.00% DE DESCUENTO EN GRANEODIN B MENTA EUCALIPTO
      C/24 TAB 5517 $104.00
```

- En la reimpresión del ticket no aparece la leyenda; solo se muestra “Esta compra acumuló: \$2.52”.

```

TARJETA DE CREDITO ACI (AC):      $340.40
Autorización : 60879U
Tarjeta      : XXXXXXXXXXXXXXX4510
No. Monedero: *****4571 Tran: 1629017844
Record Promoción: $15.93 Te faltan $84.07
Saldo total del monedero: $50.88
Esta compra acumuló: $2.52
!!! 10 AHORRO $22.07 !!!
Autorización: 1629017844
Beneficios:
Monedero
15.00% DE DESCUENTO EN GRANEODIN B MENTA EUCALIPTO
C/24 TAB 5517 $104.00

```

Nota: Para la diferencia marcada en el número de autorización, este tema ya fue detectado con anterioridad y tiene atención dentro del [FO-294 Número de autorización y transacción de monedero son diferentes al reimprimir ticket.](#)

- Tiny URL y código QR distintos
 - El ticket original, al ser generado, tiene un tiny URL y un código QR únicos.

Cuenta con 30 días naturales para solicitar su factura electrónica en sucursal donde hizo su compra o escanee el código QR con su celular o

<https://q.fdah.mx/IVUuzZHppjU>



ITU: 6ddd28e6-c31f-f011-b06e-18cf5eb27e8e



- En la reimpresión del ticket, al generar nuevamente el ticket, el tiny URL y el código QR son distintos al ticket original.

Cuenta con 30 días naturales para solicitar su factura electrónica en sucursal donde hizo su compra o escanee el código QR con su celular o

<https://q.fdah.mx/dbGxUNne19c>



ITU: 6ddd28e6-c31f-f011-b06e-18cf5eb27e8e



3.- Diagnóstico

El problema de la incorrecta reimpresión del número de tarjeta se debe a un doble enmascaramiento en *SaleManager.GetTransaccion*, donde el número ya viene enmascarado. Por otro lado, las leyendas provenientes del servicio *ORIGIS* no se tienen al momento de la reimpresión, ya que el servicio solo es consultado una vez al momento de realizar la venta, sin guardar parte de la respuesta en el store y quedándose únicamente en *POSDB_OR*.

Se realizó la identificación de los puntos dentro de la reimpresión que realiza el aplicativo, y se detectó que, al momento de recuperar y mostrar el número de tarjeta, se aplica el formato incorrecto listado al inicio.

Formato incorrecto

```
FORMA DE PAGO:
TARJETA DE CREDITO ACI (AC):          $340.40
Autorización: 60079U
Tarjeta: XXXXXXXXXXXXXXX45|0
No. Monedero: 1575 Trans 1629017844
Record Promoción: $15.93 Te faltan $84.07
Saldo total del monedero: $50.88
```

Acción donde se realiza este efecto:

Como podemos observar en la sección marcada en rojo del código, al obtener el valor del número de tarjeta se realiza un enmascaramiento adicional agregando al final del número de tarjeta un carácter “|” seguido de *keyversion*, el cual se obtiene como llave de cifrado.

```

SaleManager.cs
// recuperamos folio de autorización y el numero de tarjeta del metodo de pago en caso de ser electrónico
saleResult.TenderInfo.TenderTypeDetails.ForEach(p =>
{
    var electronicPayment = tenderManager.GetElectronicTenderByID(contextToUse, saleResult.Sale.folioTransaccion, p.codigoFormaPago, p.TenderType, p.autorizacion);

    if (p.IsThird && electronicPayment != null)
    {
        if (p.TenderType != SubTenderType.GiftCard)
        {
            string keyVersion = string.Format("{0}", electronicPayment.codigoLlaveCifrado ?? 0);
            p.CardNumber = string.Format("{0}|{1}", electronicPayment.numeroTarjeta, keyVersion);
        }
        else
        {
            p.CardNumber = electronicPayment.numeroTarjeta;
        }

        p.AdditionalData = electronicPayment.DatosAdicionales;
    }
});

```

Leyenda faltante

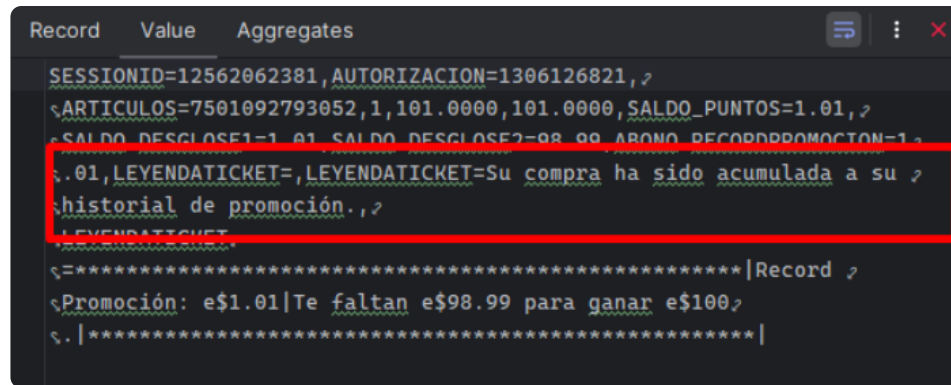
Para la leyenda faltante en la reimpresión del ticket, se identificó que esta proviene de *ORIGIS*. El consumo del servicio se realiza directamente desde el central en *OrigisManager* en el método *Commit*; por lo tanto, en una impresión inicial del ticket, esta leyenda se toma directamente de la respuesta, mientras que, al hacer la reimpresión, no se realiza ninguna petición a *ORIGIS*.

Dentro del proceso de consumo del servicio de *ORIGIS* en el método ya mencionado, la respuesta se guarda en la base de datos *POSDB_OR* en el central, en la tabla *genBitacoraMonedero*.

```

use FARMAX_CENTRAL
go
select datosRecibidos from genBitacoraMonedero;

```



```
Record  Value  Aggregates
SESSIONID=12562062381,AUTORIZACION=1306126821,¿
ARTICULOS=7501092793052,1,101.0000,101.0000,SALDO_PUNTOS=1.01,¿
SALDO_DESCLOSE1=1.01,SALDO_DESCLOSE2=98.99,ABONO_RECORDPROMOCION=1.¿
.01,LEVENDATICKET=,LEVENDATICKET=Su compra ha sido acumulada a su ¿
historial de promoción.,¿
LEVENDATICKET=
=*****|Record ¿
Promoción: e$1.01|Te faltan e$98.99 para ganar e$100¿
. |***** |
```

Código QR (no aplicable)

Para el último punto sobre el código QR distinto, esto es **correcto**, ya que la herramienta generadora de códigos QR devuelve siempre un código diferente en cada petición. Esto mismo fue comentado en su momento durante el levantamiento del punto. Por lo tanto, este último punto no aplica, ya que actualmente es el funcionamiento esperado.

4.- Propuesta de solución

Para los dos puntos aplicables a la solicitud, los cuales son el número de tarjeta y el registro de las leyendas de *ORIGIS* al momento de la reimpresión, se proponen los siguientes cambios para su solución.

Formato incorrecto

Actualmente, para la reimpresión se realiza un cambio en los últimos dígitos para los métodos de pago electrónicos, dentro de la clase *SaleManager* en el método *GetTransaccion*.

```

SaleManager.cs
// recuperamos folio de autorización y el numero de tarjeta del metodo de pago en caso de ser electrónico
saleResult.TenderInfo.TenderTypeDetails.ForEach(p =>
{
    var electronicPayment = tenderManager.GetElectronicTenderByID(contextToUse, saleResult.Sale.folioTransaccion, p.codigoFormaPago, p.TenderType, p.autorizacion);

    if (p.IsThird && electronicPayment != null)
    {
        if (p.TenderType != SubTenderType.GiftCard)
        {
            string keyVersion = string.Format("{0}", electronicPayment.codigoLlaveCifrado ?? 0);
            p.CardNumber = string.Format("{0}{1}", electronicPayment.numeroTarjeta, keyVersion);
        }
        else
        {
            p.CardNumber = electronicPayment.numeroTarjeta;
        }

        p.AdditionalData = electronicPayment.DatosAdicionales;
    }
});

```

Este proceso se está realizando de más, ya que desde el método *GetElectronicTenderByID* dentro de la clase *TenderManager*, **se recupera el número de tarjeta ya enmascarado**.

```

TenderManager.cs
public genFormasPagosElectronicas GetElectronicTenderByID(EntityConnectionCreator.ContextTypes contextType, string filter, int idPayment, string tenderType, string authorizationCode
= "")
{
    using (var context = new TenderDbContext(contextType))
    {
        var dao = new TenderDao(context);
        var logElectroincTender genFormasPagosElectronicas = dao.GetElectronicTenderByID(filter, idPayment, authorizationCode);

        // Numero de tarjeta con mascara solo mostrando los últimos 4 dígitos, para CoDi
        if (logElectroincTender != null && logElectroincTender.codigoLlaveCifrado == 0 && tenderType != SubTenderType.CoDi && tenderType != SubTenderType.WirelessCard)
        {
            logElectroincTender.numeroTarjeta = !string.IsNullOrEmpty(logElectroincTender.numeroTarjeta) && logElectroincTender.numeroTarjeta.Length > 4 ? logElectroincTender
            .numeroTarjeta.Substring(logElectroincTender.numeroTarjeta.Length - 4).PadLeft(logElectroincTender.numeroTarjeta.Length, paddingChar: 'X') : logElectroincTender
            .numeroTarjeta;
        }

        return logElectroincTender;
    }
}

```

Por lo tanto, el cambio propuesto es dentro de la clase *SaleManager* en el método *GetTransaccion*, en el cual se quitará la condición de *p.TenderType != SubTenderType.GiftCard*, pasando de manera directa el número de tarjeta que se está recuperando con su formato correcto de enmascaramiento.

```
SaleManager.cs
// recuperamos folio de autorización y el numero de tarjeta del metodo de pago en caso de ser electrónico
saleResult.TenderInfo.TenderTypeDetails.ForEach(p =>
{
    var electronicPayment = tenderManager.GetElectronicTenderByID(contextToUse, saleResult.Sale.folioTransaccion, p.codigoFormaPago, p.TenderType, p.autorizacion);

    if (p.IsThird && electronicPayment != null)
    {
        p.CardNumber = electronicPayment.numeroTarjeta;
        p.AdditionalData = electronicPayment.DatosAdicionales;
    }
});
```

```
// recuperamos folio de autorización y el numero de tarjeta del metodo de pago en caso de ser electrónico
saleResult.TenderInfo.TenderTypeDetails.ForEach(p =>
{
    var electronicPayment = tenderManager.GetElectronicTenderByID(contextToUse, saleResult.Sale.folioTransaccion, p.codigoFormaPago, p.TenderType, p.autorizacion);

    if (p.IsThird && electronicPayment != null)
    {
        if (p.TenderType != SubTenderType.GiftCard)
        {
            string keyVersion = string.Format("{0}", electronicPayment.codigollaveCifrado ?? 0);
            p.CardNumber = string.Format("{0}{1}", electronicPayment.numeroTarjeta, keyVersion);
        }
        else
        {
            p.CardNumber = electronicPayment.numeroTarjeta;
        }

        p.AdditionalData = electronicPayment.DatosAdicionales;
    }
});
```

Leyenda faltante

Para almacenar las leyendas de *ORIGIS*, se tiene la siguiente propuesta de solución debido a la estructura actual del sistema. Se propone generar una nueva tabla que almacene únicamente los parámetros de leyenda de *ORIGIS*, tomando en cuenta el mismo campo *datosRecibidos* de la tabla *genBitacoraMonedero*. Debido a que el consumo se realiza directamente en central, se contempla que la nueva tabla almacene información directamente en estos dos casos: central y store, ya que la reimpresión de tickets se realiza a nivel store y, para históricos, se aplica desde central.

Estructura de la tabla *venRespuestaOrigis*

Campo	Tipo dato	Capacidad
idRespuesta	uniqueidentifier	

folioTransaccion	nvarchar	15
codigoSucursal	int	
numeroCaja	int	
codigoCajero	int	
datosEnviados	nvarchar	1000
datosRecibidos	nvarchar	1500

Se deben realizar las modificaciones al modelo *MODEL_VEN.xml* para contemplar la nueva tabla y sus respectivas consideraciones, como la adecuación a las consultas de *EntityFramework*.

```

MODEL_VEN.xml
<!--
*****
Tabla que contiene consumo de servicio de ORIGIS
asi como respuesta y envio para reimpression de tickets

Actualizacion: 2026-07-24
Se crea la tabla
*****
-->

<table name="VENRESPUESTAORIGIS" real="venRespuestaOrigis" type="model">

  <!--PRIMARY KEYS-->

  <column name="IDRESPUESTA" real="idRespuesta" type="guid" flag="primary_key" default="newsequentialid()" properties="rowguidcol" />

  <!--COLUMNS-->

  <column name="FOLIOTRANSACCION" real="folioTransaccion" type="string" stringlength="15" flag="not_null"/>
  <column name="CODIGOSUCURSAL" real="codigoSucursal" type="int32" flag="not_null"/>
  <column name="NUMEROCAJA" real="numeroCaja" type="int32" flag="not_null"/>
  <column name="CODIGOCAJERO" real="codigoCajero" type="int32" flag="not_null" />
  <column name="DATOSENVIADOS" real="datosEnviados" stringlength="1000" type="string" flag="not_null" />
  <column name="DATOSRECIBIDOS" real="datosRecibidos" stringlength="1500" type="string" flag="null" />

  <!-- CAMPOS DE RASTREO DE SINCRONIZACION ANXO -->

  <column name="SYNC_CREADO_FECHA" real="syncCreadoFecha" type="datetime" flag="tracking_insert_date" default="getdate()" />
  <column name="SYNC_ACTUALIZADO_FECHA" real="syncActualizadoFecha" type="datetime" flag="tracking_update_date" default="getdate()" />
  <column name="SYNC_ACTUALIZADO_POR" real="syncActualizadoPor" type="string" stringlength="100" flag="tracking_update_log" default="FARMAX.DEFAULT" />
  <column name="SYNC_VERSION_REGISTRO" real="syncVersionRegistro" type="int32" flag="tracking_version" default="1" />
  <column name="SYNC_TRANSFERIDO" real="syncTransferido" type="string" stringlength="1" default="N" />

  <!--INDEXES-->
  <indexes>
    <index name="VENRESPUESTAORIGIS_IDX01" type="nonunique">
      <column name="CODIGOSUCURSAL" />
    </index>
  </indexes>

</table>

```

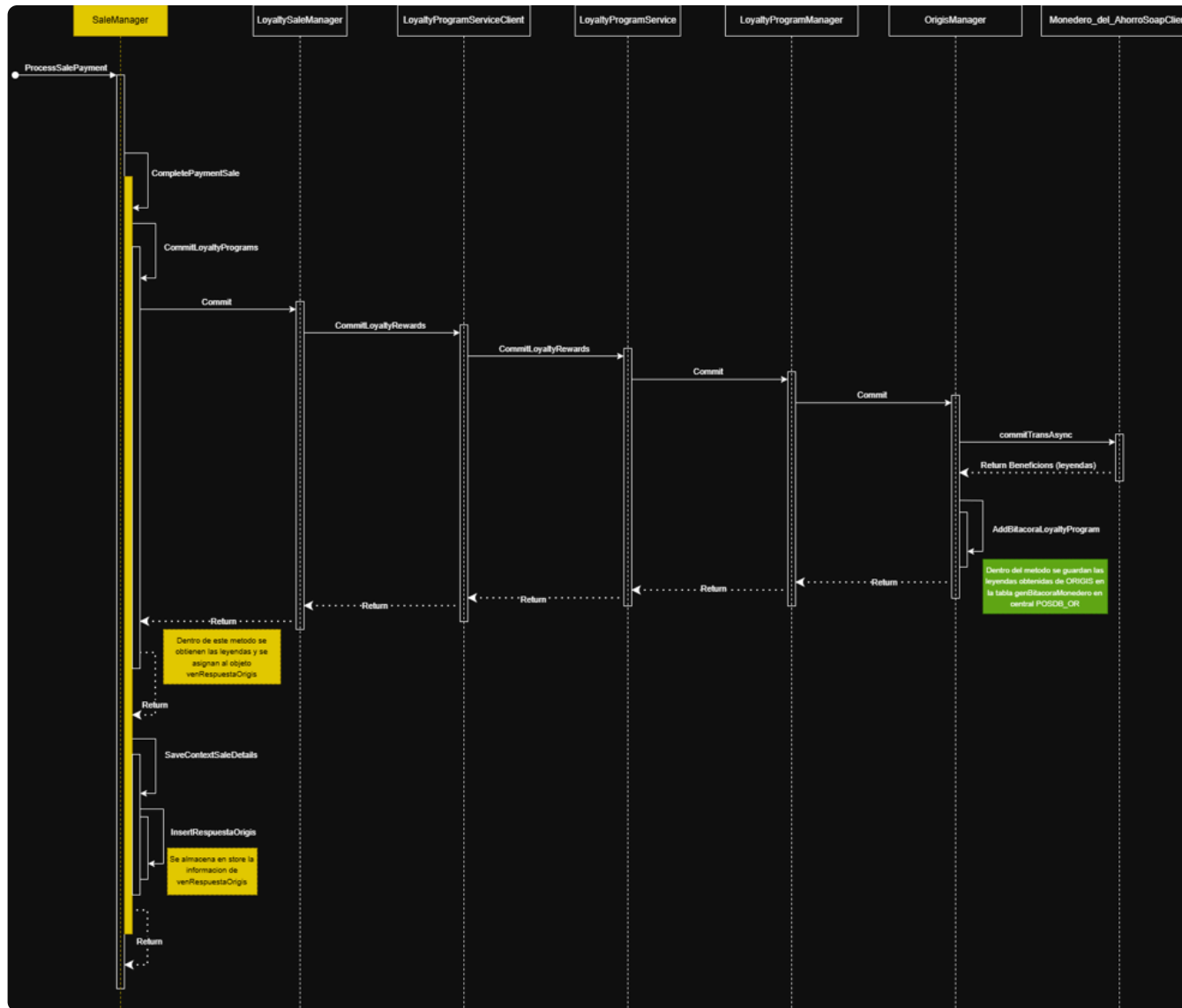
De igual forma que en el *MODEL_VEN.xml*, modificaremos el *MappingObjectDownloadModel.xml* para contemplar la bajada de *POSDB_OR* a *store* de la nueva tabla generada.

```

_MappingObjectUploadModel.xml
<table capital="false" meta="ISerializeObject;ISerializable" name="VENRESPUESTAORIGIS" real="venRespuestaOrigis">
  <!-- CAMPOS DE RASTREO DE SINCRONIZACION ANXO -->
  <attribute column="syncActualizadoFecha" default="getdate()" name="SYNC_ACTUALIZADO_FECHA" type="datetime"/>
  <attribute column="syncActualizadoPor" default="FARMAX" name="SYNC_ACTUALIZADO_POR" stringlength="100" type="string"/>
  <attribute column="syncCreadoFecha" default="getdate()" name="SYNC_CREADO_FECHA" type="datetime"/>
  <attribute column="syncTransferido" default="N" name="SYNC_TRANSFERIDO" stringlength="1" type="string"/>
  <attribute column="syncVersionRegistro" default="1" name="SYNC_VERSION_REGISTRO" type="int32"/>
  <column name="FOLIOTRANSACCION" real="folioTransaccion" type="string" stringlength="15" flag="not_null" />
  <column name="CODIGOSUCURSAL" real="codigoSucursal" type="int32" flag="not_null"/>
  <column name="NUMEROCAJA" real="numeroCaja" type="int32" flag="not_null"/>
  <column name="CODIGOCAJERO" real="codigoCajero" type="int32" flag="not_null" />
  <column name="DATOSENVIADOS" real="datosEnviados" stringlength="1000" type="string" flag="not_null" />
  <column name="DATOSRECIBIDOS" real="datosRecibidos" stringlength="1500" type="string" flag="null" />
</table>

```

El registro de las leyendas en *venRespuestaOrigis* se realizará dentro del flujo inicial de la venta. A continuación, se agrega un diagrama de secuencia donde se indica dónde se obtienen las leyendas de *ORIGIS* hasta el momento en que se recuperan para el objeto *venRespuestaOrigis*, así como el guardado que se realizará.



Para esto, se modificará dentro de la clase *SaleManager* el método *CompletePaymentSale* para obtener las leyendas provenientes de *ORIGIS*, agregándolas al nuevo objeto *venRespuestaOrigis*.

```

SaleManager.cs
// Procesa los programas de lealtad.
var resultLoyalty :ResultStruct<List<...>> = this.CommitLoyaltyPrograms(saleTransaction);

if (!resultLoyalty.HasError)
{
    if (resultLoyalty.Result != null && resultLoyalty.Result.Any()
        && saleTransaction.LoyaltyPrograms != null && saleTransaction.LoyaltyPrograms.Any(l :LoyaltyBenefits => l.LoyaltyProgramID == EnumLoyaltyProgram.Origis))
    {
        saleTransaction.LoyaltyPrograms.First().Message = resultLoyalty.Result.First().Message;

        this.AddRespuestaOrigis(new venRespuestaOrigis()
        {
            folioTransaccion = saleTransaction.Sale.folioTransaccion,
            codigoCajero = (int)saleTransaction.Sale.codigoCajero ,
            codigoSucursal = saleTransaction.Sale.codigoSucursal,
            datosEnviados = "",
            datosRecibidos = string.Join(",",
                resultLoyalty.Result.First().Ticketlegends.Select(x :string => $"LEYENDATICKET={x}"))
        });
    }

    // Actualiza la información de la forma de pago Mda.
    if (paymentDetailSale != null && paymentDetailSale.genMovimientosCajasDet.Any(x :genMovimientosCajasDet => x.estatusFormaPago == StatusTenderDetail.Applied && (x
        .genFormasDePagoCat.tipoFormaPago == SubTenderType.EWallet || x.genFormasDePagoCat.tipoFormaPago == SubTenderType.MobilePayment))) {...}
}
else{...}

```

<pre> // Procesa los programas de lealtad. var resultLoyalty = this.CommitLoyaltyPrograms(saleTransaction); if (!resultLoyalty.HasError) { if (resultLoyalty.Result != null && resultLoyalty.Result.Any() && saleTransaction.LoyaltyPrograms != null && saleTransaction.LoyaltyPrograms.Any(l => l.LoyaltyProgramID == EnumLoyaltyProgram.Origis)) { saleTransaction.LoyaltyPrograms.First().Message = resultLoyalty.Result.First().Message; } // Actualiza la información de la forma de pago Mda. if (paymentDetailSale != null && paymentDetailSale.genMovimientosCajasDet.Any(x => x .estatusFormaPago == StatusTenderDetail.Applied && (x.genFormasDePagoCat.tipoFormaPago == SubTenderType.EWallet x.genFormasDePagoCat.tipoFormaPago == SubTenderType.MobilePayment))) { // Actualiza la forma de pago monedero o monedero virtual. var paymentEwalletDetail = paymentDetailSale.genMovimientosCajasDet.First(x => (x .genFormasDePagoCat.tipoFormaPago == SubTenderType.EWallet x.genFormasDePagoCat .tipoFormaPago == SubTenderType.MobilePayment) && x.estatusFormaPago == StatusTenderDetail .Applied); // Se valida si en la venta se realizó una redención de saldo if (saleTransaction.RedemedBalance > 0) { paymentEwalletDetail.autorización = saleTransaction.Sale.idAutorizaciónMonedero; } } } </pre>	<pre> 1374 1375 // Procesa los programas de lealtad. 1376 1377 var resultLoyalty :ResultStruct<...> = this.CommitLoyaltyPrograms(saleTransaction); 1378 1379 1379 1380 if (!resultLoyalty.HasError) 1380 1381 { 1381 1382 if (resultLoyalty.Result != null && resultLoyalty.Result.Any() 1382 1383 && saleTransaction.LoyaltyPrograms != null && saleTransaction.LoyaltyPrograms.Any(l :LoyaltyBenefits => l.LoyaltyProgramID == EnumLoyaltyProgram.Origis)) 1383 1384 { 1384 1385 saleTransaction.LoyaltyPrograms.First().Message = resultLoyalty.Result.First().Message; 1385 1386 1386 1387 this.AddRespuestaOrigis(new venRespuestaOrigis() 1387 1388 { 1388 1389 folioTransaccion = saleTransaction.Sale.folioTransaccion, 1389 1390 codigoCajero = (int)saleTransaction.Sale.codigoCajero , 1390 1391 codigoSucursal = saleTransaction.Sale.codigoSucursal, 1391 1392 datosEnviados = "", 1392 1393 datosRecibidos = string.Join(",", 1393 1394 resultLoyalty.Result.First().Ticketlegends.Select(x :string => \$"LEYENDATICKET={x}")) 1394 1395 }); 1395 1396 } 1396 1397 1397 1398 // Actualiza la información de la forma de pago Mda. 1398 1399 if (paymentDetailSale != null && paymentDetailSale.genMovimientosCajasDet.Any(x 1399 1400 .genMovimientosCajasDet => x.estatusFormaPago == StatusTenderDetail.Applied && (x 1400 1401 .genFormasDePagoCat.tipoFormaPago == SubTenderType.EWallet x.genFormasDePagoCat.tipoFormaPago 1401 1402 == SubTenderType.MobilePayment))) {...} 1402 1403 } 1403 1404 else{...} </pre>
---	--

Al momento de realizar la reimpresión del ticket, se realizará la consulta en store a la nueva tabla venRespuestaOrigis para obtener las leyendas almacenadas para la venta en cuestión.


```

SaleManager.cs
// Agrega las bonificaciones en caso que existan
if (saleResult.Sale.trnRecompensasProgramaLealtad.Any())
{
    saleResult.Sale.tieneProgramaLealtad = true;
    foreach (trnRecompensasProgramaLealtad promo in saleResult.Sale.trnRecompensasProgramaLealtad)
    {
        EnumLoyaltyProgram loyaltyProgramID = this.GetLoyaltyProgramIdentifier(promo);

        saleResult.LoyaltyPrograms.Add(new LoyaltyBenefits
        {
            LoyaltyProgramID = loyaltyProgramID,
            Card = promo.folioProgramaLealtad,
            AuthorizationCode = promo.numeroAutorizacion,
            SessionID = promo.idSession,
            OthersBalance = new List<LoyaltyBalance>() // Solo se guarda para ORIGIS
        });

        if (loyaltyProgramID == EnumLoyaltyProgram.Origis)
        {
            // Solo si es ORIGIS se establecen los valores de sesión, balance, puntos,
            // redimidos e importe record promoción
            saleResult.LoyaltyPrograms.Last().SessionID = saleResult.Sale.idSessionMonedero;
            saleResult.LoyaltyPrograms.Last().Balance = saleResult.Sale.saldoActualMonedero;
            saleResult.LoyaltyPrograms.Last().OthersBalance = new List<LoyaltyBalance>
            {
                new LoyaltyBalance { Type = EnumBalanceType.Reward, Balance = saleResult.Sale.puntosGenerados ?? 0 },
                new LoyaltyBalance { Type = EnumBalanceType.RecordPromotion, Balance = saleResult.Sale.importeRecordActual ?? 0 }
            };

            // Si no existe ORIGIS se agrega a la lista de PL (El abono de puntos se considera como PL).
            if (saleResult.LoyaltyPrograms.All(p => p.LoyaltyProgramID != EnumLoyaltyProgram.Origis))
            {
                saleResult.LoyaltyPrograms.Add(new LoyaltyBenefits
                {
                    LoyaltyProgramID = EnumLoyaltyProgram.Origis,
                    Card = saleResult.Sale.folioMonedero,
                    AuthorizationCode = saleResult.Sale.idAutorizacionMonedero,
                    SessionID = saleResult.Sale.idSessionMonedero,
                    Balance = saleResult.Sale.saldoActualMonedero,
                    OthersBalance = new List<LoyaltyBalance>
                    {
                        new LoyaltyBalance { Type = EnumBalanceType.Reward, Balance = saleResult.Sale.puntosGenerados ?? 0 },
                        new LoyaltyBalance { Type = EnumBalanceType.RecordPromotion, Balance = saleResult.Sale.importeRecordActual ?? 0 }
                    }
                });
            }
        }
        else if (saleResult.LoyaltyPrograms.Count == 1 && saleResult.LoyaltyPrograms.Any(f => f.LoyaltyProgramID == EnumLoyaltyProgram.Origis))
        {
            var loyaltyProgram = saleResult.LoyaltyPrograms.First();
            loyaltyProgramID = loyaltyProgramID,
            Card = promo.folioProgramaLealtad,
            AuthorizationCode = promo.numeroAutorizacion,
            SessionID = promo.idSession,
            OthersBalance = new List<LoyaltyBalance>() // Solo se guarda para ORIGIS
        });

        if (loyaltyProgramID == EnumLoyaltyProgram.Origis)
        {
            yemRespuestaOrigis origis =
            saleDao.GetRespuestaOrigis(saleResult.Sale.folioTransaccion);
            // Solo si es ORIGIS se establecen los valores de sesión, balance, puntos
            // redimidos e importe record promoción
            saleResult.LoyaltyPrograms.Last().SessionID = saleResult.Sale.idSessionMonedero;
            saleResult.LoyaltyPrograms.Last().Balance = saleResult.Sale.saldoActualMonedero;
            saleResult.LoyaltyPrograms.Last().TicketLegends = this.ParserLeyendaALista(origis?.datosRecibidos);
            saleResult.LoyaltyPrograms.Last().OthersBalance = new List<LoyaltyBalance>
            {
                new LoyaltyBalance { Type = EnumBalanceType.Reward, Balance = saleResult.Sale.puntosGenerados ?? 0 },
                new LoyaltyBalance { Type = EnumBalanceType.RecordPromotion, Balance = saleResult.Sale.importeRecordActual ?? 0 }
            };

            // Si no existe ORIGIS se agrega a la lista de PL (El abono de puntos se considera como PL).
            if (saleResult.LoyaltyPrograms.All(p => p.LoyaltyProgramID != EnumLoyaltyProgram.Origis))
            {
                yemRespuestaOrigis origis =
                saleDao.GetRespuestaOrigis(saleResult.Sale.folioTransaccion);
                saleResult.LoyaltyPrograms.Add(new LoyaltyBenefits
                {
                    LoyaltyProgramID = EnumLoyaltyProgram.Origis,
                    Card = saleResult.Sale.folioMonedero,
                    AuthorizationCode = saleResult.Sale.idAutorizacionMonedero,
                    SessionID = saleResult.Sale.idSessionMonedero,
                    Balance = saleResult.Sale.saldoActualMonedero,
                    TicketLegends = this.ParserLeyendaALista(origis?.datosRecibidos),
                    OthersBalance = new List<LoyaltyBalance>
                    {
                        new LoyaltyBalance { Type = EnumBalanceType.Reward, Balance = saleResult.Sale.puntosGenerados ?? 0 },
                        new LoyaltyBalance { Type = EnumBalanceType.RecordPromotion, Balance = saleResult.Sale.importeRecordActual ?? 0 }
                    }
                });
            }
        }
    }
}

```

Métodos adicionales a contemplar dentro de la misma clase SaleManager.

AddRespuestaOrigis se utilizará para el guardado de las respuestas de ORIGIS, y el método ParserLeyendaALista convertirá el resultado de las leyendas almacenadas como string separado por comas en una lista.

```

SaleManager.cs

public void AddRespuestaOrigis(venRespuestaOrigis origis)
{
    using (var context = new SaleDbContext())
    {
        var saleDao = new SalesDao(context);
        saleDao.AddRespuestaOrigis(origis);
    }
}

2 usages new *
public List<string> ParserLeyendsALista(string legends)
{
    if (legends != null)
    {
        return legends.Split(',').Select(x => x.Replace("LEYENDATICKET=", string.Empty)).ToList();
    }

    return new List<string>() {"", "", ""};
}

```

<pre> /// <summary> /// Consulta la cabecera de una venta unicamente. /// </summary> /// <param name="transactionId">Identificador de transacción</param> /// <returns>Cabecera de la transacción consultada</returns> public trnTransaccionesCab GetTransactionHeaderIncludeDetails(Guid transactionId) { using (var context = new SaleDbContext()) { var saleDao = new SalesDao(context); return saleDao.GetTransactionHeader(transactionId, true); } } /// <summary> /// Metodo con el que se obtiene la venta independientemente del origen que este tiene /// </summary> /// <param name="refund">Metodos de la devolucion</param> /// <returns>Datos de la venta</returns> private bool IsSaleScheduledOrder(Guid? sadOrderId) { return (new SaleManager()).IsScheduledDeliveryService(sadOrderId); } </pre>	<pre> 7187 7188 7189 7190 7191 7192 7193 7194 7195 7196 7196 7197 7197 7198 7198 7199 7199 7200 7200 7201 7201 7202 7202 7203 7203 7204 7204 7205 7205 7206 7206 7207 7207 7208 </pre>	<pre> /// <summary> /// Metodo con el que se obtiene la venta independientemente del origen que este tiene /// </summary> /// <param name="refund">Metodos de la devolucion</param> /// <returns>Datos de la venta</returns> private bool IsSaleScheduledOrder(Guid? sadOrderId) { return (new SaleManager()).IsScheduledDeliveryService(sadOrderId); } public void AddRespuestaOrigis(venRespuestaOrigis origis) { using (var context = new SaleDbContext()) { var saleDao = new SalesDao(context); saleDao.AddRespuestaOrigis(origis); } } public List<string> ParserLeyendsALista(string legends) { if (legends != null) { return legends.Split(',').Select(x => x.Replace("LEYENDATICKET=", string.Empty)).ToList(); } return new List<string>() {"", "", ""}; } </pre>
---	--	---

A continuación, se listan los cambios en patch separados por archivo, los cuales se vieron afectados de manera colateral por la implicación del cambio propuesto, como lo es la actualización del modelo de reimpresión del ticket, contemplando las leyendas, etc.

Patch sobre la clase SaleManager.cs

```

1 Index: Farmax/Src/POS/BusinessLogic.Sale/SaleManager.cs
2 IDEA additional info:
3 Subsystem: com.intellij.openapi.diff.impl.patch.CharsetEP
4 <+>UTF-8
5 =====
6 diff --git a/Farmax/Src/POS/BusinessLogic.Sale/SaleManager.cs b/Farmax/Src/POS/BusinessLogic.S
7 --- a/Farmax/Src/POS/BusinessLogic.Sale/SaleManager.cs (revision 984eae8257aadb2575d3f79edd05
8 +++ b/Farmax/Src/POS/BusinessLogic.Sale/SaleManager.cs (date 1785431349222)
9 @@ -1379,6 +1379,16 @@
10         && saleTransaction.LoyaltyPrograms != null && saleTransaction.Loyalty
11         {
12             saleTransaction.LoyaltyPrograms.First().Message = resultLoyalty.Result
13 +
14 +             this.AddRespuestaOrigis(new venRespuestaOrigis()
15 +             {
16 +                 folioTransaccion = saleTransaction.Sale.folioTransaccion,
17 +                 codigoCajero = (int)saleTransaction.Sale.codigoCajero ,
18 +                 codigoSucursal = saleTransaction.Sale.codigoSucursal,
19 +                 datosEnviados = "",
20 +                 datosRecibidos = string.Join(",",
21 +                     resultLoyalty.Result.First().TicketLegends.Select(x => $"LEVE
22 +             });
23         }
24
25         // Actualiza la información de la forma de pago MdA.
26 @@ -3666,15 +3676,7 @@
27
28         if (p.IsThird && electronicPayment != null)
29         {
30 -             if (p.TenderType != SubTenderType.GiftCard)
31 -             {
32 -                 string keyVersion = string.Format("{0}", electronicPayment
33 -                 p.CardNumber = string.Format("{0}|{1}", electronicPayment
34 -             }
35 -             else
36 -             {
37 -                 p.CardNumber = electronicPayment.numeroTarjeta;
38 -             }
39 +             p.CardNumber = electronicPayment.numeroTarjeta;
40
41             p.AdditionalData = electronicPayment.DatosAdicionales;
42         }
43 @@ -3831,9 +3833,12 @@
44
45         if (loyaltyProgramID == EnumLoyaltyProgram.Origis)
46         {
47 +             venRespuestaOrigis origis =
48 +                 saleDao.GetRespuestaOrigis(saleResult.Sale.folioT
49             // Solo si es ORIGIS se establecen los balores de ses
50             saleResult.LoyaltyPrograms.Last().SessionId = saleRes

```

```

51         saleResult.LoyaltyPrograms.Last().Balance = saleResult.LoyaltyPrograms.Last().Balance;
52 +         saleResult.LoyaltyPrograms.Last().Ticketlegends = this.ParserLeyendsALista(origis?.datosRec);
53         saleResult.LoyaltyPrograms.Last().OthersBalance = new List<LoyaltyBalance>
54         {
55             new LoyaltyBalance { Type = EnumBalanceType.Reward, Balance = 0, Ticketlegends = null, OthersBalance = null };
56 @@ -4008,6 +4013,8 @@
57         // Si no existe ORIGIS se agrega a la lista de PL (El abono de puntos)
58         if (saleResult.LoyaltyPrograms.All(p => p.LoyaltyProgramID != EnumLoyaltyProgram.Origis))
59         {
60 +             venRespuestaOrigis origis =
61 +             saleDao.GetRespuestaOrigis(saleResult.Sale.folioTransaccion);
62             saleResult.LoyaltyPrograms.Add(new LoyaltyBenefits
63             {
64                 LoyaltyProgramID = EnumLoyaltyProgram.Origis,
65 @@ -4015,6 +4022,7 @@
66                 AuthorizationCode = saleResult.Sale.idAutorizacionMonedero,
67                 SessionId = saleResult.Sale.idSesionMonedero,
68                 Balance = saleResult.Sale.saldoActualMonedero,
69 +                 Ticketlegends = this.ParserLeyendsALista(origis?.datosRec);
70                 OthersBalance = new List<LoyaltyBalance>
71                 {
72                     new LoyaltyBalance { Type = EnumBalanceType.Reward, Balance = 0, Ticketlegends = null, OthersBalance = null };
73 @@ -7178,6 +7186,24 @@
74         {
75             return (new SaleManager()).IsScheduledDeliveryService(sadOrderId);
76         }
77 +
78 +     public void AddRespuestaOrigis(venRespuestaOrigis origis)
79 +     {
80 +         using (var context = new SaleDbContext())
81 +         {
82 +             var saleDao = new SalesDao(context);
83 +             saleDao.AddRespuestaOrigis(origis);
84 +         }
85 +     }
86 +     public List<string> ParserLeyendsALista(string legends)
87 +     {
88 +         if (legends != null)
89 +         {
90 +             return legends.Split(',').Select(x => x.Replace("LEYENDATICKET=", string.Empty)).ToList();
91 +         }
92 +         return new List<string>() { "", "", "" };
93 +     }
94 + }
95 }
96 }
97 }
98 \ No newline at end of file

```

Patch sobre la clase Sale.cs

```
1 Index: Farmax/Src/POS/BusinessLogic.Sale.DT0/Custom/Sale.cs
2 IDEA additional info:
3 Subsystem: com.intellij.openapi.diff.impl.patch.CharsetEP
4 <+>UTF-8
5 =====
6 diff --git a/Farmax/Src/POS/BusinessLogic.Sale.DT0/Custom/Sale.cs b/Farmax/Src/POS/BusinessLog
7 --- a/Farmax/Src/POS/BusinessLogic.Sale.DT0/Custom/Sale.cs (revision 984eae8257aadb2575d3f79e
8 +++ b/Farmax/Src/POS/BusinessLogic.Sale.DT0/Custom/Sale.cs (date 1785392937312)
9 @@ -271,5 +271,10 @@
10     /// Indica si el medico externo fue seleccionado
11     /// </summary>
12     public bool IsDoctorSelected { get; set; }
13 +
14 +     /// <summary>
15 +     /// Obtiene leyendas de origis
16 +     /// </summary>
17 +     public venRespuestaOrigis RespuestaOrigis { get; set; }
18 }
19 }
```

Patch sobre la clase SalesDao.cs

```
1 Index: Farmax/Src/POS/Data.Sale/DAO/SalesDao.cs
2 IDEA additional info:
3 Subsystem: com.intellij.openapi.diff.impl.patch.CharsetEP
4 <+>UTF-8
5 =====
6 diff --git a/Farmax/Src/POS/Data.Sale/DAO/SalesDao.cs b/Farmax/Src/POS/Data.Sale/DAO/SalesDao.
7 --- a/Farmax/Src/POS/Data.Sale/DAO/SalesDao.cs (revision 984eae8257aadb2575d3f79edd05e4388035
8 +++ b/Farmax/Src/POS/Data.Sale/DAO/SalesDao.cs (date 1785431349259)
9 @@ -1871,6 +1871,36 @@
10     return this.DataContext.trnTransaccionesCab.Include(x => x.trnTransaccionesDet).W
11 }
12
13 +     /// <summary>
14 +     /// Recupera de store respuestas de origis para leyendas
15 +     /// </summary>
16 +     /// <param name="folioTransaccion">folio de venta</param>
17 +     /// <returns>regresa registro venRespuestaOrigis</returns>
18 +     public venRespuestaOrigis GetRespuestaOrigis(string folioTransaccion)
19 +     {
20 +         var origis = this.DataContext.venRespuestaOrigis.Where(w => w.folioTransaccion ==
21 +         if (origis != null)
22 +         {
23 +             return origis;
24 +         }
```

```

25 +         return new venRespuestaOrigis();
26 +     }
27 +
28 +     public void AddRespuestaOrigis(venRespuestaOrigis origis)
29 +     {
30 +         try
31 +         {
32 +             this.DataContext.venRespuestaOrigis.Add(origis);
33 +             this.DataContext.SaveChanges();
34 +
35 +             Log.WriteInfo(SubModule, "AddRespuestaOrigis");
36 +         }
37 +         catch (Exception e)
38 +         {
39 +             Log.WriteException(SubModule, e, "SalesDao - AddRespuestaOrigis");
40 +         }
41 +     }
42 +
43 +     /// <summary>
44 +     /// Actualiza la Información del MDA en Offline
45 +     /// </summary>

```

Patch sobre la clase LoyaltyProgramProviderBase.cs

```

1  Index: Farmax/Src/POS/BusinessLogic.LoyaltyProgram.Provider/LoyaltyProgramProviderBase.cs
2  IDEA additional info:
3  Subsystem: com.intellij.openapi.diff.impl.patch.CharsetEP
4  <+>UTF-8
5  =====
6  diff --git a/Farmax/Src/POS/BusinessLogic.LoyaltyProgram.Provider/LoyaltyProgramProviderBase.c
7  --- a/Farmax/Src/POS/BusinessLogic.LoyaltyProgram.Provider/LoyaltyProgramProviderBase.cs      (r
8  +++ b/Farmax/Src/POS/BusinessLogic.LoyaltyProgram.Provider/LoyaltyProgramProviderBase.cs      (d
9  @@ -378,7 +378,42 @@
10
11         return result;
12     }
13 +
14 +     /// <summary>
15 +     /// Guarda las leyendas de origis para ser usadas en la reimpression
16 +     /// </summary>
17 +     /// <param name="origis">información de respuestas de origis</param>
18 +     /// <returns>retorna True si fue exitoso</returns>
19 +     protected ResultStruct<bool> AddRespuestaOrigis(venRespuestaOrigis origis)
20 +     {
21 +         ResultStruct<bool> result;
22 +         try
23 +         {
24 +             using (var context = new POSDBLoyaltyProgramContext())
25 +             {

```

```

26 +         var dao = new LoyaltyProgramDao(context);
27 +         dao.AddRespuestaOrigis(origis);
28 +     }
29
30 +         result = new ResultStruct<bool>(true);
31 +     }
32 +     catch (Exception e)
33 +     {
34 +         result = new ResultStruct<bool>(false);
35 +         result.AddErrorItem(
36 +             new ErrorItem
37 +             {
38 +                 Code = e.HResult.ToString(),
39 +                 Description = e.Message
40 +             });
41 +
42 +         Log.WriteException(SubModule, e, "LoyaltyProgramProviderBase - AddRespuestaOr
43 +         Log.WriteError(SubModule, string.Format("Error al guardar respuesta origis: v
44 +     }
45 +
46 +     return result;
47 + }
48 +
49 +     /// <summary>
50 +     /// Metodo que agrega las configuraciones del precioBase al cache
51 +     /// </summary>

```

Patch sobre la clase LoyaltyProgramDao.cs

```

1  Index: Farmax/Src/POS/Data.LoyaltyProgram/DAO/LoyaltyProgramDao.cs
2  IDEA additional info:
3  Subsystem: com.intellij.openapi.diff.impl.patch.CharsetEP
4  <+>UTF-8
5  =====
6  diff --git a/Farmax/Src/POS/Data.LoyaltyProgram/DAO/LoyaltyProgramDao.cs b/Farmax/Src/POS/Data
7  --- a/Farmax/Src/POS/Data.LoyaltyProgram/DAO/LoyaltyProgramDao.cs      (revision 984eae8257aadb25
8  +++ b/Farmax/Src/POS/Data.LoyaltyProgram/DAO/LoyaltyProgramDao.cs      (date 1785302974872)
9  @@ -48,6 +48,16 @@
10         this.DataContext.SaveChanges();
11     }
12
13 +     /// <summary>
14 +     /// Guarda las respuestas de origis
15 +     /// </summary>
16 +     /// <param name="origis">objeto con las respuestas de origis</param>
17 +     public void AddRespuestaOrigis(venRespuestaOrigis origis)
18 +     {
19 +         this.DataContext.venRespuestaOrigis.Add(origis);
20 +         this.DataContext.SaveChanges();

```

```

21 +      }
22 +
23     /// <summary>
24     /// Guarda el log de vinculaciones Mda
25     /// </summary>

```

Patch sobre la clase OrigisManager.cs

```

1  Index: Farmax/Src/POS/BusinessLogic.LoyaltyProgram.Origis/OrigisManager.cs
2  IDEA additional info:
3  Subsystem: com.intellij.openapi.diff.impl.patch.CharsetEP
4  <+>UTF-8
5  =====
6  diff --git a/Farmax/Src/POS/BusinessLogic.LoyaltyProgram.Origis/OrigisManager.cs b/Farmax/Src/
7  --- a/Farmax/Src/POS/BusinessLogic.LoyaltyProgram.Origis/OrigisManager.cs      (revision 984eae82
8  +++ b/Farmax/Src/POS/BusinessLogic.LoyaltyProgram.Origis/OrigisManager.cs      (date 178537924844
9  @@ -604,6 +604,15 @@
10
11         VersionXml = 2 // Se establece la versión 2 por cambio en el obje
12     });
13 +
14 +    this.AddRespuestaOrigis(new venRespuestaOrigis
15 +    {
16 +        folioTransaccion = benefits.FolioTransaction,
17 +        codigoSucursal = Convert.ToInt32(benefits.Branch),
18 +        numeroCaja = Convert.ToInt32(benefits.TillId),
19 +        codigoCajero = benefits.UserId,
20 +        datosEnviados = MdaDataParsers.SendDataParser(trans, MdaProgramCo
21 +        datosRecibidos = string.Join(",", accountResult.Ticketlegends.Sel
22 +    });
23 +
24     return new ResultStruct<LoyaltyBenefits>(accountResult);

```

Patch sobre la clase TransactionService.cs

```

1  Index: Farmax/Src/POS/Service.Transaction/TransactionService.cs
2  IDEA additional info:
3  Subsystem: com.intellij.openapi.diff.impl.patch.CharsetEP
4  <+>UTF-8
5  =====
6  diff --git a/Farmax/Src/POS/Service.Transaction/TransactionService.cs b/Farmax/Src/POS/Service
7  --- a/Farmax/Src/POS/Service.Transaction/TransactionService.cs      (revision 984eae8257aadb2575d3
8  +++ b/Farmax/Src/POS/Service.Transaction/TransactionService.cs      (date 1785317831925)
9  @@ -1824,6 +1824,7 @@
10
11         LoyaltyAccountID = l.Card,
12         LoyaltyProgramID = l.LoyaltyProgramID.ToString("G"),
13         AuthorizationCode = l.AuthorizationCode,
14 +        Ticketlegends = l.Ticketlegends.ToArray(),

```

```

14 Response = saleTransaction.MessageRecordLoyalty,
15
16 // Contendra si contiene o no una venta por programa de lealtad

```

Patch sobre la clase TransactionServiceClient.cs

```

1 Index: Farmax/Src/POS/Win.Sale/Services/TransactionServiceClient.cs
2 IDEA additional info:
3 Subsystem: com.intellij.openapi.diff.impl.patch.CharsetEP
4 <+>UTF-8
5 =====
6 diff --git a/Farmax/Src/POS/Win.Sale/Services/TransactionServiceClient.cs b/Farmax/Src/POS/Win
7 --- a/Farmax/Src/POS/Win.Sale/Services/TransactionServiceClient.cs (revision 984eae8257aadb25
8 +++ b/Farmax/Src/POS/Win.Sale/Services/TransactionServiceClient.cs (date 1785357152703)
9 @@ -2139,7 +2139,10 @@
10
11         PointsEarned = loyaltyProgramOrigins.Points.Where(x => x.Type
12         MessageRecordPromotion = loyaltyProgramOrigins.Response,
13         IdTransaction = loyaltyProgramOrigins.SessionID,
14 -         MessageAcomulationPoint = loyaltyProgramOrigins.Response == P
15 +         MessageAcomulationPoint = loyaltyProgramOrigins.Response == P
16 +         LegendWallet1 = loyaltyProgramOrigins.TicketLegends[0],
17 +         LegendWallet2 = loyaltyProgramOrigins.TicketLegends[1],
18 +         LegendWallet3 = loyaltyProgramOrigins.TicketLegends[2]
19     };
20
21     saleModel.TotalEWalletBalance = loyaltyProgramOrigins.Response !=

```

Patch para MODEL_VEN.xml

```

1 Index: AnxoFARMAX/DEV_SyncConfiguration/MODEL_VEN.xml
2 IDEA additional info:
3 Subsystem: com.intellij.openapi.diff.impl.patch.CharsetEP
4 <+>UTF-8
5 =====
6 diff --git a/AnxoFARMAX/DEV_SyncConfiguration/MODEL_VEN.xml b/AnxoFARMAX/DEV_SyncConfiguration
7 --- a/AnxoFARMAX/DEV_SyncConfiguration/MODEL_VEN.xml (revision 984eae8257aadb2575d3f79edd05
8 +++ b/AnxoFARMAX/DEV_SyncConfiguration/MODEL_VEN.xml (date 1785296763380)
9 @@ -2022,8 +2022,40 @@
10     </indexes>
11
12     </table>
13 -
14 +
15 +     <table name="VENRESPUESTAORIGIS" real="venRespuestaOrigis" type="model">
16 +
17 +         <!--PRIMARY KEYS-->
18 +
19 +         <column name="IDRESPUESTA" real="idRespuesta" type="guid" flag="primary_key" de

```

```

20 +
21 +
22 +
23 +
24 +
25 +
26 +
27 +
28 +
29 +
30 +
31 +
32 +
33 +
34 +
35 +
36 +
37 +
38 +
39 +
40 +
41 +
42 +
43 +
44 +
45 +
46 +
47 +
48 +
49 +
50 +

```

```

<!--COLUMNS-->

<column name="FOLIOTRANSACCION" real="folioTransaccion" type="string" stringlen
<column name="CODIGOSUCURSAL" real="codigoSucursal" type="int32" flag="not_null
<column name="NUMEROCAJA" real="numeroCaja" type="int32" flag="not_null"/>
<column name="CODIGOCAJERO" real="codigoCajero" type="int32" flag="not_null" />
<column name="DATOSENVIADOS" real="datosEnviados" stringlength="1000" type="str
<column name="DATOSRECIBIDOS" real="datosRecibidos" stringlength="1500" type="s

<!-- CAMPOS DE RASTREO DE SINCRONIZACION ANXO -->

<column name="SYNC_CREADO_FECHA" real="syncCreadoFecha" type="datetime" flag="t
<column name="SYNC_ACTUALIZADO_FECHA" real="syncActualizadoFecha" type="datetim
<column name="SYNC_ACTUALIZADO_POR" real="syncActualizadoPor" type="string" str
<column name="SYNC_VERSION_REGISTRO" real="syncVersionRegistro" type="int32" fl
<column name="SYNC_TRANSFERIDO" real="syncTransferido" type="string" stringleng

<!--INDEXES-->
<indexes>
  <index name="VENRESPUESTAORIGIS_IDX01" type="nonunique">
    <column name="CODIGOSUCURSAL" />
  </index>
</indexes>

</table>

</model>

</models>

```

Patch para el proyecto DEV_DbCommon.csproj

```

1 Index: AnxoFARMAX/DEV_DbCommon/DEV_DbCommon.csproj
2 IDEA additional info:
3 Subsystem: com.intellij.openapi.diff.impl.patch.CharsetEP
4 <+>UTF-8
5 =====
6 diff --git a/AnxoFARMAX/DEV_DbCommon/DEV_DbCommon.csproj b/AnxoFARMAX/DEV_DbCommon/DEV_DbCommon.csproj
7 --- a/AnxoFARMAX/DEV_DbCommon/DEV_DbCommon.csproj      (revision 984eae8257aadb2575d3f79edd05e438)
8 +++ b/AnxoFARMAX/DEV_DbCommon/DEV_DbCommon.csproj      (date 1785213296615)
9 @@ -542,6 +542,7 @@
10    <Compile Include="VENFECHAOPERACION.cs" />
11    <Compile Include="VENPAGOSERVICIOSBITACORAREG.cs" />
12    <Compile Include="VENPAGOSERVICIOSREG.cs" />
13 +  <Compile Include="VENRESPUESTAORIGIS.cs" />
14    <Compile Include="VENRETIROSDENOMINACIONESREG.cs" />
15    <Compile Include="VENRETIROSDOCUMENTOSREG.cs" />

```

Patch para el LoyaltyProgramModel

```

1 Index: Farmax/Src/POS/Data.LoyaltyProgram/LoyaltyProgramModel.Context.cs
2 IDEA additional info:
3 Subsystem: com.intellij.openapi.diff.impl.patch.CharsetEP
4 <+>UTF-8
5 =====
6 diff --git a/Farmax/Src/POS/Data.LoyaltyProgram/LoyaltyProgramModel.Context.cs b/Farmax/Src/P
7 --- a/Farmax/Src/POS/Data.LoyaltyProgram/LoyaltyProgramModel.Context.cs (revision 984eae8257a
8 +++ b/Farmax/Src/POS/Data.LoyaltyProgram/LoyaltyProgramModel.Context.cs (date 1785425751326)
9 @@ -23,6 +23,12 @@
10      : base(EntityConnectionCreator.DefaultInstance.CreateEntityConnection("LoyaltyPr
11      {
12      }
13 +
14 +     public POSDBLoyaltyProgramContext(EntityConnectionCreator.ContextTypes context)
15 +     : base(EntityConnectionCreator.DefaultInstance.CreateEntityConnection("LoyaltyPr
16 +     {
17 +         this.Configuration.LazyLoadingEnabled = false;
18 +     }
19
20     protected override void OnModelCreating(DbModelBuilder modelBuilder)
21     {
22 @@ -33,6 +39,7 @@
23     public virtual DbSet<genProgramasLealtadCat> genProgramasLealtadCat { get; set; }
24     public virtual DbSet<genEquivalenciaFormaDePago> genEquivalenciaFormaDePago { get; s
25     public virtual DbSet<genBitacoraMonedero> genBitacoraMonedero { get; set; }
26 +     public virtual DbSet<venRespuestaOrigis> venRespuestaOrigis { get; set; }
27     public virtual DbSet<genVinculacionesMDABitacoraReg> genVinculacionesMDABitacoraReg
28     }
29 }
30 Index: Farmax/Src/POS/Data.LoyaltyProgram/LoyaltyProgramModel.edmx.diagram
31 IDEA additional info:
32 Subsystem: com.intellij.openapi.diff.impl.patch.CharsetEP
33 <+>UTF-8
34 =====
35 diff --git a/Farmax/Src/POS/Data.LoyaltyProgram/LoyaltyProgramModel.edmx.diagram b/Farmax/Src
36 --- a/Farmax/Src/POS/Data.LoyaltyProgram/LoyaltyProgramModel.edmx.diagram (revision 984eae8
37 +++ b/Farmax/Src/POS/Data.LoyaltyProgram/LoyaltyProgramModel.edmx.diagram (date 17853026976
38 @@ -9,6 +9,7 @@
39     <EntityTypeShape EntityType="POSDBModel.genProgramasLealtadCat" Width="1.5" PointX="
40     <EntityTypeShape EntityType="POSDBModel.genEquivalenciaFormaDePago" Width="3.5" Poin
41     <EntityTypeShape EntityType="POSDBModel.genVinculacionesMDABitacoraReg" Width="1.5"
42 +     <EntityTypeShape EntityType="POSDBModel.venRespuestaOrigis" Width="1.75" PointX="5.6
43     </Diagram>
44     </edmx:Diagrams>
45     </edmx:Designer>

```

```

46 Index: Farmax/Src/POS/Data.LoyaltyProgram/LoyaltyProgramModel.edmx
47 IDEA additional info:
48 Subsystem: com.intellij.openapi.diff.impl.patch.CharsetEP
49 <+>UTF-8
50 =====
51 diff --git a/Farmax/Src/POS/Data.LoyaltyProgram/LoyaltyProgramModel.edmx b/Farmax/Src/POS/Data
52 --- a/Farmax/Src/POS/Data.LoyaltyProgram/LoyaltyProgramModel.edmx      (revision 984eae8257aadb2
53 +++ b/Farmax/Src/POS/Data.LoyaltyProgram/LoyaltyProgramModel.edmx      (date 1785302697631)
54 @@ -32,6 +32,18 @@
55         <Property Name="fechaTransaccion" Type="datetime" Nullable="false" />
56         <Property Name="versionCentral" Type="int" Nullable="false" />
57     </EntityType>
58 +    <EntityType Name="venRespuestaOrigis">
59 +        <Key>
60 +            <PropertyRef Name="idRespuesta" />
61 +        </Key>
62 +        <Property Name="idRespuesta" Type="uniqueidentifier" StoreGeneratedPattern="Ident
63 +        <Property Name="folioTransaccion" Type="nvarchar" MaxLength="15" Nullable="false"
64 +        <Property Name="codigoSucursal" Type="int" Nullable="false" />
65 +        <Property Name="numeroCaja" Type="int" Nullable="false" />
66 +        <Property Name="codigoCajero" Type="int" Nullable="false" />
67 +        <Property Name="datosEnviados" Type="nvarchar" MaxLength="1000" Nullable="false" /
68 +        <Property Name="datosRecibidos" Type="nvarchar" MaxLength="1500" />
69 +    </EntityType>
70     <EntityType Name="genProgramaLealtadBitacoraReg">
71         <Key>
72             <PropertyRef Name="idBitacora" />
73 @@ -87,6 +99,7 @@
74         <EntitySet Name="genProgramaLealtadBitacoraReg" EntityType="Self.genProgramaLealta
75         <EntitySet Name="genProgramasLealtadCat" EntityType="Self.genProgramasLealtadCat"
76         <EntitySet Name="genBitacoraMonedero" EntityType="Self.genBitacoraMonedero" Schema
77 +        <EntitySet Name="venRespuestaOrigis" EntityType="Self.venRespuestaOrigis" Schema="
78         <EntitySet Name="genVinculacionesMDABitacoraReg" EntityType="Self.genVinculaciones
79     </EntityContainer>
80     </Schema></edmx:StorageModels>
81 @@ -98,6 +111,7 @@
82         <EntitySet Name="genProgramasLealtadCat" EntityType="POSDBModel.genProgramasLealta
83         <EntitySet Name="genEquivalenciaFormaDePago" EntityType="POSDBModel.genEquivalenci
84         <EntitySet Name="genBitacoraMonedero" EntityType="POSDBModel.genBitacoraMonedero"
85 +        <EntitySet Name="venRespuestaOrigis" EntityType="POSDBModel.venRespuestaOrigis" />
86         <EntitySet Name="genVinculacionesMDABitacoraReg" EntityType="POSDBModel.genVincula
87     </EntityContainer>
88     <EntityType Name="genProgramaLealtadBitacoraReg">
89 @@ -141,6 +155,18 @@
90         <Property Name="fechaTransaccion" Type="DateTime" Nullable="false" Precision="3" /
91         <Property Name="versionCentral" Type="Int32" Nullable="false" />
92     </EntityType>
93 +    <EntityType Name="venRespuestaOrigis">
94 +        <Key>
95 +            <PropertyRef Name="idRespuesta" />

```

```

96 +         </Key>
97 +         <Property Name="idRespuesta" Type="Guid" Nullable="false" annotation:StoreGenerate
98 +         <Property Name="folioTransaccion" Type="String" Nullable="false" MaxLength="15" Fi
99 +         <Property Name="codigoSucursal" Type="Int32" Nullable="false" />
100 +         <Property Name="numeroCaja" Type="Int32" Nullable="false" />
101 +         <Property Name="codigoCajero" Type="Int32" Nullable="false" />
102 +         <Property Name="datosEnviados" Type="String" Nullable="false" MaxLength="1000" Fix
103 +         <Property Name="datosRecibidos" Type="String" MaxLength="1500" FixedLength="false"
104 +     </EntityType>
105     <EntityType Name="genProgramasLealtadCat">
106         <Key>
107             <PropertyRef Name="codigoProgramaLealtad" />
108 @@ -223,6 +249,19 @@
109             <ScalarProperty Name="codigoSucursal" ColumnName="codigoSucursal" />
110             <ScalarProperty Name="folioTransaccion" ColumnName="folioTransaccion" />
111             <ScalarProperty Name="idBitacora" ColumnName="idBitacora" />
112 +         </MappingFragment>
113 +     </EntityTypeMapping>
114 + </EntitySetMapping>
115 + <EntitySetMapping Name="venRespuestaOrigis">
116 +     <EntityTypeMapping TypeName="POSSDBModel.venRespuestaOrigis">
117 +         <MappingFragment StoreEntitySet="venRespuestaOrigis">
118 +             <ScalarProperty Name="datosRecibidos" ColumnName="datosRecibidos" />
119 +             <ScalarProperty Name="datosEnviados" ColumnName="datosEnviados" />
120 +             <ScalarProperty Name="codigoCajero" ColumnName="codigoCajero" />
121 +             <ScalarProperty Name="numeroCaja" ColumnName="numeroCaja" />
122 +             <ScalarProperty Name="codigoSucursal" ColumnName="codigoSucursal" />
123 +             <ScalarProperty Name="folioTransaccion" ColumnName="folioTransaccion" />
124 +             <ScalarProperty Name="idRespuesta" ColumnName="idRespuesta" />
125         </MappingFragment>
126     </EntityTypeMapping>
127 </EntitySetMapping>

```

Patch para POSSaleModel

```

1 Index: Farmax/Src/POS/Data.Sale/POSSaleModel.Context.cs
2 IDEA additional info:
3 Subsystem: com.intellij.openapi.diff.impl.patch.CharsetEP
4 <+>UTF-8
5 =====
6 diff --git a/Farmax/Src/POS/Data.Sale/POSSaleModel.Context.cs b/Farmax/Src/POS/Data.Sale/POSS
7 --- a/Farmax/Src/POS/Data.Sale/POSSaleModel.Context.cs (revision 984eae8257aadb2575d3f79edd0
8 +++ b/Farmax/Src/POS/Data.Sale/POSSaleModel.Context.cs (date 1785377851315)
9 @@ -81,5 +81,6 @@
10     public virtual DbSet<genCFDIObjetoImpuesto> genCFDIObjetoImpuesto { get; set; }
11     public virtual DbSet<genCFDITipoExportacion> genCFDITipoExportacion { get; set; }
12     public virtual DbSet<sadOrdenFormasPago> sadOrdenFormasPago { get; set; }
13 +    public virtual DbSet<venRespuestaOrigis> venRespuestaOrigis { get; set; }
14     }

```

```

15 }
16 Index: Farmax/Src/POS/Data.Sale/POSSaleModel.edmx.diagram
17 IDEA additional info:
18 Subsystem: com.intellij.openapi.diff.impl.patch.CharsetEP
19 <+>UTF-8
20 =====
21 diff --git a/Farmax/Src/POS/Data.Sale/POSSaleModel.edmx.diagram b/Farmax/Src/POS/Data.Sale/PO
22 --- a/Farmax/Src/POS/Data.Sale/POSSaleModel.edmx.diagram      (revision 984eae8257aadb2575d3f79
23 +++ b/Farmax/Src/POS/Data.Sale/POSSaleModel.edmx.diagram      (date 1785392679730)
24 @@ -94,6 +94,7 @@
25         <EntityTypeShape EntityType="POSDbModel.sadRegistroLiquidacion" Width="1.5" PointX="
26         <EntityTypeShape EntityType="POSDbModel.sadOrdenArticulosDespachados" Width="2.5" Po
27         <EntityTypeShape EntityType="POSDbModel.sadOrdenFormasPago" Width="1.5" PointX="13.3
28 +         <EntityTypeShape EntityType="POSDbModel.venRespuestaOrigis" Width="1.5" PointX="20.3
29         </Diagram>
30     </edmx:Diagrams>
31 </edmx:Designer>
32 Index: Farmax/Src/POS/Data.Sale/POSSaleModel.edmx
33 IDEA additional info:
34 Subsystem: com.intellij.openapi.diff.impl.patch.CharsetEP
35 <+>UTF-8
36 =====
37 diff --git a/Farmax/Src/POS/Data.Sale/POSSaleModel.edmx b/Farmax/Src/POS/Data.Sale/POSSaleMod
38 --- a/Farmax/Src/POS/Data.Sale/POSSaleModel.edmx      (revision 984eae8257aadb2575d3f79edd05e43
39 +++ b/Farmax/Src/POS/Data.Sale/POSSaleModel.edmx      (date 1785392601280)
40 @@ -763,6 +763,23 @@
41         <Property Name="idTransaccion" Type="uniqueidentifier" Nullable="false" />
42         <Property Name="codigoValesEmitidos" Type="uniqueidentifier" Nullable="false" />
43     </EntityType>
44 +    <EntityType Name="venRespuestaOrigis">
45 +        <Key>
46 +            <PropertyRef Name="idRespuesta" />
47 +        </Key>
48 +        <Property Name="idRespuesta" Type="uniqueidentifier" Nullable="false" />
49 +        <Property Name="folioTransaccion" Type="nvarchar" MaxLength="15" Nullable="false"
50 +        <Property Name="codigoSucursal" Type="int" Nullable="false" />
51 +        <Property Name="numeroCaja" Type="int" Nullable="false" />
52 +        <Property Name="codigoCajero" Type="int" Nullable="false" />
53 +        <Property Name="datosEnviados" Type="nvarchar" MaxLength="1000" Nullable="false" /
54 +        <Property Name="datosRecibidos" Type="nvarchar" MaxLength="1500" />
55 +        <Property Name="syncCreadoFecha" Type="datetime" Nullable="false" />
56 +        <Property Name="syncActualizadoFecha" Type="datetime" Nullable="false" />
57 +        <Property Name="syncActualizadoPor" Type="nvarchar" MaxLength="100" Nullable="fals
58 +        <Property Name="syncVersionRegistro" Type="int" Nullable="false" />
59 +        <Property Name="syncTransferido" Type="nchar" MaxLength="1" Nullable="false" />
60 +    </EntityType>
61     <EntityType Name="venTurnosCajaReg">
62         <Key>
63             <PropertyRef Name="codigoTurno" />
64 @@ -1457,6 +1474,7 @@

```

```

65         <EntitySet Name="venDevolucionesCab" EntityType="Self.venDevolucionesCab" Schema="
66         <EntitySet Name="venDevolucionesDet" EntityType="Self.venDevolucionesDet" Schema="
67         <EntitySet Name="venEmisionValeTransaccion" EntityType="Self.venEmisionValeTransac
68 +         <EntitySet Name="venRespuestaOrigis" EntityType="Self.venRespuestaOrigis" Schema="
69         <EntitySet Name="venTurnosCajaReg" EntityType="Self.venTurnosCajaReg" Schema="dbo"
70         <EntitySet Name="venValeEmitidoCanjeParcial" EntityType="Self.venValeEmitidoCanjeP
71         <EntitySet Name="venValeEmitidoDevolucion" EntityType="Self.venValeEmitidoDevoluci
72 @@ -1861,6 +1879,7 @@
73         <EntitySet Name="genCFDIObjetoImpuesto" EntityType="POSDBModel.genCFDIObjetoImpues
74         <EntitySet Name="genCFDITipoExportacion" EntityType="POSDBModel.genCFDITipoExporta
75         <EntitySet Name="sadOrdenFormasPago" EntityType="POSDBModel.sadOrdenFormasPago" />
76 +         <EntitySet Name="venRespuestaOrigis" EntityType="POSDBModel.venRespuestaOrigis" />
77     </EntityContainer>
78     <EntityType Name="cnfTicketsCat">
79         <Key>
80 @@ -3244,7 +3263,24 @@
81         <Property Name="codigoSucursal" Type="Int32" Nullable="false" />
82         <Property Name="referenciaTarjeta" Type="String" MaxLength="256" FixedLength="fals
83         <Property Name="puntos" Type="Decimal" Precision="18" Scale="4" />
84 -     </EntityType>
85 +     </EntityType>
86 +     <EntityType Name="venRespuestaOrigis">
87 +         <Key>
88 +             <PropertyRef Name="idRespuesta" />
89 +         </Key>
90 +         <Property Name="idRespuesta" Type="Guid" Nullable="false" />
91 +         <Property Name="folioTransaccion" Type="String" Nullable="false" MaxLength="15" Fi
92 +         <Property Name="codigoSucursal" Type="Int32" Nullable="false" />
93 +         <Property Name="numeroCaja" Type="Int32" Nullable="false" />
94 +         <Property Name="codigoCajero" Type="Int32" Nullable="false" />
95 +         <Property Name="datosEnviados" Type="String" Nullable="false" MaxLength="1000" Fix
96 +         <Property Name="datosRecibidos" Type="String" MaxLength="1500" FixedLength="false"
97 +         <Property Name="syncCreadoFecha" Type="DateTime" Nullable="false" Precision="3" />
98 +         <Property Name="syncActualizadoFecha" Type="DateTime" Nullable="false" Precision="
99 +         <Property Name="syncActualizadoPor" Type="String" Nullable="false" MaxLength="100"
100 +         <Property Name="syncVersionRegistro" Type="Int32" Nullable="false" />
101 +         <Property Name="syncTransferido" Type="String" Nullable="false" MaxLength="1" Fixe
102 +     </EntityType>
103 </Schema>
104 </edmx:ConceptualModels>
105 <!-- C-S mapping content -->
106 @@ -4124,6 +4160,24 @@
107         <ScalarProperty Name="pagoTipoTarjeta" ColumnName="pagoTipoTarjeta" />
108         <ScalarProperty Name="codigoFormaPago" ColumnName="codigoFormaPago" />
109         <ScalarProperty Name="sadOrdenId" ColumnName="sadOrdenId" />
110 +     </MappingFragment>
111 + </EntityTypeMapping>
112 + </EntitySetMapping>
113 + <EntitySetMapping Name="venRespuestaOrigis">
114 +     <EntityTypeMapping TypeName="POSDBModel.venRespuestaOrigis">

```

```

115 +         <MappingFragment StoreEntitySet="venRespuestaOrigis">
116 +             <ScalarProperty Name="syncTransferido" ColumnName="syncTransferido" />
117 +             <ScalarProperty Name="syncVersionRegistro" ColumnName="syncVersionRegistro" />
118 +             <ScalarProperty Name="syncActualizadoPor" ColumnName="syncActualizadoPor" />
119 +             <ScalarProperty Name="syncActualizadoFecha" ColumnName="syncActualizadoFecha" />
120 +             <ScalarProperty Name="syncCreadoFecha" ColumnName="syncCreadoFecha" />
121 +             <ScalarProperty Name="datosRecibidos" ColumnName="datosRecibidos" />
122 +             <ScalarProperty Name="datosEnviados" ColumnName="datosEnviados" />
123 +             <ScalarProperty Name="codigoCajero" ColumnName="codigoCajero" />
124 +             <ScalarProperty Name="numeroCaja" ColumnName="numeroCaja" />
125 +             <ScalarProperty Name="codigoSucursal" ColumnName="codigoSucursal" />
126 +             <ScalarProperty Name="folioTransaccion" ColumnName="folioTransaccion" />
127 +             <ScalarProperty Name="idRespuesta" ColumnName="idRespuesta" />
128 +         </MappingFragment>
129 +     </EntityTypeMapping>
130 </EntitySetMapping>

```

Además, se deben contemplar las clases nuevas VENRESPUESTAORIGIS en el proyecto DEV_DbCommon y venRespuestaOrigis para BusinessLogic.LoyaltyProgram.DTO y BusinessLogic.Sale.DTO.

5.- Impacto

La solución implica una propuesta más robusta y escalable a lo largo del tiempo; sin embargo, es la solución que implica más consideraciones, ya que impacta directamente sobre el modelo, la base de datos y la sincronización de información. Esto evitará la dependencia de la conexión en *POSDB_OR* mediante webCentrales para la obtención de las leyendas.

6.- Conclusión

Tras el análisis de los incidentes **75 (FO-295)** y **76 (FO-299)**, se identificaron tres diferencias entre el ticket original y su reimpresión: el número de tarjeta, las leyendas de *ORIGIS* y el código QR.

Se determinó que el **número de tarjeta incorrecto** se debe a un doble enmascaramiento en el método *SaleManager.GetTransaccion*, donde el número de tarjeta ya enmascarado recibe un formato adicional erróneo al agregar el carácter “|” y el *keyversion*. La solución consiste en eliminar dicha condición para pasar directamente el número ya enmascarado obtenido de *TenderManager.GetElectronicTenderByID*.

Respecto a las **leyendas faltantes de ORIGIS**, la causa raíz es que estas solo se consultan durante la venta inicial y no se persisten para su uso en reimpresiones. La solución consiste en agregar una nueva tabla *venRespuestaOrigis*, almacenando las leyendas durante el flujo de venta y eliminando la dependencia de *POSDB_OR* vía *webCentrales* en la reimpresión. Esta solución implica cambios en el modelo *MODEL_VEN.xml*, la actualización de la caja y las consultas de *EntityFramework*.

En cuanto al **código QR distinto**, se concluyó que es el comportamiento esperado de la herramienta generadora, que devuelve un código diferente en cada petición, por lo que no se considera un defecto.