

Causa raíz Forma de pago 0 en commitTrans hacia Origis

Autor: Huilver Nolasco Aguilar

Fecha: Marzo del 2025

Contenido

Descripción general.....	3
Causa.....	3
Solución.....	6

Descripción general

Se detectó que para la venta V10982000088983 que corresponde al pedido 7007373420APP se está mandando en el método de commitTrans hacia origis la forma de pago 0

Causa

Primero se revisan los datos generales de la venta, la orden y sus formas de pago.

```
--consulta de cabecera de venta
select idTransaccion, fechaTransaccion, sadOrdenId from trnTransaccionesCab where folioTransaccion='V10982000088983';
```

idTransaccion	fechaTransaccion	sadOrdenId
596E48EF-5EE2-EF11-95E0-10E7C6374B88	2025-02-03 12:44:48.247	FF719F1B-87CF-4FC4-892B-83CE310D0617

```
--cabecera de orden asociado a la venta
select sadOrdenId, ordenFolio, codigoFormaPago, monedero from sadOrdenes
where sadOrdenId = (select sadOrdenId from trnTransaccionesCab where folioTransaccion='V10982000088983');
```

sadOrdenId	ordenFolio	codigoFormaPago	monedero
FF719F1B-87CF-4FC4-892B-83CE310D0617	7007373420APP	41	9690000002657

```
--formas de pago de la orden
select sadOrdenId, fp.codigoFormaPago, fpcat.descripcion, fpcat.tipoFormaPago, monedero, referenciaTarjeta from sadCentralOrdenFormasPago fp
join genFormasDePagoCat fpcat on fp.codigoFormaPago=fpcat.codigoFormaPago
where fp.sadOrdenId=(select sadOrdenId from trnTransaccionesCab where folioTransaccion='V10982000088983');
```

sadOrdenId	codigoFormaPago	descripcion	tipoFormaPago	monedero	referenciaTarjeta
FF719F1B-87CF-4FC4-892B-83CE310D0617	41	PAYPAL	SAD3		

En la que se observa que la orden fue pagada por **PAYPAL** y el tipo asociado a esa forma de pago es **SAD3**

Despues vemos el log del monedero del método commitTrans, en donde se observa que efectivamente a origis se mando la forma de pago 0

```
224 [2025-02-03 12:42:47:236], FARMAX_USC, MONEDERO, INFO, Request: <s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
225 <s:Header>
226 <Action s:mustUnderstand="1" xmlns="http://schemas.microsoft.com/ws/2005/05/addressing/none">https://pos.monederodelahorro.net/online2/commitTrans</Action>
227 </s:Header>
228 <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
229 <commitTrans xmlns="https://pos.monederodelahorro.net/online2/">
230 <Tarjeta>9690000002657</Tarjeta>
231 <SucId>1098</SucId>
232 <CsJaid>200</CsJaid>
233 <TerminalId>175327</TerminalId>
234 <SessionId>14352336926</SessionId>
235 <Articulos>7501298209067,1,747,747|0650240057946,2,22.90002400,45.80004800|7506472801310,1,10.0000,10.0000</Articulos>
236 <NumTicket>V10982000088983</NumTicket>
237 <SaldoRedimido>0</SaldoRedimido>
238 <ImporteTotal>802.8000</ImporteTotal>
239 <FechaOperacion>20250203</FechaOperacion>
240 <EntregaDomicilio>2</EntregaDomicilio>
241 <FormaPago>0</FormaPago>
242 <AuthCode>098c9ff72fb6db0843d61e81bcbd2bd</AuthCode>
243 </commitTrans>
244 </s:Body>
245 </s:Envelope>
```

En términos generales lo que ocurre es que a origis se le manda un *sFormaPago* mediante una equivalencia de tipo forma de pago, dicha equivalencia se obtiene de la siguiente tabla

499	=	
500	✓	<code>select * from genEquivalenciaFormaDePago;</code>
501		

TENDER_ID_HOME_DELIVERY_PAYPAL = 41; 7	POSDB_OR.dbo.genFormasDePagoCat	POSD
--	---------------------------------	------

formapago	equivalenciaFormaPago
1 AMEX	3
2 CODI	7
3 CPAY	9
4 CRED	5
5 EFEC	1
6 MONE	6
7 PMOV	6
8 TCRE	3
9 TDEB	3
10 TPVI	3
11 TREG	3
12 VALE	4
13 WALL	8

En donde se puede observar que el tipo SAD3 no está considerado en estas equivalencias

Por otro lado cuando se obtienen estas equivalencias se hace mediante un método que da un elemento de mapeo por defecto cuando no está presente alguna entrada en la lista de equivalencias, por lo que al ejecutar `payments.FirstOrDefault().Value` regresa **0**

```

C# OrigisManager x C# LoyaltyProgramManager C# LoyaltyProgramService C# LoyaltySession C# LoyaltyBenefits C# FactoryIn
C# NovartisManager C# Reference C# ILoyaltyProgram C# LoyaltyProgramDao C# TransactionService.partial Web C
> Q- GetPaymentMethod x Cc W .* 1/1 ↑ ↓ 🔍 :
1411 private string GetCodePaymentMethod(List<string> payment)
1412 {
1413     try
1414     {
1415         using (var context = new POSDBLoyaltyProgramContext())
1416         {
1417             var dao = new LoyaltyProgramDao(context);
1418             paymetMethod = dao.GetPaymentMethod();
1419         }
1420     }
1421     ...
1422     var codePayment = new List<string>();
1423     payment.ForEach(c =>
1424     {
1425         var payments :List<KeyValuePair<string,int>> = paymetMethod.Where(x :KeyValuePair<string,int> => x.Key == c).ToList();
1426         if (payments != null)
1427         {
1428             codePayment.Add(payments.FirstOrDefault().Value.ToString());
1429         }
1430     });
1431     return string.Join(",", codePayment);
1432 }

```

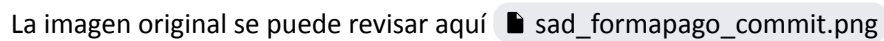
Y al hacer el commitTrans termina mandandose 0

```

C# OrigisManager x C# LoyaltyProgramManager C# LoyaltyProgramService C# LoyaltyBenefits C# FactoryInstance C# NovartisManager C# Reference C# ILoyaltyProg
Web C# Enumerable C# KeyValuePair'2 C# String
416 /// <returns>Regresa la información de la tarjeta validada en el objeto <see cref="LoyaltyTransaction"/></returns>
417 public override async Task<ResultStruct<LoyaltyBenefits>> Commit(LoyaltyTransaction benefits)
418 {
419     try
420     {
421         using (var serviceClient = new Monedero_del_AhorroSoapClient(endpointConfigurationName: "Monedero_del_AhorroSoap", checkConnection: false, enableOrigisLog: true))
422         {
423             var accountResult = new LoyaltyBenefits();
424             // se verifica si la transaccion ya fue procesada, si ya fue procesada se regresa el resultado obtenido por proveedor
425             var checkIfExists :ResultStruct<List<string>> = this.GetAproveLogLoyaltyProgram(new CancelTokens { BusinessId = Convert.ToInt32(benefits.Branch), tillId = Convert.ToInt32(benefits.TillId) });
426             if (checkIfExists.Result.Any(t :igenProgramaLealtadBitacoraReg => t.tipoPrograma == this.ConvertEnumLoyaltyToDBLoyalty(benefits.LoyaltyProgramID))) {...}
427             else
428             {
429                 if (benefits.ShoppingBasket.Any(p :LoyaltyItem => p.IsRejected || p.LoyaltyProgramID == EnumLoyaltyProgram.Origis)) {...}
430                 // En caso de no traer metodos de pago se agrega el default 1
431                 benefits.PaymentMethods = benefits.PaymentMethods == null ? new List<string> { "1" } : benefits.PaymentMethods;
432                 this.MapPropertyts(sourceObj: benefits, targetObj: accountResult);
433                 //usado para el logeo
434                 LoyaltyBenefits trans = new LoyaltyBenefits();
435                 this.MapPropertyts(sourceObj: benefits, targetObj: trans);
436                 // Se manda procesaro solo si no se a procesado con anterioridad --MDABLINDAJE setVentaOffline no forma parte del flow con monedero nuevo
437                 DataSet result;
438                 result = !benefits.IsOfflineSale
439                     ? await serviceClient.commitTransAsync(sTarjeta: benefits.Card, sSucia: benefits.Branch, benefits.TillId,
440                     sFormaPago: this.GetCodePaymentMethod(benefits.PaymentMethods), this.authenticationCode) //Task<DataSet>
441                     : null;
442             }
443         }
444     }
445     catch (Exception ex)
446     {
447         // ...
448     }
449 }

```

Para un flujo más detallado se agrega el diagrama de secuencia, en donde se muestran las principales interacciones y notas donde la forma de pago va viajando entre componentes



Se debe agregar la equivalencia de **SA3** en caso de requerir un valor diferente en Orígis, los valores actuales son:

	formaPago		equivalenciaFormaPago	
1	AMEX			3
2	CODI			7
3	CPAY			9
4	CRED			5
5	EFEC			1
6	MONE			6
7	PMOV			6
8	TCRE			3
9	TDEB			3
10	TPVI			3
11	TREG			3
12	VALE			4
13	WALL			8

Para agregar una nueva entrada se debe agregar un job en
 AnxoFARMAX\DEV_SyncConfiguration\PACK_GENEQUIVALENCIAFORMADEPAGO.xml con la nueva
 forma de pago **SAD3** de ser el caso

```

C# CentralTransactions  C# FETxWSGetTransactionDetails  <> PACK_GENEQUIVALENCIAFORMADEPAGO x
1  <?xml version="1.0" encoding="utf-8" ?>
2  <transfer>
3
4  <jobs>
5
6  <job name="GENEQUIVALENCIAFORMADEPAGO_V01" type="data_load">
7
8  > <mappings database="POSDB_OR" table="GENEQUIVALENCIAFORMADEPAGO" ...>
12
13  <data_objects>
14  <data_object formaPago="AMEX" equivalenciaFormaPago="3" />
15  <data_object formaPago="CODI" equivalenciaFormaPago="7" />
16  <data_object formaPago="CRED" equivalenciaFormaPago="5" />
17  <data_object formaPago="EFEC" equivalenciaFormaPago="1" />
18  <data_object formaPago="MONE" equivalenciaFormaPago="6" />
19  <data_object formaPago="PMOV" equivalenciaFormaPago="6" />
20  <data_object formaPago="ICRE" equivalenciaFormaPago="3" />
21  <data_object formaPago="IDEB" equivalenciaFormaPago="3" />
22  <data_object formaPago="IPVI" equivalenciaFormaPago="3" />
23  <data_object formaPago="TREG" equivalenciaFormaPago="3" />
24  <data_object formaPago="VALE" equivalenciaFormaPago="4" />
25  </data_objects>
26  </job>
27
28  <job name="GENEQUIVALENCIAFORMADEPAGO_V02" type="data_load">
29
30  > <mappings database="POSDB_OR" table="GENEQUIVALENCIAFORMADEPAGO" ...>
34
35  <data_objects>
36  <data_object formaPago="WALL" equivalenciaFormaPago="8" />
37  </data_objects>
38  </job>
39
40
41  <job name="GENEQUIVALENCIAFORMADEPAGO_V03" type="data_load">
42
43  > <mappings database="POSDB_OR" table="GENEQUIVALENCIAFORMADEPAGO" ...>
47
48  <data_objects>
49  <data_object formaPago="CPAY" equivalenciaFormaPago="9" />
50  </data_objects>
51  </job>
52  </jobs>

```