Causa raíz NO se envía el folio de orden en setInicioTrans2 hacia Origis

Autor: Huilver Nolasco Aguilar

Fecha: Marzo del 2025

Contenido

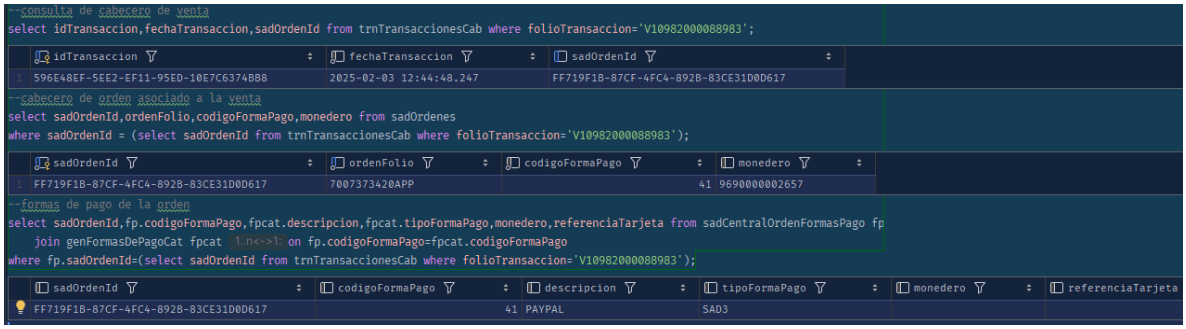
Descripción general

Recientemente se agregó un nuevo método en origis llamado setInicioTrans2 el cual soporta nuevos campos relacionados con la orden, en el caso particular del folio de orden el dato que debe mandarse es el campo sPedidoNumero, sin embargo se detectó que para algunas ventas que son de SAD, este dato viaja en 0 y para otras dicho dato si viaja correctamente.

Causa

La venta que se tomó para análisis es la venta V10982000088983 que corresponde al pedido 7007373420APP.

Lo primero que se hace es revisar los datos generales del pedido, así como sus formas de pago



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

idTransaccion	fechaTransaccion	sadOrdenId
596E48EF-5EE2-EF11-95ED-10E7C6374BB8	2025-02-03 12:44:48.247	FF719F1B-87CF-4FC4-892B-83CE31D0D617

```
--cabecero 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-83CE31D0D617	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  1..n<->1  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-83CE31D0D617	41	PAYPAL	SAD3		

Como se puede observar en la imagen anterior esta venta corresponde a una orden que fue pagada con PAYPAL(41) y por lo tanto es correcto que el campo **monedero y referencia tarjeta *no tengan dato***

Despues se procede a corroborar que lo reportado en el log de setInicioTrans2 de origis contra el log de monedero que se tiene para estar seguro que se trata del mismo caso.

setInicioTrans2	sTarjeta=9690000002657, sSucId=1098, sCajaId=20, sTerminalId=175327, sOrigen=2, sPedidoNumero=0, sReferenciaNumero=0, sCarritoNumero=0, sAuthCode=d98c9ff72fb6dbd0843d61e81bcdb2bd

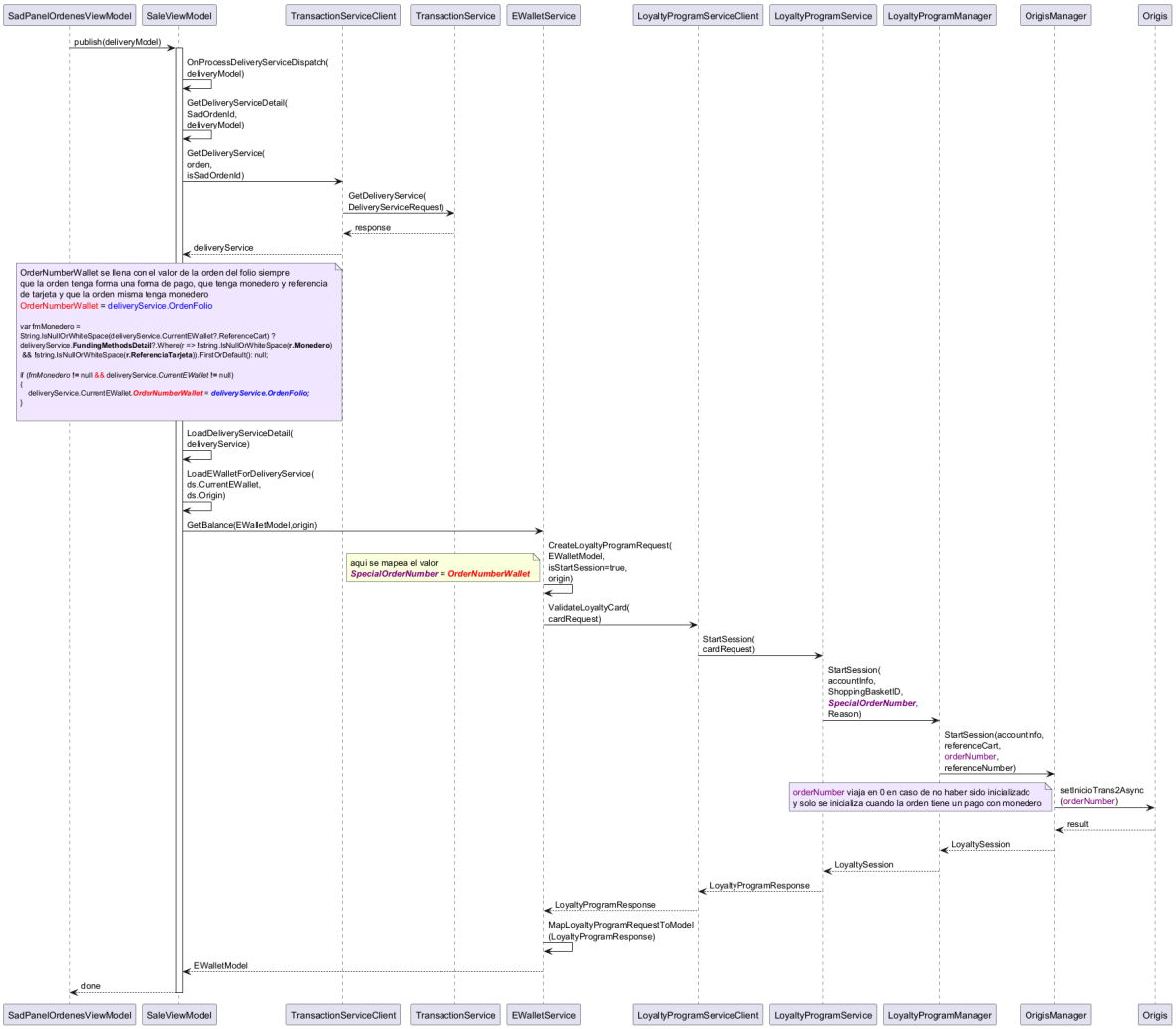
Dado que en la operación de commitTrans se registra de ser el caso el número de monedero, el folio de autorización y el número de venta, con esto se puede corroborar que se trata del mismo caso

Se observa que el número de tarjeta 9690000002657 y el numero de autorizacion d98c9ff72fb6dbd0843d61e81bcdb2bd coinciden, por lo tanto la venta V10982000088983 es la correcta para el análisis.

A continuación se describe de manera general cómo es que con una orden SAD se interactúa con Origis enfocado al método setInicioTrans2 .

Despues de que la orden ha llegado a la sucursal para su atención, dicha orden puede ser visualizada en el panel de órdenes de las cajas, una vez que se atiende la orden esta debe generar una comanda pero no sin antes haber obtenido la información de la orden, es decir su información general, el detalle de productos así como sus formas de pago. Despues de obtener dicha información esta se coloca dentro del flujo del POS para la captura de productos y su despacho, justo en ese mapeo de los datos de la orden hacia los datos de la venta y al ejecutar la comanda, se dispara el inicio de sesión con origis y es ahí donde el número de pedido(folio orden) se notifica, sin embargo en el mapeo establecido entre la orden y la venta asociada, el número de pedido se mapea solo bajo ciertas condiciones que básicamente son, que la orden tenga un monedero asociado y que tenga una forma de pago mda, si esto no se cumple la consecuencia es que termina mandando 0 por defecto.

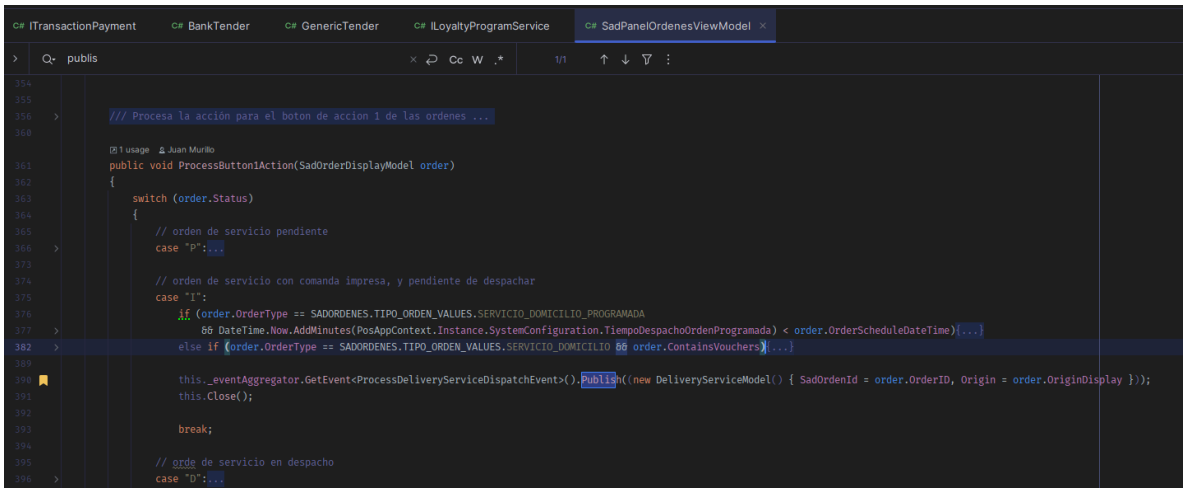
Se agrega un diagrama de secuencia para tener más detalle:



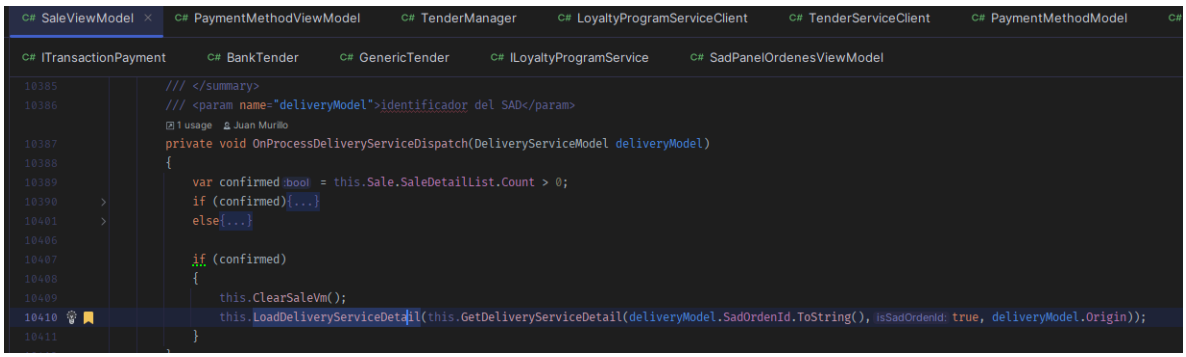
La imagen original se puede ver aquí sad_enviofolioorden_setiniciotrans.png

En código los lugares claves son

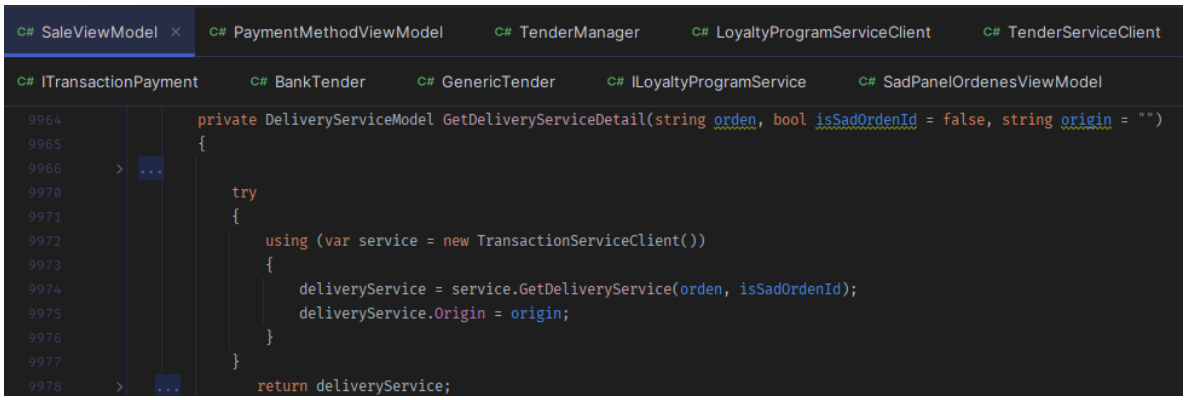
1.-Generación de comanda de la orden



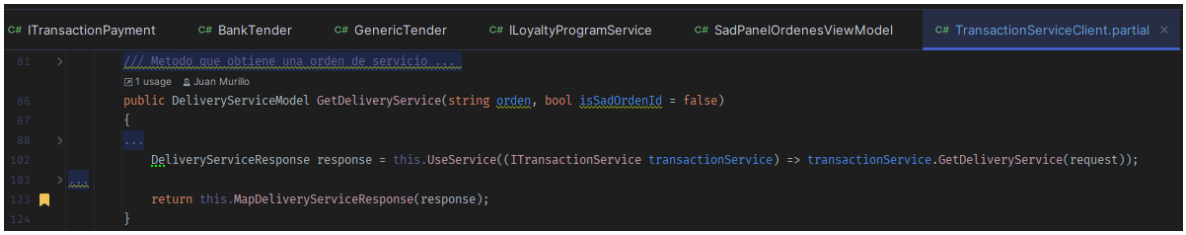
2.-Inicia la atención de la orden



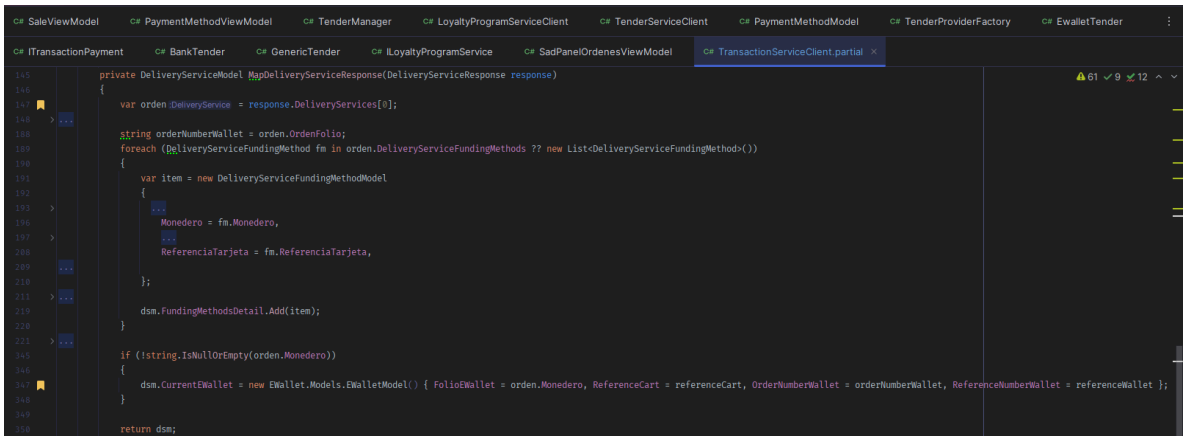
3.-Se consulta la información de la orden mediante la solicitud de un servicio de till service



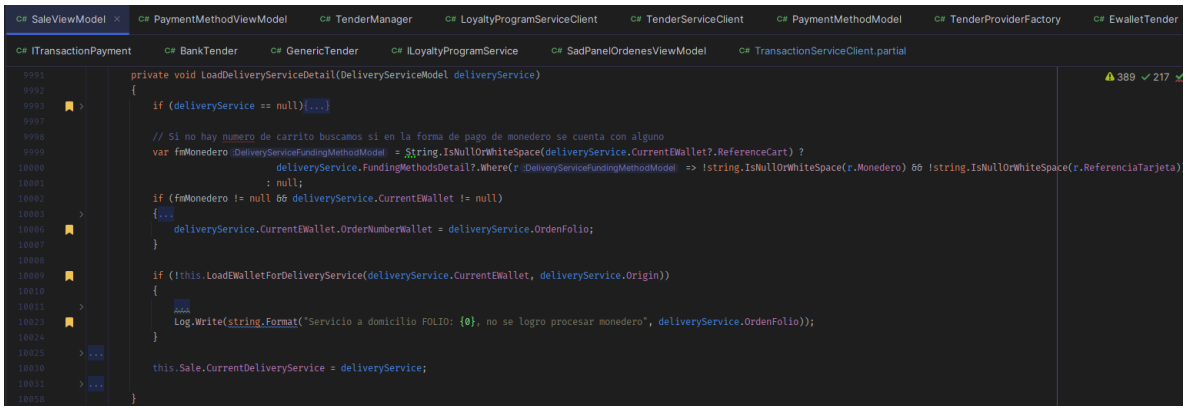
4.-Se consulta el servicio



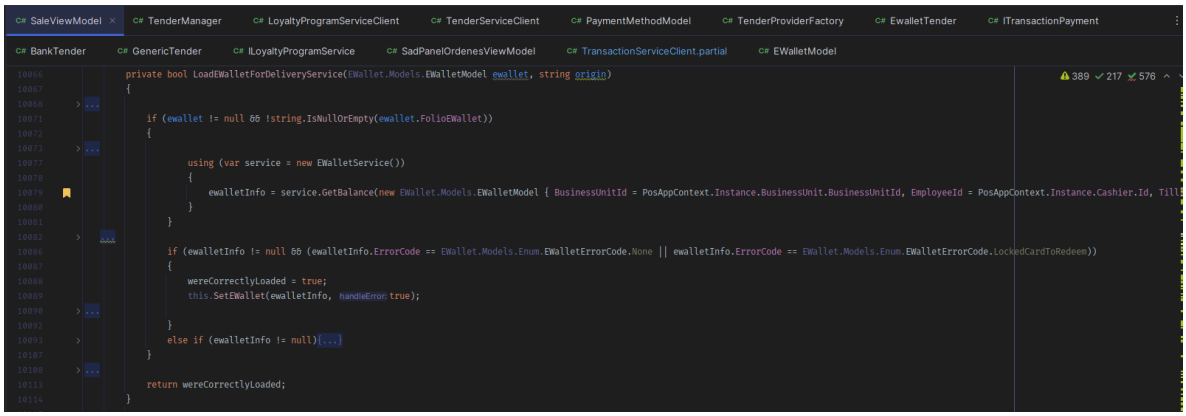
5.-Se hace el mapeo de los datos de la orden hacia un response, haciendo notar el mapeo relevante de los datos que más adelante se usan, por ejemplo está el monedero de la forma de pago de la orden, así como su referencia de tarjeta y los datos del monedero de la orden en el currentEwallet

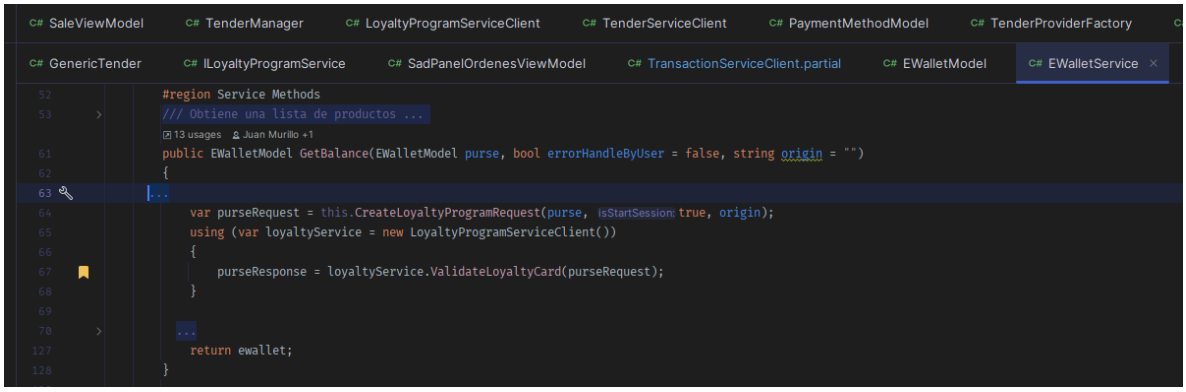


6.-Se cargan los datos de la orden para que sea procesada por el POS, en donde se puede observar que el llenado de OrdenNumberWallet depende de que exista una forma de pago de la orden que tenga monedero y que tenga referenciaTarjeta y que la orden tenga los datos del monedero en currentEwallet. Sin embargo la Orden en cuestión fue pagada con PayPal y **su forma de pago no tiene monedero ni referencia de tarjeta** así que el dato OrdenNumberWallet **no se llena**

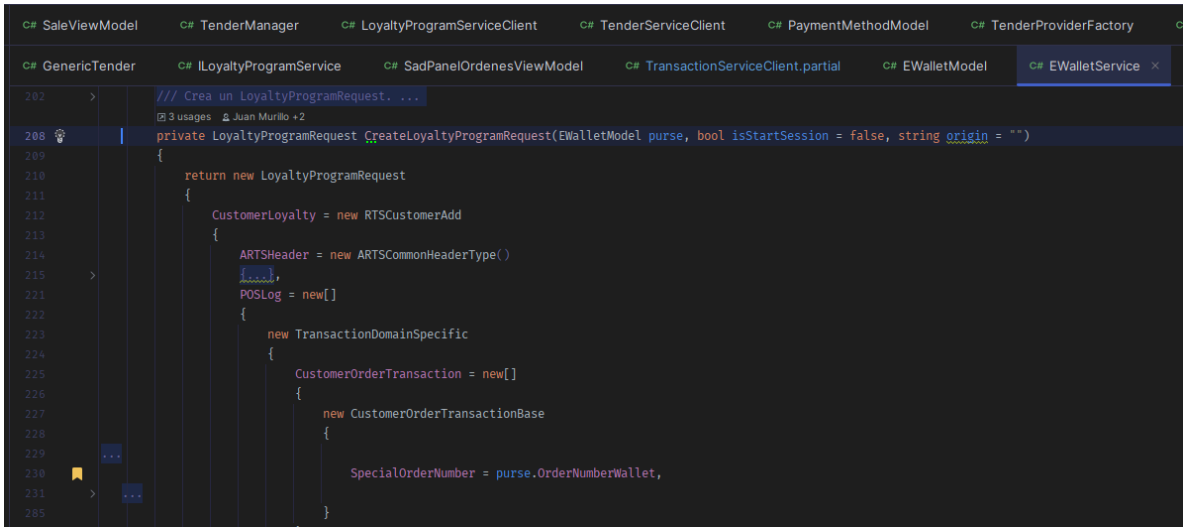


7.-Se consultan los datos del monedero hacia Origis en caso de que se cuente con el monedero, para despues colocar dicha información en el POS

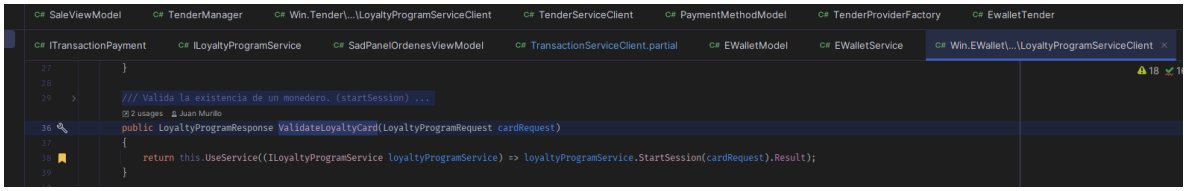




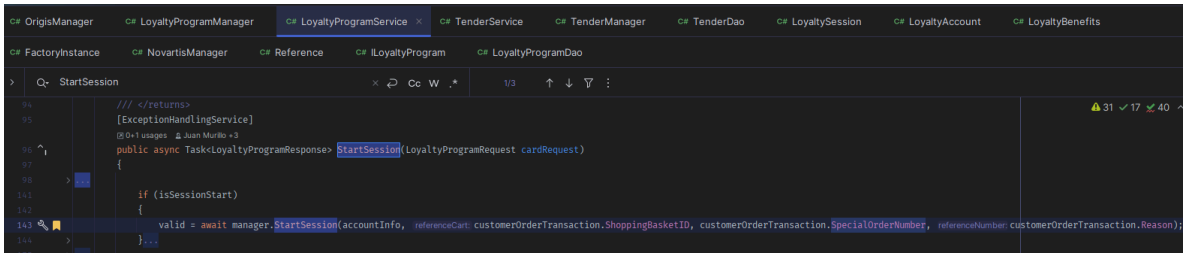
Pero antes de se llamar al servicio se genera el request mediante un mapeo en el que ahora el OrdenNumberWallet es depositado en SpecialOrdenNumber

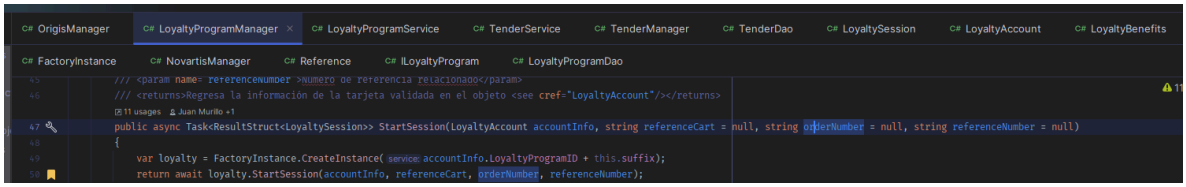


El servicio que se termina invocando es StartSession de la clase LoyaltyProgramServiceClient

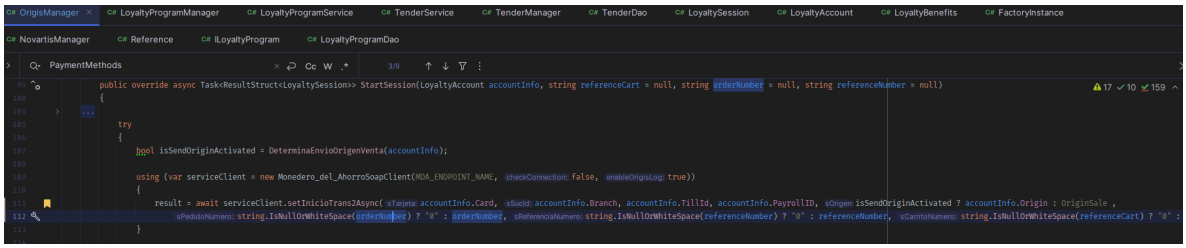


8.-Se inicia la sesión con origis mandando el dato SpecialOrderNumber que debería contener el folio de la orden





9.-Se manda llamar el metodo setInicioTrans2Async de origis, se puede observar que en caso de venir nulo o vacío se manda por defecto 0, que como hemos visto anteriormente orderNumber=SpecialOrderNumber y para esta orden en particular viene lleno debido al punto **6**



Solución

Se propone cambiar en el método FAhorro.POS.Win.Sale.ViewModels.SaleViewModel.LoadDeliveryServiceDetail la siguiente seccion de código:

```
// Si no hay numero de carrito buscamos si en la forma de pago de monedero se cuenta con alguno
var fmMonedero = String.IsNullOrWhiteSpace(deliveryService.CurrentEWallet?.ReferenceCart) ?
                    deliveryService.FundingMethodsDetail?.Where(r => !string.IsNullOrWhiteSpace(r.Monedero) &&
!string.IsNullOrWhiteSpace(r.ReferenciaTarjeta)).FirstOrDefault()
                  : null;
if (fmMonedero != null && deliveryService.CurrentEWallet != null)
{
   deliveryService.CurrentEWallet.ReferenceCart = fmMonedero.ReferenciaTarjeta;
   deliveryService.CurrentEWallet.ReferenceNumberWallet = fmMonedero.PagoFolioAutorizacion;
   deliveryService.CurrentEWallet.OrderNumberWallet = deliveryService.OrdenFolio;
}
```

Por esta otra:

```
var currentWallet = deliveryService.CurrentEWallet;
if (currentWallet != null){
   // Si no hay numero de carrito buscamos si en la forma de pago de monedero se cuenta con alguno
   if (string.IsNullOrWhiteSpace(currentWallet.ReferenceCart)){
       var fmMonedero =
           deliveryService.FundingMethodsDetail?.Where(r => !string.IsNullOrWhiteSpace(r.Monedero) &&
!string.IsNullOrWhiteSpace(r.ReferenciaTarjeta))
                  .FirstOrDefault();
       if (fmMonedero != null && !string.IsNullOrWhiteSpace(fmMonedero.ReferenciaTarjeta) &&
!string.IsNullOrWhiteSpace(fmMonedero.PagoFolioAutorizacion)){
           currentWallet.ReferenceCart = fmMonedero.ReferenciaTarjeta;
           currentWallet.ReferenceNumberWallet = fmMonedero.PagoFolioAutorizacion;
       }
   }
   currentWallet.OrderNumberWallet = deliveryService.OrdenFolio;
}
```

que básicamente significa que coloques *OrderNumberWallet* solo condicionado a que *deliveryService*.*CurrentEWallet* no sea nulo y los datos *ReferenceCart* y *ReferenceNumberWallet* de *deliveryService*.*CurrentEWallet* se traten de llenar con los datos de la forma de pago si encuentra alguna que los tenga, pero solo si *ReferenceCart* de *deliveryService*.*CurrentEWallet* no trae dato

Se anexa comparativa del cambio propuesta vs la versión actual