Causa raíz No se permite realizar la devolución de piezas de regalo origis en ventas SAD PAYPAL

Autor: Huilver Nolasco Aguilar

Fecha: Abril del 2025

Contenido

Descripción general.....3
Causa.....3
Solución.....7

Descripción general

Actualmente hay un error que hace que las piezas de regalo otorgadas por monedero no sean registrados en FARMAX para órdenes sad pagadas con PAYPAL, para tal error se generó un hotfix que pudiera solventar este tema, las pruebas realizadas fueron exitosas. Sin embargo para revisar un flujo completo del ciclo de una venta se intentó realizar la devolución de dicha venta pero no se logró obteniendo el mensaje **"Este producto no está incluído en el folio de venta o pedido SAD."** cuando se escaneo el producto de regalo

Causa

Para entender lo que está pasando es necesario primero revisar los datos de la venta, en donde se observa que la venta que la venta tuvo una pieza de regalo por parte de MDA que corresponde al producto 408684808(G AHORRO IBUPROFENO 600MG/20TAB 0383)

```
---consulta de datos generales de orden
select sadOrdenId,ordenFolio,pagadaEcommerce,CONCAT(o.codigoFormaPago,':',fp.descripcion) as forma_pago
from sadCentralOrdenes o join genFormasDePagoCat fp on o.codigoFormaPago = fp.codigoFormaPago
where sadOrdenId=(select sadOrdenId from trnTransaccionesCab where folioTransaccion='V22932000038850');
```

sadOrdenId	ordenFolio	pagadaEcommerce	forma_pago
852F8504-73EF-4618-810F-9968FB76B9BA	7009528700APP	true	41:PAYPAL

```
---consulta de datos generales de venta
select idTransaccion,sadOrdenId,folioTransaccion,subTotal,iva,cambio,folioMonedero,idSesionMonedero,totalCantidadVenta
from trnTransaccionesCab  where folioTransaccion='V22932000038850';
```

idTransaccion	sadOrdenId	folioTransaccion	subTotal	iva	cambio	folioMonedero	idSesionMonedero	totalCantidadVenta
54B184BD-800E-F011-946F-BC0FF36E20A3	852F8504-73EF-4618-810F-9968FB76B9BA	V22932000038850	6.8966	1.1035	0.0000	9600390157264	14427809201	2

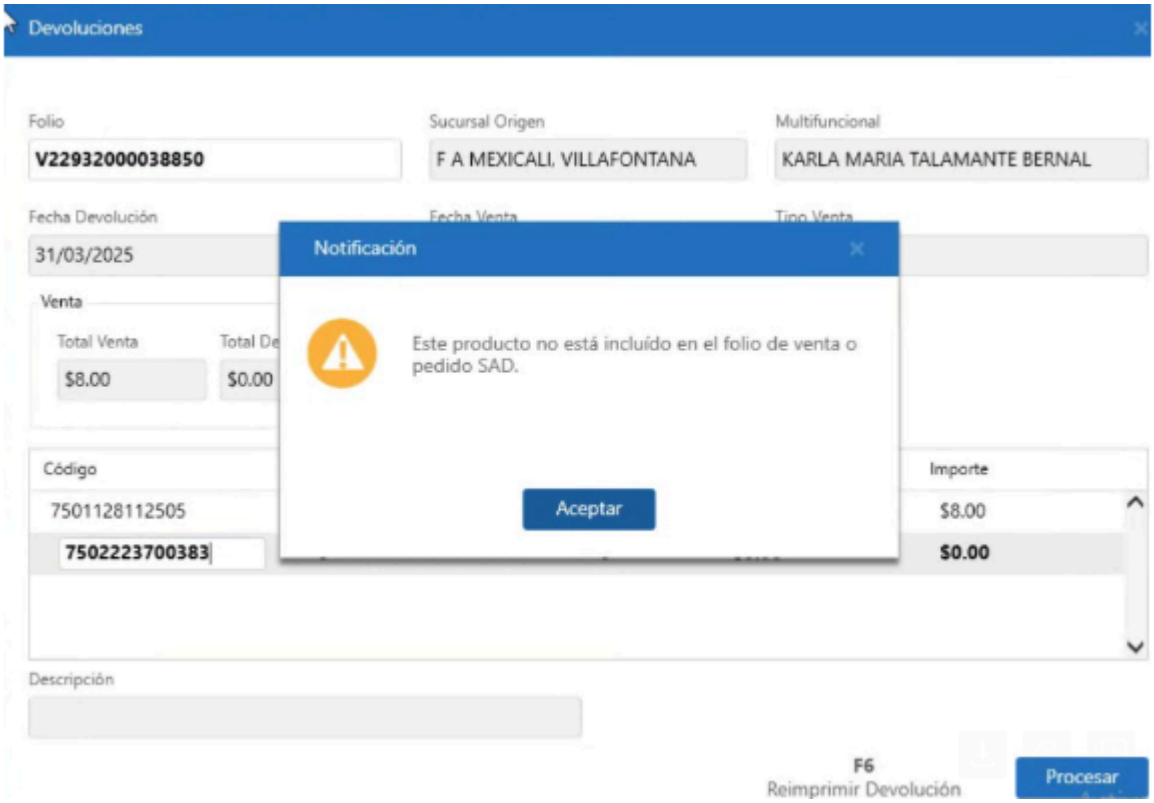
```
---consulta de datos generales de detalle de venta
select idTransaccion,renglon,codigoProducto,codigoRelacionado,precioVenta,cantidadVendida,cantidadBonificada
from trnTransaccionesDet where idTransaccion=(select idTransaccion from trnTransaccionesCab where folioTransaccion='V22932000038850');
```

	idTransaccion	renglon	codigoProducto	codigoRelacionado	precioVenta	cantidadVendida	cantidadBonificada
1	54B184BD-800E-F011-946F-BC0FF36E20A3	1	1780461972	7501128112505	6.8966	1.0000	0.0000
2	54B184BD-800E-F011-946F-BC0FF36E20A3	2	408684808	7502223700383	120.0000	0.0000	1.0000

```
---consulta de datos generales de recompensas de lealtad
select idTransaccion,renglon,CONCAT(l.codigoProducto,':',prod.descripcion) as prod,folioProgramaLealtad,programaLealtad,numeroAutorizacion,cantidad
from trnRecompensasProgramaLealtad l join dbo.genProductosCat prod on l.codigoProducto = prod.codigoProducto where idTransaccion=(select idTransaccion from trnTransaccionesCab where folioTransaccion='V22932000038850');
```

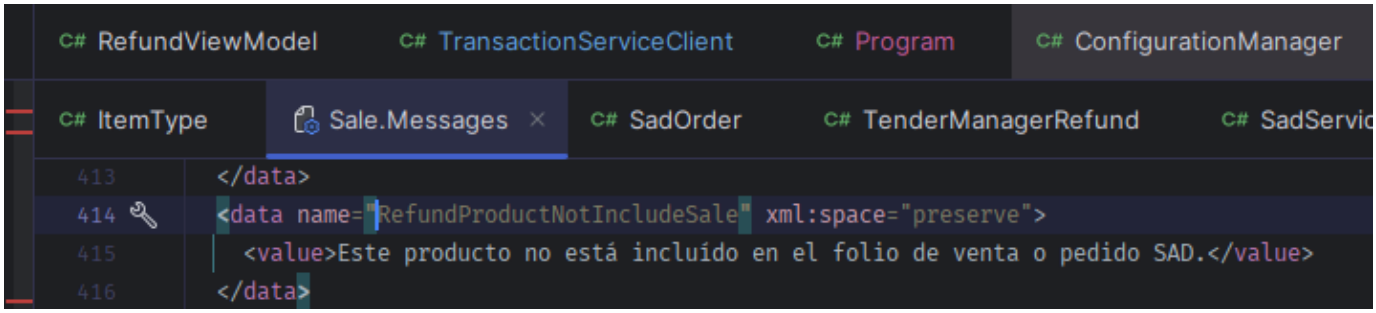
idTransaccion	renglon	prod	folioProgramaLealtad	programaLealtad	numeroAutorizacion	cantidad
54B184BD-800E-F011-946F-BC0FF36E20A3	2	408684808:G AHORRO IBUPROFENO 600MG/20TAB 0383	9600390157264	ORI	1622282355	1

Adicional tenemos la imagen de la incidencia que nos indica el mensaje **"Este producto no está incluído en el folio de venta o pedido SAD."**



Por lo que se procede a revisar el código la causa que origina este mensaje.

El mensaje está definido en Farmax\Src\POS\Win.Resources\Resources\Sale.Messages.resx



Y es usado por Farmax\Src\POS\Win.Sale\ViewModels\RefundViewModel.cs en donde se muestra que para que ocurra el mensaje debe ocurrir que al buscar el sku en la venta asociada a la devolución, NO se encuentre dicho sku

```
C# RefundViewModel ×   C# TransactionServiceClient   C# Program   C# RefundModel   C# ItemService   C# Sale.Messages.Designer   C# RefundService
Sale.Messages   C# SadOrder   C# TenderManagerRefund   C# SadServiceClient   C# ISadService   Localization Manager

3128            // Buscar el producto en la venta
3129            // tendria que ser al objeto de la venta con todos sus datos
3130            var saleItemData :RefundDetailModel = this.GetSaleItemData(inputSku);
3131
3132            if (saleItemData == null)
3133            {
3134                // FE4 Producto no pertenece al folio de venta o pedido SAD.
                    this.ShowMessageNew( title: Messages.NotificationHeaderText, message: Messages.RefundProductNotIncludeSale, image: MessageIcon.Warning);
3136                this.RefreshAndPrepareToCaptureNew();
3137                return;
3138            }
```

Sin embargo como se vio, la venta si contiene el detalle con el producto 7502223700383

```
---consulta de datos generales de detalle de venta
select idTransaccion,renglon,codigoProducto,codigoRelacionado,precioVenta,cantidadVendida,cantidadBonificada
from trnTransaccionesDet where idTransaccion=(select idTransaccion from trnTransaccionesCab where folioTransaccion='V22932000038850');
```

	idTransaccion	renglon	codigoProducto	codigoRelacionado	precioVenta	cantidadVendida	cantidadBonificada
1	54B184BD-800E-F011-946F-BC0FF36E20A3	1	1780461972	7501128112505	6.8966	1.0000	0.0000
2	54B184BD-800E-F011-946F-BC0FF36E20A3	2	408684808	7502223700383	120.0000	0.0000	1.0000

De regreso al código se observa que se busca el sku proporcionado dentro de la lista de detalles de venta pero que sean de tipo Product

```
C# RefundViewModel ×   C# TransactionServiceClient   C# Program   C# RefundModel   C# ItemService   C# Sale.Messages.Designer   C# RefundService   C# ItemType
Sale.Messages   C# SadOrder   C# TenderManagerRefund   C# SadServiceClient   C# ISadService   Localization Manager

3720        private RefundDetailModel GetSaleItemData(string codeProduct)                                   133 122 266
3721        {
3722            var saleItem :SaleDetailModel =
3723                this.Refund.Sale.SaleDetailList.Where(p :SaleDetailModel => p.Type == Item.Models.Enums.ItemType.Product) // IEnumerable<SaleDetailModel>
3724                    .FirstOrDefault(x :SaleDetailModel => x.DefaultRelatedProductID == codeProduct || x.RelatedProductID == codeProduct);
3725
3726            if (saleItem == null)
3727            {
3728                return null;
3729            }
3730            ...
3740            if (this.Refund.Sale.SaleType == EnumSaleType.Credit && this.refund.Sale.SaleDetailList.Where(x :SaleDetailModel => x.ProductId == saleItem.ProductId && x.Type == Item.Models.Enums.
3744            else if (saleItem.Promotions != null && saleItem.Promotions.Any()){...}
3748            ...
3750            return new RefundDetailModel
3751            {...};
3774        }
```

```
// ------------------------------------------------------------------------------------------...
namespace FAhorro.POS.Win.Item.Models.Enums
{
    /// Enum para mostrar el combo de los tipos de venta ...
    383 usages  Juan Murillo  11 exposing APIs
    public enum ItemType
    {
        /// Tipo Producto ...
        Product = 1,
...
        Promotion,
...
        LoyaltyPromotion,
...
        Bonus,
```

Por lo que se procede a obtener la respuesta del servicio producto de la consulta a la bd, en la siguiente imagen se muestran los datos relevantes obtenidos mediante el servicio, donde se muestra que regresa dos detalle de venta, uno con type 1 que corresponde a tipo Product y otro con type 3 que corresponde al tipo LoyaltyPromotion, que justamente es el sku 7502223700383 que fue el producto de regalo.


```
{
    "SaleFolio": "V22932000038850",
    [],
    {
        "ItemsCount": 1.0000,
        "SaleDetailList":
        [
            {
                "IsSugeridoAceptado": false,
                "SendPromoToServer": true,
                "TotalSalePriceWithDiscountWithOutLoyaltyPromotions": 0.0,
                "CodeDepartment": 2,
                "IsGift": false,
                "Type": 1,
                "PromotionId": 0,
                "ExcludeRecord": false,
                "IsRejected": false,
                "RejectedReason": null,
                "QuantityLoyaltySuggestion": 0,
                "FlagDirectBenefit": false,
                "ProductId": 1780461972,
                "Description": "G AHORRO JERINGA INSULINA 31GX6MM 0.3 ML 2505",
                "RelatedProductID": "7501128112505",
                "DefaultRelatedProductID": "7501128112505",
                "Quantity": 0.0000,
                "Promotions": null,
                "MessageLoyaltyPromotionInfo": null,
            },
            {
                "IsSugeridoAceptado": false,
                "SendPromoToServer": true,
                "TotalSalePriceWithDiscountWithOutLoyaltyPromotions": 120.00,
                "CodeDepartment": 0,
                "IsGift": false,
                "Type": 3,
                "PromotionId": 0,
                "ExcludeRecord": false,
                "IsRejected": false,
                "RejectedReason": null,
                "QuantityLoyaltySuggestion": 0,
                "FlagDirectBenefit": false,
                "ProductId": 408684808,
                "Description": null,
                "RelatedProductID": "7502223700383",
                "DefaultRelatedProductID": null,
                "Quantity": 1.0,
                "Promotions": null,
                "MessageLoyaltyPromotionInfo": null,
            }
        ],
        "SuggestedItems":
        [],
        "RealSaleDetailList":
```


Así que **deliberadamente no** se consideran los productos de regalo por **lealtad**

Solución

Se puede pensar que hacer menos restrictiva la consulta puede solucionar el problema, sin embargo se observa que no es la única condición que debe quitarse

ItemType | RefundViewModel | TransactionServiceClient | Program | RefundModel | ItemService | Sale.Messages.Designer | RefundService

SadServiceClient | ISadService | Localization Manager

```
18                  this.ShowMessageNew(Messages.TitleOfWarning, message: Messages.RefundProductNotExists, image: MessageIcon.Warning);
19                  this.RefreshAndPrepareToCaptureNew();
20                  return;
21              }
22
23              var saleItemDetails :IEnumerable<SaleDetailModel> = this.Refund.Sale.SaleDetailList.Where(p :SaleDetailModel => (p.Type == Item.Models.Enums.ItemType.Product)
24                  && (p.DefaultRelatedProductID == inputSku || p.RelatedProductID == inputSku));
25
26              bool isSadDeliveryCancel = this.Refund.HasSadNoDelivery;
27
28              // Buscar el producto en la venta
29              // tendría que ser al objeto de la venta con todos sus datos
30              var saleItemData :RefundDetailModel = this.GetSaleItemData(inputSku);
31
32              if (saleItemData == null)
33              {
34                  // FE4 Producto no pertenece al folio de venta o pedido SAD.
35                  this.ShowMessageNew( title: Messages.NotificationHeaderText, message: Messages.RefundProductNotIncludeSale, image: MessageIcon.Warning);
36                  this.RefreshAndPrepareToCaptureNew();
37                  return;
38              }
39
40              // valida si el producto pertenece a una promoción de origis si es asi no permite que se agregue a la devolución
41              if (!isSadDeliveryCancel && this.ValidateIfCurrentproductContainsOrigisPromotion(saleItemData.DefaultRelatedCode))
42              {
43                  this.ShowMessageNew( title: Messages.NotificationHeaderText, Messages.messageIfProductContainsOrigisPromotion, image: MessageIcon.Warning);
44                  this.RefreshAndPrepareToCaptureNew();
45                  return;
46              }
```

```
C# ItemType   C# RefundViewModel   C# TransactionServiceClient   C# Program   C# RefundModel   C# ItemService   C# Sale.Mess
C# SadServiceClient   C# ISadService   Localization Manager
pay                                                                  33/209
965            }
966
967            /// <summary>
968            /// Método que dispara al proceso de devolución.
969            /// </summary>
               1 usage   Juan Murillo +1
970            private void OnProcessRefund()
971            {
972                if (this.ValidateStoreServicesConnection(this.Refund))
973                {
974                    return;
975                }
976
977
978                if (this.Refund.HasSadNoDelivery
979                    || (this.Refund.HasSad
980                        && this.refund.SadOrder.OrderType != SADORDENES.TIPO_ORDEN_VALUES.LABORATORY
981                        && !this.Refund.SadOrder.PayOnEcommerce
982                        && !this.RefundTM.TenderAllowedForPartialRefund.Contains(this.Refund.SadOrder.TenderCode)))
983                {
984
985                    //Se valida si se deben de cargar productos vales interbodega a la devolucion
986                    if (!this.LoadRefundProdVoucher(this.Refund))
987                    {
```

Adicional a que se debe reportar a origis dicha pieza devuelta por lo que esto debiera resolverse al finalizar la ejecución del requerimiento ***Devoluciones Planes de Lealtad (Piezas gratis)*** que actualmente se encuentra en ejecución.