	Incidencia – Error en neteo de productos al realizar devolución con más de 2 productos con la misma promoción	
Autor	Versión	Fecha de elaboración
Antonio Escorcía	1.0	22-06-2024

Captura de productos en devoluciones

Descripción general

Al iniciar el proceso de devolución es necesario capturar el folio de la venta a devolver para obtener su información, una vez se ha capturado la venta se procede a ingresar los SKUs de los artículos correspondientes, si el artículo capturado cuenta con una promoción, es requerido escanear todos los artículos con el mismo SKU y todos los artículos que tengan promoción (aunque no tengan el mismo SKU y tengan una promoción distinta), en esta parte del flujo de devoluciones aún no se persiste la devolución.

El flujo de devoluciones se lleva a cabo actualizando y persistiendo información en distintos pasos:

1. Se solicita el folio de venta y los productos a devolver.
2. **Se ingresan los SKUs de los artículos a devolver.**
3. Se crea el folio de devolución con el estado de “N” y se persiste en la base de datos.
4. Se solicitan los montos de devolución.
5. Se solicita el motivo de devolución.
6. Se actualiza la devolución al estado de “R”, se procesan y persisten los métodos de pago, si todos fueron aplicados se actualiza el estado de la devolución a “A”.
7. Se actualiza el estado de la transacción al estado “D” si todos los productos de la transacción fueron devueltos.

Adjunto se encuentra un documento de nombre [Devolucion - ingreso de datos - diagrama de secuencia.drawio.png](#) donde es posible apreciar mejor parte de este flujo.

Diagnóstico

El error ocurre en el paso (2) del flujo, se da al intentar devolver una transacción con más de una promoción en distintos SKUs, ocurre al ingresar el segundo SKU con promoción. Se intenta realizar la devolución múltiples veces en las que se obtiene el mismo mensaje de error, como evidencia se puede consultar el log adjunto [20240611_FARMAX_TILL_SERVER_CORE.XLOG](#), desde la línea 504 hasta la línea 525, donde se expresa que ocurrió un error al netear los productos, esto ocurre 8 veces:

```
[2024-06-11 21:30:33:141],FARMAX_TILL_SERVER,CORE,ERROR,[ONEXCEPTION],ONEXCEPTION: Method:
ProcessRefundQuery - Exception: El neteo de productos arrojó una cantidad negativa. El neteo de productos
arrojó una cantidad negativa.
en
FAhorro.POS.BusinessLogic.PriceCalculate.PromotionsRefoundManager.AuditPromotionRefound(IList`1
itemsConfiguration, IList`1 originalSaleItems, IList`1 committedReturns, ItemQuantity newItemToReturn,
IList`1 previouslyReturnedItemsInTransaction, IList`1 coupons)

en FAhorro.POS.BusinessLogic.Refund.RefundManager.GetItemsToRefundOnPromotion(ContextTypes
contextType, Int32 businessUnitId, String saleNumber, RefoundItem returnedItem, List`1
previouslyReturnedItemsInTransaction)

en
FAhorro.POS.Service.Transaction.TransactionService.ProcessRefundQuery(TransactionRefundProcessRequest
request)
```



Incidencia – Error en neteo de productos al realizar devolución con más de 2 productos con la misma promoción

Autor	Versión	Fecha de elaboración
Antonio Escorcía	1.0	22-06-2024

El error ocurre en `PromotionsRefoundManager`, en la línea 317:

```
C# RefundManager.cs  C# PromotionsRefoundManager.cs  C# TransactionService.cs

314
315     if (remainingSaleDetails.Any(d:RefoundItem => d.Quantity < 0))
316     {
317         throw new InvalidOperationException(message: "El neteo de productos arrojó una cantidad negativa.");
318     }
319
```


En la siguiente [figura](#) se puede observar parte de los motivos del fallo:

```
C# RefundManager.cs  C# PromotionsRefoundManager.cs  C# TransactionService.cs

275
276 // Cantidades netas por producto y promoción antes de devolver el nuevo producto.
277 // Estos serán los registros que se tomarán como base. Se aplicará el motor de promociones sobre las cantidades netas y después se compararan los resultados del motor de promociones
278 // para determinar si se debe solicitar la devolución de algún artículo adicional.
279 var remainingSaleDetails = i.RefoundItems.Select(i:RefoundItem => new RefoundItem
280 {
281     ProductId = i.ProductID,
282     PromotionId = i.PromotionId,
283     Coupon = i.Coupon,
284     RegularSalesUnitPrice = i.UnitPrice,
285     ActualSalesUnitPrice = i.ModifySalePrice,
286     Quantity = Convert.ToInt32(i.Quantity),
287     LoyaltyProgram = i.LoyaltyProgram,
288 });
289 .Union(previouslyReturnedItems) // 1
290 .GroupBy(d:RefoundItem => new { d.ProductId, d.PromotionId, d.Coupon, SalePrice = d.ActualSalesUnitPrice, RegularPrice = d.RegularSalesUnitPrice, LoyaltyProgram = d.LoyaltyProgram }) // 2
291 .Select(g:IGrouping<ProductId, PromotionId, Coupon>, => new RefoundItem
292 {
293     ProductId = g.Key.ProductId,
294     PromotionId = g.Key.PromotionId,
295     Coupon = g.Key.Coupon,
296     RegularSalesUnitPrice = g.Key.RegularPrice,
297     ActualSalesUnitPrice = g.Key.SalePrice,
298     Quantity = g.Sum(d:RefoundItem => d.Quantity), // 3
299     LoyaltyProgram = g.Key.LoyaltyProgram,
300 }) // 4
301 .Where(d:RefoundItem => d.Quantity != 0) // 5
302 .ToList();
```

1. Se utiliza el detalle de la venta después de un mapeo (realizado en **`RefundManager.ConvertDetailsToPriceCalculateItems()`**).
2. Se realiza una unión con la lista de artículos que han sido ingresados por el usuario hasta el momento.
3. Se agrupa por una serie de campos, incluyendo “LoyaltyProgram”.
4. Se suman las cantidades (Quantity) en las agrupaciones, causando que algunas lleguen a 0.
5. Se filtran de la lista los elementos con una cantidad (Quantity) de 0.

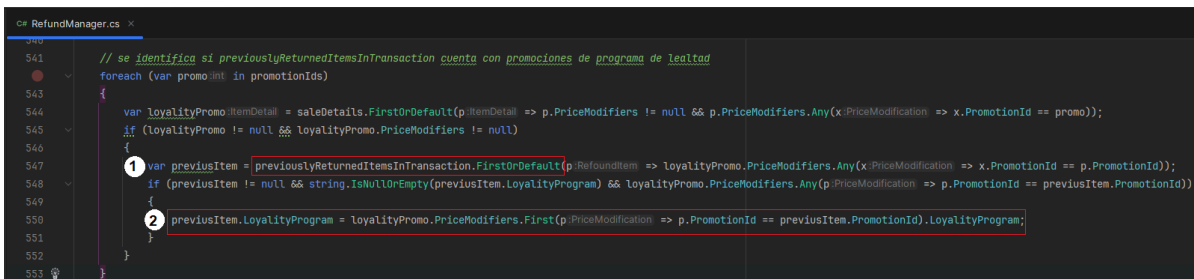
El error ocurre porque la propiedad `LoyaltyProgram` proveniente de los artículos del usuario se encuentra con el valor de “null” pero su elemento correspondiente en el detalle de la venta tiene la propiedad de `LoyaltyProgram` con el valor de “” (cadena vacía), aunque realmente no haya tenido una promoción.

	Incidencia – Error en neteo de productos al realizar devolución con más de 2 productos con la misma promoción	
Autor	Versión	Fecha de elaboración
Antonio Escorcía	1.0	22-06-2024

Posibles causas

- Error lógico en mapeo de datos

El error se da en **RefundManager.cs** en el método **GetItemsToRefundOnPromotion (línea 547)** por un error lógico en un mapeo de datos:



```

541 // se identifica si previouslyReturnedItemsInTransaction cuenta con promociones de programa de lealtad
542 foreach (var promoInt in promotionIds)
543 {
544     var loyaltyPromoItemDetail = saleDetails.FirstOrDefault(p.ItemDetail => p.PriceModifiers != null && p.PriceModifiers.Any(x.PriceModification => x.PromotionId == promo));
545     if (loyaltyPromo != null && loyaltyPromo.PriceModifiers != null)
546     {
547         1 var previousItem = previouslyReturnedItemsInTransaction.FirstOrDefault(p.RefoundItem => loyaltyPromo.PriceModifiers.Any(x.PriceModification => x.PromotionId == p.PromotionId));
548         if (previousItem != null && string.IsNullOrEmpty(previousItem.LoyaltyProgram) && loyaltyPromo.PriceModifiers.Any(p.PriceModification => p.PromotionId == previousItem.PromotionId))
549         {
550             2 previousItem.LoyaltyProgram = loyaltyPromo.PriceModifiers.First(p.PriceModification => p.PromotionId == previousItem.PromotionId).LoyaltyProgram;
551         }
552     }
553 }

```

[Ver en pantalla completa](#)

1. De los artículos anteriormente ingresados por el usuario por medio del POS se toma el primer artículo (no todos) que coincida con la promoción.
2. Se le asigna la cadena de programa de lealtad a la propiedad LoyaltyProgram.

El problema se da porque se asume que solo un artículo podría llevar una promoción específica, esta lógica funciona con 1 o 2 artículos con la misma promoción, **al agregar un 3er artículo o más (caso de la incidencia)** ocurre un problema de integridad ya que los artículos a partir del 2do no tendrán asignado su campo de **LoyaltyProgram** correspondiente, causando una agrupación errónea y dando lugar a un error de neteo.



Incidencia – Error en neteo de productos al realizar devolución con más de 2 productos con la misma promoción

Autor	Versión	Fecha de elaboración
Antonio Escorcía	1.0	22-06-2024


Posibles soluciones

1. Es posible resolver este problema sustituyendo el código que realiza el mapeo en la clase **RefundManager.cs** en el método **GetItemsToRefundOnPromotion (línea 547)**, asignando el valor original proveniente de la promoción a todos los artículos en lugar de solamente asignarlo al primer artículo:

```
540
541 // se identifica si previouslyReturnedItemsInTransaction cuenta con promociones de programa de lealtad
542 foreach (var promoId in promotionIds)
543 {
544     var loyaltyPromoItemDetail = saleDetails.FirstOrDefault(pItemDetail => p.PriceModifiers != null && p.PriceModifiers.Any(x => x.PromotionId == promoId));
545     if (loyaltyPromo == null || loyaltyPromo.PriceModifiers == null) continue;
546
547     bool ItemHasPromo(RefoundItem item)
548     {
549         return loyaltyPromo.PriceModifiers.Any(modifier => item.PromotionId == modifier.PromotionId)
550             && string.IsNullOrEmpty(item.LoyaltyProgram);
551     }
552
553     1 var itemsWithPromo : IEnumerable<RefoundItem> = previouslyReturnedItemsInTransaction.Where(ItemHasPromo);
554     foreach (var item in itemsWithPromo)
555     {
556         bool IsCurrentItemPromo(PricingModification modifier) => item.PromotionId == modifier.PromotionId;
557         var appliedPromo : PricingModification = loyaltyPromo.PriceModifiers.First(IsCurrentItemPromo);
558         2 item.LoyaltyProgram = appliedPromo.LoyaltyProgram;
559     }
560 }
```

[Ver en pantalla completa](#)

1. Se obtienen todos los artículos que tengan dicha promoción.
2. Se asigna su etiqueta de programa de lealtad respectivamente.

	<p>Incidencia – Error en neteo de productos al realizar devolución con más de 2 productos con la misma promoción</p>	
Autor	Versión	Fecha de elaboración
Antonio Escorcía	1.0	22-06-2024

Casos relacionados

[Devoluciones - Error de neteo de productos](#)