

			
Autor	Versión	Fecha de elaboración	Última edición
Angel Fernando	1.0	05/09/2025	11/09/2025

Control de versiones

Versión	Fecha	Autor	Descripción
1.0	05/09/2025	Ángel	Tarjeta de Crédito Debito ACI
1.1	11/09/2025	Eduardo Ochoa	Se realiza análisis y explicación de la transacción

Sucursal	2610
Caja	22
Venta	V26102200109788
Orden SAD	N/A
Fecha Transacción	2025-08-28 16:41:32.000

			
Autor	Versión	Fecha de elaboración	Última edición
Angel Fernando	1.0	05/09/2025	11/09/2025

Descripción general

CIUDAD_OPERATIVA	MONTERREY ORIENTE
SUCURSAL	F A MONTERREY OTE, CORPORATIVO
SUCURSAL_ID	2610
AFILIACIÓN AMEX	8134845577
AFILIACIÓN BANAMEX	1133538
FECHA	25-agosto-25
HORA	20
TIPO_VENTA	POS
FORMA_DE_PAGO	TARJETA DE DEBITO ACI (AD)
AUTORIZACION	004595
CAJA	22
NRO_TICKET	22_35909788
IMPORTE_CON_IVA	400.00

Se reporta que se obtiene la venta V26102200109788 donde al obtener los reportes se muestra como faltante y es una venta que no está cobrada. Y se requiere el ticket de venta para enviarlos aclaración con el banco.

Se solicita apoyar para validar como se puede rastrear si es cancelación, ya que validando los voucher físicamente y en farmax con opción a impresión no aparece el ticket 22_35909788

				
Autor	Versión	Fecha de elaboración		Última edición
Angel Fernando	1.0	05/09/2025		11/09/2025

Diagnóstico

Se valida el folio de venta V26102200109788 en la base de datos de Caja, Store y Central, donde se obtiene información de las siguientes tablas;

- 1) En la primera tabla **trnTransaccionesCab** se puede visualizar los datos como el folio de venta, la fecha que fue realizada la compra, el subtotal, el IVA, el estatus donde nos indica que se encuentra en C = Cancelada, el campo estaActivo es una bandera donde también nos indica con 1 o 0 si fue procesada y por ultimo el campo folioMonedero donde viene en null.

	(No column name)	codigoMovimientoCaja	folioTransaccion	codigoSucursal	numeroCaja	fechaTransaccion	subTotal	iva	status	estaActivo	folioMonedero
1	trnTransaccionesCab	3D4E0BDF-2182-F011-9671-8469936A491A	V26102200109788	2610	22	2025-08-25 20:10:50.707	344.8276	55.1724	C	0	NULL

Ilustración 1 TrnTransaccionesCab.

- 2) En la segunda tabla **trnTransaccionesDet** se puede observar el producto con SKU 6583760104008, el precio de venta de \$344.8276 y la cantidad vendida que fue 1, con descripción “PAQUETE AMIGO SL 400”.

	(No column name)	idTransaccion	renglon	codigoProducto	codigoRelacionado	unidad	precioRegular	precioVenta	precioMaximo	cantidadVendida	tasalva	importeIva
1	trntransaccionesdet	3C4E0BDF-2182-F011-9671-8469936A491A	1	921401149	6583760104008	0	344.8276	344.8276	344.8276	1.0000	16.0000	55.1724

Ilustración 2 TrnTransaccionesDet

- 3) En la tercer tabla **genMovimientosCajasdet** se encuentra el método de pago que fue realizado donde vemos que fue con el código (12) = Tarjeta de Débito ACI, con un monto de \$400.00 pesos, numero de autorización 004595 y su estatus de la forma de pago esta en C = Cancelada.

	(No column name)	codigoMovimientoCaja	codigoFormaPago	importePago	importeComisionBancaria	importeIva	tipoDeCambio	autorizacion	documento	estatusFormaPago
1	genMovimientosCajasdet	3D4E0BDF-2182-F011-9671-8469936A491A	12	400.0000	0.0000	0.0000	1.0000	004595	NULL	C

Ilustración 3 genMovimientosCajasdet

Esto nos indica que la transacción no fue transaccionada con éxito y fue cancelada.

			
Autor	Versión	Fecha de elaboración	Última edición
Angel Fernando	1.0	05/09/2025	11/09/2025

Traza de transacción en Farmax

Donde podemos observar que hubo dos transacciones simultaneas V261022001109788 y V261022001109787

Análisis del Flujo de Estados:

- **Estado N:** Se creó la venta como "Nueva"
 - **Estado R:** Cambió a "Pendiente de pago"
 - **Estado V:** (Temporal) Procesada como "Vigente"
 - **Estado C:** Finalmente cambió a "Cancelada"

Se creó el método `CancelSelectedPendingSales` que indica que la venta fue cancelada, cambiando el estatus a C.

GET Technologies			
Autor	Versión	Fecha de elaboración	Última edición
Angel Fernando	1.0	05/09/2025	11/09/2025

Revisamos el log del AnxoPinPad donde se guardan los registros con pagos de tarjeta bancaria ya sea Crédito/Debito. No se encuentra registro alguno con el importe pagado por la cantidad de 400, sin embargo se encuentra una autorización por la cantidad de \$505 con el mismo número de autorización mencionado en el reporte 004595.

A continuación se anexa el log.

```
-----  
CC1 request - card processing start  
[2025-08-25 20:10:51:851],FARMAX_POS,ANXOPINPAD,INFO,,  
Amount: $505  
Counter: 22  
Payment Type: Credit/debit tender ID: 12  
Transaction Code: D Default  
[2025-08-25 20:11:01:511],FARMAX_POS,ANXOPINPAD,INFO,,TIMING1,ObtieneTarjeta,00:00:09.66  
[2025-08-25 20:11:01:511],FARMAX_POS,ANXOPINPAD,INFO,,TIMING1,CierraVentana,00:00:00.00  
[2025-08-25 20:11:01:513],FARMAX_POS,ANXOPINPAD,INFO,  
[2025-08-25 20:11:01:513],FARMAX_POS,ANXOPINPAD,INFO,,Loaded configuration shown  
[2025-08-25 20:11:01:513],FARMAX_POS,ANXOPINPAD,INFO,, Implementation: AnxoImplementation  
[2025-08-25 20:11:01:513],FARMAX_POS,ANXOPINPAD,INFO,, Serial port: 9  
[2025-08-25 20:11:01:513],FARMAX_POS,ANXOPINPAD,INFO,, Serial port PRESENT: True  
[2025-08-25 20:11:01:513],FARMAX_POS,ANXOPINPAD,INFO,, Serial port baud rate: 19200  
[2025-08-25 20:11:01:513],FARMAX_POS,ANXOPINPAD,INFO,, Serial port parity: None  
[2025-08-25 20:11:01:513],FARMAX_POS,ANXOPINPAD,INFO,, Serial port data bits: 8  
[2025-08-25 20:11:01:513],FARMAX_POS,ANXOPINPAD,INFO,, Serial port stop bits: One  
[2025-08-25 20:11:01:513],FARMAX_POS,ANXOPINPAD,INFO,, Timeout PINpad ACK: 5  
[2025-08-25 20:11:01:513],FARMAX_POS,ANXOPINPAD,INFO,, Timeout PINpad Q4: 20  
[2025-08-25 20:11:01:513],FARMAX_POS,ANXOPINPAD,INFO,, Timeout PINpad C0: 30  
[2025-08-25 20:11:01:513],FARMAX_POS,ANXOPINPAD,INFO,, Timeout PINpad C1: 45  
[2025-08-25 20:11:01:513],FARMAX_POS,ANXOPINPAD,INFO,, Timeout PINpad C4 chip 45  
[2025-08-25 20:11:01:513],FARMAX_POS,ANXOPINPAD,INFO,, Timeout PINpad CPO 30  
[2025-08-25 20:11:01:513],FARMAX_POS,ANXOPINPAD,INFO,, Auth server: 192.168.152.17  
[2025-08-25 20:11:01:513],FARMAX_POS,ANXOPINPAD,INFO,, Auth server port: 7000  
[2025-08-25 20:11:01:513],FARMAX_POS,ANXOPINPAD,INFO,, Timeout auth: 30  
[2025-08-25 20:11:01:513],FARMAX_POS,ANXOPINPAD,INFO,, Credit/debit tender ID: 12  
[2025-08-25 20:11:01:513],FARMAX_POS,ANXOPINPAD,INFO,, AMEX tender ID: 14  
[2025-08-25 20:11:01:513],FARMAX_POS,ANXOPINPAD,INFO,,*****  
[2025-08-25 20:11:01:718],FARMAX_POS,ANXOPINPAD,INFO,,TIMING2,SendAuthMessage,00:00:00.20,OK  
[2025-08-25 20:11:01:718],FARMAX_POS,ANXOPINPAD,WARNING,,SUC:2610|CADA:00000022|FOL:109787 [TARJ:*****9866 MTO:50500 ARQC:ECE4EEEDF6B55653|RC:00 AUT:004595]  
[2025-08-25 20:11:02:068],FARMAX_POS,ANXOPINPAD,INFO,,AUTH approved  
[2025-08-25 20:11:05:359],FARMAX_POS,ANXOPINPAD,INFO,,TIMING1,enviaRetencionVenta,00:00:03.84  
[2025-08-25 20:11:05:359],FARMAX_POS,ANXOPINPAD,INFO,,TIMING1,CierraVentana,00:00:00.00  
[2025-08-25 20:15:14:378],FARMAX_POS,ANXOPINPAD,INFO,
```

Este monto de \$505 coincide con lo que nos indica farmacias donde valido con el integrador ACI que si se aprobó con el numero de aprobación que fue asignado pero no corresponde al total de la transacción V261022001109788 que era por el monto de \$400.

			
Autor	Versión	Fecha de elaboración	Última edición
Angel Fernando	1.0	05/09/2025	11/09/2025

F A MTCW	08/25/2025	0400 - Debit	\$ 505.00	00 - Approved	VISA Debito	418914*****9866 109787 22	80
F A MTCW	08/25/2025	0200 - Debit	\$ 505.00	00 - Approved	VISA Debito	418914*****9866 109787 22	130
CORPORATIVO	20.15.13.627	Reversal					
CORPORATIVO	20.11.00.793	Sale					

Transaction Information		Customer Information	
Reference #:	110246196465	Name:	
Transaction Type:	0400 - Debit Reversal	Address:	
Store:	F.A.MTCW CORPORATIVO	Phone Number:	
Response:	00 - Approved		
Host Response:	00 - APROGADA 004585		
Date-Time:	08/25/2025 20:15:13.827		
Customer Identification			
ID Type	ID Number	Card Number Token	
VISA Debit	4169140056369865		
Transaction Detail			
Authorization			
Transaction Amount:	\$ 505.00	Controller:	00000022
Original Amount:	\$ 505.00	Terminal:	22
Reversed Amount:	\$ 505.00	Clerk:	
Settlement Batch:	6871	Merchant Type:	Pharmacy
POS Date:	08/25/2025 20:15:14	Auth ID:	004585
POS Condition Code:	Customer Present	Auth Source:	Online
POS Entry Mode:	Chip Read Chip Device	Authorized by:	BASE24
STAN:	109787	Settle to:	Banamex
Acquiring Inst ID:	1133538	Acquiring Inst ID 2:	
Transaction ID:		Receipt Number:	
Settlement Completion:	No	Transaction Currency Code:	Mexican Peso (484)
Fraud Authorizer:		Check Rating:	
Fraud Response Code:		Settlement Date:	

Revisando la transacción que se realizó de manera simultánea V261022001109787 podemos darnos cuenta que tiene el mismo número de tarjeta que la venta V261022001109788, también presenta un estatus de cancelada y el monto de la transacción es por la cantidad de \$105 que sumado a el monto de la transacción V261022001109788 de \$400 nos da el total de \$505 que podemos ver en los logs de la pinpad y que aprobó el integrador ACI

Autor	Versión	Fecha de elaboración	Última edición
Angel Fernando	1.0	05/09/2025	11/09/2025

Causa del problema

Revisando el código fuente de la versión 1000560712 donde ocurrió el reporte podemos ver lo siguiente:

PUNTO DE ENTRADA: Método LoadPaymentMethodView()

Cuando el cajero presionó el botón para proceder al pago, se ejecutó este método

```
#region refactor
/// <summary>
/// Prepara la venta para pasar a las formas de pago.
/// </summary>
[ 7 usages  ↳ Juan Murillo +4
private void LoadPaymentMethodView()
{
    this.Sale.RecalculateTotals();
    Log.Write("Ejecutando LoadPaymentMethodView...");
    this.paymentsViewNavigationCanceled = false;

    if (this.Sale.RealSaleDetailList.Any() && this.Sale.RealSaleDetailList.Count() == 1
        && this.Sale.RealSaleDetailList.Any(x :SaleDetailModel) => x.RelatedProductID == PosApplicationContext.Instance.CodArticuloRecargaxVenta))
    {
        this.NotificationManager.ShowMessage( title: Messages.NotificationHeaderText, message: Messages.NotAddedProductForSale, image: MessageIcon.Warning);
        return;
    }
}
```

Aquí recalcula los totales de la venta en este punto: Sale.Total = \$505.00 (productos + recarga)

Despues el código en el método UpSertSale demuestra que se realizó una actualización en el código para hacer una separación de la venta de forma automática cuando se trata de una venta normal y una recarga haciendo clonación y separación de productos normales:

```
//Proceso para dividir la venta en dos partes (Venta Normal y Venta Recargas)
if (PosApplicationContext.Instance.GridVentaArticulosConRecarga)
{
    SaleModel saleNormales = new SaleModel();
    SaleModel saleRecargas = new SaleModel();

    //Clonamos el objeto Sale y eliminamos el articulo de recargas para que queden solo los normales y
    despues recalculamos valores
    saleNormales = sale.CustomDeepClone();

    //si hay en el grid de ventas una recarga activa, la elimina del listado saleNormales
    if (!string.IsNullOrEmpty(PosApplicationContext.Instance.CodArticuloRecargaxVenta))
    {
        // Validar que la lista no sea nula o vacia
        if (saleNormales.SaleDetailList != null && saleNormales.SaleDetailList.Any())
        {
            var index = saleNormales.SaleDetailList.ToList().FindIndex(x => x.RelatedProductID == PosApplicationContext.Instance.CodArticuloRecargaxVenta);
            if (index >= 0) // Validar que el indice sea válido
            {
                saleNormales.SaleDetailList.RemoveAt(index);
            }
        }
        if (saleNormales.RealSaleDetailList != null && saleNormales.RealSaleDetailList.Any())
        {
```

Autor	Versión	Fecha de elaboración	Última edición
Angel Fernando	1.0	05/09/2025	11/09/2025

```

        var index = saleNormales.RealSaleDetailList.ToList().FindIndex(x => x.RelatedProductID ==
PosApplicationContext.Instance.CodArticuloRecargaxVenta);
        if (index >= 0) // Validar que el indice sea válido
        {
            saleNormales.RealSaleDetailList.RemoveAt(index);
        }
    }
    saleNormales.RecalculateTotals();

    if (PosApplicationContext.Instance.VentasNormalesyRecargasSeparadas.Any(x => x.TipoVenta == "VentaNormales"))
    {
        this.SaleNormal.SaleIdentifier =
PosApplicationContext.Instance.VentasNormalesyRecargasSeparadas.FirstOrDefault(x => x.TipoVenta ==
"VentaNormales").Idticket;
        this.SaleNormal.TransactionCode =
PosApplicationContext.Instance.VentasNormalesyRecargasSeparadas.FirstOrDefault(x => x.TipoVenta ==
"VentaNormales").IdTransaccion;
    }

    SaveTransactionResult resultNormales = SaveTransactionResult.Successful;
    //Si no hay registros de venta normal, se genera una transaccion de lo contrario se actualiza
    var saleResultN = TransactionServiceClient.SaveTransaction(saleNormales, out resultNormales);

    if (resultNormales == SaveTransactionResult.Successful)
    {
        if (saleResultN.Result.CurrentEWallet != null && saleResultN.Result.CurrentEWallet.ErrorCode ==
EWallet.Models.Enum.EWalletErrorCode.LockedCardToRedeem
        && saleResultN.Result.CurrentEWallet.IsLockedToRedeemByArticle)
        {
            saleResultN.Result.CurrentEWallet.IsLockedToRedeemByArticle = false;
            this.ValidateEWalletLockedForRedemption();
        }

        if (saleNormales.CouponInformation != null && saleNormales.CouponInformation.Any()
        && !saleResultN.Result.SaleDetailList.Where(f => sale.CouponInformation.Any(z => z.PromotionId ==
f.PromotionId)).Any())
        {
            this.NotificationManager.ShowMessage(Messages.TitleOfWarning, Messages.PromotionDuplicate,
MessageIcon.Warning);
        }

        bool replacePromotion = false;
        promotionIds.ForEach(promo =>
        {
            if (!saleResultN.Result.SaleDetailList.Any(p => p.PromotionId == promo.PromotionId))
            {
                replacePromotion = true;
            }
        });
        if (replacePromotion)
        {
            this.NotificationManager.ShowMessage(Messages.TitleOfWarning, Messages.PromotionReplacePL,
MessageIcon.Warning);
        }
    }
}

```

en ese método podemos ver como guarda la venta de productos normales en las líneas

```

SaveTransactionResult resultNormales = SaveTransactionResult.Successful;
//Si no hay registros de venta normal, se genera una transaccion de lo contrario se actualiza
var saleResultN = TransactionServiceClient.SaveTransaction(saleNormales, out resultNormales);

```

Que es donde se retorna la venta V261022001109787 como SaleIdentifier = "V261022001109787"

Y se registra en memoria esta primera venta para después continuar con Preparación y guardado de la recarga

Autor	Versión	Fecha de elaboración	Última edición
Angel Fernando	1.0	05/09/2025	11/09/2025

```
//Clonamos el objeto Sale y eliminamos los articulos normales para que quede solo el de recargas y despues
//recalculemos valores
saleRecargas = sale.CustomDeepClone();

if (!string.IsNullOrEmpty(PosApplicationContext.Instance.CodArticuloRecargaxVenta))
{
    // Validar SaleDetailList
    if (saleRecargas.SaleDetailList != null && saleRecargas.SaleDetailList.Any())
    {
        for (int i = saleRecargas.SaleDetailList.Count - 1; i >= 0; i--)
        {
            if (saleRecargas.SaleDetailList[i].RelatedProductID != PosApplicationContext.Instance.CodArticuloRecargaxVenta)
            {
                saleRecargas.SaleDetailList.RemoveAt(i);
            }
        }
    }

    // Validar RealSaleDetailList
    if (saleRecargas.RealSaleDetailList != null && saleRecargas.RealSaleDetailList.Any())
    {
        for (int i = saleRecargas.RealSaleDetailList.Count - 1; i >= 0; i--)
        {
            if (saleRecargas.RealSaleDetailList[i].RelatedProductID != PosApplicationContext.Instance.CodArticuloRecargaxVenta)
            {
                saleRecargas.RealSaleDetailList.RemoveAt(i);
            }
        }
    }
}

//Se indica que venta con articulos + recarga, el modelo de recarga sera de tipo TA / Servicios
saleRecargas.PaymentType = "TA";
saleRecargas.RecalculateTotals();

//Si no hay registros de venta de servicio recarga, se genera una transaccion
//Se envia a guardar la venta de la recarga
SaveTransactionResult resultRecarga;
saleRecargas.CurrentEWallet = null;
saleRecargas.SaleIdentifier = null;
saleRecargas.TransactionCode = null;

var saleResultR = TransactionServiceClient.SaveTransaction(saleRecargas, out resultRecarga);

var resultUpSertR = new ResultStruct<bool>(resultRecarga == SaveTransactionResult.Successful);
```

Ahi es cuando el servidor guarda y retorna la segunda venta de la recarga SaleIdentifier = "V261022001109788".

```
resultUpSertR.AddErrorItemList(saleResultR.ErrorItemList);

if (resultRecarga == SaveTransactionResult.Successful)
    PosApplicationContext.Instance.VentasNormalesyRecargasSeparadas.Add(new
PosApplicationContext.SaleNormalRecargasModel
{
    TipoVenta = "VentaRecarga",
    IdTransaccion = saleResultR.Result.TransactionCode,
    Idticket = saleResultR.Result.SaleIdentifier,
    MontoRecarga = System.Math.Round(saleRecargas.Total, 2)
});

this.SaleRecharge = saleRecargas;

}

this.SaleNormal = saleNormales;
this.Sale = saleNormales;
this.Sale.RecalculateTotals();
```

			
Autor	Versión	Fecha de elaboración	Última edición
Angel Fernando	1.0	05/09/2025	11/09/2025

Después de guardar ambas transacciones por separado continuamos en el método BuildPaymentMethodParameters() donde detecta que hay ventas separadas

```
var ventasSeparadas = PosApplicationContext.Instance.VentasNormalesyRecargasSeparadas.Any();
var gridConRecarga = PosApplicationContext.Instance.GridVentaArticulosConRecarga;

Log.Write($"[{nameof(SaleViewModel)}.{nameof(BuildPaymentMethodParameters)}] (9468): ### Se cargara el
Modelo de venta - ({ventasSeparadas ? "Recarga + Normal" : "Normal"})");
navParameters.Add("SaleNormal", ventasSeparadas ? this.SaleNormal : this.Sale);

Log.Write($"[{nameof(SaleViewModel)}.{nameof(BuildPaymentMethodParameters)}] (380): ### Seteo el Modelo de
venta SaleNormal dentro de navParameters");
```

aquí podemos ver que están los logs que confirman que fue el flujo que siguieron las transacciones:

```
[2025-08-25 20:10:45:762],FARMAX_POS,CORE,ERROR,[BUILDPAYMENTMETHODPARAMETERS],[SaleViewModel.BuildPaymentMethodParameters](9468): ### Se cargara el Modelo de venta - Recarga + Normal
[2025-08-25 20:10:45:762],FARMAX_POS,CORE,ERROR,[BUILDPAYMENTMETHODPARAMETERS],[SaleViewModel.BuildPaymentMethodParameters](380): ### Seteo el Modelo de venta SaleNormal dentro de navParameters
[2025-08-25 20:10:45:763],FARMAX_POS,CORE,ERROR,[ISNAVIGATIONTARGET],[SaleViewModel.PaymentMethodView] (9468): ### Se cargara el Modelo de venta SaleNormal dentro de la pantalla PaymentMethodView
[2025-08-25 20:10:45:763],FARMAX_POS,CORE,ERROR,[ISNAVIGATIONTARGET],[SaleViewModel.PaymentMethodView] (386): ### Se cargara el Modelo de venta SaleNormal
[2025-08-25 20:10:45:763],FARMAX_POS,CORE,ERROR,[ISNAVIGATIONTARGET],[PaymentMethodViewModel.IINavigationTarget] (380): ### Seteo el Modelo de venta SaleNormal
[2025-08-25 20:10:45:763],FARMAX_POS,CORE,ERROR,[ISNAVIGATIONTARGET],[PaymentMethodViewModel.IINavigationTarget] (386): ### Se cargara el Modelo de venta TenderManager SaleNormal
[2025-08-25 20:10:50:835],FARMAX_POS,CORE,ERROR,[APPLYPAYMENTS],[TenderManager.ApplyPayments] (740): Aplicando pago TARJETA DE DEBITO ACI (AD)
[2025-08-25 20:10:50:836],FARMAX_POS,CORE,ERROR,[APPLYPAYMENTS],[TenderManager.ApplyPayments] (756): Intentando pagar TARJETA DE DEBITO ACI (AD).
```

Y vemos que combina ambas para el cobro total de \$505.00

```
this.SaleTenderManager.Sale = this.Sale;

if (gridConRecarga)
    navParameters.Add("SaleRecharge", this.SaleRecharge);

if (ventasSeparadas && gridConRecarga)
{
    this.SaleTenderManager.Sale.SaleDetailList.Add(this.SaleRecharge.SaleDetailList.FirstOrDefault());
    this.SaleTenderManager.Sale.UpdateModelTotals();
}

navParameters.Add("Parameters", paramters);
navParameters.Add("TransactionTenderManager", this.SaleTenderManager);

return navParameters;
}
```

después en el método IsNavigationTarget Recibe ambos modelos de venta
SaleModel = venta de \$400.00

SaleRechargeModel = venta de \$105.00

Suma ambos totales para el cobro \$400 + \$105 = \$505.00

```
if (PosApplicationContext.Instance.VentasNormalesyRecargasSeparadas.Any())
{
    Log.Write($"[{nameof(PaymentMethodViewModel)}.{nameof(IsNavigationTarget)}] (386): ### Se cargara el
Modelo de venta SaleModel y SaleRechargeModel, en venta mas recarga");

    this.SaleModel = (SaleModel)navigationContext.Parameters["SaleNormal"];
    this.SaleRechargeModel = PosApplicationContext.Instance.GridVentaArticulosConRecarga ?
(SaleModel)navigationContext.Parameters["SaleRecharge"] : null;

    if (this.SaleRechargeModel != null)
        parameters.Total += this.SaleRechargeModel.Total;
}
```

			
Autor	Versión	Fecha de elaboración	Última edición
Angel Fernando	1.0	05/09/2025	11/09/2025

lo cual también podemos validar en los logs que fue lo que ocurrió:

```
[2025-08-25 20:10:45:763],FARMAX_POS_CORE,ERROR,[INVESTIGATIONTARGET],[PaymentMethodTypeModel] [NavigationTarget](386): ### Se cargara el Modelo de venta SaleModel y SaleRechargeModel, en venta mas recarga
[2025-08-25 20:10:45:763],FARMAX_POS_CORE,ERROR,[APPLYPAYMENTS],[TenderManager.ApplyPayments](740): Aplicando pago TARJETA DE DEBITO ACI (AD)
[2025-08-25 20:10:50:836],FARMAX_POS_CORE,ERROR,[APPLYPAYMENTS],[TenderManager.ApplyPayments](846): Intentando pagar TARJETA DE DEBITO ACI (AD).
[2025-08-25 20:10:51:804],FARMAX_POS_CORE,ERROR,[APPLYPAYMENTS],[TenderManager.ApplyPayments](805): Aplicando TARJETA DE DEBITO ACI (AD).
[2025-08-25 20:10:51:804],FARMAX_POS_CORE,ERROR,[APPLY],[Banktender.AcnInv](1137): Aplicando TARJETA DE DEBITO ACI (AD).
```

PROCESAMIENTO EN LA PINPAD

Un único cobro de \$505.00

Autorización única: 004595

Dos referencias internas: V261022001109787 y V261022001109788

Cuando se procesó la reversa:

El sistema detectó que había dos transacciones vinculadas

Aplicó la reversa a V261022001109787 (productos)

Aplicó la reversa a V261022001109788 (recarga)

Se procesó una reversa total de \$505.00

Resumen de flujo:

1. Cajero ingresa venta mixta → Total \$505
2. Sistema detecta configuración GridVentaArticulosConRecarga = true
3. Clona venta y crea saleNormales (\$400)
4. Guarda saleNormales → Genera V261022001109787
5. Clona venta y crea saleRecargas (\$105)
6. Guarda saleRecargas → Genera V261022001109788
7. Envía ambos modelos a formas de pago
8. Suma totales para pinpad: \$400 + \$105 = \$505
9. Pinpad cobra \$505 con autorización 004595
10. Ocurre reversa
11. Sistema reversa V261022001109787
12. Sistema reversa V261022001109788
13. Confirma reversa total de \$505

			
Autor	Versión	Fecha de elaboración	Última edición
Angel Fernando	1.0	05/09/2025	11/09/2025

Conclusión

El comportamiento observado el 25/08/2025 a las 20:15:13.627 NO ES UN ERROR, sino el FUNCIONAMIENTO ESPERADO del sistema según el nuevo requerimiento EDPU_Recargas implementado en la versión 10.5.60705.0.

En este caso cuando una venta contiene:

Productos farmacéuticos (TARJETA DE DEBITO ACI - SKU 2610)

Servicios de telecomunicaciones (Recarga PAQUETE AMIGO SL 400)

El sistema automáticamente las SEPARA EN DOS TRANSACCIONES DISTINTAS, cada una con su propio identificador.