 <div>Ventas con status diferentes</div>		
Autor	Versión	Fecha de elaboración
Juan Pérez	1.0	17-07-2024

Descripción general


Se detectan dos ventas con folio (**V20792000036839**, **V12282200020032**) donde encontramos que el movimiento del pago se encuentra como cancelado esto registra un identificador donde el campo es **codigoMovimientoCaja** en la tabla **genMovimientosCajaDet**.

	(No column name)	codigoMovimientoCaja	codigoFormaPago	importePago	importeComisionBancaria	importeIVA	tipoDeCambio	autorizacion	documento	estatusFormaPago
1	genMovimientosCajaDet	CF9D34A8-9C17-EF11-B814-C8D9D21CD986	12	401.0000	0.0000	0.0000	1.0000	NULL	NULL	C

Al continuar el análisis en la base de datos se obtuvo el registro total de ventas realizadas por el corte de turno y se encuentra que para estas dos ventas se registraron dos veces, pero con diferente **codigoMovimientoCaja** donde uno de los registros su forma de pago ya está con estatus **A = Aplicada**.

	codigoMovimientoCaja	codigoTurno	folioOperacion	folioCorte	tipoTransaccion	fechaActualizacion	codigoFormaPago	importePago	estatusFormaPago
17	C73312D8-9D17-EF11-B2F8-C8D9D21CD986	557A0E8B-7117-EF11-B814-C8D9D21CD986	V20792000036839	CT2079200000703	V	2024-05-21 12:13:42.647	12	401.0000	A
64	CF9D34A8-9C17-EF11-B814-C8D9D21CD986	557A0E8B-7117-EF11-B814-C8D9D21CD986	V20792000036839	CT2079200000703	V	2024-05-21 12:05:14.283	12	401.0000	C


Como se muestra en las dos imágenes se puede visualizar que el primer registro de la venta fue a las **2024-05-21 12:05 p.m.** y el segundo a las **2024-05-21 12:13:42 p.m.** tenemos una diferencia de 8 minutos, el detalle aquí es que no actualizo el segundo movimiento donde la forma de pago ya está como aplicada.

	Ventas con status diferentes	
Autor	Versión	Fecha de elaboración
Juan Pérez	1.0	17-07-2024

Flujo afectado

El flujo afectado en este caso, es el flujo de ventas, en el flujo de ventas la venta sufre distintos cambios de estado, los cambios de estado se presentan de la siguiente manera:

- (Etapa 1) N: Venta sin pagos o nueva**
 Se han seleccionado los productos y **se procede a la pantalla de métodos de pago**, en esta transición la venta se persiste en base de datos con **estado N** y **aún no se ha creado un movimiento de caja**.
- (Etapa 2) R: Pendiente de pagos**
Se seleccionan los métodos de pago y **se empiezan a aplicar/procesar**, la venta permanece en este estado hasta que todos los métodos de pago sean procesados, al iniciar el procesamiento de los métodos de pago la venta se persiste en base de datos con el **estado R** y **se crea su respectivo movimiento de caja**, se crean también los detalles de movimiento de caja, el estado inicial de un detalle de movimiento de caja es **P** hasta que este sea procesado por completo.
- (Etapa 3) V: Válida/Finalizada**
 La venta **se ha procesado con éxito**, ya sea con todos los métodos de pagos aplicados o algunos métodos de pagos cancelados y otros aplicados, los métodos de pago que se intentaron utilizar o que quedaron pendientes (**P**) se cancelan.

	Ventas con status diferentes	
Autor	Versión	Fecha de elaboración
Juan Pérez	1.0	17-07-2024


Diagnóstico

El problema es **debido a un apagón** en medio de la etapa **(Etapa 2)** del flujo de ventas, se apaga tanto la caja como la máquina de la sucursal causando que la transacción se persista con estado **R** y su respectivo **movimiento de caja** en la **base de datos de la caja** pero el proceso de sincronización de **caja → sucursal** no logra sincronizar los cambios a este estado antes de que ocurra el apagón por lo que la transacción en la base de datos de la sucursal se queda en la etapa **(Etapa 1)**, la transacción permanece con estado **N** y **sin movimiento de caja** en la **base de datos de la sucursal** y con estado **R** y **con movimiento de caja** en la **base de datos de la caja**, dando lugar a un problema de consistencia de datos.

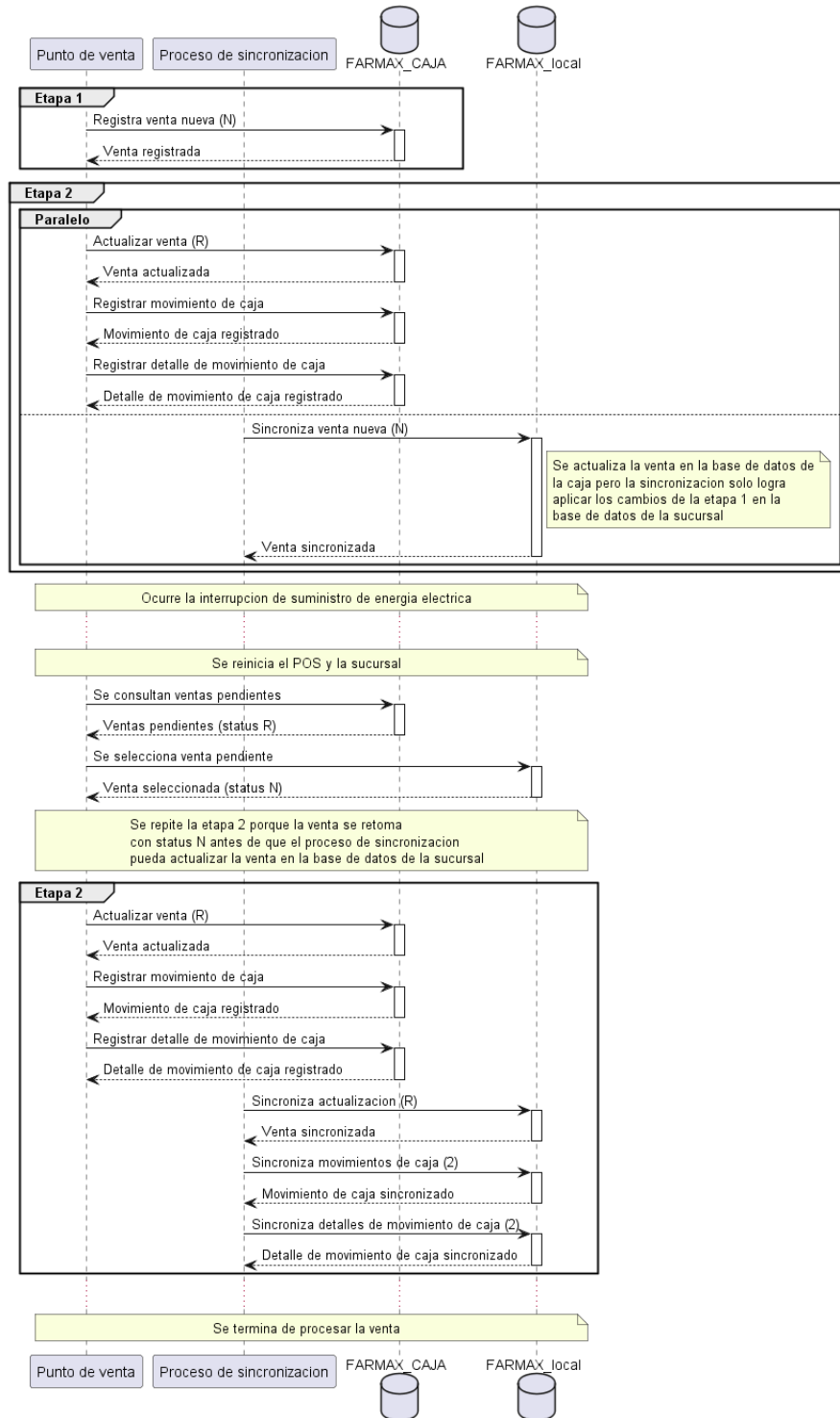
Esto quiere decir que el apagón se dio cuando un método de pago ya había sido seleccionado, en este caso, la **tarjeta de débito ACI** con **codigo 12**, pero no se pudo procesar por el apagón, causando que el detalle de movimiento de caja creado con estado **P** no se llega a procesar, este luego se ve cancelado (**pasa a estado C**) al finalizar la venta **(Etapa 3)**.

Al reiniciar el punto de venta se retoma la transacción, consultando su estado desde la **base de datos de la sucursal**, por lo que retoma la venta con status **N** y **sin movimiento de caja**, lo que causa que se **repita el proceso de la etapa (Etapa 2)**.


En la siguiente sección se discute cómo se llega a esta conclusión.

	Ventas con status diferentes	
Autor	Versión	Fecha de elaboración
Juan Pérez	1.0	17-07-2024

A continuación se presenta un diagrama de secuencia de alto nivel de lo ocurrido:



[Ver en pantalla completa](#)

	Ventas con status diferentes	
Autor	Versión	Fecha de elaboración
Juan Pérez	1.0	17-07-2024

Procedimiento de diagnóstico

Se revisan los logs para determinar las posibles causas de la generación de un nuevo movimiento de caja para una misma transacción.


Se revisan los logs correspondientes a la **caja 20** de la **sucursal 2079**:

- Se revisa el log [20240521_FARMAX_TILL_SERVER_SALE.XLOG](#) para determinar el recorrido que tuvo la venta, desglosando lo encontrado a partir de la línea 807 hasta la línea 854, se tiene:

```
[2024-05-21 12:05:08:614],FARMAX_TILL_SERVER,SALE,INFO,,Venta :: Metodo:AddSale
FolioVenta:V20792000036839 Estado:N
[2024-05-21 12:05:08:621],FARMAX_TILL_SERVER,SALE,INFO,,Donativo ::
Metodo:UpdateSaleHeader FolioVenta:V20792000036839 Estado:N Monto:
IdBeneficiencia: FolioDonativo:
[2024-05-21 12:05:08:621],FARMAX_TILL_SERVER,SALE,INFO,,Venta ::
Metodo:UpdateSaleHeader FolioVenta:V20792000036839 Estado:N
[2024-05-21 12:05:08:627],FARMAX_TILL_SERVER,SALE,INFO,,Venta ::
Metodo:UpdateSaleAfterWriteSaleDetails FolioVenta:V20792000036839
Estado:N
[2024-05-21 12:05:08:633],FARMAX_TILL_SERVER,SALE,INFO,,Donativo ::
Metodo:AddOrUpdateSale FolioVenta:V20792000036839 Estado:N Monto:
IdBeneficiencia: FolioDonativo:
[2024-05-21 12:05:08:633],FARMAX_TILL_SERVER,SALE,INFO,,Venta ::
Metodo:AddOrUpdateSale FolioVenta:V20792000036839 Estado:N
[2024-05-21 12:05:08:640],FARMAX_TILL_SERVER,SALE,INFO,,Donativo ::
Metodo:UpdateSaleHeader FolioVenta:V20792000036839 Estado:N Monto:
IdBeneficiencia: FolioDonativo:
[2024-05-21 12:05:08:641],FARMAX_TILL_SERVER,SALE,INFO,,Venta ::
Metodo:UpdateSaleHeader FolioVenta:V20792000036839 Estado:N
[2024-05-21 12:05:08:644],FARMAX_TILL_SERVER,SALE,INFO,,Venta ::
Metodo:UpdateSaleAfterWriteSaleDetails FolioVenta:V20792000036839
Estado:N
```

Se observa que la venta se crea correctamente, con el estado de **N (Nueva sin pagos)**, esto significa que se muestra la pantalla de métodos de pago pero que aún no se ha seleccionado un método de pago, luego se tiene:

```
[2024-05-21 12:05:14:347],FARMAX_TILL_SERVER,SALE,INFO,,Donativo ::
Metodo:UpdateSaleHeader FolioVenta:V20792000036839 Estado:R Monto:
IdBeneficiencia: FolioDonativo:
[2024-05-21 12:05:14:347],FARMAX_TILL_SERVER,SALE,INFO,,Venta ::
Metodo:UpdateSaleHeader FolioVenta:V20792000036839 Estado:R
```

	Ventas con status diferentes	
Autor	Versión	Fecha de elaboración
Juan Pérez	1.0	17-07-2024

```
[2024-05-21 12:05:14:353],FARMAX_TILL_SERVER,SALE,INFO,,Donativo ::
Metodo:UpdateSaleHeader FolioVenta:V20792000036839 Estado:R Monto:
IdBeneficiencia: FolioDonativo:
[2024-05-21 12:05:14:353],FARMAX_TILL_SERVER,SALE,INFO,,Venta ::
Metodo:UpdateSaleHeader FolioVenta:V20792000036839 Estado:R
[2024-05-21 12:05:14:357],FARMAX_TILL_SERVER,SALE,INFO,,Venta ::
Metodo:UpdateSaleAfterWriteSaleDetails FolioVenta:V20792000036839
Estado:R
```

Se observa que los métodos de pago se inician a procesar, por lo que la venta pasa a status **R (Pendiente de pagos)**, esto significa que un método de pago fue seleccionado y se encuentra en procesamiento, luego se tiene:


```
[2024-05-21 12:12:15:992],FARMAX_TILL_SERVER,SALE,INFO,,Venta ::
Metodo:GetTransactionsByStatus FolioVenta:V20792000036839 Estado:R
[2024-05-21 12:12:35:962],FARMAX_TILL_SERVER,SALE,INFO,,Venta ::
Metodo:GetTransactionsByStatus FolioVenta:V20792000036839 Estado:R
[2024-05-21 12:12:38:832],FARMAX_TILL_SERVER,SALE,INFO,,Venta ::
Metodo:GetTransactionsByStatus FolioVenta:V20792000036839 Estado:R
[2024-05-21 12:12:56:803],FARMAX_TILL_SERVER,SALE,INFO,,Venta ::
Metodo:GetTransaction FolioVenta:V20792000036839 Estado:N
[2024-05-21 12:13:31:981],FARMAX_TILL_SERVER,SALE,INFO,,Venta ::
Metodo:GetTransactionsByStatus FolioVenta:V20792000036839 Estado:R
```

Aquí ocurre algo extraño, el tiempo entre la última entrada de la sección anterior y la primera entrada de esta sección es de ~7 minutos, además de que el método **GetTransactionsByStatus** es utilizado para consultar transacciones y no para continuar procesando la venta, esto puede ser debido a un apagón, interrumpiendo el procesamiento de la venta y cerrando el POS, al abrir el POS nuevamente la venta se retoma como venta pendiente.

Cabe destacar también que los estados de la transacción entre los métodos de consulta **GetTransactionsByStatus** y **GetTransaction** no coinciden, siendo los estados obtenidos **R** y **N** respectivamente, por lo que puede ser indicio de problemas de consistencia de datos.

Se revisa qué bases de datos consultan estos métodos:

- El método **GetTransactionsByStatus(...)** se encuentra en **SalesDao.cs**, en **POS.Services**, y se utiliza con **SaleDbContext.cs**, en el método **GetTransactionsByStatus(...)** en **SaleManager.cs**:

	Ventas con status diferentes	
Autor	Versión	Fecha de elaboración
Juan Pérez	1.0	17-07-2024

```

2 usages Juan Murillo
public List<DT0.TransactionHeader> GetTransactionsByStatus(SaleFilter filter)
{
    using (var context = new SaleDbContext())
    {
        var saleDao = new SalesDao(context);
        var sales :List<TransactionHeader> = saleDao.GetTransactionsByStatus(filter);

        foreach (var item :TransactionHeader in sales)
        {
            item.status = item.status != SaleStatus.PaymentsPending && item.codigoMovimientoCaja != null
        }

        return sales.Select(this.NewTrnTransaccionesCab).Where(w :TransactionHeader => w != null).ToList();
    }
}

```

[Ver en pantalla completa](#)

Este utiliza el método de **CreateEntityConnection(...)** de **EntityConnectionCreator.cs** para definir su conexión:

```

70 usages Juan Murillo *
22 public SaleDbContext()
23 : base( existingConnection: EntityConnectionCreator.DefaultInstance, CreateEntityConnection
24 modelName: "POSSaleModel", Assembly.GetExecutingAssembly(), contextOwnsConnection: true)
25 {
26     this.Configuration.LazyLoadingEnabled = false;
27 }

```

[Ver en pantalla completa](#)


Este se conecta hacia la caja:

```

C# SadService.cs C# BusinessLogicGetOrderInfo.cs C# SalesDao.cs C# SaleManager.cs C# POSSaleModel.Context.cs C# EntityConnectionCreator.cs x
21 public sealed class EntityConnectionCreator
217 public EntityConnection CreateEntityConnection(string modelName, Assembly assembly)
222     valueFactory: k:string =>
225         paths:new string[]
229             "res://*/" + modelName + ".msl",
230         },
231         assembliesToConsider:new Assembly[] { assembly });
232     });
233
234     String connectionString = null;
235
236     if(_currentEnv == ContextEnvs.Store)
237     {
238         connectionString = _connectionTillString;
239     }
240     else
241     {
242         connectionString = _connectionCentralString;
243     }
244
245     return new EntityConnection(metadata, new SqlConnection(connectionString));
246 }
247

```

[Ver en pantalla completa](#)

	Ventas con status diferentes	
Autor	Versión	Fecha de elaboración
Juan Pérez	1.0	17-07-2024

- El método **GetTransaction** también definido en **SalesDao.cs**, este se utiliza en el método **GetTransaction(...)** en **SaleManager.cs**:

```

82      public partial class SaleManager
3343     public SaleTransaction GetTransaction(Guid saleId, string saleFolio, string folioDevolucion, int busin
3352         throw new BusinessException(message:"La base de datos de la sucursal no esta disponible");
3353     }
3354
3355     EntityConnectionCreator.ContextTypes contextToUse = EntityConnectionCreator.ContextTypes.Store;
3356
3357     using (var context = new SaleDbContext(contextToUse))
3358     {
3359         var saleDao = new SalesDao(context);
3360         int saleBusinessUnitId;

```

[Ver en pantalla completa](#)

Aquí también se utiliza **SaleDbContext** pero especificando que se conecte a la base de datos de la sucursal:

```

20      public partial class SaleDbContext : DbContext
31      {
32          public SaleDbContext(EntityConnectionCreator.ContextTypes context)
33              : base(existingConnection: EntityConnectionCreator.DefaultInstance.CreateEntityConnection(modelName:"POSSaleModel", Assembly.GetExecutingAssembly().context))
34          {
35              this.Configuration.LazyLoadingEnabled = false;

```


[Ver en pantalla completa](#)

```

21      public sealed class EntityConnectionCreator
258     public EntityConnection CreateEntityConnection(string modelName, Assembly assembly, ContextTypes context)
274
275
276     string connectionString = null;
277
278     if (_currentEnv == ContextEnvs.Store)
279     {
280         if(context == ContextTypes.Store)
281         {
282             connectionString = _connectionStoreString;
283         }
284         else
285         {
286             connectionString = _connectionTillString;
287         }
288     }
289     else
290     {
291         if(context == ContextTypes.CentralFLD )
292         {
293             connectionString = _connectionCentralFLDString;
294         }
295         else
296         {
297             connectionString = _connectionCentralString;
298         }
299     }

```


[Ver en pantalla completa](#)

	Ventas con status diferentes	
Autor	Versión	Fecha de elaboración
Juan Pérez	1.0	17-07-2024

Por lo que se puede concluir que hay una inconsistencia entre la base de datos de la caja y la sucursal, esto debido al apagón, esto se verifica en la siguiente sección.

- Se procede a revisar el log [20240521 FARMAX POS.XLOG](#), para confirmar el posible apagón en el POS, a partir de la línea 1093:

```
[2024-05-21 12:05:00:424],FARMAX_POS,,INFO,,ShellViewModel.ProcessPendingMovements
- No se encontraron transacciones con turno vacio
[2024-05-21
12:08:36:560],FARMAX_POS,,INFO,,*****
***** FARMAX_POS *****
[2024-05-21 12:08:36:872],FARMAX_POS,,INFO,,Log Level:Detail
[2024-05-21 12:08:36:872],FARMAX_POS,,INFO,,Log Time:Local
[2024-05-21 12:08:36:872],FARMAX_POS,,INFO,,Log Module:FARMAX_POS
[2024-05-21 12:08:36:872],FARMAX_POS,,INFO,,Operation System 64 bits:FALSE
[2024-05-21 12:08:36:872],FARMAX_POS,,INFO,,Current Process 64 bits:FALSE
[2024-05-21 12:08:36:872],FARMAX_POS,,INFO,,Computer Name:FA2079C10
[2024-05-21 12:08:36:872],FARMAX_POS,,INFO,,Platform:Win32NT ServicePack:Service
Pack 1 Version:Microsoft Windows NT 6.1.7601 Service Pack 1
[2024-05-21 12:08:36:872],FARMAX_POS,,INFO,,Processors:2
[2024-05-21 12:08:36:872],FARMAX_POS,,INFO,,Working
Directory:C:\ProgramData\FAhorro\FARMAX\InstalledVersions\10.5.60682.6\POS
[2024-05-21 12:08:36:872],FARMAX_POS,,INFO,,UserName:caja
[2024-05-21 12:08:36:872],FARMAX_POS,,INFO,,Domain:FA2079C10
[2024-05-21 12:08:36:872],FARMAX_POS,,INFO,,CLR Version:4.0.30319.42000
[2024-05-21
12:08:36:872],FARMAX_POS,,INFO,,*****
*****
[2024-05-21
12:10:56:574],FARMAX_POS,,INFO,,*****
***** FARMAX_POS *****
[2024-05-21 12:10:56:730],FARMAX_POS,,INFO,,Log Level:Detail
[2024-05-21 12:10:56:730],FARMAX_POS,,INFO,,Log Time:Local
[2024-05-21 12:10:56:730],FARMAX_POS,,INFO,,Log Module:FARMAX_POS
[2024-05-21 12:10:56:730],FARMAX_POS,,INFO,,Operation System 64 bits:FALSE
[2024-05-21 12:10:56:730],FARMAX_POS,,INFO,,Current Process 64 bits:FALSE
[2024-05-21 12:10:56:730],FARMAX_POS,,INFO,,Computer Name:FA2079C10
[2024-05-21 12:10:56:730],FARMAX_POS,,INFO,,Platform:Win32NT ServicePack:Service
Pack 1 Version:Microsoft Windows NT 6.1.7601 Service Pack 1
[2024-05-21 12:10:56:730],FARMAX_POS,,INFO,,Processors:2
[2024-05-21 12:10:56:730],FARMAX_POS,,INFO,,Working
Directory:C:\ProgramData\FAhorro\FARMAX\InstalledVersions\10.5.60682.6\POS
[2024-05-21 12:10:56:730],FARMAX_POS,,INFO,,UserName:caja
[2024-05-21 12:10:56:730],FARMAX_POS,,INFO,,Domain:FA2079C10
[2024-05-21 12:10:56:730],FARMAX_POS,,INFO,,CLR Version:4.0.30319.42000
[2024-05-21
12:10:56:730],FARMAX_POS,,INFO,,*****
*****
[2024-05-21 12:12:10:704],FARMAX_POS,,INFO,,Se considera que el usuario tiene una
sesion abierta. Cajero:181529 Caja:20 Empleado:181529
[2024-05-21 12:12:32:482],FARMAX_POS,,INFO,,Se considera que el usuario tiene una
sesion abierta. Cajero:181529 Caja:20 Empleado:181529
```

 <h2>Ventas con status diferentes</h2>		
Autor	Versión	Fecha de elaboración
Juan Pérez	1.0	17-07-2024

Según la primera entrada y la última, hubo una inactividad en el POS entre las 12:05:00 PM y las 12:12:32 PM.

Las últimas dos entradas señalan que el pos se reinició y el turno fue reutilizado:

```
[2024-05-21 12:12:10:704],FARMAX_POS,,INFO,,Se considera que el usuario tiene una
sesion abierta. Cajero:181529 Caja:20 Empleado:181529
[2024-05-21 12:12:32:482],FARMAX_POS,,INFO,,Se considera que el usuario tiene una
sesion abierta. Cajero:181529 Caja:20 Empleado:181529
```

Para determinar la causa de esta inactividad se revisan los eventos del sistema utilizando la utilidad de línea de comandos del Visor de de eventos de windows **wevtutil**, ejecutando el siguiente comando:

```
wevtutil qe System /q:"*[System[(EventID=1074 or EventID=1076 or EventID=6006 or
EventID=6008 or EventID=41) and TimeCreated[@SystemTime>='2024-05-21T00:00:00.000Z' and
@SystemTime<'2024-05-22T00:00:00.000Z']]]" /f:text
```

Este comando consulta los eventos que cumplan con el siguiente criterio:

- Que sean del sistema (**qe System**).
- Que el EventID sea cualquiera de los siguientes:
 - 1074: Indica que una aplicación ha iniciado el reinicio o apagado del sistema.
 - 1076: Provee mayor información sobre un reinicio o apagado del sistema.
 - 6006: Indica que el servicio del Event Log fue detenido.
 - 6008: Indica que el sistema se apaga de manera inesperada.
 - 41: Indica que el sistema se reinicia sin un apagado limpio.
- Que haya ocurrido en el transcurso del día 21-05-2024.

Se obtiene el siguiente resultado:

```
Event[3]:
Log Name: System
Source: Microsoft-Windows-Kernel-Power
Date: 2024-05-21T12:06:41.958
Event ID: 41
Task: N/A
Level: Critico
Opcode: Información
Keyword: N/A
User: S-1-5-18
User Name: NT AUTHORITY\SYSTEM
Computer: FA2079C10
Description:
Se reinició el sistema sin apagarlo limpiamente primero. Este error puede produc
irse si el sistema dejó de responder, se bloqueó o se interrumpió el suministro
eléctrico de forma inesperada.
C:\Windows\System32\config\systemprofile\
```

Copy

Queue: Juan Antonio Perez

Type: Session

Priority: Medium

Time in this Queue: 0:00:18

Customer Name: [Pinned] FA2079C10

Computer Name: FA2079C10

Operating System: Windows 7 Professio...

Support Issue:

Time in the System: 0:00:19

Uptime: 8:40:55

Required Skills:

IP Address: 10.111.2.4

Company Name:

Company Code:

External Key:


Details:

[Ver en pantalla completa](#)

Se guardan los resultados en el archivo [shutdown_logs.txt](#):

...

```
Event[2]:
Log Name: System
Source: EventLog
Date: 2024-05-21T12:06:59.000
```

	Ventas con status diferentes	
Autor	Versión	Fecha de elaboración
Juan Pérez	1.0	17-07-2024

Event ID: 6008
 Task: N/A
 Level: Error
 Opcode: N/A
 Keyword: Clásico
 User: N/A
 User Name: N/A
 Computer: FA2079C10
 Description:

El cierre anterior del sistema a las 12:04:45 p.m. del 21/05/2024 resultó inesperado.

Event[3]:
 Log Name: System
 Source: Microsoft-Windows-Kernel-Power
 Date: **2024-05-21T12:06:41.958**
 Event ID: 41
 Task: N/A
 Level: Crítico
 Opcode: Información
 Keyword: N/A
 User: S-1-5-18
 User Name: NT AUTHORITY\SYSTEM
 Computer: FA2079C10
 Description:

Se reinició el sistema sin apagarlo limpiamente primero. Este error puede producirse si el sistema dejó de responder, se bloqueó o se interrumpió el suministro eléctrico de forma inesperada.

Estos eventos demuestran que el equipo fue apagado de manera inesperada, probablemente por una interrupción de suministro eléctrico al equipo.


- Se revisa el log [20240521_FARMAX_TILL_SERVER_SYNCLIBRARY.XLOG](#) de sincronización para determinar si la base de datos de la sucursal logró sincronizar todos los cambios de la transacción, a partir de la línea 1942, se tiene:

[2024-05-21 12:05:12:609],FARMAX_TILL_SERVER,SYNCLIBRARY,INFO,,[SYNC_TILL_TX_DATA] Se guardaran 1 registro(s) de la tabla:'trnTransaccionesCab' en la base de datos de la sucursal

[2024-05-21 12:05:12:696],FARMAX_TILL_SERVER,SYNCLIBRARY,INFO,,[SYNC_TILL_TX_DATA] Se guardaran 5 registro(s) de la tabla:'trnTransaccionesDet' en la base de datos de la sucursal

[2024-05-21 12:13:06:397],FARMAX_TILL_SERVER,SYNCLIBRARY,INFO,,SYNC.TILL.SERVICE: Servicio iniciado

Esto indica que la base de datos recibió un registro de venta y 5 registros de detalles de venta a las 12:05:12 PM.

	Ventas con status diferentes	
Autor	Versión	Fecha de elaboración
Juan Pérez	1.0	17-07-2024

Cabe destacar que la base de datos no recibe un nuevo registro o actualizaciones debido al apagón, se ve en el log que este servicio se inicia nuevamente hasta las 12:13:06 PM.

Revisando nuevamente el log [20240521_FARMAX_TILL_SERVER_SALE.XLOG](#), en la línea 815, esto indica que la transacción recibida fue en estado N:


```
[2024-05-21      12:05:08:644],FARMAX_TILL_SERVER,SALE,INFO,,Venta      ::
Metodo:UpdateSaleAfterWriteSaleDetails FolioVenta:V20792000036839 Estado:N
[2024-05-21      12:05:14:347],FARMAX_TILL_SERVER,SALE,INFO,,Donativo  ::
Metodo:UpdateSaleHeader      FolioVenta:V20792000036839      Estado:R      Monto:
IdBeneficiencia: FolioDonativo:
[2024-05-21  12:05:14:347],FARMAX_TILL_SERVER,SALE,INFO,,Venta  :: Metodo:UpdateSaleHeader
FolioVenta:V20792000036839 Estado:R
```

Puesto que la transacción se actualiza a estado R a las 12:05:14 PM, esto quiere decir que el servicio de sincronización no logró recibir los últimos cambios de la transacción, permaneciendo en estado N en la base de datos de la sucursal.

Revisando nuevamente el log [20240521_FARMAX_TILL_SERVER_SYNCLIBRARY.XLOG](#) en la línea 1945:


```
[2024-05-21      12:14:06:683],FARMAX_TILL_SERVER,SYNCLIBRARY,INFO,,[SYNC_TILL_TX_DATA]  Se
guardaran 1 registro(s) de la tabla:'segUsuariosReg' en la base de datos de la sucursal
[2024-05-21      12:14:07:604],FARMAX_TILL_SERVER,SYNCLIBRARY,INFO,,[SYNC_TILL_TX_DATA]  Se
guardaran 1 registro(s) de la tabla:'genFormasPagosElectronicas' en la base de datos de la
sucursal
[2024-05-21      12:14:08:424],FARMAX_TILL_SERVER,SYNCLIBRARY,INFO,,[SYNC_TILL_TX_DATA]  Se
guardaran 1 registro(s) de la tabla:'venTurnosCajaReg' en la base de datos de la sucursal
[2024-05-21      12:14:08:674],FARMAX_TILL_SERVER,SYNCLIBRARY,INFO,,[SYNC_TILL_TX_DATA]  Se
guardaran 2 registro(s) de la tabla:'genMovimientosCajasCab' en la base de
datos de la sucursal
[2024-05-21      12:14:09:997],FARMAX_TILL_SERVER,SYNCLIBRARY,INFO,,[SYNC_TILL_TX_DATA]  Se
guardaran 2 registro(s) de la tabla:'genMovimientosCajasDet' en la base de
datos de la sucursal
[2024-05-21      12:14:12:577],FARMAX_TILL_SERVER,SYNCLIBRARY,INFO,,[SYNC_TILL_TX_DATA]  Se
guardaran 1 registro(s) de la tabla:'trnTransaccionesCab' en la base de
datos de la sucursal
[2024-05-21      12:14:13:467],FARMAX_TILL_SERVER,SYNCLIBRARY,INFO,,[SYNC_TILL_TX_DATA]  Se
guardaran 2 registro(s) de la tabla:'trnTransaccionesDet' en la base de datos de
la sucursal
```

Aquí se observa que se reciben los movimientos de caja y los detalles de movimientos de caja, en este caso fueron dos movimientos de caja, con un detalle cada uno, correspondientes a la misma transacción.

	Ventas con status diferentes	
Autor	Versión	Fecha de elaboración
Juan Pérez	1.0	17-07-2024

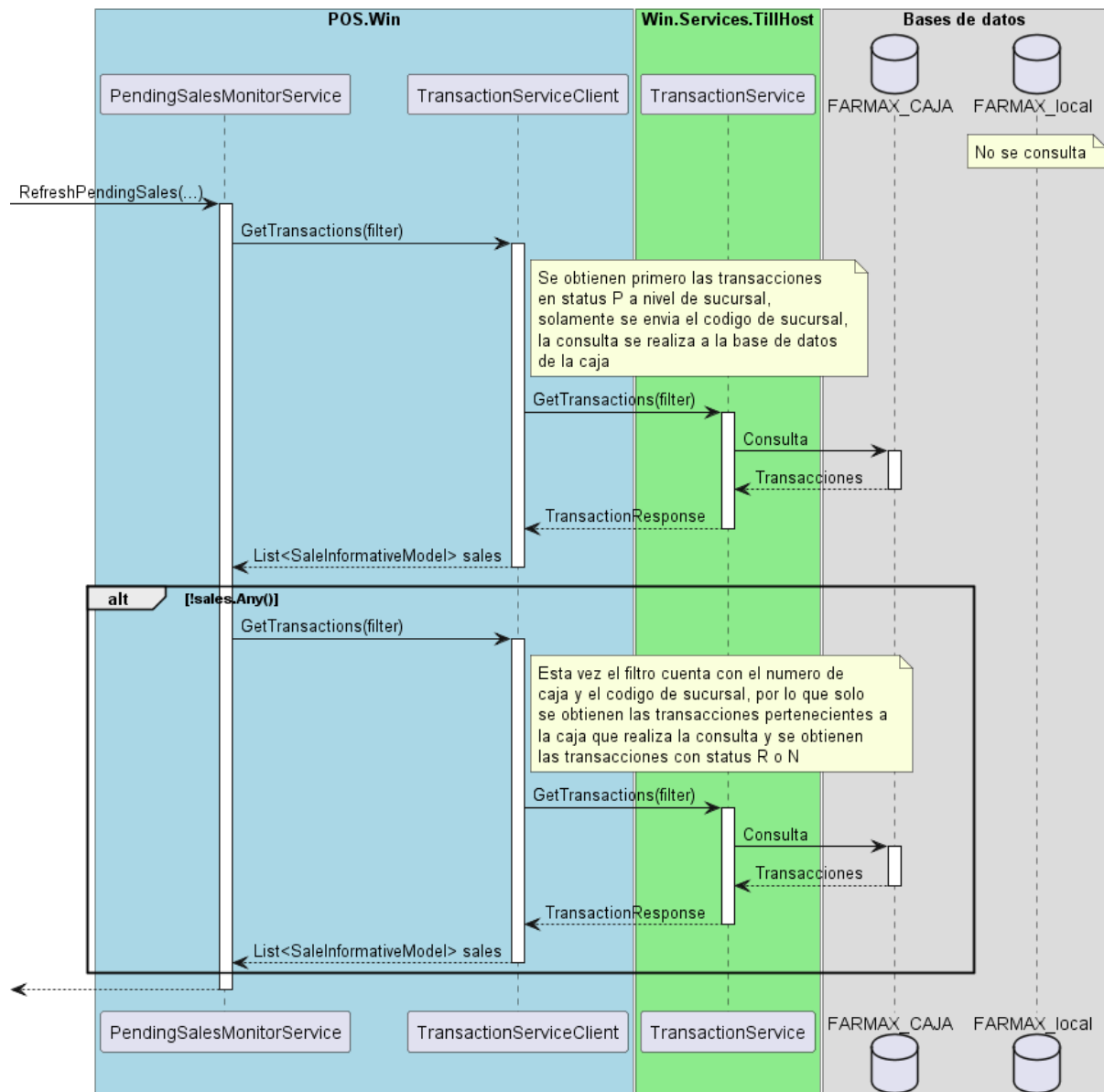
Posibles causas

Al consultar la venta pendiente a retomar desde la base de datos de la sucursal no se revisa si esta ya fue sincronizada, dando lugar a la posibilidad de retomar la venta con datos inconsistentes, **lo que causa que la (etapa 2) del flujo de ventas se repita.**


 <h1>Ventas con status diferentes</h1>		
Autor	Versión	Fecha de elaboración
Juan Pérez	1.0	17-07-2024

Posibles soluciones

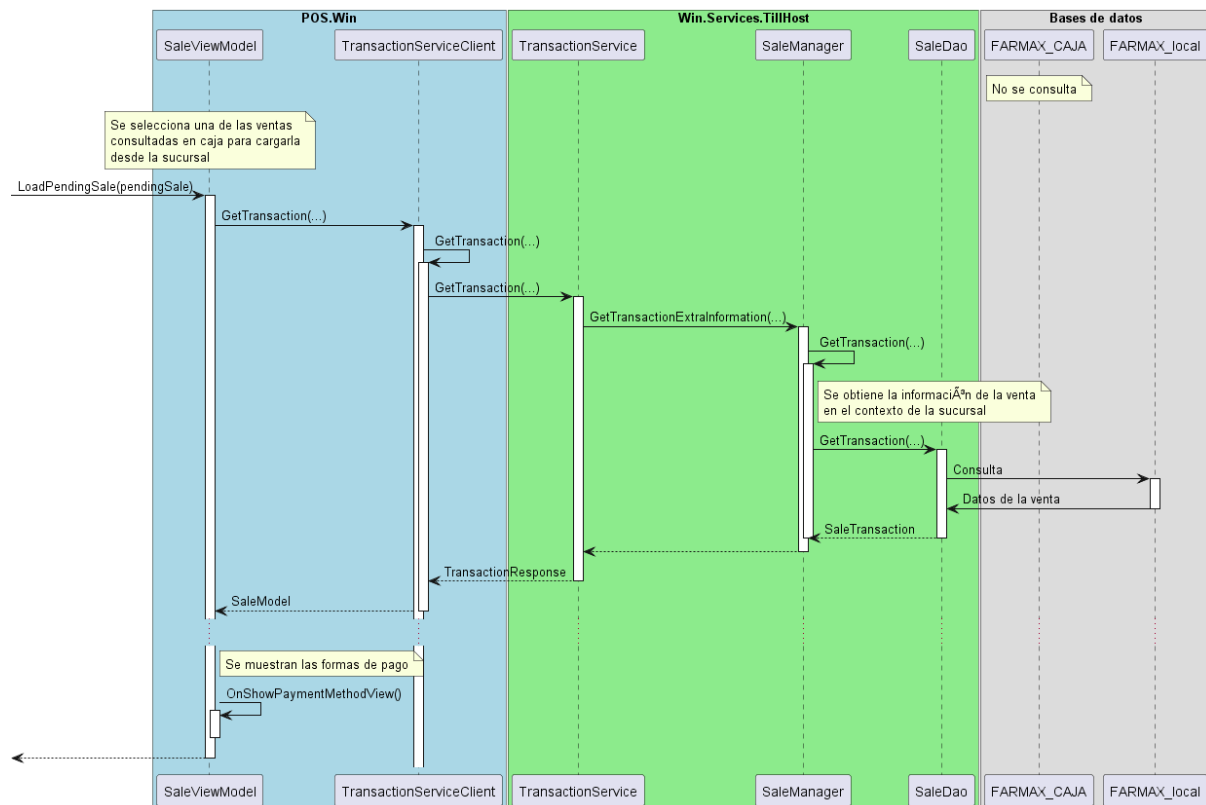
Al consultar las ventas pendientes primero se revisa la base de datos de la caja, esto ocurre al iniciar la aplicación de POS, al retomar una venta o al finalizar una venta, entre otras situaciones:



[Ver en pantalla completa](#)


 <h1>Ventas con status diferentes</h1>		
Autor	Versión	Fecha de elaboración
Juan Pérez	1.0	17-07-2024

Al seleccionar una venta se consulta la base de datos de la sucursal para poder obtener toda la información disponible de la transacción:



[Ver en pantalla completa](#)

En el metodo **LoadPendingSale(pendingSale)** de la clase **SaleViewModel** se recibe la venta seleccionada como **pendingSale**, esta instancia contiene la información de la venta con respecto a la **base de datos de la caja**, luego se consulta toda la información de la transacción en la **base de datos de la sucursal**, como se puede observar en el diagrama de secuencia, es posible realizar una validación comparando los estados de la transacción tanto en la base de datos de la caja como en la base de datos de la sucursal para evitar que se retome hasta que el proceso de sincronización pueda conseguir que la base de datos de la sucursal sea consistente con la de la caja:

	Ventas con status diferentes	
Autor	Versión	Fecha de elaboración
Juan Pérez	1.0	17-07-2024

```

C# SaleViewModel.cs x C# LogSubModules.cs C# INotificationManager.cs C# RemoteProxy.cs
122 public sealed class SaleViewModel : ViewModelBase, INavigationAware
6687 private void LoadPendingSale(PendingSaleModel pendingSale)
6695     var giftCardProduct = false;
6696     SaleModel pendingSaleModel;
6697     using (TransactionServiceClient client = new TransactionServiceClient())
6698     {
6699         pendingSaleModel = client.GetTransaction(idSale, transactionFolio: string.Empty, PosAppContext.Ins
6700     }
6701
6702     // Verifica si la transacción ya fue sincronizada con la base de datos de la sucursal
6703     if (pendingSale.Status != pendingSaleModel.Status)
6704     {
6705         this.NotificationManager.ShowMessage(
6706             title: "Advertencia",
6707             message: "Existe una inconsistencia entre la caja y la sucursal." +
6708                 Environment.NewLine +
6709                 "No es posible retomar la venta, intente de nuevo en un momento.",
6710             image: MessageIcon.Warning);
6711
6712         Log.WriteWarning(
6713             submodule: LogSubModules.SubModule_Sale,
6714             message: "Existe una inconsistencia entre la caja y la sucursal. " +
6715                 $"Estado en caja: {pendingSale.Status}, estado en sucursal: {pendingSaleModel.Status}");
6716
6717         return;
6718     }

```

[Ver en pantalla completa](#)

Esto verifica que el estado de la transacción sea la misma en ambas bases de datos antes de poder ser retomada, evitando así que se cree otro movimiento de caja y un detalle extra.