	Incidencia – Venta finalizada con forma de pago pendiente		
Autor	Versión	Fecha de elaboración	Fecha de última actualización
Juan Pérez	1.0	31-10-2024	31-10-2024

Control de versiones

Versión	**Fecha**	**Autor**	**Descripción**
1.0	31-10-2024	Juan Pérez	Se analiza el problema, se encuentra causa raíz y se propone una solución.

GET Technologies	Incidencia – Venta finalizada con forma de pago pendiente		
Autor	Versión	Fecha de elaboración	Fecha de última actualización
Juan Pérez	1.0	31-10-2024	31-10-2024

Descripción general

Se identifica una venta con folio **V25752100092966**, esta venta corresponde a la sucursal **2575** y a la caja **21**, la venta se realiza el **13 de octubre del año 2024**, se encuentra que la venta finaliza correctamente ya que es actualizada a su **status V (Válida)**, este **status** se alcanza una vez la venta se finaliza. 11 o

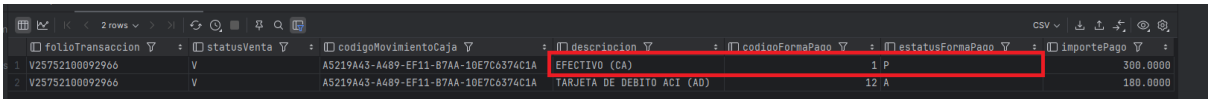
Esta venta se realiza utilizando dos formas de pago, la forma de pago de **“EFECTIVO (CA)”** con **id 1** y la forma de pago **“TARJETA DE DEBITO ACI (AD)”** con **id 12**.

Se encuentra que la forma de pago **“EFECTIVO (CA)”** permanece en status **P** aún cuando la venta se finaliza, es decir, la venta llega a **status V**, al llegar a este estado, las formas de pago deberían estar en **status A** si se procesan de manera correcta o **status C** si ocurre algún problema con la forma de pago específica.

Esto se evidencia utilizando la siguiente consulta:

```sql
DECLARE @folio NVARCHAR(15) = N'V25752100092966';

-- Venta - Cabecero
SELECT
   trn.folioTransaccion,
   trn.status AS statusVenta,
   trn.codigoMovimientoCaja,
   formaPago.descripcion,
   formaPago.codigoFormaPago,
   movDet.estatusFormaPago,
   movDet.importePago
FROM dbo.trnTransaccionesCab trn
        INNER JOIN genMovimientosCajasDet movDet
                   ON trn.codigoMovimientoCaja = movDet.codigoMovimientoCaja
        INNER JOIN genFormasDePagoCat formaPago
                   ON movDet.codigoFormaPago = formaPago.codigoFormaPago
WHERE trn.folioTransaccion = @folio;
```

Ver consulta



	folioTransaccion	statusVenta	codigoMovimientoCaja	descripcion	codigoFormaPago	estatusFormaPago	importePago
1	V25752100092966	V	A5219A43-A489-EF11-B7AA-10E7C6374C1A	EFECTIVO (CA)	1	P	300.0000
2	V25752100092966	V	A5219A43-A489-EF11-B7AA-10E7C6374C1A	TARJETA DE DEBITO ACI (AD)	12	A	180.0000

Ver figura

GET Technologies	Incidencia – Venta finalizada con forma de pago pendiente		
Autor	Versión	Fecha de elaboración	Fecha de última actualización
Juan Pérez	1.0	31-10-2024	31-10-2024

Diagnóstico

Revisión de logs

Flujo de venta

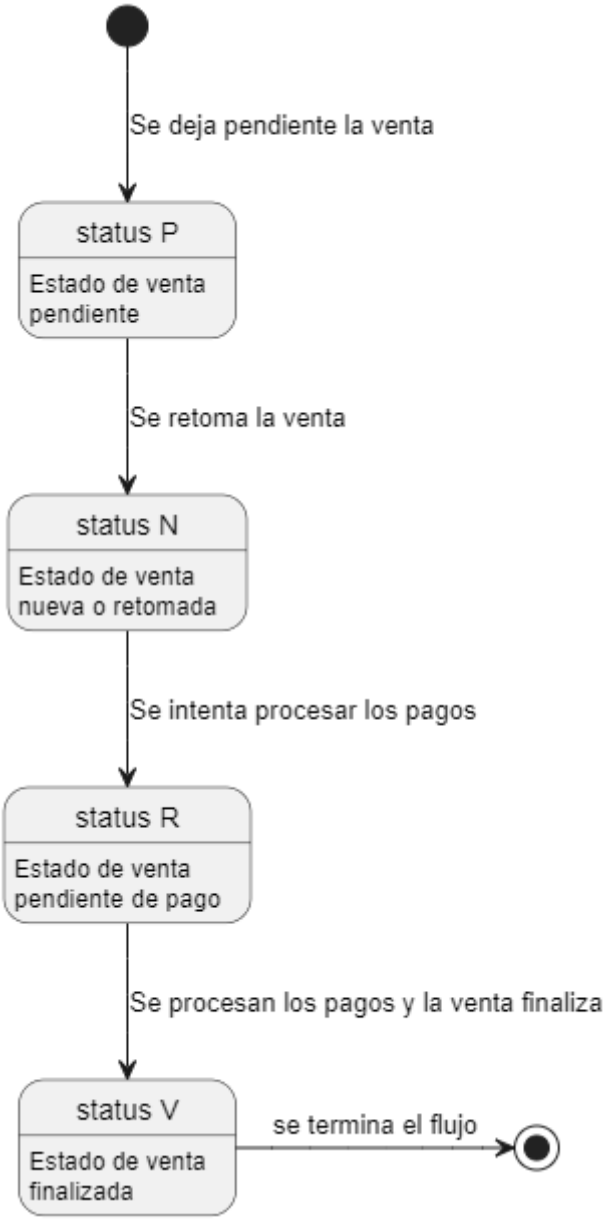
Se revisa el flujo que tuvo la venta al momento de ser procesada, para ello se revisa el log llamado 20241013_FARMAX_TILL_SERVER_SALE.XLOG, este log está ubicado en **%ProgramData%/FAhorro/FARMAX_TILL_SERVER** y contiene la traza de los distintos cambios de estado que puede sufrir una venta, aquí se puede observar que la venta sigue un flujo normal, los **"..."** significan que se omiten algunas líneas hasta llegar a la que se presenta después, se tiene a partir de la **línea 989**:

```
[2024-10-13 14:44:46:759],FARMAX_TILL_SERVER,SALE,INFO,,Venta :: Metodo:AddSale FolioVenta:V25752100092966 Estado:P

...

[2024-10-13 14:46:09:324],FARMAX_TILL_SERVER,SALE,INFO,,Venta :: Metodo:UpdateSaleStatus FolioVenta:V25752100092966 Estado:N

...

[2024-10-13 14:46:48:364],FARMAX_TILL_SERVER,SALE,INFO,,Donativo :: Metodo:UpdateSaleHeader FolioVenta:V25752100092966 Estado:R Monto: IdBeneficiencia: FolioDonativo:

...

[2024-10-13 14:47:43:965],FARMAX_TILL_SERVER,SALE,ERROR,[REMOVEPENDINGSALEINFORMATIONEWALLET],Excepción SaleManager-RemovePendingSaleInformationEwallet. Excepción:El valor no puede ser nulo.
       Nombre del parámetro: entity Stacktrace: en System.Data.Entity.Utilities.Check.NotNull[T](T value, String parameterName)
       en System.Data.Entity.DbSet`1.Remove(TEntity entity)
       en FAhorro.POS.Data.Sale.DAO.SalesDao.RemovePendingSaleInformationEwallet(trnTransaccionVentaPendiente pendingSale, Guid idTransaccion)
       en FAhorro.POS.BusinessLogic.Sale.SaleManager.RemovePendingSaleInformationEwallet(trnTransaccionVentaPendiente id, Guid idTransaccion)
[2024-10-13 14:47:43:969],FARMAX_TILL_SERVER,SALE,INFO,,Donativo :: Metodo:UpdateSaleHeader FolioVenta:V25752100092966 Estado:V Monto: IdBeneficiencia: FolioDonativo:
```

La excepción encontrada en medio del flujo de venta se explica más adelante, después del flujo de venta.

Aquí se observa que el flujo de la venta fue el siguiente:

GET Technologies	Incidencia – Venta finalizada con forma de pago pendiente		
Autor	Versión	Fecha de elaboración	Fecha de última actualización
Juan Pérez	1.0	31-10-2024	31-10-2024



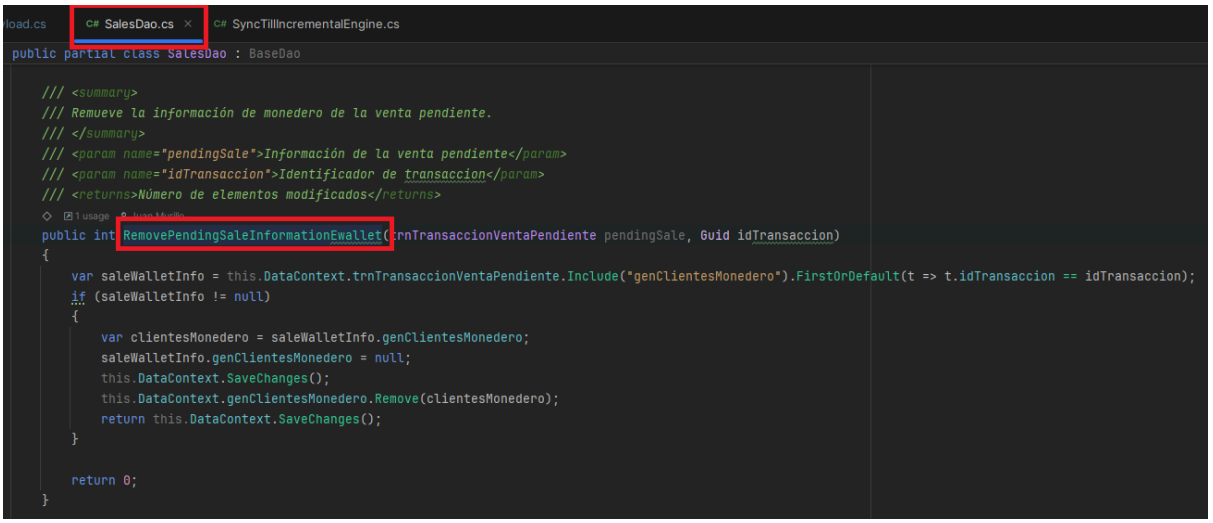
Ver diagrama

El flujo de la venta **no presenta nada fuera de lo habitual**, es el flujo que normalmente siguen las ventas pendientes al ser retomadas para su finalización, por lo que **no se detecta ninguna anomalía**, al menos con respecto a los cambios de estado de la venta.

Se observa también en el mismo log, a partir de la **línea 1023** que ocurre una excepción en medio del flujo de la venta:

```
[2024-10-13 14:47:43:965],FARMAX_TILL_SERVER,SALE,ERROR,[REMOVEPENDINGSALEINFORMATIONEWALLET],Excepción SaleManager-RemovePendingSaleInformationEwallet. Excepción:El valor no puede ser nulo.
        Nombre del parámetro: entity Stacktrace: en System.Data.Entity.Utilities.Check.NotNull[T](T value, String parameterName)
        en System.Data.Entity.DbSet`1.Remove(TEntity entity)
        en FAhorro.POS.Data.Sale.DAO.SalesDao.RemovePendingSaleInformationEwallet(trnTransaccionVentaPendiente pendingSale, Guid idTransaccion)
        en FAhorro.POS.BusinessLogic.Sale.SaleManager.RemovePendingSaleInformationEwallet(trnTransaccionVentaPendiente id, Guid idTransaccion)
```

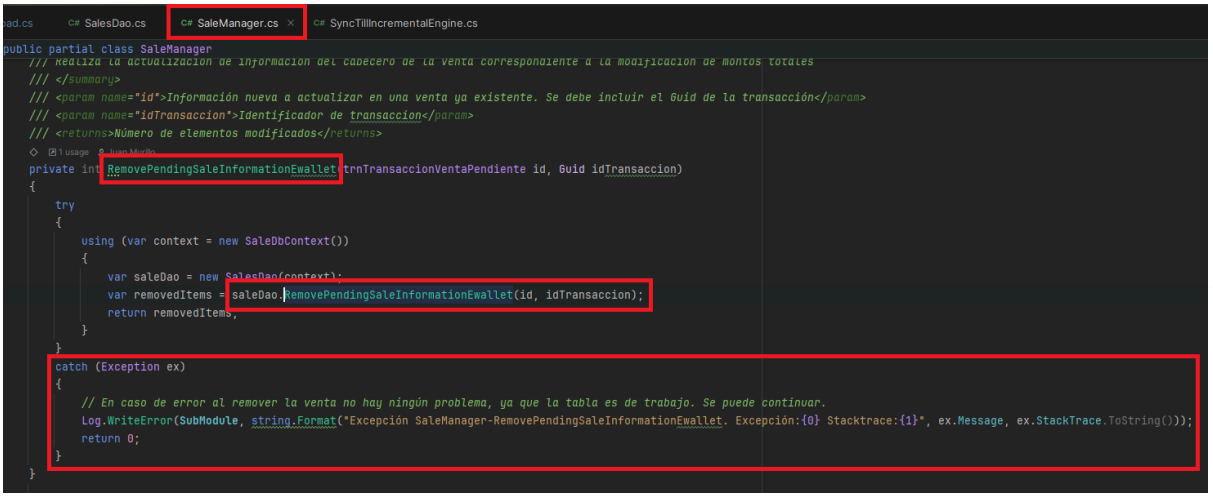
Esta excepción es manejada dentro del flujo y no causa una interrupción, el error se da en la solución **POS.Services**, en el proyecto **Win.Services.TillHost**, en el archivo **SalesDao.cs**:



Ver figura

Lo que hace este método es borrar el registro de **genClientesMonedero** relacionado a una venta pendiente, este solamente se llama en **SaleManager.cs**, en el método **RemovePendingSaleInformationEwallet(...)**, las excepciones que puedan ocurrir son manejadas dentro del mismo método por lo que cualquier flujo externo que lo utilice no se ve interrumpido por las posibles excepciones:

GET Technologies	Incidencia – Venta finalizada con forma de pago pendiente		
Autor	Versión	Fecha de elaboración	Fecha de última actualización
Juan Pérez	1.0	31-10-2024	31-10-2024



Ver figura

Se determina que la excepción no tiene un impacto en el flujo.

Ahora es necesario revisar el procesamiento de las formas de pago, se revisa el log llamado 20241013_FARMAX_POS_CORE.XLOG, este se ubica en **%ProgramData%/FAhorro/FARMAX_POS/**, este log puede contener las operaciones realizadas con las formas de pago, no es muy detallado pero se presentan las excepciones que pueden ocurrir al intentar procesar una forma de pago, se tiene a partir de la **línea 750**:

```
[2024-10-13                14:47:08:631],FARMAX_POS,CORE,ERROR,[APPLYPAYMENTS],#########(3)
QPSLimitAmount: '250', PaymentType: 'TDEB' payment.IsQPS='True'
[2024-10-13                14:47:08:631],FARMAX_POS,CORE,ERROR,[APPLYPAYMENTS],#########(4)
QPSLimitAmount: '250', PaymentType: 'TDEB' payment.IsQPS='True'
[2024-10-13 14:47:08:644],FARMAX_POS,CORE,ERROR,[APPLYPAYMENTS],ApplyPayments - Message: La
forma de pago TARJETA DE DEBITO ACI (AD) no pudo ser aplicado., StackTrace:       en
FAhorro.POS.Win.Tender.TenderManager.ApplyPayments()
[2024-10-13                14:47:43:093],FARMAX_POS,CORE,ERROR,[APPLYPAYMENTS],#########(3)
QPSLimitAmount: '250', PaymentType: 'TDEB' payment.IsQPS='True'
[2024-10-13                14:47:43:093],FARMAX_POS,CORE,ERROR,[APPLYPAYMENTS],#########(4)
QPSLimitAmount: '250', PaymentType: 'TDEB' payment.IsQPS='True'
[2024-10-13                14:47:43:093],FARMAX_POS,CORE,ERROR,[APPLYPAYMENTS],#########(3)
QPSLimitAmount: '250', PaymentType: 'EFEC' payment.IsQPS='False'
[2024-10-13                14:47:43:094],FARMAX_POS,CORE,ERROR,[APPLYPAYMENTS],#########(4)
QPSLimitAmount: '250', PaymentType: 'EFEC' payment.IsQPS='False'
[2024-10-13   14:49:07:556],FARMAX_POS,CORE,ERROR,[TRYGETBUSINESSERRORMESSAGE],Usuario   y/o
contraseña inválidos.
```

Se observa que la primera forma de pago que se aplica es la tarjeta de débito, se observa también que ocurrió un error en el primer intento de procesamiento para la forma de pago de tarjeta de débito, sin embargo, la tarjeta luego se procesa correctamente junto con el pago de efectivo, se observa que **no ocurrió una excepción al procesar el efectivo**.

Se revisa el log del pin pad 20241013_FARMAX_POS_ANXOPINPAD.XLOG, puesto que ocurrió una excepción en el primer intento de procesar el método de pago de tarjeta de

débito, el log se encuentra en **%ProgramData%/FAhorro/FARMAX_POS**, este contiene las operaciones realizadas en el pin pad, se observa a partir de la **línea 703**:

```
[2024-10-13 14:46:50:044],FARMAX_POS,ANXOPINPAD,INFO,,
        ------------------------------------------------
        CC1 request - card processing start
        ------------------------------------------------
[2024-10-13 14:46:50:044],FARMAX_POS,ANXOPINPAD,INFO,,
        Amount: $200
        Counter: 21
        Payment Type: Credit/debit tender ID: 12
        Transaction Code: Q QPS

[2024-10-13 14:46:59:673],FARMAX_POS,ANXOPINPAD,INFO,,TIMING1,ObtieneTarjeta,00:00:09.63
[2024-10-13 14:46:59:673],FARMAX_POS,ANXOPINPAD,INFO,,TIMING1,CierraVentana,00:00:00.00
[2024-10-13 14:46:59:674],FARMAX_POS,ANXOPINPAD,INFO,,
        ************************************************************
[2024-10-13 14:46:59:674],FARMAX_POS,ANXOPINPAD,INFO,,Loaded configuration shown
[2024-10-13 14:46:59:674],FARMAX_POS,ANXOPINPAD,INFO,, Implementation:          AnxoImplementation
[2024-10-13 14:46:59:674],FARMAX_POS,ANXOPINPAD,INFO,, Serial port:             9
[2024-10-13 14:46:59:674],FARMAX_POS,ANXOPINPAD,INFO,, Serial port PRESENT:     True
[2024-10-13 14:46:59:674],FARMAX_POS,ANXOPINPAD,INFO,, Serial port baud rate:  19200
[2024-10-13 14:46:59:674],FARMAX_POS,ANXOPINPAD,INFO,, Serial port parity:      None
[2024-10-13 14:46:59:674],FARMAX_POS,ANXOPINPAD,INFO,, Serial port data bits:  8
[2024-10-13 14:46:59:674],FARMAX_POS,ANXOPINPAD,INFO,, Serial port stop bits:  One
[2024-10-13 14:46:59:674],FARMAX_POS,ANXOPINPAD,INFO,, Timeout PINpad ACK:      5
[2024-10-13 14:46:59:674],FARMAX_POS,ANXOPINPAD,INFO,, Timeout PINpad Q4:       20
[2024-10-13 14:46:59:674],FARMAX_POS,ANXOPINPAD,INFO,, Timeout PINpad CC0:      30
[2024-10-13 14:46:59:674],FARMAX_POS,ANXOPINPAD,INFO,, Timeout PINpad CC1:      45
[2024-10-13 14:46:59:675],FARMAX_POS,ANXOPINPAD,INFO,, Timeout PINpad CC4 chip 45
[2024-10-13 14:46:59:675],FARMAX_POS,ANXOPINPAD,INFO,, Timeout PINpad CP0       30
[2024-10-13 14:46:59:675],FARMAX_POS,ANXOPINPAD,INFO,, Auth server:             192.168.152.17
[2024-10-13 14:46:59:675],FARMAX_POS,ANXOPINPAD,INFO,, Auth server port:        7000
[2024-10-13 14:46:59:675],FARMAX_POS,ANXOPINPAD,INFO,, Timeout auth:            30
[2024-10-13 14:46:59:675],FARMAX_POS,ANXOPINPAD,INFO,, Credit/debit tender ID: 12
[2024-10-13 14:46:59:675],FARMAX_POS,ANXOPINPAD,INFO,, AMEX tender ID:          14
[2024-10-13
14:46:59:675],FARMAX_POS,ANXOPINPAD,INFO,,************************************************************
[2024-10-13 14:47:00:032],FARMAX_POS,ANXOPINPAD,INFO,,TIMING2,SendAuthMessage,00:00:00.35,OK
[2024-10-13
14:47:00:049],FARMAX_POS,ANXOPINPAD,WARNING,,SUC:2575|CAJA:00000021|FOL:092966|TARJ:**********8414|MTO:20000|
ARQC:A1D8B43D3DC0614C|RC:51|AUT:000000
[2024-10-13 14:47:00:396],FARMAX_POS,ANXOPINPAD,WARNING,,AUTH NOT approved
[2024-10-13 14:47:03:675],FARMAX_POS,ANXOPINPAD,INFO,,TIMING1,EnviaPeticionVenta,00:00:03.99
[2024-10-13 14:47:03:675],FARMAX_POS,ANXOPINPAD,INFO,,TIMING1,CierraVentana,00:00:00.00
```

Se observa que el primer intento fue por el monto de **MX$200**, pero que este no fue aprobado.

Se revisa el siguiente intento, a partir de la línea 764:

```
[2024-10-13 14:47:28:323],FARMAX_POS,ANXOPINPAD,INFO,,
        ------------------------------------------------
        CC1 request - card processing start
        ------------------------------------------------
[2024-10-13 14:47:28:323],FARMAX_POS,ANXOPINPAD,INFO,,
        Amount: $180
        Counter: 21
        Payment Type: Credit/debit tender ID: 12
        Transaction Code: Q QPS

[2024-10-13 14:47:38:020],FARMAX_POS,ANXOPINPAD,INFO,,TIMING1,ObtieneTarjeta,00:00:09.69
[2024-10-13 14:47:38:020],FARMAX_POS,ANXOPINPAD,INFO,,TIMING1,CierraVentana,00:00:00.00
[2024-10-13 14:47:38:021],FARMAX_POS,ANXOPINPAD,INFO,,
        ************************************************************
```

```
[2024-10-13 14:47:38:021],FARMAX_POS,ANXOPINPAD,INFO,,Loaded configuration shown
[2024-10-13 14:47:38:021],FARMAX_POS,ANXOPINPAD,INFO,, Implementation:          AnxoImplementation
[2024-10-13 14:47:38:021],FARMAX_POS,ANXOPINPAD,INFO,, Serial port:             9
[2024-10-13 14:47:38:021],FARMAX_POS,ANXOPINPAD,INFO,, Serial port PRESENT:     True
[2024-10-13 14:47:38:021],FARMAX_POS,ANXOPINPAD,INFO,, Serial port baud rate:  19200
[2024-10-13 14:47:38:021],FARMAX_POS,ANXOPINPAD,INFO,, Serial port parity:      None
[2024-10-13 14:47:38:021],FARMAX_POS,ANXOPINPAD,INFO,, Serial port data bits:  8
[2024-10-13 14:47:38:021],FARMAX_POS,ANXOPINPAD,INFO,, Serial port stop bits:  One
[2024-10-13 14:47:38:022],FARMAX_POS,ANXOPINPAD,INFO,, Timeout PINpad ACK:      5
[2024-10-13 14:47:38:022],FARMAX_POS,ANXOPINPAD,INFO,, Timeout PINpad Q4:       20
[2024-10-13 14:47:38:022],FARMAX_POS,ANXOPINPAD,INFO,, Timeout PINpad CC0:      30
[2024-10-13 14:47:38:022],FARMAX_POS,ANXOPINPAD,INFO,, Timeout PINpad CC1:      45
[2024-10-13 14:47:38:022],FARMAX_POS,ANXOPINPAD,INFO,, Timeout PINpad CC4 chip 45
[2024-10-13 14:47:38:022],FARMAX_POS,ANXOPINPAD,INFO,, Timeout PINpad CP0       30
[2024-10-13 14:47:38:022],FARMAX_POS,ANXOPINPAD,INFO,, Auth server:             192.168.152.17
[2024-10-13 14:47:38:022],FARMAX_POS,ANXOPINPAD,INFO,, Auth server port:        7000
[2024-10-13 14:47:38:022],FARMAX_POS,ANXOPINPAD,INFO,, Timeout auth:            30
[2024-10-13 14:47:38:022],FARMAX_POS,ANXOPINPAD,INFO,, Credit/debit tender ID: 12
[2024-10-13 14:47:38:022],FARMAX_POS,ANXOPINPAD,INFO,, AMEX tender ID:          14
[2024-10-13
14:47:38:022],FARMAX_POS,ANXOPINPAD,INFO,,************************************************************
[2024-10-13 14:47:38:295],FARMAX_POS,ANXOPINPAD,INFO,,TIMING2,SendAuthMessage,00:00:00.27,OK
[2024-10-13
14:47:38:295],FARMAX_POS,ANXOPINPAD,WARNING,,SUC:2575|CAJA:00000021|FOL:092966|TARJ:************8414|MTO:18000|
ARQC:C2E7C7EBA2C1261E|RC:00|AUT:430575
[2024-10-13 14:47:38:641],FARMAX_POS,ANXOPINPAD,INFO,,AUTH approved
[2024-10-13 14:47:41:920],FARMAX_POS,ANXOPINPAD,INFO,,TIMING1,EnviaPeticionVenta,00:00:03.89
[2024-10-13 14:47:41:920],FARMAX_POS,ANXOPINPAD,INFO,,TIMING1,CierraVentana,00:00:00.00
```

Se observa que esta vez fue procesado por el monto de **MX$180** y esta vez fue aprobado, por lo que la excepción anterior se da porque el primer intento no fue aprobado.

Se determina que las formas de pago fueron procesadas correctamente y que la excepción encontrada no interrumpe ni corrompe el flujo de ninguna manera.

GET Technologies	Incidencia – Venta finalizada con forma de pago pendiente		
Autor	Versión	Fecha de elaboración	Fecha de última actualización
Juan Pérez	1.0	31-10-2024	31-10-2024

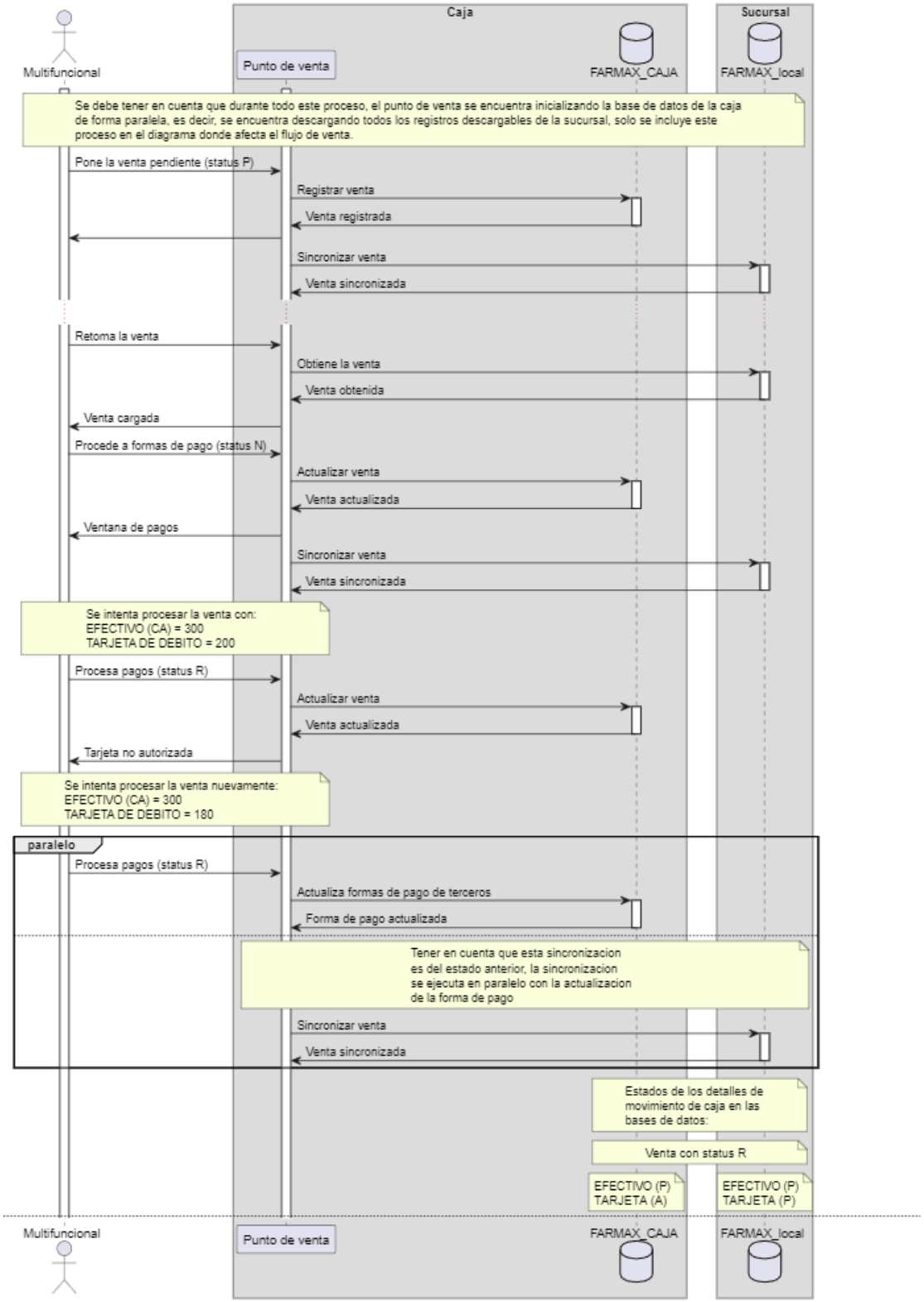
Sincronización de bases de datos

Se determina que el flujo de venta no sufre ningún tipo de anomalía, por lo que se procede a revisar el log de sincronización, este log se llama **20241013_FARMAX_TILL_SERVER_SYNCLIBRARY.XLOG** y se ubica en **%ProgramData%/FAhorro/FARMAX_TILL_SERVER**, en este log se presentan las operaciones de sincronización realizadas tanto para subida y como para descarga de datos entre la caja y la sucursal, se observa que al momento de procesar las formas de pago de la venta, al mismo tiempo, se está inicializando la base de datos de la caja, a partir de la **línea 3628**:

```
[2024-10-13 14:47:24:502],FARMAX_TILL_SERVER,SYNCLIBRARY,INFO,,[SYNC_TILL_INIT_DATA]
Se descargara la tabla:'genMovimientosCajasDet' con el servidor de sucursal
[2024-10-13 14:47:27:925],FARMAX_TILL_SERVER,SYNCLIBRARY,INFO,,[SYNC_TILL_INIT_DATA]
Se recibieron 117985 registro(s) de la tabla:'genMovimientosCajasDet'
[2024-10-13 14:47:42:776],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
[2024-10-13 14:47:43:031],FARMAX_TILL_SERVER,SYNCLIBRARY,INFO,,[SYNC_TILL_TX_DATA] Se guardaran 1 registro(s) de la tabla:'genMovimientosCajasDet' en la base de datos de la sucursal
[2024-10-13 14:47:43:087],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
[2024-10-13 14:48:05:150],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
```

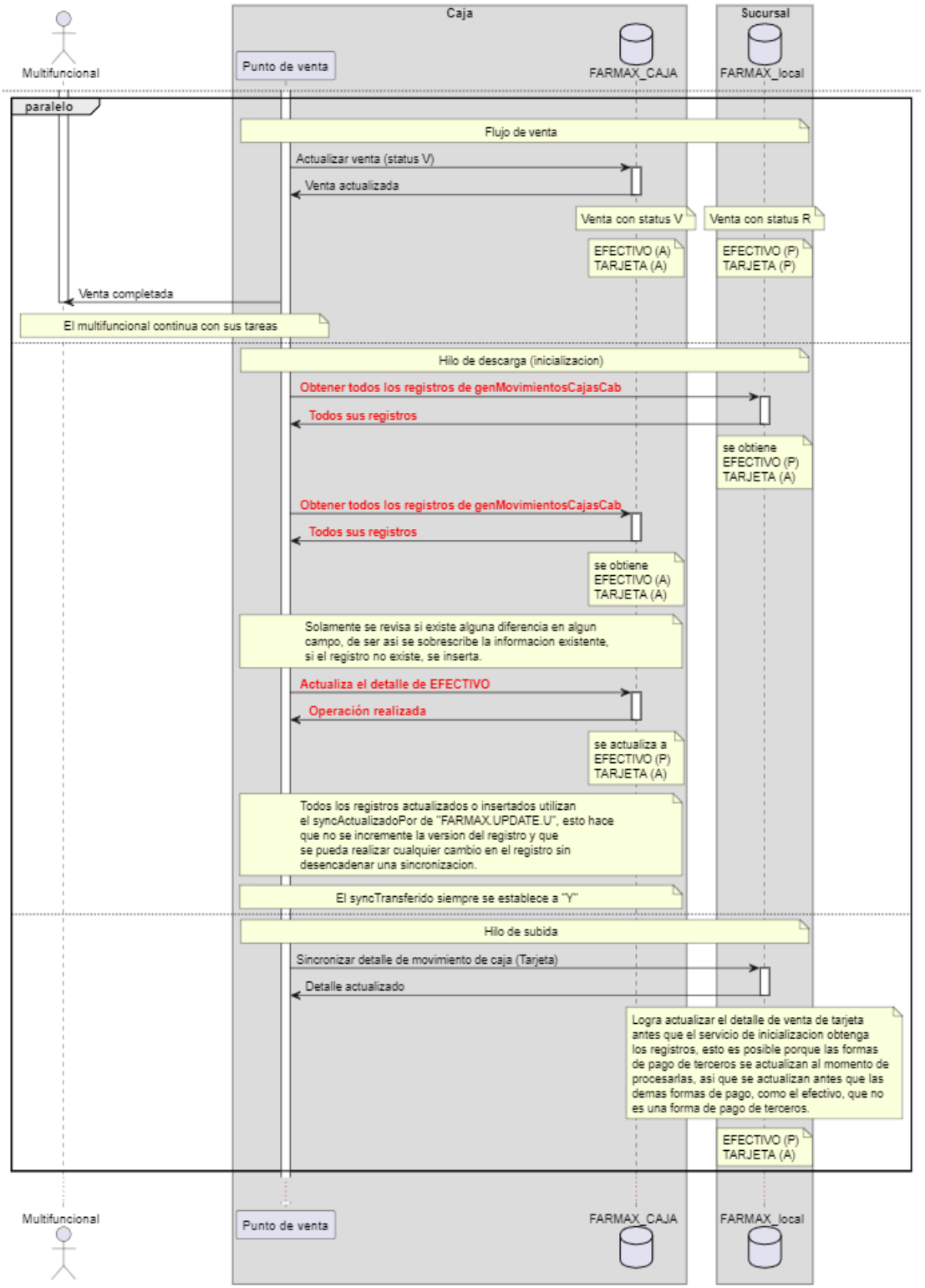
Se determina que la caja se encontraba descargando todos los registros de **genMovimientosCajasDet** antes de mandar la última versión de la venta a la base de datos de la sucursal, se debe considerar también que esta descarga tarda alrededor de 30 segundos, en estos 30 segundos se insertan nuevos registros y se actualizan los registros existentes, con esta información se puede representar el escenario de lo que ocurrió utilizando un diagrama de secuencia de alto nivel:

GET Technologies	Incidencia – Venta finalizada con forma de pago pendiente		
Autor	Versión	Fecha de elaboración	Fecha de última actualización
Juan Pérez	1.0	31-10-2024	31-10-2024



Ver diagrama (página 1)

	Incidencia – Venta finalizada con forma de pago pendiente		
Autor	Versión	Fecha de elaboración	Fecha de última actualización
Juan Pérez	1.0	31-10-2024	31-10-2024



Ver diagrama (página 2)

Todos estos procesos se ejecutan en paralelo, causando el problema de sobreescritura de datos.

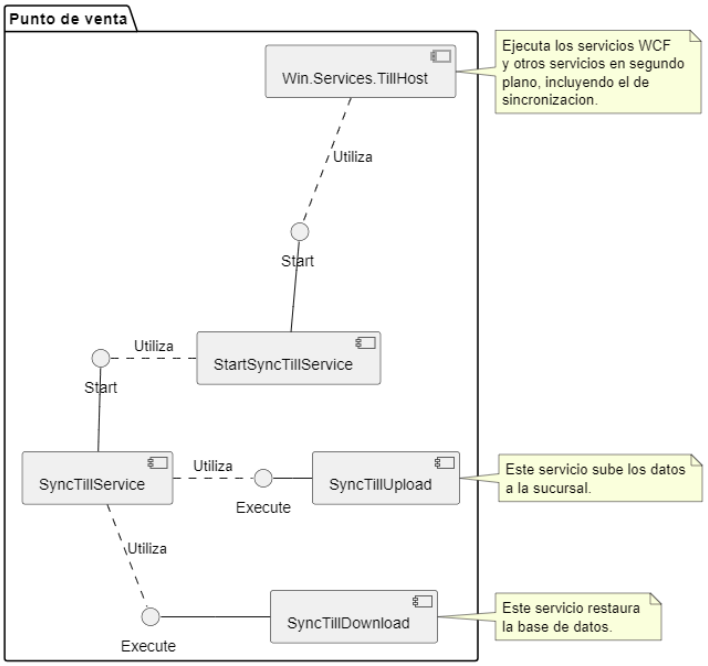
Restauración de base de datos de caja

Se explica brevemente qué es y donde ocurre la inicialización de base de datos.

En la solución **POS.Services**, en el proyecto **Farmax.SyncTillLibrary** se encuentra el servicio **SyncTillService.cs**, este servicio ejecuta dos servicios llamados:

- **SyncTillDownload**
  - Se encarga de descargar datos desde la base de datos de la sucursal a la base de datos de la caja.
  - Otra de sus responsabilidades es inicializar la base de datos en caso de que sea necesario, este proceso también es conocido como modo o proceso de restauración y consiste en descargar todos los registros de todas las tablas descargables desde de la base de datos de la sucursal a la base de datos de la caja, incluso si estos ya existen en la base de datos de la caja.
- **SyncTillUpload**
  - Se encarga de subir datos desde la base de datos de la caja hacia la base de datos de la sucursal.

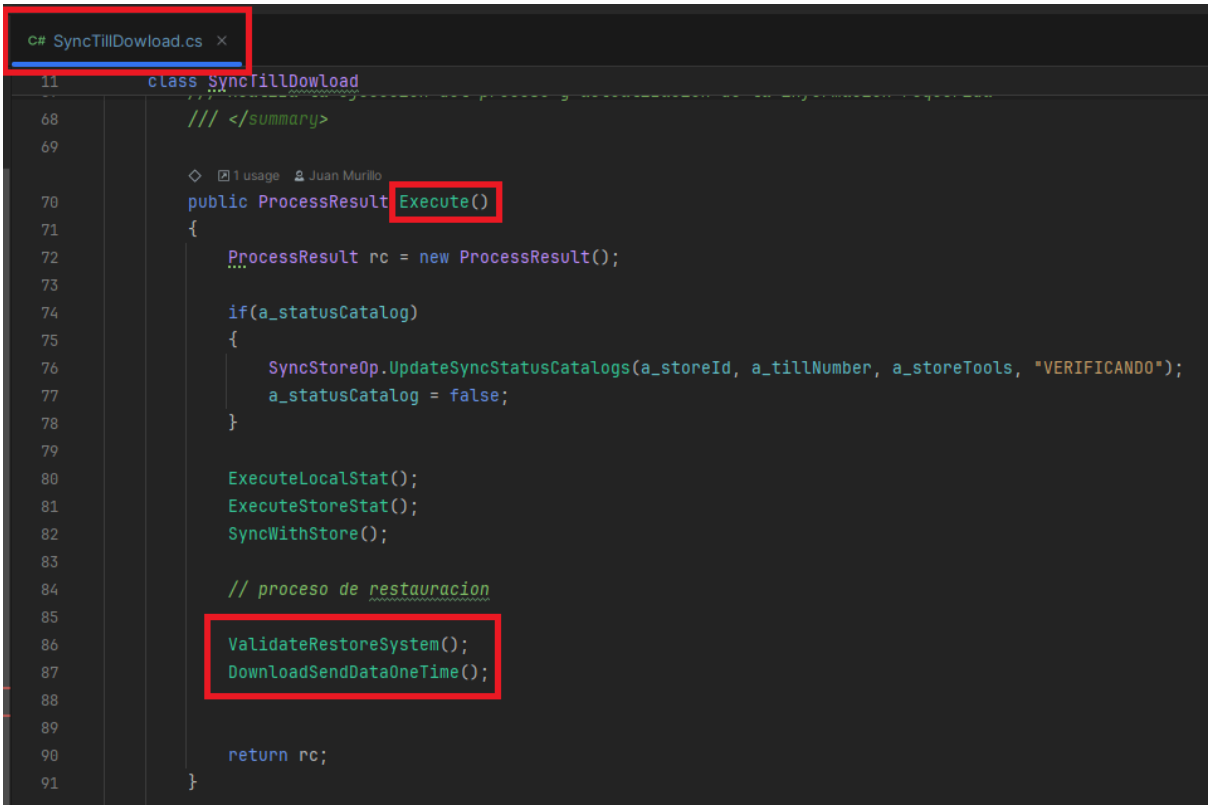
Se presenta en el siguiente diagrama de componentes la arquitectura de estos servicios:



Ver diagrama

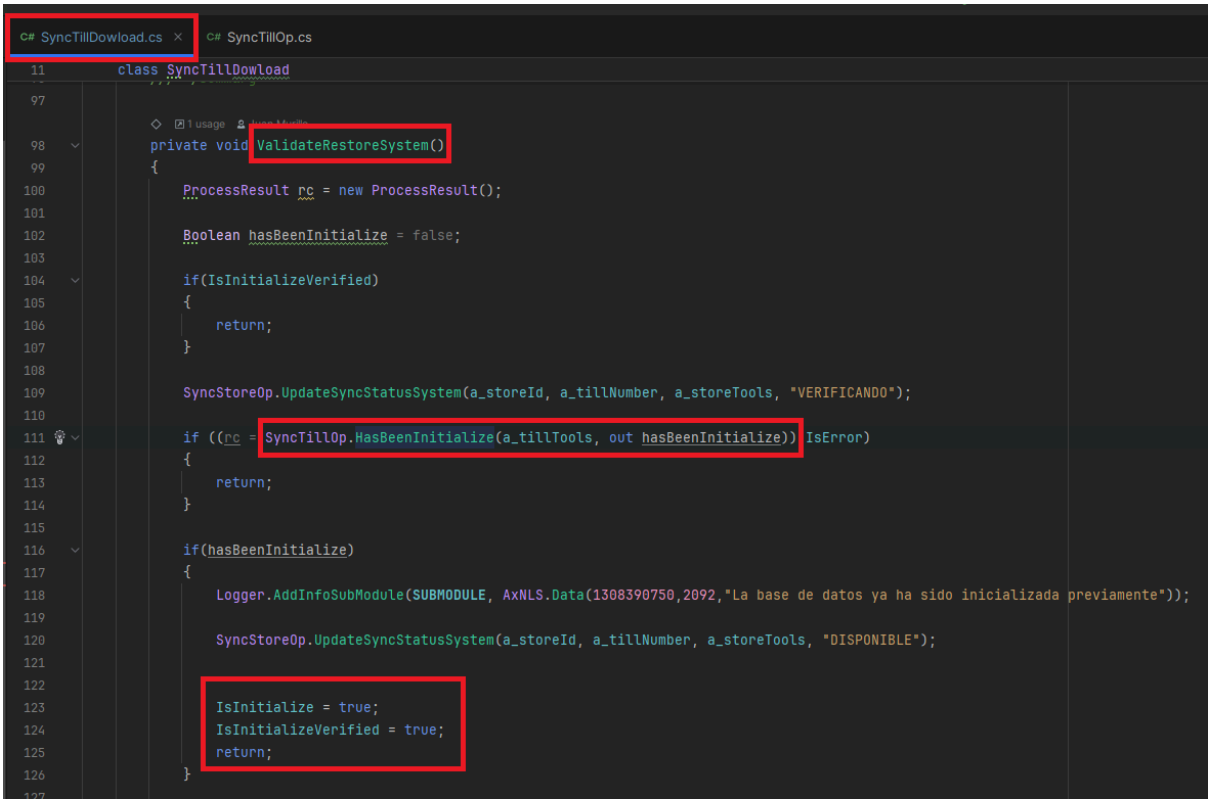
	Incidencia – Venta finalizada con forma de pago pendiente		
Autor	Versión	Fecha de elaboración	Fecha de última actualización
Juan Pérez	1.0	31-10-2024	31-10-2024

En el servicio **SyncTillDownload**, en el método **Execute()**, se realiza la validación para saber si se debe realizar la restauración de base de datos de caja y posteriormente se realiza la restauración en caso de ser necesario, esta validación ocurre en **ValidateRestoreSystem()** y la ejecución de restauración ocurre en **DownloadSendDataOneTime()**:



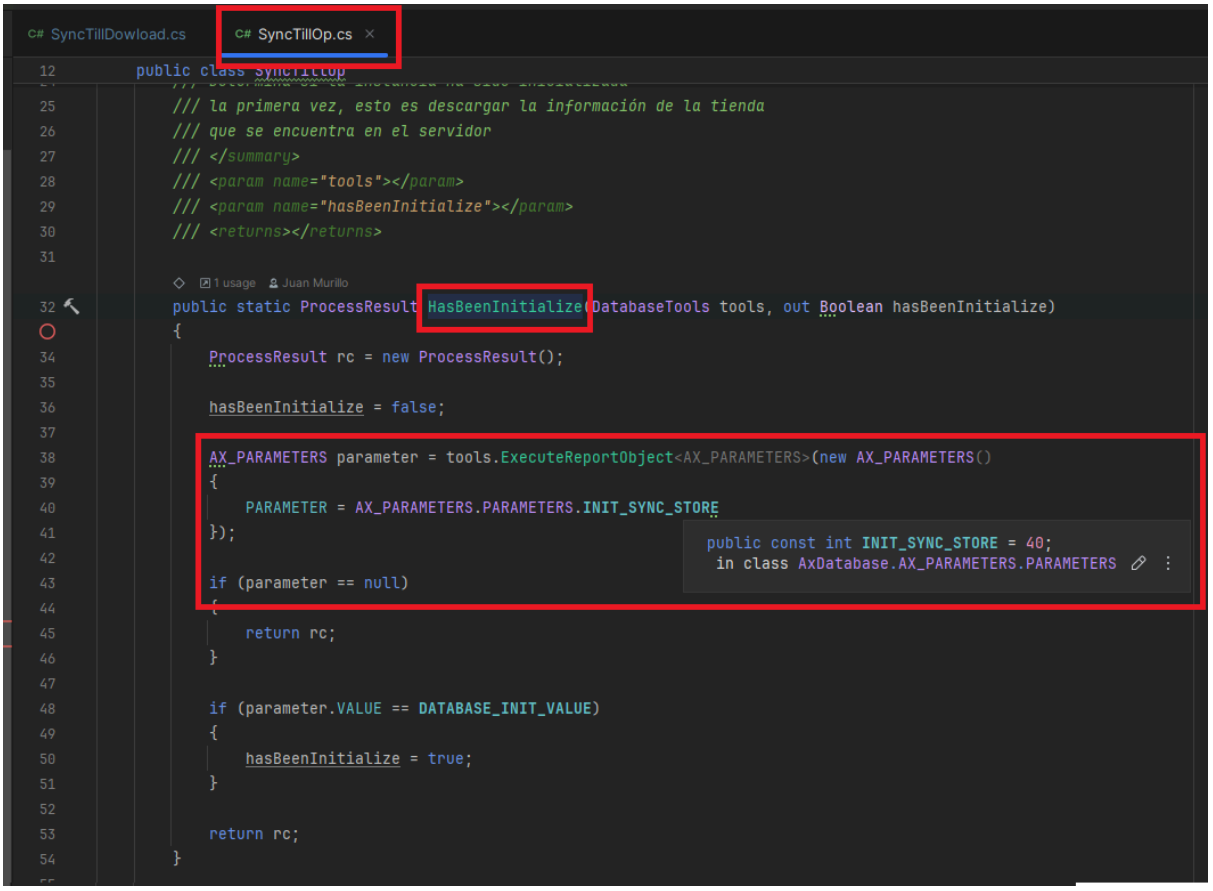
Ver figura

Lo que se realiza en **ValidateRestoreSystem()** es revisar los valores en la tabla **AX_PARAMETERS**, para el registro con el campo **PARAMETER** con valor de **40**:



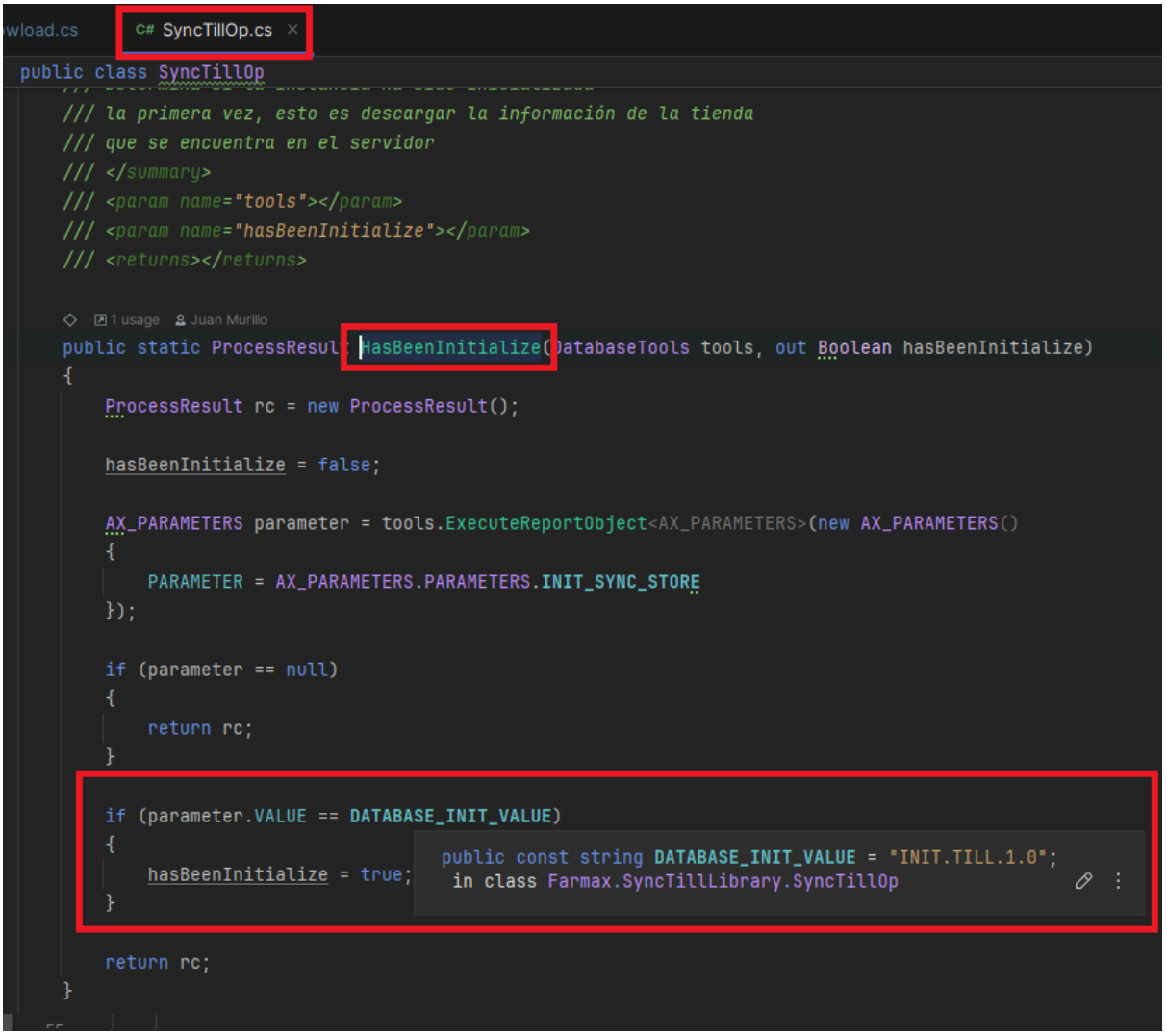
Ver figura

	Incidencia – Venta finalizada con forma de pago pendiente		
Autor	Versión	Fecha de elaboración	Fecha de última actualización
Juan Pérez	1.0	31-10-2024	31-10-2024



Ver figura

GET Technologies	Incidencia – Venta finalizada con forma de pago pendiente		
Autor	Versión	Fecha de elaboración	Fecha de última actualización
Juan Pérez	1.0	31-10-2024	31-10-2024



Ver figura

Esta información se puede consultar de la siguiente manera:

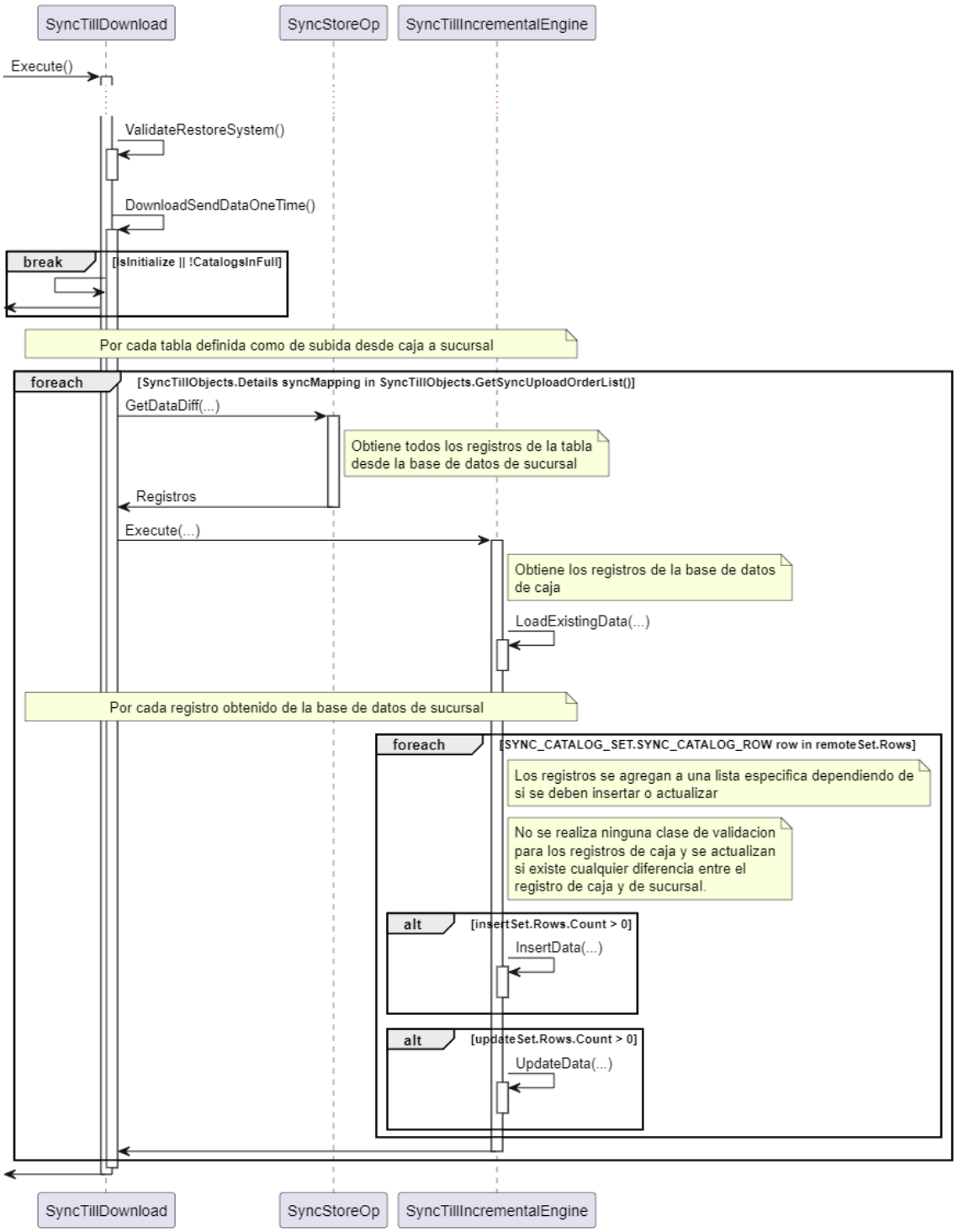
```
SELECT
   *
FROM AX_PARAMETERS
WHERE PARAMETER = 40;
```

Ver consulta

Si el registro no existe o su campo **VALUE** es diferente a la cadena **“INIT.TILL.1.0”** se considera que la base de datos necesita ser restaurada.

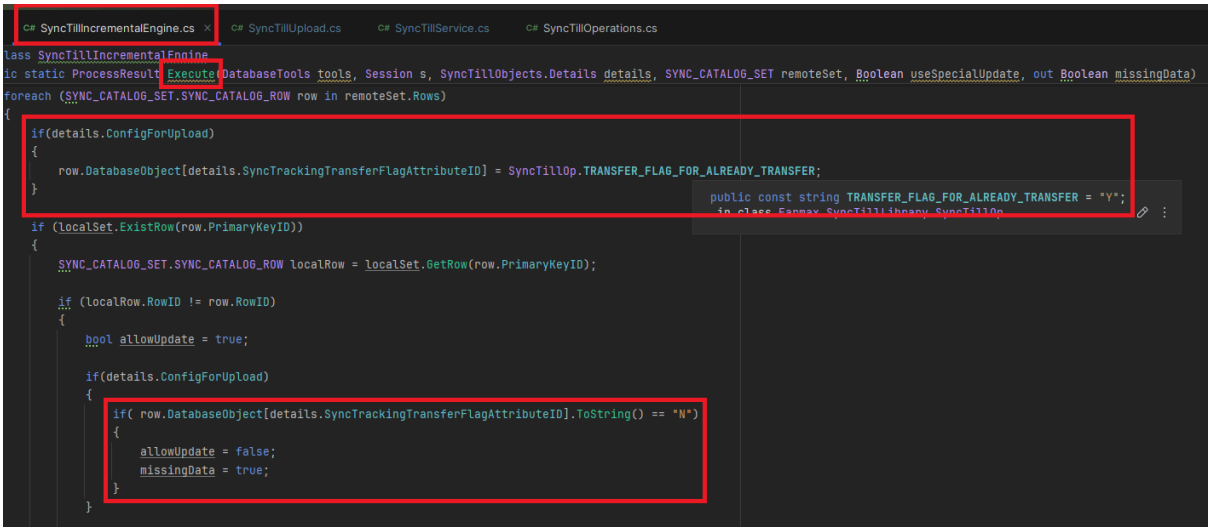
GET Technologies	Incidencia – Venta finalizada con forma de pago pendiente		
Autor	Versión	Fecha de elaboración	Fecha de última actualización
Juan Pérez	1.0	31-10-2024	31-10-2024

En el método **DownloadSendDataOneTime()** se descargan los datos desde la base de datos de la sucursal a la base de datos de la caja, primero se revisa si ya está inicializada la base de datos, si no, se procede a inicializar, esto se presenta a nivel técnico en el siguiente diagrama de secuencia:



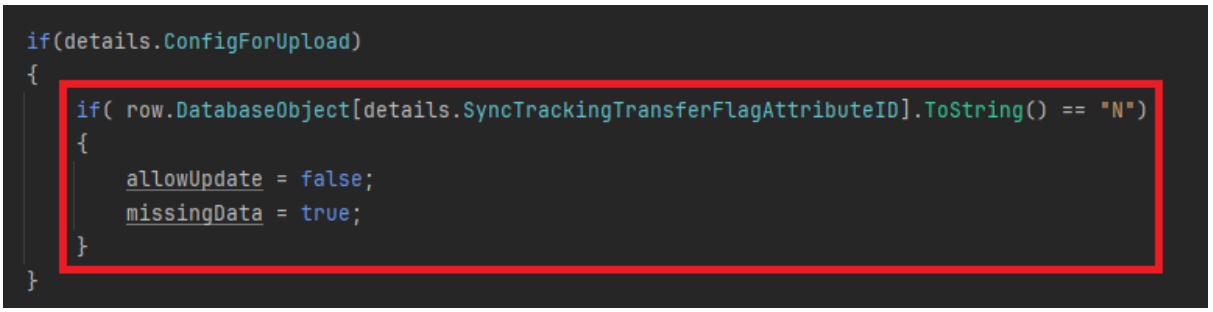
Ver diagrama

Como se menciona en el diagrama de secuencia, no se realiza ninguna validación con respecto a los registros existentes en la base de datos de caja, aunque existen validaciones para los registros de base de datos de sucursal:



Ver figura

Aunque después hay una validación para saber si el registro de sucursal no se ha transferido:



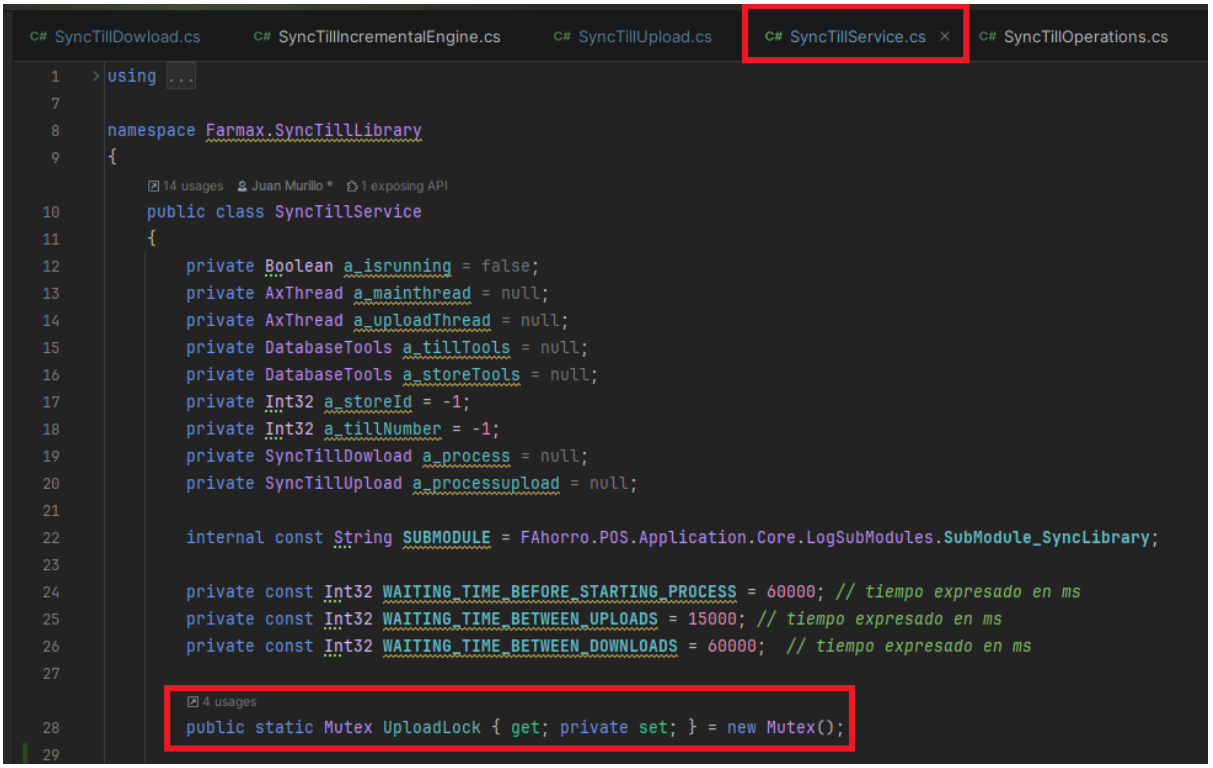
Ver figura

Esta validación **nunca se evalúa a true** puesto que unas líneas de código antes el valor se establece a **“Y” (transferido)**:

Ver figura

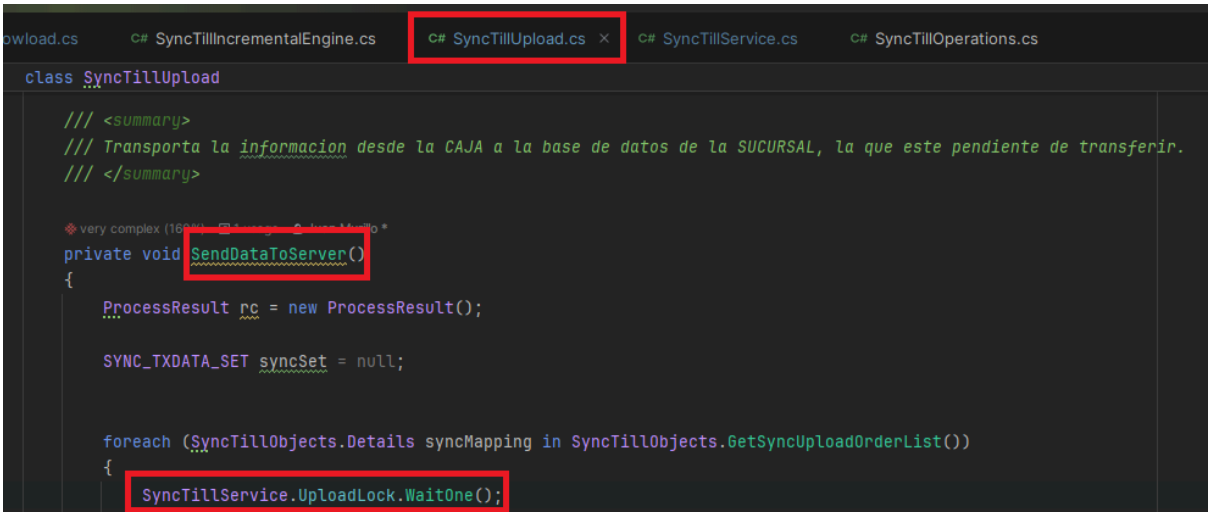
Es posible sincronizar el hilo de restauración de base de datos en **SyncTillDownload** con el hilo de subida de datos en **SyncTillUpload**, existe un **Mutex** en **SyncTillService** que se utiliza en **SyncTillUpload** para sincronizar el proceso de subida:

	Incidencia – Venta finalizada con forma de pago pendiente		
Autor	Versión	Fecha de elaboración	Fecha de última actualización
Juan Pérez	1.0	31-10-2024	31-10-2024



[Ver figura]

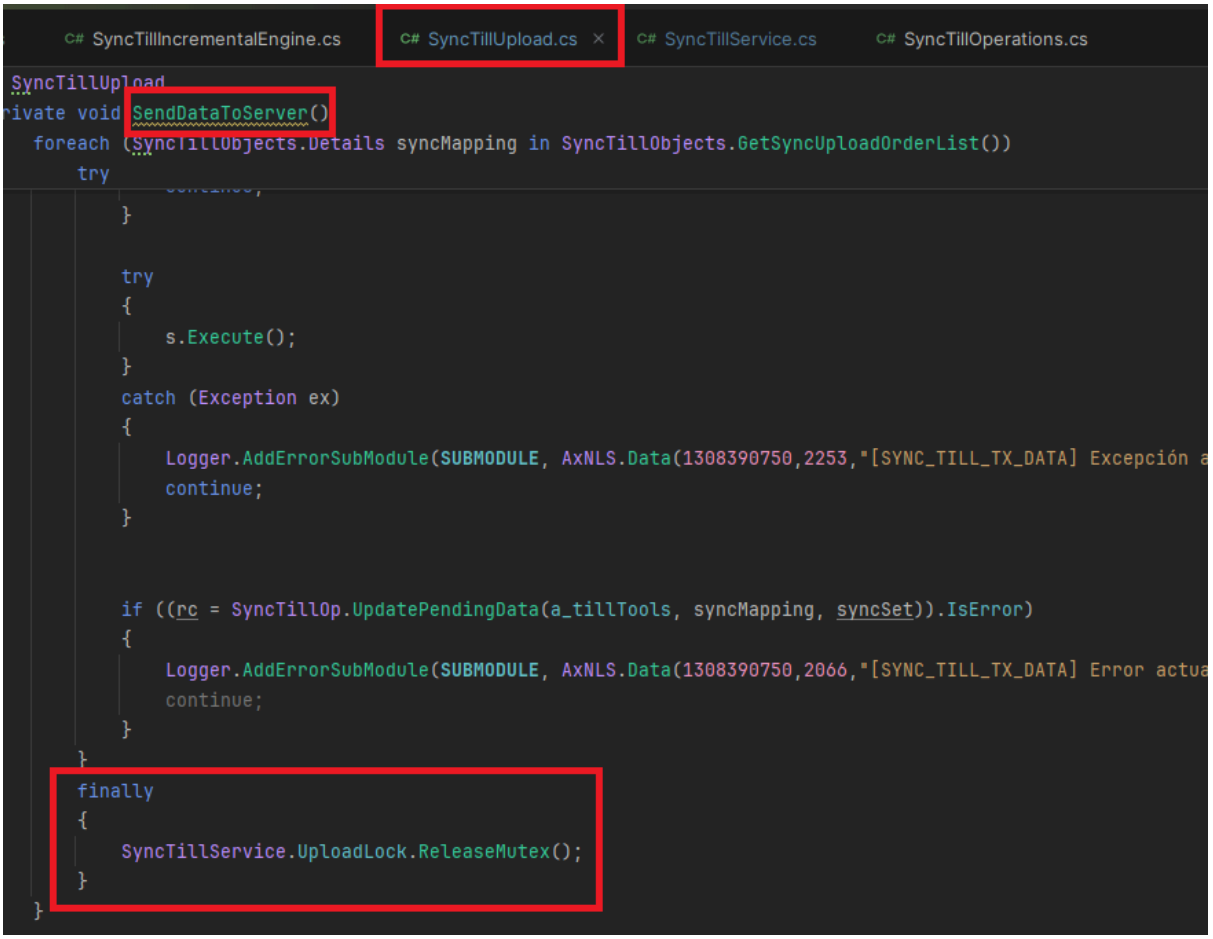
Se observa que se bloquea por cada tabla a subir:



[Ver figura]

GET Technologies	Incidencia – Venta finalizada con forma de pago pendiente		
Autor	Versión	Fecha de elaboración	Fecha de última actualización
Juan Pérez	1.0	31-10-2024	31-10-2024

Luego de terminar con la tabla se libera:



Ver figura

GET Technologies	Incidencia – Venta finalizada con forma de pago pendiente		
Autor	Versión	Fecha de elaboración	Fecha de última actualización
Juan Pérez	1.0	31-10-2024	31-10-2024

Posibles causas

El estado del detalle de movimiento de caja pendiente en una venta finalizada **se debe a que el proceso de inicialización de base de datos de caja recibió información desactualizada** de la transacción realizada desde la **base de datos de la sucursal**, la información recibida es de **antes de poder sincronizar el detalle de movimiento de caja**, por lo que se **utiliza una versión en la que el detalle de movimiento de caja para la forma de pago en efectivo aún se encontraba pendiente**, pero **desde la perspectiva del punto de venta, la venta fue realizada correctamente** y la **venta fue actualizada en la base de datos de caja**, para luego ser **sobrescrita por el proceso de inicialización**, **esto ocurre después de que la venta finaliza**.

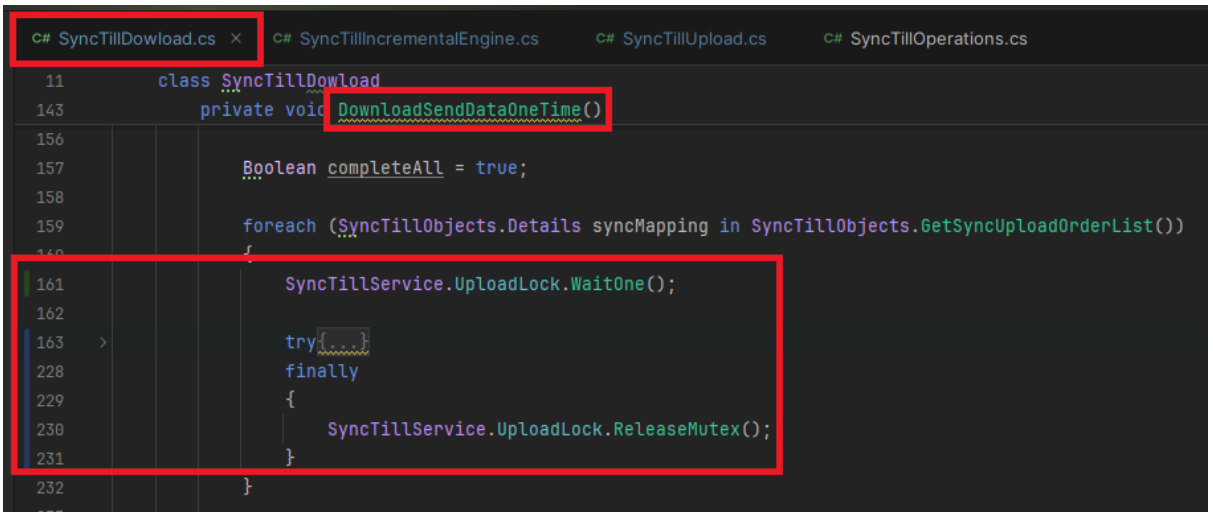
Los problemas solamente se dan con el detalle de movimiento de caja porque **genMovimientosCajaDet es la tabla que se estaba descargando en ese momento**, la incidencia ocurre en una situación muy excepcional donde distintos hilos y procesos se encontraban trabajando con la misma fuente de datos y compitiendo para realizar sus operaciones de actualización.

Si bien los motores de bases de datos implementan bloqueos a distintos niveles (filas, páginas, tablas, etc.) para las operaciones de lectura y escritura, los procesos de la incidencia que utilizan la base de datos no están al tanto uno del otro, no se sincronizan de ninguna manera lo que puede llevar a este comportamiento errático.

GET Technologies	Incidencia – Venta finalizada con forma de pago pendiente		
Autor	Versión	Fecha de elaboración	Fecha de última actualización
Juan Pérez	1.0	31-10-2024	31-10-2024

Posible solución

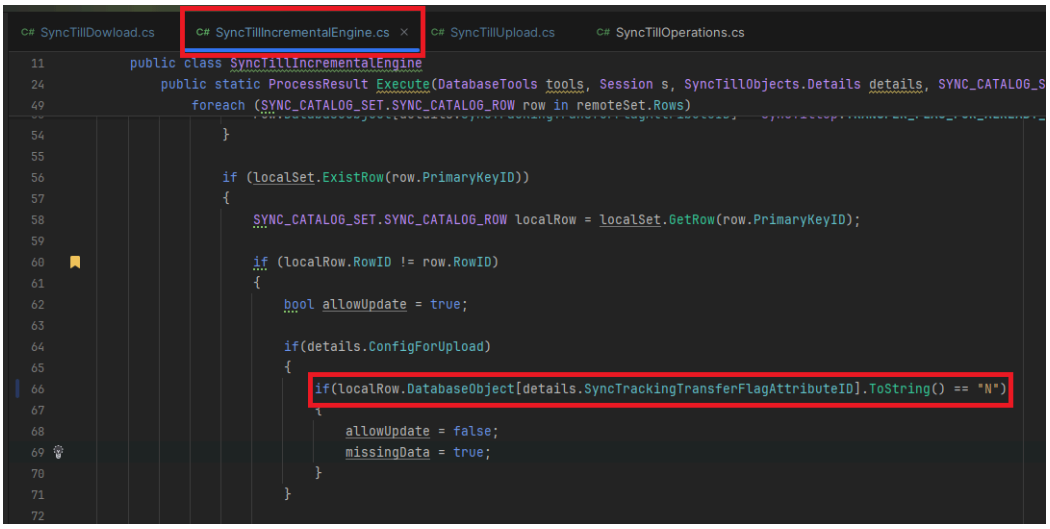
Para sincronizar el servicio de **SyncTillDownload** con el de **SyncTillUpload**, se puede utilizar el **Mutex** en **SyncTillService** llamado **UploadLock**, se podría bloquear la subida antes de descargar una tabla:



Ver figura

Se encierra todo el código del foreach en un try-finally cuyo único propósito es asegurarse de que el bloqueo se libera después de finalizar de procesar una tabla, tenga éxito o no esta operación, para permitir al servicio de sincronización continuar con su procesamiento.

Bloquear la subida permite comparar los registros de la base de datos de caja con la confianza de que estos no van a sufrir un cambió realizado por el servicio de subida de datos a sucursal:



Ver figura