



INCGET - 20250213-01 - Doble commit en Origis con orden de servicio a domicilio con monedero electrónico

Autor	Versión	Fecha de elaboración	Fecha de última actualización
Juan Pérez	1.3	14-02-2025	19-02-2025

Control de versiones

Versión	Fecha	Autor	Descripción
1.0	14-02-2025	Juan Pérez	Se analiza el problema y se encuentra causa raíz.
1.1	16-02-2025	Juan Pérez	Se especifican las secciones de código más importantes en el flujo afectado.
1.2	18-02-2025	Juan Pérez	Se determina mitigación del problema, la solución raíz queda pendiente.
1.3	19-02-2025	Juan Pérez	Se reestructura la sección de mitigación del problema.

	INCGET - 20250213-01 - Doble commit en Origis con orden de servicio a domicilio con monedero electrónico		
Autor	Versión	Fecha de elaboración	Fecha de última actualización
Juan Pérez	1.3	14-02-2025	19-02-2025

Incidencia

Información general

Sucursal	2191
Caja	20
Versión de software	10.5.60702.19
Folio venta	V21912000038455
Monedero	9600233812350

	INCGET - 20250213-01 - Doble commit en Origis con orden de servicio a domicilio con monedero electrónico		
Autor	Versión	Fecha de elaboración	Fecha de última actualización
Juan Pérez	1.3	14-02-2025	19-02-2025

Descripción

Se encuentra que en algunas ventas a domicilios se realizan dos operaciones ***commitTrans*** seguidas con el servicio de ***Origis*** al despachar la orden, este error no permite finalizar el despacho ya que el segundo commit marca un error con código ***42 Venta No Aplicada: La transacción ya existe [042.3]***, este código ocurre cuando se realiza un commit en una venta que ya fue registrada en el servicio de Origis utilizando la operación ***commitTrans***, como ocurre en la incidencia.

Esto puede observarse en la tabla *genBitacoraMonedero*, donde se registran las operaciones de *commitTrans* y *reverseTrans*, se utiliza la siguiente consulta:

```
SELECT * FROM dbo.genBitacoraMonedero  
WHERE codigoSucursal = 2191  
    AND folioTransaccion = 'V21912000038455'  
ORDER BY syncCreadoFecha;
```

Donde se obtienen los siguientes resultados:

The screenshot shows a database development interface with two main panes. The top pane is a code editor with the following SQL query:

```
SELECT * FROM dbo.genBitacoraMonedero
WHERE codigoSucursal = 2191
AND folioTransaccion = 'V21912000038455'
ORDER BY syncCreadoFecha;
```

The bottom pane displays the results of the query as a table:

	1	2
idBitacora	F6718F1C-14EA-EF11-80E2-0050568B4381	FD718F1C-14EA-EF11-80E2-0050568B4381
folioTransaccion	V21912000038455	V21912000038455
codigoSucursal	2191	2191
numeroCaja	20	20
codigoCajero	24308	24308
numeroTarjeta	9600233812350	9600233812350
operacion	C	C
datosEnviamos	sSessionId=14365713777,sArticulos=7502223703070,1,44,44,sSaldoRedimido...	sSessionId=14365713777,sArticulos=7502223703070,1,44,44,sSaldoRedimido...
datosRecibidos	SESSIONID=14365713777,AUTORIZACION=1607876292,ARTICULOS=7502223703070...	SESSIONID=14365713777...
pagoMonedero	44,0000	44,0000
codigoError	0	42
mensaje	Venta No Aplicada: La transacción ya existe [042.3]	
confirmaReversa	<null>	<null>
fechaTransaccion	2025-02-13 08:09:27.000	2025-02-13 08:09:29.393
versionCentral	1000400291	1000400291
syncCreadoFecha	2025-02-13 08:09:26.317	2025-02-13 08:09:27.310
syncActualizadoFecha	2025-02-13 08:09:26.317	2025-02-13 08:09:27.310
syncActualizadoPor	FARMAX.DEFAULT	FARMAX.DEFAULT
syncVersionRegistro	1	1
syncTransferido	I	I

The results show two rows of data. The first row corresponds to the transaction ID V21912000038455, and the second row corresponds to a subsequent update or reversal. The 'mensaje' column for the second row contains the error message "Venta No Aplicada: La transacción ya existe [042.3]".

Donde:

- **folioTransaccion**: es el folio de la venta.
 - **operacion**: es el tipo de operación realizada, C = Commit y R = Reversa.

	INCGET - 20250213-01 - Doble commit en Origis con orden de servicio a domicilio con monedero electrónico		
Autor	Versión	Fecha de elaboración	Fecha de última actualización
Juan Pérez	1.3	14-02-2025	19-02-2025

- **codigoError**: el código de error recibido de la operación, si hubo error.
- **mensaje**: el mensaje de error, si hubo error.
- **syncCreadoFecha**: momento en el que el registro es creado en esta base de datos específica.

Como se puede observar, según el campo **syncCreadoFecha**, se realizan 2 commits con una diferencia de 1 segundo.

	INCGET - 20250213-01 - Doble commit en Origis con orden de servicio a domicilio con monedero electrónico		
Autor	Versión	Fecha de elaboración	Fecha de última actualización
Juan Pérez	1.3	14-02-2025	19-02-2025

Reproducción del caso

Versión de software utilizada	10.5.60702.19
Ambiente	Desarrollo local (personal)
Recursos	<ul style="list-style-type: none"> - Payload para crear orden en servicio web de ordenes de servicio a domicilio: <u>CrearOrdenSadEnWebService.json</u>. - Script de bloqueo de tabla <i>genProgramaLealtadBitacoraReg:</i> <u>BloquearTabla.sql</u>
Precondiciones	<ul style="list-style-type: none"> - Tener una orden de servicio a domicilio pendiente de despacho con pago de monedero electrónico. - Tener un monedero con suficiente saldo electrónico para cubrir el monto total de la orden. - Activar el buffer de bitácora de operaciones de programa de lealtad (<i>genProgramaLealtadBitacoraReg</i>) en web service central para su persistencia asíncrona. - Simular alta concurrencia en tabla <i>genProgramaLealtadBitacoraReg</i> bloqueandola de manera artificial.
Pasos para reproducir	<ul style="list-style-type: none"> - Imprimir comanda de la orden de servicio a domicilio. - Iniciar despacho de orden de servicio a domicilio. - Ingresar artículos de la orden de servicio a domicilio. - Finalizar despacho de la orden de servicio a domicilio.

Para **activar el buffer de bitácora de operaciones de programa de lealtad**, se debe establecer en el **Web.config**, en la sección **appSettings**, el valor **ActivateLoyaltyBuffer** a **true**, esto hace que los registros sean persistidos de forma asíncrona.



INCGET - 20250213-01 - Doble commit en Origis con orden de servicio a domicilio con monedero electrónico

Autor	Versión	Fecha de elaboración	Fecha de última actualización
Juan Pérez	1.3	14-02-2025	19-02-2025

```
<appSettings>
...
<add key="ActivateLoyaltyBuffer" value="true" />
</appSettings>
```

Se imprime la comanda

The screenshot shows a computer interface for managing service-to-home orders. At the top, it displays 'Bienvenido a Farmacias del Ahorro' and 'Te Atiende CAJERO'. A central window titled 'Servicio a Domicilio - Panel de Ordenes' lists 13 pending orders (Folios) from 13/02/2025 08:04 a.m. to 13/02/2025 08:04 a.m. The columns include Folio, Fecha, Estado, Transcurrido, Acciones, and Prioridad. The 'Acciones' column contains buttons for Comanda, Cancelar, Soporte, and Info. The 'Prioridad' column shows all entries as NORMAL. Below the table, there are filters for 'Menos 15 minutos', 'Entre 15 minutos y 20 minutos', and 'Mayor a 20 minutos'. At the bottom, it shows 'Dejaste de Ganar \$0.00', 'Total de esta compra \$0.99', and 'Total a Pagar \$0.00'. Navigation keys F2 through F11 are visible at the bottom left.



INCGET - 20250213-01 - Doble commit en Origis con orden de servicio a domicilio con monedero electrónico

Autor	Versión	Fecha de elaboración	Fecha de última actualización
Juan Pérez	1.3	14-02-2025	19-02-2025

Versión: 10.5.60701.2 CAJA: 10 07:23 PM 14/02/2023

Bienvenido a Farmacias del Ahorro
Te Atiende CAJERO

Servicio a Domicilio - Panel de Ordenes

Pendientes Marketplace Por Des Imprimir Comanda

Folio	Fecha
2000000012APP	13/02/20
2000000011APP	13/02/20
2000000015APP	13/02/20
2000000020APP	13/02/20
2000000019APP	13/02/20
2000000016APP	13/02/20
2000000009APP	13/02/20
2000000018APP	13/02/20
2000000014APP	13/02/20
2000000010APP	13/02/2025 08:04 a. m.

Todas Vales Interbodega

Info	NORMAL
Info	NORMAL

Cancelar Imprimir

Menos 15 minutos Entre 15 minutos y 20 minutos Mayor a 20 minutos

Dejaste de Ganar \$0.00 Total de esta compra \$0.00 Total a Pagar \$0.00

F2 Buscar F3 Reimpresión F4 Servicios F5 Monedero F7 Recetas F9 Última Venta F11 Cotizaciones Ctrl + F10 Verificador

Servicio a Domicilio
PENDIENTES: 13
POR DESPACHAR: 0
POR ENTREGAR: 0
POR DEVOLVER: 0

\$0.00 +

Precio Máximo Precio Ahorro

Versión: 10.5.60701.2 CAJA: 10 07:23 PM 14/02/2025

Bienvenido a Farmacias del Ahorro
Te Atiende CAJERO

FÁRATICOS 1.1
POR DESPRECiar 0
POR ENTREGAR 0
POR DEVOLVER 0

Servicio a Domicilio \$0.00 +

Precio Máximo Precio Ahorro

SKU

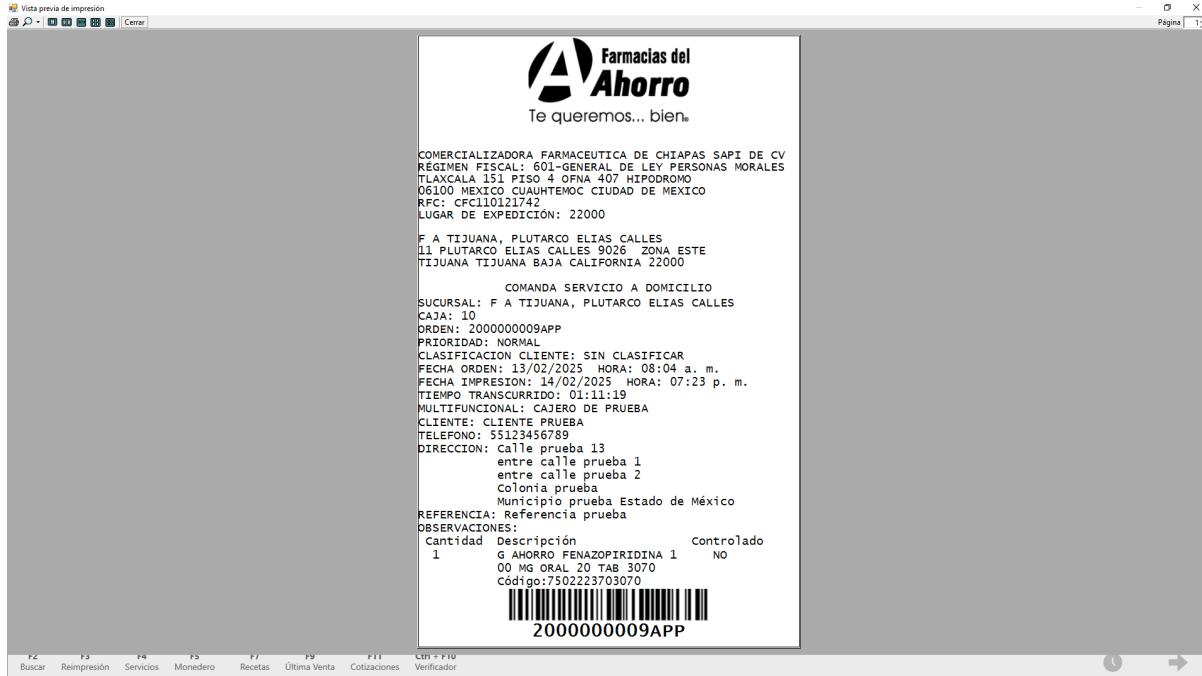
Servicio a Domicilio - Panel de Órdenes

Pendientes	Marketplace	Por Des	Imprimir Comanda	Todas	Vales Interbodega										
Folio	Fecha				Prioridad										
2000000012APP	13/02/20		FOLIO: 2000000009APP CLIENTE: CLIENTE PRUEBA FECHA y HORA: 13/02/2025 08:04 a.m. PRIORIDAD: NORM	Info	NORMAL										
2000000011APP	13/02/20			Info	NORMAL										
2000000015APP	13/02/20	1	SKU: 750	Info	NORMAL										
2000000020APP	13/02/20			Info	NORMAL										
2000000019APP	13/02/20			Info	NORMAL										
2000000016APP	13/02/20			Info	NORMAL										
2000000009APP	13/02/20			Info	NORMAL										
2000000018APP	13/02/20			Info	NORMAL										
2000000014APP	13/02/20			Info	NORMAL										
2000000010APP	13/02/2025 08:04 a.m.		PENDIENTE	Comanda	Cancelar Soporte										
Se procesó la comanda Aceptar Cancelar Imprimir															
<input type="checkbox"/> Menos 15 minutos Entre 15 minutos y 20 minutos Mayor a 20 minutos															
Dejaste de Ganar \$0.00															
Total de esta compra \$0.00															
Total a Pagar \$0.00															
F2	Buscar	Reimpresión	F4	Servicios	F5	Monedero	F7	Recetas	F9	Última Venta	F11	Cotizaciones	Ctrl + F10	Verificador	



INCGET - 20250213-01 - Doble commit en Origis con orden de servicio a domicilio con monedero electrónico

Autor	Versión	Fecha de elaboración	Fecha de última actualización
Juan Pérez	1.3	14-02-2025	19-02-2025



Se inicia el despacho de la orden

The screenshot shows the Farmacias del Ahorro Service to Home panel. At the top, it displays the version (10.5.60701.2), the cash register (CAJA: 10), and the date (07:23 PM 14/02/2025). On the right, there's a summary of pending deliveries: 13 pending, 3 to dispatch, 0 to deliver, 0 to return, and 0 to refund.

The main area is titled "Servicio a Domicilio - Panel de Órdenes". It lists four pending delivery orders:

Folio	Fecha	Estado	Transcurrido	Acciones	Prioridad	Tipo
2000000008APP	13/02/2025 08:04 a.m.	POR DESPACHAR	00:11:20	Despacho Cancelar Soporte Info	NORMAL	SI
2000000009APP	13/02/2025 08:04 a.m.	POR DESPACHAR	00:11:20	Despacho Cancelar Soporte Info	NORMAL	SI
2000000007APP	13/02/2025 08:04 a.m.	POR DESPACHAR	00:11:20	Despacho Cancelar Soporte Info	NORMAL	SI
2000000004APP	13/02/2025 08:04 a.m.	POR DESPACHAR	00:11:20	Despacho Cancelar Soporte Info	NORMAL	SI

Below the table, there are filters for delivery time: "Menos de 15 minutos" (unchecked), "Entre 15 minutos y 20 minutos" (checked), and "Mayor a 20 minutos" (unchecked).

At the bottom, there are summary boxes: "Dejaste de Ganar \$0.00", "Total de esta compra \$0.00", and "Total a Pagar \$0.00".



INCGET - 20250213-01 - Doble commit en Origis con orden de servicio a domicilio con monedero electrónico

Autor	Versión	Fecha de elaboración	Fecha de última actualización
Juan Pérez	1.3	14-02-2025	19-02-2025

Versión: 10.5.60701.2 CAJA: 10 07:23 PM 14/02/2025

¡Hola PRUEBA MONEDERO! 🎉 @
Te Atiende CAJERO

7502223703070

Saldo en Monedero e\$100.00 \$0.00
Te faltan \$100.00 para ganar \$100.00

Servicio a domicilio FOLIO: 2000000009APP

\$0.00 + 📦

Cant. Productos Precio Máximo Precio Ahorro

Al Monedero \$0.00 Total de esta compra \$0.00 Total a Pagar \$0.00

F2 Buscar F3 Reimpresión F4 Servicios F5 Monedero F7 Recetas F9 Última Venta F11 Cotizaciones Ctrl + F10 Verificador

Versión: 10.5.60701.2 CAJA: 10 07:23 PM 14/02/2025

¡Hola PRUEBA MONEDERO! 🎉 @
Te Atiende CAJERO

7502223703070

Saldo en Monedero e\$100.00 \$0.00
Te faltan \$100.00 para ganar \$100.00

Servicio a domicilio FOLIO: 2000000009APP

\$0.00 + 📦

Cant. Productos Precio Máximo Precio Ahorro

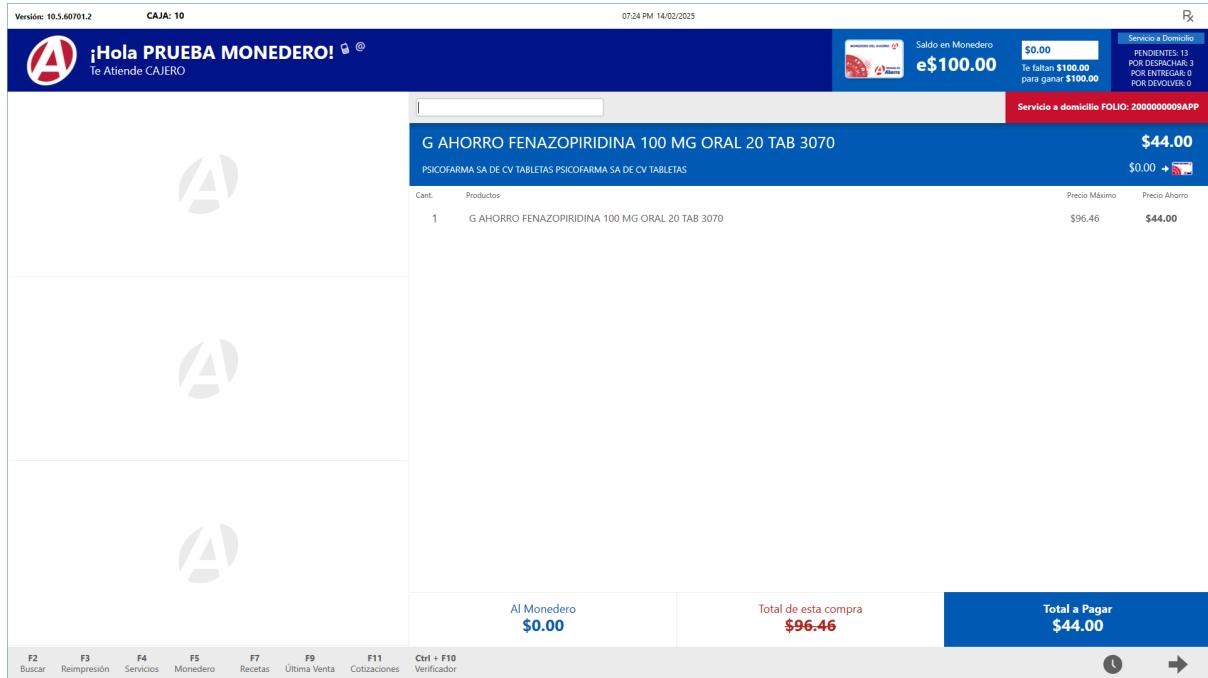
* Acumula 4 y gana 1. Ilimitado. Vigencia a Dec 31, 2025.

Aceptar

Al Monedero \$0.00 Total de esta compra \$0.00 Total a Pagar \$0.00

F2 Buscar F3 Reimpresión F4 Servicios F5 Monedero F7 Recetas F9 Última Venta F11 Cotizaciones Ctrl + F10 Verificador

	INCGET - 20250213-01 - Doble commit en Origis con orden de servicio a domicilio con monedero electrónico		
Autor	Versión	Fecha de elaboración	Fecha de última actualización
Juan Pérez	1.3	14-02-2025	19-02-2025



Antes de despachar la orden, se ejecutan las dos primeras líneas del script [BloquearTabla.sql](#) para simular alta concurrencia de operaciones con monedero, se ejecuta únicamente las dos primeras líneas del script.

```

1 BEGIN TRANSACTION;
2 SELECT * FROM genProgramaLealtadBitacoraReg WITH (XLOCK HOLDLOCK);
3 ROLLBACK TRANSACTION;

```

The screenshot shows the SSMS interface with the 'Database Explorer' and 'Files' panes. In the 'Output' pane, the results of the executed script are shown:

idBitacora	idTransaccion	tipoPrograma	tipoOperacion	codigoSucursal	numeroCaja	folioTransaccion	folioProg
1	83CEB55F-48EA-EF11-87A1-F43808267C81	ORI	T	1553	10	V15531000005602	V155310000005
2	9D364BF3-26EB-EF11-87A4-E89C250F7201	ORI	T	1553	10	V15531000005665	V155310000005
3	E4965616-27EB-EF11-87A4-E89C250F7201	ORI	T	1553	10	V15531000000413	D15531000000
4	10F516AF-2AE8-EF11-87A4-E89C250F7201	ORI	T	1553	10	V15531000005668	V155310000005
5	10F516AF-2AE8-EF11-87A4-E89C250F7201	ORI	R	1553	10	V15531000005668	V155310000005
6	8FB450FD-37EB-EF11-87A4-E89C250F7201	ORI	T	1553	10	V15531000005669	V155310000005
7	7BE29094-38EB-EF11-87A4-E89C250F7201	ORI	T	1553	10	V15531000005670	V155310000005

Se continua el despacho



INCGET - 20250213-01 - Doble commit en Origis con orden de servicio a domicilio con monedero electrónico

Autor	Versión	Fecha de elaboración	Fecha de última actualización
Juan Pérez	1.3	14-02-2025	19-02-2025

Versión: 10.5.60701.2 CAJA: 10 07:24 PM 14/02/2025 Rx

¡Hola PRUEBA MONEDERO! 🌐 Te Atiende CAJERO

Saldo en Monedero e\$100.00 \$0.00 PENDIENTES: 13 POR DESPACHAR: 3 POR ENTREGAR: 0 POR DEVOLVER: 0

Servicio a domicilio FOLIO: 2000000009APP

G AHORRO FENAZOPIRIDINA 100 MG ORAL 20 TAB 3070 \$44.00

PSICOFARMA SA DE CV TABLETAS PSICOFARMA SA DE CV TABLETAS

Cant.	Productos	Precio Máximo	Precio Ahorro
1	G AHORRO FENAZOPIRIDINA 100 MG ORAL 20 TAB 3070	\$96.46	\$44.00

Al Monedero \$0.00 Total de esta compra \$96.46 Total a Pagar \$44.00

F2 Buscar F3 Reimpresión F4 Servicios F5 Monedero F7 Recetas F9 Última Venta F11 Cotizaciones Ctrl + F10 Verificador F10 Forma de Pago

Versión: 10.5.60701.2 CAJA: 10 07:24 PM 14/02/2025 Rx

¡Hola PRUEBA MONEDERO! 🌐 Te Atiende CAJERO

Saldo en Monedero e\$100.00 \$0.00 PENDIENTES: 13 POR DESPACHAR: 3 POR ENTREGAR: 0 POR DEVOLVER: 0

Servicio a domicilio FOLIO: 2000000009APP

G AHORRO FENAZOPIRIDINA 100 MG ORAL 20 TAB 3070 \$44.00

PSICOFARMA SA DE CV TABLETAS PSICOFARMA SA DE CV TABLETAS

Cant.	Productos	Precio Máximo	Precio Ahorro
1	G AHORRO FENAZOPIRIDINA 100 MG ORAL 20 TAB 3070	\$96.46	\$44.00

Notificación

¿Desea capturar los datos del médico?

Cancelar Aceptar

Al Monedero \$0.00 Total de esta compra \$96.46 Total a Pagar \$44.00

F2 Buscar F3 Reimpresión F4 Servicios F5 Monedero F7 Recetas F9 Última Venta F11 Cotizaciones Ctrl + F10 Verificador F10 Forma de Pago

Se esperan ~2 segundos y se realiza el rollback de la transacción en la sesión que bloquea la tabla, ejecutando únicamente la 3ra línea del script.



INCGET - 20250213-01 - Doble commit en Origis con orden de servicio a domicilio con monedero electrónico

Autor	Versión	Fecha de elaboración	Fecha de última actualización
Juan Pérez	1.3	14-02-2025	19-02-2025

The screenshot shows a database interface with several panes. The top pane displays a transaction log with three entries:

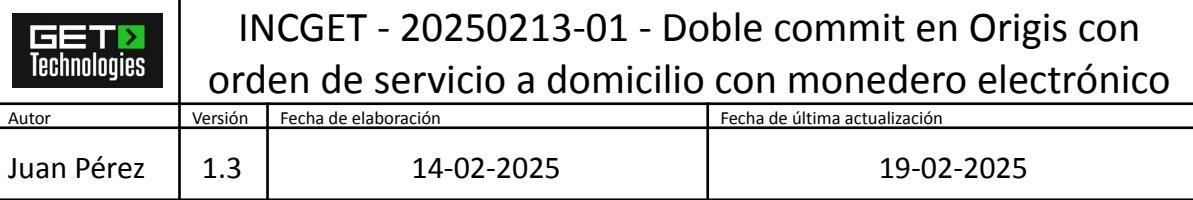
```
1 BEGIN TRANSACTION;
2 SELECT * FROM genProgramaLealtaBitacoraReg WITH (XLOCK HOLDLOCK)
3 ROLLBACK TRANSACTION;
```

The bottom pane shows a table titled "bitacora" with 9 rows of data. The columns include idTransaccion, tipoPrograma, tipoOperacion, codigoSucursal, numeroCaja, folioTransaccion, and folioProgr. The data is as follows:

idTransaccion	tipoPrograma	tipoOperacion	codigoSucursal	numeroCaja	folioTransaccion	folioProgr
1 83CEB55F-46EA-EF11-87A1-F43808267C81	ORI	T	1553	10 V1553100000562	V1553100000005c	
2 9D36d6F3-26EB-EF11-87A4-E89c250F7201	ORI	T	1553	10 V1553100000565	V1553100000005d	
3 E4956516-27EB-EF11-87A4-E89c250F7201	ORI	T	1553	10 D15531000000413	D15531000000000	
4 10F516A9-2AE8-EF11-87A4-E89c250F7201	ORI	T	1553	10 V1553100000568	V1553100000005e	
5 10F516A9-2AE8-EF11-87A4-E89c250F7201	ORI	R	1553	10 V1553100000568	V1553100000005f	
6 8FB450DF-37EB-EF11-87A4-E89c250F7201	ORI	T	1553	10 V1553100000569	V1553100000005g	
7 7BEC29094-38EB-EF11-87A4-E89c250F7201	ORI	T	1553	10 V15531000005670	V1553100000005k	
8 48203B6A-3AEB-EF11-87A4-E89c250F7201	ORI	T	1553	10 V15531000005672	V1553100000005l	
9 48203B6A-3AEB-EF11-87A4-E89c250F7201	ORI	R	1553	10 V15531000005672	V1553100000005m	

Ocurre el error de doble commit y se realiza una reversa automatica de la transacción.

The screenshot shows the Origis POS system interface. At the top, it says "¡Hola PRUEBA MONEDERO! Te Atiende CAJERO". The main screen displays a product for "G AHORRO FENAZOPIRIDINA 100 MG ORAL 20 TAB 3070" with a price of \$44.00. Below the product, there is an "Error" dialog box with the message: "La venta no puede ser finalizada, las formas de pago aplicadas serán canceladas." At the bottom, the payment summary shows "Al Monedero \$0.00", "Total de esta compra \$96.46", and "Total a Pagar \$44.00".





INCGET - 20250213-01 - Doble commit en Origis con orden de servicio a domicilio con monedero electrónico

Autor	Versión	Fecha de elaboración	Fecha de última actualización
Juan Pérez	1.3	14-02-2025	19-02-2025

The screenshot shows the Fiddler Classic interface with several network requests listed in the main pane. A modal dialog box is open in the center, displaying the message: "La venta no puede ser finalizada, las formas de pago aplicadas serán canceladas." (The sale cannot be finalized, applied payment methods will be canceled.)

At the bottom left, a status bar indicates "Capturing" and "All Processes". The bottom right corner shows the system tray with icons for battery, signal, volume, and time (10:25 PM, ENG, 14/2/2023).

	INCGET - 20250213-01 - Doble commit en Origis con orden de servicio a domicilio con monedero electrónico		
Autor	Versión	Fecha de elaboración	Fecha de última actualización
Juan Pérez	1.3	14-02-2025	19-02-2025

Diagnóstico

Versión de software utilizada	10.5.60702.19
Se reproduce el error	Si
Se corrige el error	No

Flujo de incidencia

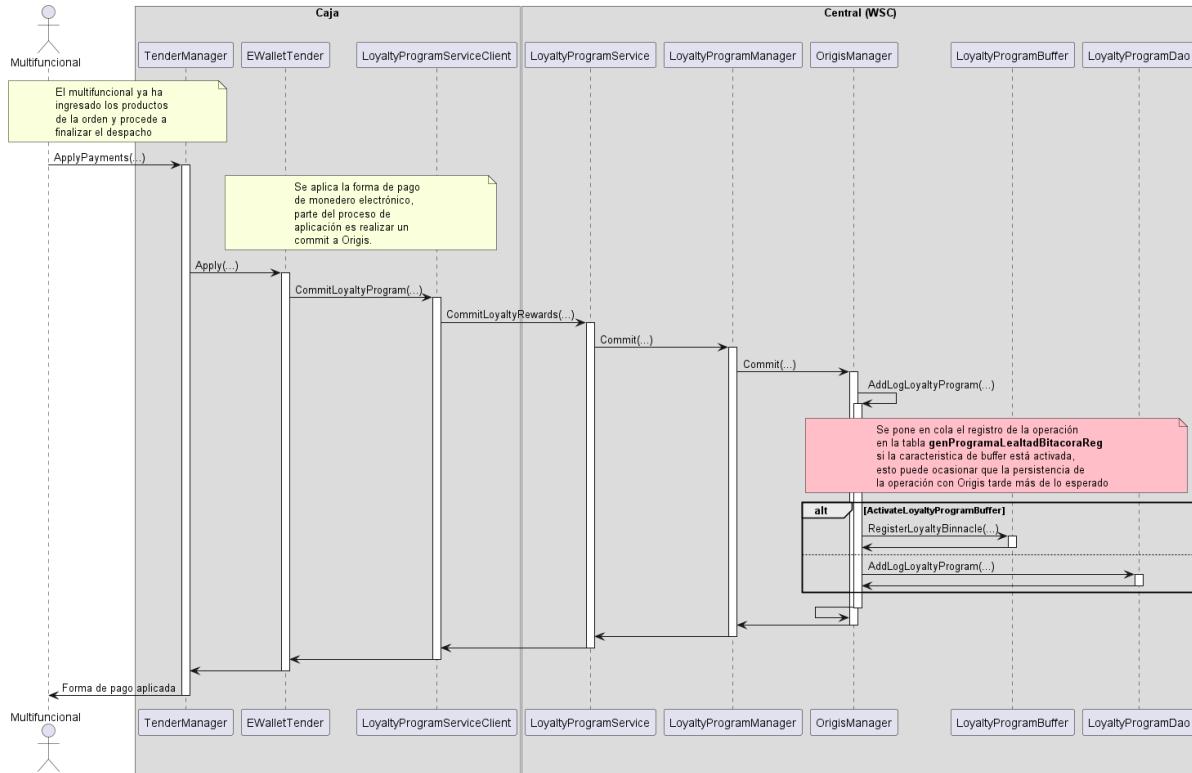
Cuando se procesa una orden SAD con forma de pago de monedero electrónico, esta es procesada antes de finalizar la venta, en su flujo de procesamiento se realiza el **commitTrans** con **Origis**, antes de finalizar la venta, este commit debería generar un registro en la tabla **genProgramaLealtadBitacoraReg**, este registro se puede utilizar en el flujo de la operación de commit para evitar realizar un segundo commit por error, si se realiza la operación de commit para una transacción que ya se registró previamente, se revisa la tabla **genProgramaLealtadBitacoraReg** para obtener el registro que contenga la información del commit realizado anteriormente y así evitar realizar un segundo commit por error.

Existe una particularidad al persistir los registros de **genProgramaLealtadBitacoraReg**, actualmente, existe una funcionalidad que permite persistir estos registros de manera asíncrona, es una funcionalidad que se activa por medio del **Web.config**, estableciendo el valor de la propiedad **ActivateLoyaltyBuffer** a **true**, este valor es **true** en los nodos de **WSC** en el ambiente de producción, lo que significa que **los registros se persisten de manera asíncrona** para esta tabla.

El flujo de la incidencia se explica en los siguientes diagramas de secuencia:

INCGET - 20250213-01 - Doble commit en Origis con orden de servicio a domicilio con monedero electrónico

Autor	Versión	Fecha de elaboración	Fecha de última actualización
Juan Pérez	1.3	14-02-2025	19-02-2025



[Ver figura](#)

Al aplicar la forma de pago con monedero electrónico, **se realiza el primer commit de Origis**, este commit se realiza de forma correcta, el pago es aplicado y el commit es registrado en Origis, esto se realiza en la siguiente sección de código:



INCGET - 20250213-01 - Doble commit en Origis con orden de servicio a domicilio con monedero electrónico

Autor	Versión	Fecha de elaboración	Fecha de última actualización
Juan Pérez	1.3	14-02-2025	19-02-2025

```
C# EwalletTender.cs x C# ElectronicTenderTransactionClient.cs C# TenderManager.cs C# ITransactionPayment.cs
30     public class EwalletTender : TransactionPaymentBase, ITransactionPayment
179     public bool Apply(PaymentMethodModel paymentMethod, string transactionId, DateTime fechaTransacion, bool isRetry)
182         if (paymentMethod.EWallet == null && this.saleModel == null)
183         {
184             throw new InvalidOperationException("Missing e-wallet object");
185         }
186
187         if (paymentMethod.IsWalletAlreadyPaid)
188         {
189             try
190             {
191                 using (var client = new LoyaltyProgramServiceClient())
192                 {
193                     var response = client.CommitLoyaltyProgram(this.saleModel, paymentMethod.Amount, paymentMethod.WalletAlreadyPaidOrigin);
194                     if (response.HasError)
195                     {
196                         // Se realiza el RollBack del Monedero
197                         client.RollbackLoyaltyRewards(this.saleModel);
198
199                         return false;
200                     }
201                     else
202                     {
203                         this.ApplyElectronicPayment(paymentMethod, transactionId);
204                         this.saleModel.CurrentEWallet.AuthorizationNumber = ((SaleBase)response.Result.Customer.POSLog.Single()).CustomerOrderTra
205                         if (response.Result.Customer.ARTHHeader.Description.Any())
206                         {
207                             this.saleModel.CurrentEWallet.ExitUserMessage = response.Result.Customer.ARTHHeader.Description.First().Value;
208                         }
209                         return true;
210                     }
211                 }
212             }
213         }
214     }
```

Ubicación: Farmax/Src/POS/Win.Tender/ElectronicPayments/EwalletTender.cs

Luego, como se observa en el diagrama de flujo, el registro puede ser persistido de inmediato o de manera asíncrona dependiendo de una bandera de configuración que se obtiene desde el archivo **Web.config**.



INCGET - 20250213-01 - Doble commit en Origis con orden de servicio a domicilio con monedero electrónico

Autor	Versión	Fecha de elaboración	Fecha de última actualización
Juan Pérez	1.3	14-02-2025	19-02-2025

```
public abstract class LoyaltyProgramProviderBase : ILoyaltyProgram
{
    /// <summary>
    /// <param name="log">información de la bitácora</param>
    /// <returns>retorna True si fue exitoso</returns>
    protected ResultStruct<bool> AddLogLoyaltyProgram(OpenProgramaLealtadBitacoraReg log)
    {
        log.fechaActualizacion = DateTime.Now;

        ResultStruct<bool> result;
        try
        {
            if (ActivateLoyaltyProgramBuffer)
            {
                GENPROGRAMALEALTADBITACORAREG entry = new GENPROGRAMALEALTADBITACORAREG()
                {
                    IDTRANSAACION = log.idTransaccion,
                    TIPOPROGRAMA = log.tipoPrograma,
                    TIPOOPERACION = log.tipoOperacion,
                    CODIGOSUCURSAL = log.codigoSucursal,
                    NUMEROCAJA = log.numeroCaja,
                    FOLIOTRANSACCION = log.folioTransaccion,
                    FOLIOPROGRAMA = log.folioPrograma,
                    FECHAOPERACION = log.fechaOperacion,
                    TARJETALEALTAD = log.tarjetaLealtad,
                    NUMEROAUTORIZACION = log.numeroAutorizacion,
                    DATOSSOLICITUD = log.datosSolicitud,
                    DATOSRESPUESTA = log.datosRespuesta,
                    CODIGOUSUARIO = log.codigoUsuario,
                    FECHAACTUALIZACION = log.fechaActualizacion,
                    VERSIONXML = log.VersionXml
                };
                LoyaltyProgramBuffer.RegisterLoyaltyBinnacle(entry);
            }
        }
    }
}
```

Ubicación:

Farmax/Src/POS/BusinessLogic.LoyaltyProgram.Provider/LoyaltyProgramProviderBase.cs

Esto registra el procesamiento de la bitácora en una cola para su posterior persistencia en base de datos:

	INCGET - 20250213-01 - Doble commit en Origis con orden de servicio a domicilio con monedero electrónico		
Autor	Versión	Fecha de elaboración	Fecha de última actualización
Juan Pérez	1.3	14-02-2025	19-02-2025

```

 12  public class LoyaltyProgramBuffer
 13
 14      /// <summary>
 15      /// Método para guardar la bitácora de programa de lealtad
 16      /// </summary>
 17      /// <param name="dataManagement">Obtiene la instancia de la base de datos central</param>
 18      public static void SaveLoyaltyBinnacle(DatabaseTools tools, int gty)
 19      {
 20          int count = 0;
 21
 22          List<GENPROGRAMALEALTADBITACORAREG> regsToSend = new List<GENPROGRAMALEALTADBITACORAREG>();
 23      }

```

Ubicación: Farmax/Src/Framework/Application.Core/LoyaltyProgramBuffer.cs

Esta cola de procesamiento inserta en base de datos un lote de 50 registros en cada iteración y existe un tiempo de espera de un segundo entre iteraciones, aquí se puede observar el tiempo de espera, se encuentra en duro:

```

 7
 8     namespace FAhorro.POS.Services.Host
 9     {
 10         /// <summary>
 11         /// Clase para ejecutar la descarga de logs.
 12         /// </summary>
 13         public class LoyaltyProgramTask : BackgroundTask
 14         {
 15             private static AxObjectOrientedDatabase.DatabaseTools _tools = null;
 16             /// <summary>
 17             /// Constructor de la clase de descarga de logs.
 18             /// </summary>
 19             /// <param name="logTimeOut">Tiempo de espera para conectar con el servidor de logs.</param>
 20             /// <param name="logServer">Nombre del servidor de logs.</param>
 21             /// <param name="logServerPort">Puerto del servidor de logs.</param>
 22
 23             public LoyaltyProgramTask(AxObjectOrientedDatabase.DatabaseTools tools) : base("LOYALTY_PROGRAM_TASK", 1)
 24             {
 25                 _tools = tools;
 26             }

```

Ubicación: Farmax/Src/POS/Services.Host/LoyaltyProgramTask.cs

Se observa que la cantidad de registros por lote tiene el valor de 50 registros por defecto:

	INCGET - 20250213-01 - Doble commit en Origis con orden de servicio a domicilio con monedero electrónico		
Autor	Versión	Fecha de elaboración	Fecha de última actualización
Juan Pérez	1.3	14-02-2025	19-02-2025

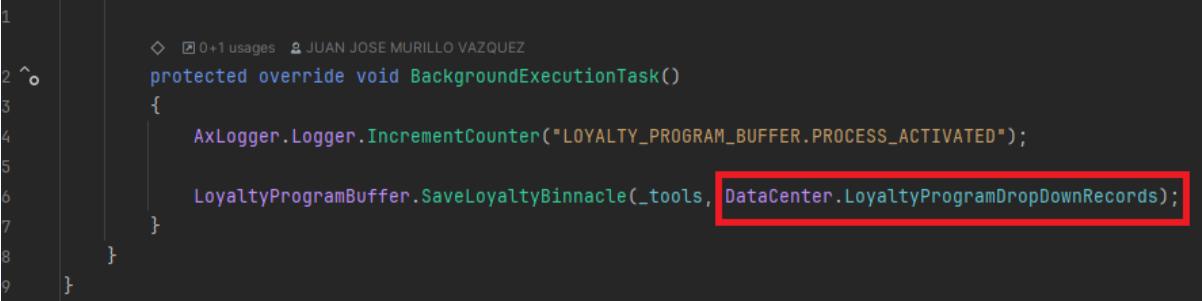


```

10     public class DataCenter
11     {
12         public static string LoggerApp = "FARMAX_WSC",
13         public static string LogFolder = @"c:\programdata\FAhorro";
14         public static AxLogger.LogLevel LogLevel = AxLogger.LogLevel;
15         public static bool EnforceCertificates = true;
16         public static bool ActivateLogBuffer = false;
17         public static int LogBufferChunkSize = 5000;
18         public static bool ActivateLogService = false;
19         public static Int32 LogServerTimeOut = 5;
20         public static bool LogServerEnabled = false;
21         public static Int32 LogServerPort = 4527;
22         public static String ServerName = "FARMAX_WSC";
23         public static String LogServer = "localhost";
24         public static int LogBufferMaxSize = 20000;
25         public static bool LogPerHour = false;
26         public static bool ActivateLoyaltyProgramBuffer = false;
27         public static Int32 LoyaltyProgramDropDownRecords = 50;
28     }

```

Ubicación: Farmax/Src/POS/Services.Host/DataCenter.cs



```

1
2     protected override void BackgroundExecutionTask()
3     {
4         AxLogger.Logger.IncrementCounter("LOYALTY_PROGRAM_BUFFER.PROCESS_ACTIVATED");
5
6         LoyaltyProgramBuffer.SaveLoyaltyBinnacle(_tools, DataCenter.LoyaltyProgramDropDownRecords);
7     }
8 }

```

Ubicación: Farmax/Src/POS/Services.Host/LoyaltyProgramTask.cs

Se puede ver que la cantidad del lote de registros a insertar es configurable por medio de un parámetro de configuración del **Web.config**:



INCGET - 20250213-01 - Doble commit en Origis con orden de servicio a domicilio con monedero electrónico

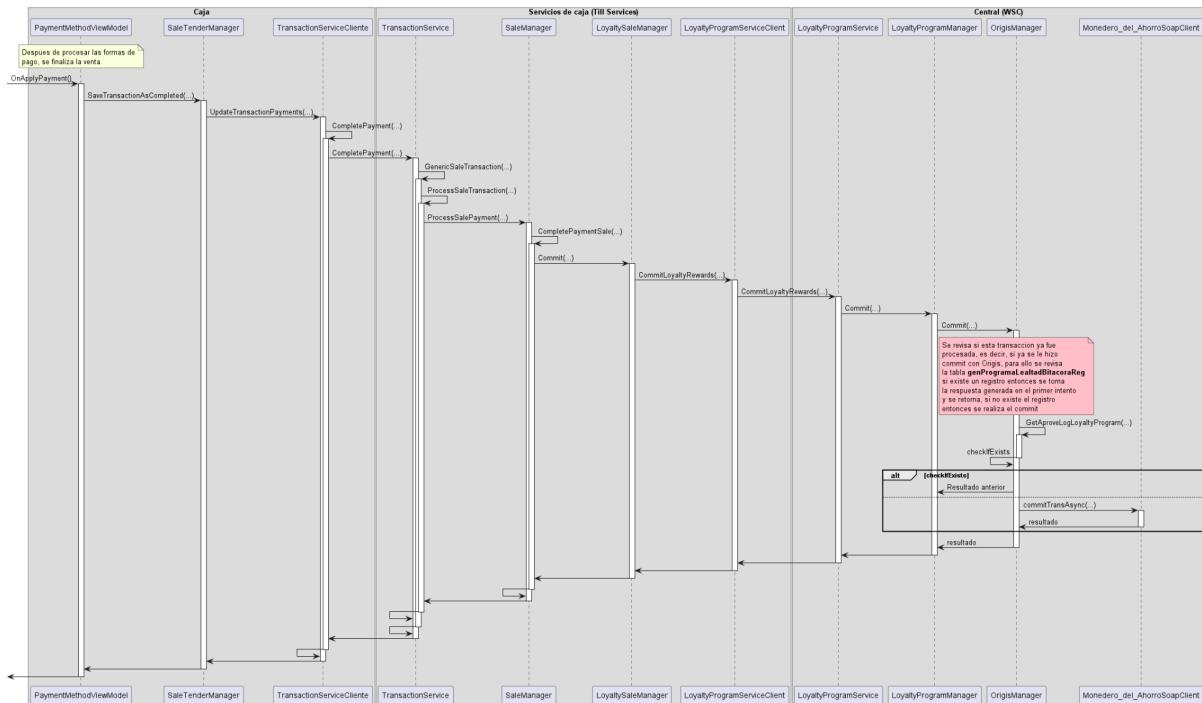
Autor	Versión	Fecha de elaboración	Fecha de última actualización
Juan Pérez	1.3	14-02-2025	19-02-2025

```
C# LoyaltyProgramBuffer.cs    C# LoyaltyProgramTask.cs    C# DataCenter.cs X    C# StartSyncTillService.cs    C# Global.asax.cs

10     public class DataCenter
11
12         // Determina cuantos registros se pueden descargar en forma simultanea
13         try
14         {
15             String loyaltyProgramRecords = System.Configuration.ConfigurationManager.AppSettings["LoyaltyProgramDropDownRecords"];
16             if (loyaltyProgramRecords != null && loyaltyProgramRecords.Length > 0)
17             {
18                 int size = -1;
19
20                 if (Int32.TryParse(loyaltyProgramRecords, out size) && size >= 0)
21                 {
22                     LoyaltyProgramDropDownRecords = size;
23                 }
24             }
25         }
26         catch (Exception) { }
```

Ubicación: Farmax/Src/POS/Services.Host/DataCenter.cs

Si el registro no existe en la tabla de la bitácora **genProgramaLealtadBitacoraReg**, o no se ha guardado aún debido a un alto nivel de concurrencia, es posible que la transacción se considere como no procesada y por consecuencia se realice un segundo commit, como se muestra en el siguiente diagrama de secuencia:



[Ver figura](#)

	INCGET - 20250213-01 - Doble commit en Origis con orden de servicio a domicilio con monedero electrónico		
Autor	Versión	Fecha de elaboración	Fecha de última actualización
Juan Pérez	1.3	14-02-2025	19-02-2025

Se observa que el registro debe existir en base de datos antes de realizar el segundo intento de commit, es posible que debido a la alta concurrencia el registro aún no sea guardado en base de datos, por lo que la transacción se considera como no procesada y se realiza el segundo intento de commit.

```
cr# OrigisManager.cs < 
36     public class OrigisManager : LoyaltyProgramProviderBase
417     public override async Task<ResultStruct<LoyaltyBenefits>> Commit(LoyaltyTransaction benefits)
419         try
421             using (var serviceClient = new Monedero_del_AhorroSoapClient("Monedero_del_AhorroSoap", false, true))
423                 var accountResult = new LoyaltyBenefits();
424
425                 // se verifica si la transaccion ya fue procesada... si ya fue procesada se regresa el resultado obtenido por proveedor
426                 var checkIfExists = await GetAproveLogLoyaltyProgram(new CancelTokens { BusinessId = Convert.ToInt32(benefits.Branch), tillId = 1 });
427
428                 if (checkIfExists.Result.Any(t => t.tipoPrograma == this.ConvertEnumLoyaltyToDBLoyalty(benefits.LoyaltyProgramID)))
429                 {
430                     accountResult = SerializerData.XmlDataDeserialize<LoyaltyBenefits>(checkIfExists.Result.Single(t => t.tipoPrograma == this.ConvertEnumLoyaltyToDBLoyalty(benefits.LoyaltyProgramID)).data);
431                 }
432             return accountResult;
433         }
434     }
435 }
```

Ubicación: Farmax/Src/POS/BusinessLogic.LoyaltyProgram.Origis/OrigisManager.cs



INCGET - 20250213-01 - Doble commit en Origis con orden de servicio a domicilio con monedero electrónico

Autor	Versión	Fecha de elaboración	Fecha de última actualización
Juan Pérez	1.3	14-02-2025	19-02-2025

Posibles causas

El registro que evita que se realice el segundo commit aún no había sido persistido en la base de datos, esto se debe a que la funcionalidad de buffer para la persistencia asíncrona de los registros para la tabla **genProgramaLealtadBitacoraReg** se encuentra activa, esto quiere decir que el registro no se guarda en base de datos de inmediato.

Utilizando la consulta:

```
SELECT * FROM dbo.genBitacoraMonedero
WHERE codigoSucursal = 2191
AND folioTransaccion = 'V21912000038455'
ORDER BY syncCreadoFecha;
```

Es posible ver que el segundo commit se realiza antes de que el registro de la operación en **genProgramaLealtadBitacoraReg** sea persistido en base de datos.

	1	2	3
idBitacora	F6718F1C-14EA-EF11-80E2-0050568B43B1	FD718F1C-14EA-EF11-80E2-0050568B43B1	56
folioTransaccion	V21912000038455	V21912000038455	V2
codigoSucursal	2191	2191	21
numeroCaja	20	20	20
codigoCajero	24308	24308	24
numeroTarjeta	9600233812350	9600233812350	96
operacion	C	C	R
datosEnviados	sSessionId=14365713777,sArticulos=750223703070,1,44,44,sSal...	sSessionId=14365713777,sArticulos=750223703070,1,44,44,sSal...	ss
datosRecibidos	SESSIONID=14365713777,AUTORIZACION=1607876292,ARTICULOS=7502...	<null>	SE
pagoMonedero	44.0000	44.0000	0.
codigoError	0	42	0
mensaje		Venta No Aplicada: La transacción ya existe [042.3]	
confirmaReversa	<null>	<null>	tr
fechaTransaccion	2025-02-13 08:09:27.000	2025-02-13 08:09:29.393	20
versionCentral	1000400291	1000400291	10
syncCreadoFecha	2025-02-13 08:09:26.317	2025-02-13 08:09:27.310	20
syncActualizadoFecha	2025-02-13 08:09:26.317	2025-02-13 08:09:27.310	20
syncActualizadoPor	FARMAX.DEFAULT	FARMAX.DEFAULT	FA
syncVersionRegistro	1	1	1
syncTransferido	I	I	I

Este es realizado a las **08:09:27.310**.

Se revisa la hora de persistencia para el registro en **genProgramaLealtadBitacoraReg** utilizando el siguiente script:

```
SELECT * FROM dbo.genProgramaLealtadBitacoraReg
WHERE codigoSucursal = 2191
AND numeroCaja = 20
AND idTransaccion = '07AE6D21-14EA-EF11-A77B-ECB1D743BE7C';
```



INCGET - 20250213-01 - Doble commit en Origis con orden de servicio a domicilio con monedero electrónico

Autor	Versión	Fecha de elaboración	Fecha de última actualización
Juan Pérez	1.3	14-02-2025	19-02-2025

	1	2
idBitacora	96729663	96729688
idTransaccion	07AE6D21-14EA-EF11-A77B-ECB1D743BE7C	07AE6D21-14EA-EF11-A77B-ECB1D743BE7C
tipoPrograma	ORI	ORI
tipoOperacion	T	R
codigoSucursal	2191	2191
numeroCaja	20	20
folioTransaccion	V21912000038455	V21912000038455
folioPrograma	V21912000038455	V21912000038455
fechaOperacion	2025-02-13 08:09:27.000	2025-02-13 08:09:27.000
tarjetaLealtad	9600233812350	9600233812350
numeroAutorizacion	1607876292	1607876292
datosSolicitud	<LoyaltyTransaction xmlns="http://schemas.d	<LoyaltyBenefits xmlns="http://schemas.data
datosRespuesta	<LoyaltyBenefits xmlns="http://schemas.data	<LoyaltyCancelTransaction xmlns="http://sche
codigoUsuario	24308	24308
fechaActualizacion	2025-02-13 08:09:30.247	2025-02-13 08:09:36.853
VersionXml	2	2
syncCreadoFecha	2025-02-13 08:09:31.057	2025-02-13 08:09:38.087
syncActualizadoFecha	2025-02-13 08:09:31.057	2025-02-13 08:09:38.087
syncActualizadoPor		
syncVersionRegistro	1	1

Se observa que el registro fue persistido a las **08:09:31.057**, eso es alrededor de 4 segundos después del segundo commit con Origis, por lo que no había forma de que la transacción fuera considerada como procesada al momento de intentar realizar el segundo commit.

	INCGET - 20250213-01 - Doble commit en Origis con orden de servicio a domicilio con monedero electrónico		
Autor	Versión	Fecha de elaboración	Fecha de última actualización
Juan Pérez	1.3	14-02-2025	19-02-2025

Mitigación del problema

Aumentar tamaño de lote y reducir tiempo de procesamiento de buffer de bitácora de programa de lealtad

Se propone mitigar el problema de concurrencia aumentando la cantidad de registros por lote para el buffer de bitácora de **genProgramaLealtadBitacoraReg**, esto permitirá procesar más registros en la misma cantidad de tiempo, lo que puede reducir la cantidad de veces que ocurre la incidencia.

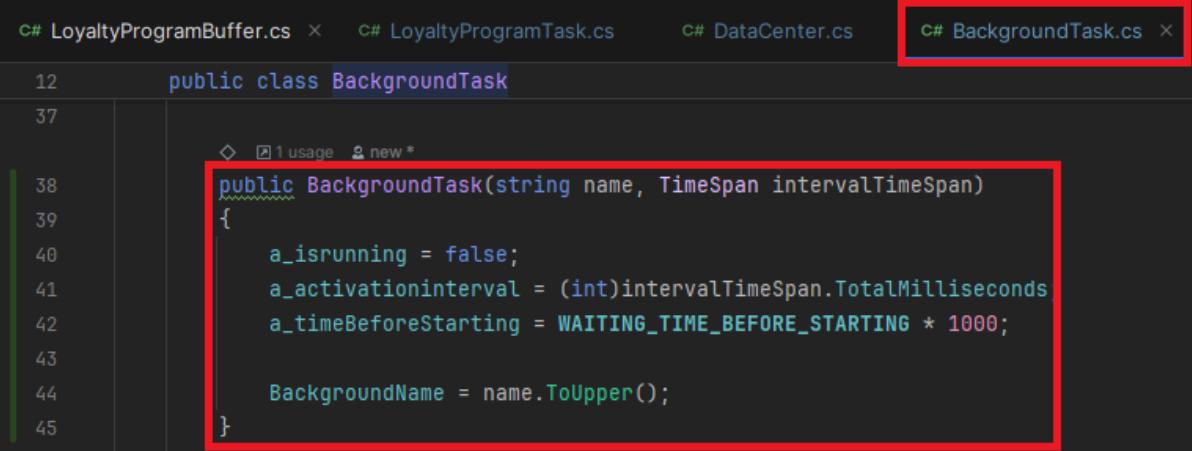
Actualmente, existe una configuración que se puede utilizar para cambiar el tamaño del lote de registros de bitácora a procesar por iteración, este valor se establece por medio del **Web.config** en **Servicios Web Centrales** y se establece al iniciar la aplicación:

```
<configuration>
    ...
    <appSettings>
        ...
        <add key="LoyaltyProgramDropDownRecords" value="100" />
    </appSettings>
</configuration>
```

Esta funcionalidad **ya se encuentra implementada actualmente**, con esto es suficiente para que la cantidad de registros a procesar por cada iteración cambie a **100**.

Cómo cambio, se propone una manera de configurar el intervalo de iteraciones por medio de un **TimeSpan**, lo que permite definir el intervalo a escala de milisegundos, se agrega un nuevo constructor en **BackgroundTask** que permita definir el intervalo por medio de un **TimeSpan**:

	INCGET - 20250213-01 - Doble commit en Origis con orden de servicio a domicilio con monedero electrónico		
Autor	Versión	Fecha de elaboración	Fecha de última actualización
Juan Pérez	1.3	14-02-2025	19-02-2025



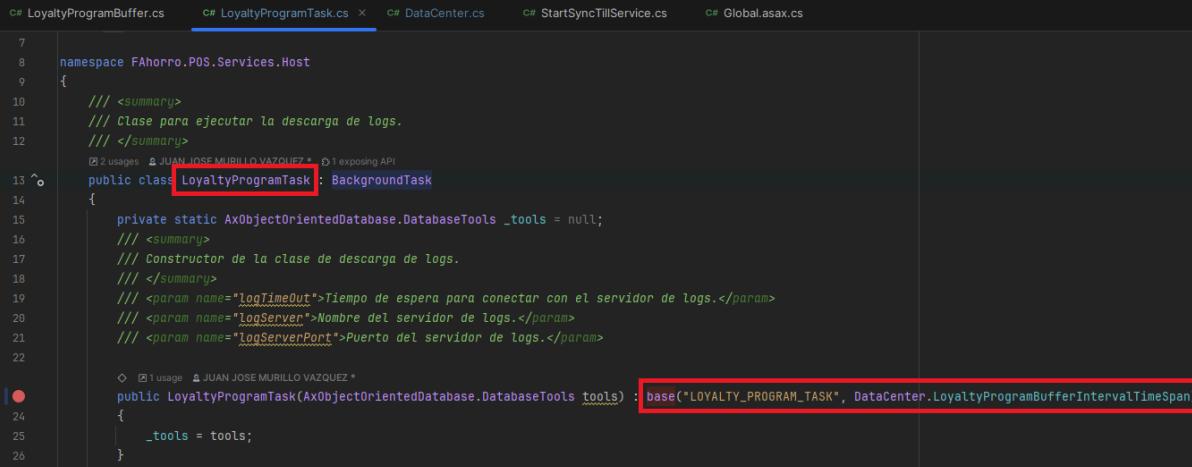
```

C# LoyaltyProgramBuffer.cs × C# LoyaltyProgramTask.cs × C# DataCenter.cs × C# BackgroundTask.cs ×
12     public class BackgroundTask
37
38         ◇ 1 usage 2 new*
39             public BackgroundTask(string name, TimeSpan intervalTimeSpan)
40             {
41                 a_isrunning = false;
42                 a_activationinterval = (int)intervalTimeSpan.TotalMilliseconds;
43                 a_timeBeforeStarting = WAITING_TIME_BEFORE_STARTING * 1000;
44
45                 BackgroundName = name.ToUpper();
}

```

Ubicación: AnxoFARMAX/Common/BackgroundTask.cs

Este constructor de **BackgroundTask** es utilizado por **LoyaltyProgramTask** para definir el tiempo de intervalo utilizando un **TimeSpan**, este proceso es el que se encarga de procesar los lotes de los registros de la bitácora:



```

C# LoyaltyProgramBuffer.cs C# LoyaltyProgramTask.cs × C# DataCenter.cs C# StartSyncTillService.cs C# Global.asax.cs
7
8     namespace FAhorro.POS.Services.Host
9     {
10
11         /// <summary>
12         /// Clase para ejecutar la descarga de logs.
13         /// </summary>
14         [2 usages & JUAN JOSE MURILLO VAZQUEZ * 1 exposing API]
15         public class LoyaltyProgramTask : BackgroundTask
16         {
17             private static AxObjectOrientedDatabase.DatabaseTools _tools = null;
18
19             /// <summary>
20             /// Constructor de la clase de descarga de logs.
21             /// </summary>
22             /// <param name="logTimeOut">Tiempo de espera para conectar con el servidor de logs.</param>
23             /// <param name="logServer">Nombre del servidor de logs.</param>
24             /// <param name="logServerPort">Puerto del servidor de logs.</param>
25
26             ◇ 1 usage & JUAN JOSE MURILLO VAZQUEZ *
27             public LoyaltyProgramTask(AxObjectOrientedDatabase.DatabaseTools tools) : base("LOYALTY_PROGRAM_TASK", DataCenter.LoyaltyProgramBufferIntervalTimeSpan)
28             {
29                 _tools = tools;
}

```

Ubicación: Farmax/Src/POS/Services.Host/LoyaltyProgramTask.cs

Se define un valor por defecto para el intervalo de 1 segundo, valor que se encontraba antes en duro:



```

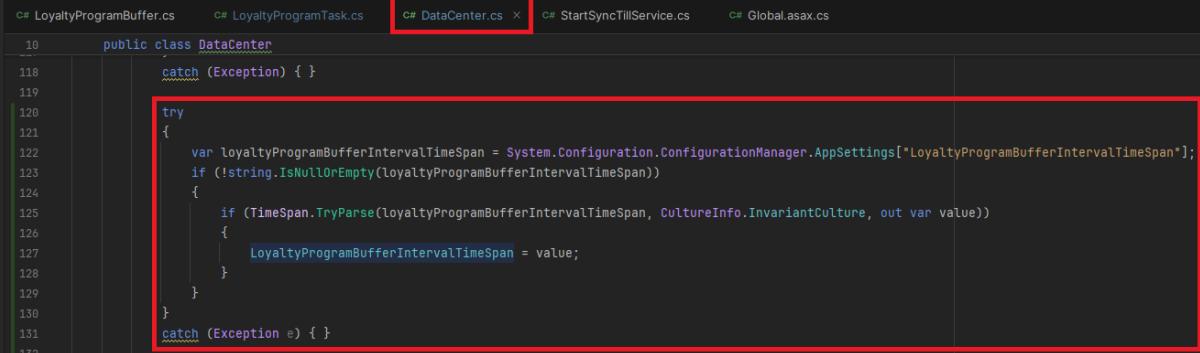
C# LoyaltyProgramBuffer.cs C# LoyaltyProgramTask.cs C# DataCenter.cs × C# StartSyncTillService.cs C# Global.asax.cs
10     public class DataCenter
11     {
12         public static Int32 LoyaltyProgramDropDownRecords = 50;
13
14         public static TimeSpan LoyaltyProgramBufferIntervalTimeSpan = TimeSpan.FromSeconds(1);
}

```

Ubicación: Farmax/Src/POS/Services.Host/DataCenter.cs

	INCGET - 20250213-01 - Doble commit en Origis con orden de servicio a domicilio con monedero electrónico		
Autor	Versión	Fecha de elaboración	Fecha de última actualización
Juan Pérez	1.3	14-02-2025	19-02-2025

Se carga la configuración desde el **Web.config**, utilizando el valor del parámetro **LoyaltyProgramBufferIntervalTimeSpan**:



```

10  public class DataCenter
11
12      catch (Exception) { }
13
14      try
15      {
16          var loyaltyProgramBufferIntervalTimeSpan = System.Configuration.ConfigurationManager.AppSettings["LoyaltyProgramBufferIntervalTimeSpan"];
17          if (!string.IsNullOrEmpty(loyaltyProgramBufferIntervalTimeSpan))
18          {
19              if (TimeSpan.TryParse(loyaltyProgramBufferIntervalTimeSpan, CultureInfo.InvariantCulture, out var value))
20              {
21                  LoyaltyProgramBufferIntervalTimeSpan = value;
22              }
23          }
24      }
25      catch (Exception e) { }
26

```

Ubicación: Farmax/Src/POS/Services.Host/DataCenter.cs

Esta configuración se puede especificar en el **Web.config** de la siguiente manera:

```

<configuration>
    ...
    <appSettings>
        ...
        <add key="LoyaltyProgramBufferIntervalTimeSpan"
value="00:00:00.500" />
    </appSettings>
</configuration>

```

Esta configuración establece el tiempo de espera entre cada iteración de procesamiento a 500 milisegundos.



INCGET - 20250213-01 - Doble commit en Origis con orden de servicio a domicilio con monedero electrónico

Autor	Versión	Fecha de elaboración	Fecha de última actualización
Juan Pérez	1.3	14-02-2025	19-02-2025

Operación de Origis de verificación de estado de orden

A **reservas del proveedor Origis**, podría ser posible que **Origis** implemente una operación de precommit antes de realizar un commit con el servicio de Origis, esta operación podría proveer de la información necesaria para determinar si es necesario realizar un commit o si la transacción ya se encuentra procesada, podría utilizarse para evitar realizar un segundo commit por error sin depender de los registros en la bitácora **genProgramaLealtadBitacoraReg**.

Esto podría implementarse antes de realizar el commit con Origis en **OrigisManager**, en el método **Commit**:

```
C# OrigisManager.cs
36     public class OrigisManager : LoyaltyProgramProviderBase
37
38     public override async Task<ResultStruct<LoyaltyBenefits>> Commit(LoyaltyTransaction benefits)
39
40     // Se manda procesado solo si no se a procesado con anterioridad --MDABLINDAJE setVentaOffline no funciona
41     DataSet result;
42     result = !benefits.IsOfflineSale
43     ? await serviceClient.commitTransAsync(benefits.Card, benefits.Branch, benefits.TillId,
44         benefits.PayrollID, benefits.SessionId,
45         this.MapBenefitsProducts(benefits.ShoppingBasket),
46         benefits.FolioTransaction, benefits.Redemption.ToString(CultureInfo.InvariantCulture),
47         benefits.SaleTotal.ToString(CultureInfo.InvariantCulture),
48         benefits.OperationDate.ToString("yyyyMMdd"),
49         benefits.IsHomeDelivery ? benefits.HomeDeliveryType.ToString() : "0",
50         this.GetCodePaymentMethod(benefits.PaymentMethods), this.authenticationCode)
51     : await serviceClient.setVentaOfflineAsync(benefits.Card, benefits.Branch, benefits.TillId,
52         benefits.PayrollID, this.MapBenefitsProducts(benefits.ShoppingBasket),
53         benefits.FolioTransaction, benefits.SaleTotal.ToString(CultureInfo.InvariantCulture),
54         benefits.OperationDate.ToString("yyyyMMdd"),
55         this.GetCodePaymentMethod(benefits.PaymentMethods), this.authenticationCode);
56
57 }
```

Ubicación: Farmax/Src/POS/BusinessLogic.LoyaltyProgram.Origis/OrigisManager.cs

En el mismo lugar, a **reservas del proveedor Origis**, podría ser posible que Origis, al realizar la operación **commiTans**, devuelva la misma información aunque haya ocurrido un código de **error 42**, con mensaje “**Venta No Aplicada: La transacción ya existe [042.3]**”, de esta manera podría continuar el flujo cuando ocurra ese error, utilizando la información recibida del servicio.