
	Dclose generado con ventas pendientes		
Autor	Versión	Fecha de elaboración	Última edición
Eduardo Ochoa	1.3	07-04-2025	25-09-2025

Control de versiones

Versión	Fecha	Autor	Descripción
1.0	08-04-2025	Eduardo Ochoa	Análisis y propuesta en proceso síncrono dclose
1.1	09-04-2025	Eduardo Ochoa	Se agregaron reintentos automáticos de sincronización
1.2	03-06-2025	Eduardo Ochoa	Se agrega diagramas del proceso de apertura de turno con autorización, sincronización y dclose así como la propuesta de solución.
1.3	25-09-2025	Eduardo Ochoa	Se realiza funcionalidad de configuración en base de datos para reintentos , configuración de permisos y ajustes en mensajes mostrados al usuario

Sucursal	NA
Caja	Todas
Venta	NA
Version	Todas

	Dclose generado con ventas pendientes		
Autor	Versión	Fecha de elaboración	Última edición
Eduardo Ochoa	1.3	07-04-2025	25-09-2025


Descripción general

Se han reportado casos en los que existen turnos abiertos en las cajas que no se reflejan en la base de datos del store. Esta situación ocurre cuando la sincronización entre la base local (caja) y el store falla por pérdida de conectividad, errores en el proceso o interrupciones del servicio. Como consecuencia, el sistema permite ejecutar cierres de día (dclose) sin contemplar estos turnos activos no sincronizados.

```

C# ClosingDayOp.cs x C# ClosingDayService.cs C# TillDataCenter.cs C# UpdateStoreService.cs
> Q- VENTURNOSCAJAREG x ↺ Cc W .* 7/7 ↑ ↓ 🔍 :
13      class ClosingDayOp
158      public static ProcessResult CloseWorkingDay()
188          new VENTURNOSCAJAREG()
190              COD10030003AL = storeConfig.StoreNumber,
191              IDFECHAOPERACION = workingDay.IDFECHAOPERACION
192          });
193
194      Boolean shiftClosed = true;
195      VENTURNOSCAJAREG shiftTill = null;
196
197      foreach (VENTURNOSCAJAREG shift in shiftCreated)
198      {
199          if (shift.ESTADOACTUAL != VENTURNOSCAJAREG.ESTADOS.CLOSED)
200          {
201              shiftClosed = false;
202              shiftTill = shift;
203              break;
204          }
205      }

```

	Dclose generado con ventas pendientes		
Autor	Versión	Fecha de elaboración	Última edición
Eduardo Ochoa	1.3	07-04-2025	25-09-2025

Diagnóstico

El registro de turnos es asíncrono entre la caja y el store, lo que permite que un turno abierto en la base local no llegue al store por fallos de red, errores de sincronización o interrupciones en el servicio SyncTillService. Al no estar reflejados los turnos abiertos en el store, el sistema puede ejecutar cierres de día sin considerar estos turnos.


Esto se verifica en el método `GenerateUploadSequenceMappingTable()` de la clase `SyncTillObjects`, el cual define la lista de entidades que serán sincronizadas desde las cajas hacia el store.

```
private static void GenerateUploadSequenceMappingTable()
{
    AddUpload(new GENCAJASCAT_V1());
    AddUpload(new CNFFUNCIONESCAJACAB_V1());

    AddUpload(new SEGUSUARIOSCAT_V1());
    AddUpload(new SEGUSUARIOSREG_V1());
    AddUpload(new SEGHISTORIACONTRASENASREG_V1());

    AddUpload(new GENDOCTORESCAT_V1());
    AddUpload(new GENDIRECCIONESDOCTORES_V1());

    AddUpload(new GENFOLIADORESCAJASREG_V1());
    AddUpload(new GENFOLIADORESSUCURSALESREG_V1());
    AddUpload(new GENFORMASPAGOSELECTRONICAS_V03());
    AddUpload(new GENLOTESDET_V1());
    AddUpload(new GENPROCESOSERRORESMENSAJESBITACORAREG_V1());
    AddUpload(new VENTURNOSCAJAREG_V1());
}
```

	<h1 style="text-align: center;">Dclose generado con ventas pendientes</h1>		
Autor	Versión	Fecha de elaboración	Última edición
Eduardo Ochoa	1.3	07-04-2025	25-09-2025

Causa del problema

El problema se origina en que la apertura de turnos depende de un proceso de sincronización asíncrono que no garantiza la persistencia inmediata en el store. Ante cualquier interrupción, el turno puede quedar registrado solo en la caja, sin reflejarse en el store, lo que permite que se realicen cierres de día sin contemplar turnos aún abiertos.

Posible solución

Para garantizar la integridad de los datos entre la base de datos local (caja) y la base del store, se propone modificar el proceso de apertura de turnos para que la sincronización con el store se intente hacer de manera inmediata y ventana de **autorización del responsable de turno**) o **Responsable de Farmacia** cuando la sincronización inmediata no se logre, permitiendo que el usuario **multifuncional continúe operando** mientras se reintenta la sincronización en segundo plano.

La solución consiste en invocar explícitamente el método SyncOperations del servicio ServiceAdapter inmediatamente después de registrar un nuevo turno desde el método StartOfDayTillSettle en FARMAX_CAJA. Si la sincronización falla, el sistema solicita autorización del **Responsable de Turno (RT)** o **Responsable de Farmacia(RF)** y así continuar operando sin sincronización con la base de datos del store pero seguir intentando sincronizar en segundo plano.

Implementación sugerida en services.

- El método StartOfDayTillSettle fue modificado para devolver un objeto ResultStruct<Guid> en lugar de un Guid directo para mandar errores encapsulados desde la capa DAO.


```

/// <summary>
/// Apertura un nuevo turno de caja.
/// </summary>
/// <param name="item">Información del turno.</param>
/// <param name="currentOperationDate">Contiene la información de la fecha de operacion</param>
/// <param name="currentShift">Contiene la información del turno correspondiente</param>
/// <returns>Identificador del turno abierto sino guid vacío</returns>
[usage Juan Murillo]
public ResultStruct<Guid> StartOfDayTillSettle(venTurnosCajaReg item, venFechaOperacion currentOperationDate, genTurnos currentShift)
{
    var result = tillSettleDao.StartOfDayTillSettle(item, currentOperationDate, currentShift);

    if (result.HasError)
    {
        return result;
    }

    return new ResultStruct<Guid>(result.Result);
}

```

	Dclose generado con ventas pendientes		
Autor	Versión	Fecha de elaboración	Última edición
Eduardo Ochoa	1.3	07-04-2025	25-09-2025


- La inserción del turno en la base de datos local se realiza dentro de una transacción, Si ocurre un error al guardar, se hace rollback.

```

using (var transaction = this.DataContext.Database.BeginTransaction())
{
    try
    {
        this.DataContext.SaveChanges();
        transaction.Commit();
    }
    catch (Exception ex)
    {
        FAhorro.POS.Application.Core.Log.WriteError(SubModule,
            message: string.Format("Error al guardar en la base de datos local: {0}", ex.ToString()));
        transaction.Rollback();

        throw;
    }
}

```

	Dclose generado con ventas pendientes		
Autor	Versión	Fecha de elaboración	Última edición
Eduardo Ochoa	1.3	07-04-2025	25-09-2025

Implementación de configuración dinámica para reintentos de sincronización:

Se agregaron parámetros configurables desde la base de datos mediante los códigos 1045 y 1046 en la tabla `cnfConfiguracionAgrupacionSucursalesCnf`, con sus respectivos valores en la tabla `genConfiguracionAgrupacionSucursalesReg`. La configuración opera **por agrupación de sucursales**, estableciendo:

- **Código 1045:** Número de reintentos de sincronización
- **Código 1046:** Tiempo de espera entre reintentos en milisegundos

Lógica de configuración jerárquica:

1. El sistema identifica la agrupación (`idAgrupacion`) a la que pertenece cada sucursal
2. Busca configuración específica para esa agrupación
3. Si no existe configuración específica, aplica los valores por defecto globales (4 reintentos con intervalos de 1000ms)


Esta implementación permite que **múltiples sucursales de una misma agrupación** compartan la misma configuración de reintentos, facilitando la gestión centralizada por zonas o características operativas similares.

```
var configurationManager = new ConfigurationManager();
int reintentos = 4; // default
int espera = 1000; // default

var syncRetriesConfig :List<PosAppConfiguration> = configurationManager.GetBusinessUnitConfiguration(
    item.codigoSucursal, codeConfiguration: 1045);
if (syncRetriesConfig.Any() && int.TryParse(syncRetriesConfig.First().Value, out int configReintentos))
    reintentos = configReintentos;

var syncDelayConfig :List<PosAppConfiguration> = configurationManager.GetBusinessUnitConfiguration(
    item.codigoSucursal, codeConfiguration: 1046);
if (syncDelayConfig.Any() && int.TryParse(syncDelayConfig.First().Value, out int configEspera))
    espera = configEspera;

int intentos = 0;
bool syncExitosa = false;
```

	<h2 style="text-align: center;">Dclose generado con ventas pendientes</h2>		
Autor	Versión	Fecha de elaboración	Última edición
Eduardo Ochoa	1.3	07-04-2025	25-09-2025

El sistema implementa un bucle de reintentos que ejecuta SyncOperations() hasta alcanzar el éxito o el límite configurado. Entre cada intento fallido aplica una espera de Thread.Sleep(espera) y registra el error. Si la respuesta es exitosa (response.IsOK) finaliza el proceso; caso contrario incrementa el contador e intenta nuevamente.

```
//Intentamos sincronizar
try
{
    while (intentos < reintentos && !syncExitosa)
    {
        try
        {
            var client = new ServiceAdapter();
            var syncRequest = new SyncOperationsRequest
            {
                StoreID = item.codigoSucursal,
                TableName = VENTURNOSCAJAREG.DatabaseMappingObject.TableName
            };


            var response = client.SyncOperations(syncRequest);

            if (response.IsOK)
            {
                syncExitosa = true;
                break;
            }
            else
            {
                intentos++;

                FAhorro.POS.Application.Core.Log.WriteWarning(SubModule,
                    message: string.Format("Intento {0}/{1}: No se pudo sincronizar el turno con el store. Caja:{2}, Tabla:{3}, Detalle:{4}",
                        intentos, reintentos, item.numeroCaja,
                        VENTURNOSCAJAREG.DatabaseMappingObject.TableName,
                        response.ErrorMessage));

                if (intentos < reintentos)
                {
                    System.Threading.Thread.Sleep(espera);
                }
            }
        }
        catch (Exception ex)
        {
            intentos++;
            FAhorro.POS.Application.Core.Log.WriteError(SubModule,
                message: string.Format("Error en intento {0}/{1} de sincronización: {2}",
                    intentos, reintentos, ex.Message));

            if (intentos < reintentos)
            {
                System.Threading.Thread.Sleep(espera);
            }
        }
    }
}
```

	Dclose generado con ventas pendientes		
Autor	Versión	Fecha de elaboración	Última edición
Eduardo Ochoa	1.3	07-04-2025	25-09-2025

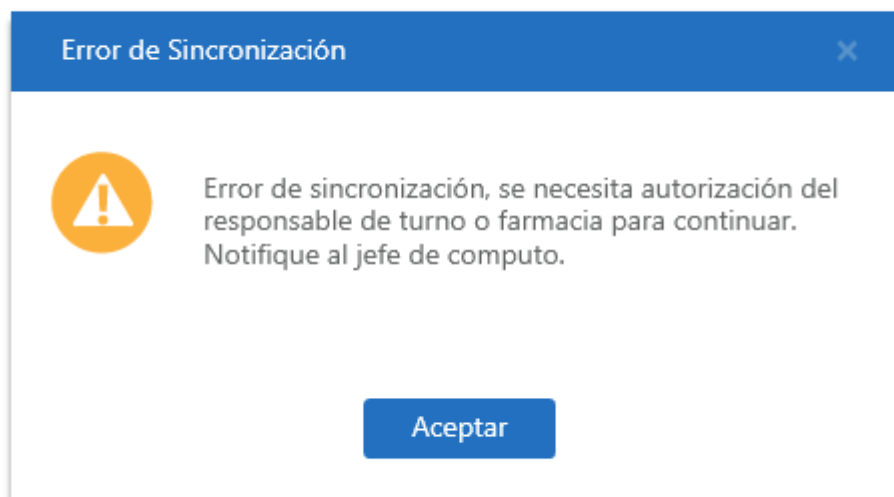
```
[2025-09-18 15:09:36:400], FARMAX_TILL_SERVER, TILL, WARNING, , Intento 1/4: No se pudo sincronizar el turno con el store. Caja:22, Tabla:venTurnosCajaReg, Detalle:La base de datos no esta disponible.
[2025-09-18 15:09:36:457], FARMAX_TILL_SERVER, TILL, WARNING, , Intento 2/4: No se pudo sincronizar el turno con el store. Caja:22, Tabla:venTurnosCajaReg, Detalle:La base de datos no esta disponible.
[2025-09-18 15:10:16:519], FARMAX_TILL_SERVER, TILL, WARNING, , Intento 3/4: No se pudo sincronizar el turno con el store. Caja:22, Tabla:venTurnosCajaReg, Detalle:La base de datos no esta disponible.
[2025-09-18 15:10:36:589], FARMAX_TILL_SERVER, TILL, WARNING, , Intento 4/4: No se pudo sincronizar el turno con el store. Caja:22, Tabla:venTurnosCajaReg, Detalle:La base de datos no esta disponible.
[2025-09-18 15:10:36:589], FARMAX_TILL_SERVER, TILL, WARNING, , Sincronización fallida. Turno guardado localmente requiere autorización. Caja: 22, Turno: 69ef5ec1-d394-f011-a293-e08f4cc457ec
```


-Si, después de todos los intentos, la sincronización no se logra, el sistema retorna un error SYNC_AUTH_REQUIRED con el TURNO_ID

```
if (!syncExitosa)
{
    FAhorro.POS.Application.Core.Log.WriteWarning(SubModule,
        message: string.Format("Sincronización fallida. Turno guardado localmente requiere autorización. Caja: {0}, Turno: {1}",
            item.numeroCaja, item.codigoTurno));

    var result = new ResultStruct<Guid>(item.codigoTurno);
    result.AddErrorItem(new ErrorItem
    {
        Code = "SYNC_AUTH_REQUIRED"
    });
    return result;
}
```

- Se muestra un aviso de que no logro sincronizar la caja con el store:



	Dclose generado con ventas pendientes		
Autor	Versión	Fecha de elaboración	Última edición
Eduardo Ochoa	1.3	07-04-2025	25-09-2025

Se implementó un nuevo permiso (ID 1108) asociado a la función 1102 del módulo Sale para "Autorización para continuar operación sin sincronización de turno". Este permiso se asignó a los roles de Responsable de Farmacia (RF) y Responsable de Turno (RT), permitiendo que usuarios con estos perfiles puedan autorizar la continuación de operaciones cuando falle la sincronización.

El código valida si existe un error de sincronización (SYNC_AUTH_REQUIRED) y, en caso afirmativo, solicita autorización del usuario mediante `AuthorizationManager.RequestAuthorization()` con el permiso `AuthorizationForTurnSyncFailure = "1108"`. Si el usuario no tiene los permisos necesarios, se registra en log y se muestra una notificación de error.

```

if (response?.ControlTender?.ARTSHeader?.Response?.BusinessError != null)
{
    var syncError = response.ControlTender.ARTSHeader.Response.BusinessError // ResponseCommonDataBusinessError[]
        .FirstOrDefault(e :ResponseCommonDataBusinessError => e.Code == "SYNC_AUTH_REQUIRED");


    if (syncError != null)
    {
        NotificationManager infoNotification = new NotificationManager();
        infoNotification.ShowMessage(
            title: "Error de Sincronización",
            message: "Error de sincronización, se necesita autorización del responsable de turno o farmacia para continuar. Notifique al jefe de computo.",
            image: MessageIcon.Warning
        );

        var authorizedUser object = AuthorizationManager.AuthorizationService.RequestAuthorization(
            permissionKey: AuthorizationManager.AuthorizationForTurnSyncFailure, msg: null);
        if (authorizedUser is FAhorro.POS.Win.Security.User userAuthorizer)
        {
            Log.WriteInfo($"Usuario {userAuthorizer.Id} ({userAuthorizer.UserId}) autorizó continuar sin sincronización de turno.");


            var businessErrors :List<ResponseCommonDataBusinessError> = response.ControlTender.ARTSHeader.Response.BusinessError.ToList();
            response.ControlTender.ARTSHeader.Response.BusinessError = businessErrors // List<ResponseCommonDataBusinessError>
                .Where(e :ResponseCommonDataBusinessError => e.Code != "SYNC_AUTH_REQUIRED") // IEnumerable<ResponseCommonDataBusinessError>
                .ToArray(); // ResponseCommonDataBusinessError[]
        }
    }
}
else

```

- Al cerrar el mensaje de advertencia o aceptarlo se solicita autorización del responsable de turno.

	<h2 style="text-align: center;">Dclose generado con ventas pendientes</h2>		
Autor	Versión	Fecha de elaboración	Última edición
Eduardo Ochoa	1.3	07-04-2025	25-09-2025

Autorización
✕



Usuario

Contraseña

Aceptar

- Si el RT o RF autoriza continuar sin comunicación con el store, se desplegará una ventana con fondo rojo y un mensaje de advertencia indicando que el sistema está offline:



Versión: 10.5.80510.0
CAJA: 22
03:00 PM 18/09/2025
🔍

Bienvenido a farmacias del Ahorro

Te Atiende CAJERO 2


Servicio al Cliente

PENDIENTES: 0
POR DESPACHAR: 0
POR ENTREGAR: 0
POR DEVOLVER: 0

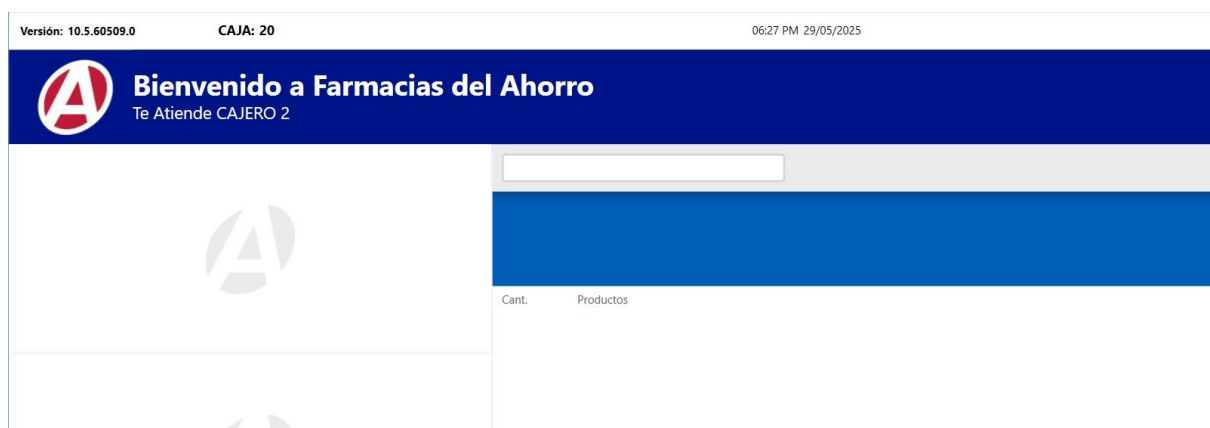




\$0.00 ➔

Cant.	Productos	Precio Máximo	Precio Ahorro

	Dclose generado con ventas pendientes		
Autor	Versión	Fecha de elaboración	Última edición
Eduardo Ochoa	1.3	07-04-2025	25-09-2025

Una vez se restablece la comunicación con el store el fondo rojo en el número de caja desaparecerá:



	<h2 style="text-align: center;">Dclose generado con ventas pendientes</h2>		
Autor	Versión	Fecha de elaboración	Última edición
Eduardo Ochoa	1.2	07-04-2025	03-06-2025

Estado de las tablas de caja y tienda antes de sincronizar:

```

select
codigoTurno,fechaHoraInicioTurno, fechaHoraFinalTurno,cajaDisponible, reutilizarTurno, estadoActual, syncCreadoFecha, syncActualizadoFecha
from FARMAX_CAJA.[dbo].venTurnosCajaReg order by syncActualizadoFecha desc
select
codigoTurno,fechaHoraInicioTurno, fechaHoraFinalTurno,cajaDisponible, reutilizarTurno, estadoActual, syncCreadoFecha, syncActualizadoFecha
from [FARMAX_localT].[dbo].venTurnosCajaReg order by syncActualizadoFecha desc

```

	codigoTurno	fechaHoraInicioTurno	fechaHoraFinalTurno	cajaDisponi...	reutilizarTu...	estadoAct...	syncCreadoFecha	syncActualizadoFecha
1	60A6D2B9-EF10-F011-A23F-E08F4CC457EC	2025-04-03 18:56:24.903	2025-04-08 09:04:25.303	1	0	C	2025-04-03 18:57:05.927	2025-04-08 09:04:25.307
2	237E34B0-4B0F-F011-A23F-E08F4CC457EC	2025-04-01 16:50:21.143	2025-04-03 18:50:20.783	1	2	C	2025-04-01 16:50:21.167	2025-04-03 18:50:20.790
3	813178D8-100F-F011-A23E-E08F4CC457EC	2025-04-01 09:49:08.387	2025-04-01 16:45:15.350	1	4	C	2025-04-01 09:49:08.410	2025-04-01 16:45:15.360
4	BA2EE4C3-950E-F011-A23E-E08F4CC457EC	2025-03-31 19:08:05.670	2025-04-01 09:42:46.603	1	1	C	2025-03-31 19:08:05.790	2025-04-01 09:42:46.607
5	818D8DA2-640E-F011-A23E-E08F4CC457EC	2025-03-31 13:16:24.483	2025-03-31 16:44:47.177	1	2	C	2025-03-31 13:16:24.517	2025-03-31 16:44:47.180
	codigoTurno	fechaHoraInicioTurno	fechaHoraFinalTurno	cajaDisponi...	reutilizarTu...	estadoAct...	syncCreadoFecha	syncActualizadoFecha
1	60A6D2B9-EF10-F011-A23F-E08F4CC457EC	2025-04-03 18:56:24.903	2025-04-08 09:04:25.303	1	0	C	2025-04-03 18:57:05.927	2025-04-08 09:04:25.307
2	237E34B0-4B0F-F011-A23F-E08F4CC457EC	2025-04-01 16:50:21.143	2025-04-03 18:50:20.783	1	2	C	2025-04-01 16:50:21.167	2025-04-03 18:50:20.790
3	39484A95-5008-F011-A23D-E08F4CC457EC	2025-03-23 19:37:45.097	2025-04-01 16:49:37.717	1	1	C	2025-03-23 19:37:45.357	2025-04-01 16:49:37.717
4	F53344EA-B4D7-ED11-92A8-862E52002284	2023-04-10 08:32:34.993	2025-04-01 16:49:13.467	1	1	C	2023-04-10 10:33:27.977	2025-04-01 16:49:13.470
5	813178D8-100F-F011-A23E-E08F4CC457EC	2025-04-01 09:49:08.387	2025-04-01 16:45:15.350	1	4	C	2025-04-01 09:49:08.410	2025-04-01 16:45:15.360


Después de agotar los intentos de sincronización pasa a solicitar la autorización a continuación se muestra el estado de las tablas antes de hacer un rollback por falta de autorización del responsable de turno:

```

select
codigoTurno,fechaHoraInicioTurno, fechaHoraFinalTurno,cajaDisponible, reutilizarTurno, estadoActual, syncCreadoFecha, syncActualizadoFecha
from FARMAX_CAJA.[dbo].venTurnosCajaReg order by syncActualizadoFecha desc
select
codigoTurno,fechaHoraInicioTurno, fechaHoraFinalTurno,cajaDisponible, reutilizarTurno, estadoActual, syncCreadoFecha, syncActualizadoFecha
from [FARMAX_localT].[dbo].venTurnosCajaReg order by syncActualizadoFecha desc

```

	codigoTurno	fechaHoraInicioTurno	fechaHoraFinalTurno	cajaDisponi...	reutilizarTu...	estadoAct...	syncCreadoFecha	syncActualizadoFecha
1	5B09571C-2516-F011-A245-E08F4CC457EC	2025-04-10 10:01:34.897	NULL	0	0	A	2025-04-10 10:01:50.353	2025-04-10 10:01:50.353
2	60A6D2B9-EF10-F011-A23F-E08F4CC457EC	2025-04-03 18:56:24.903	2025-04-08 09:04:25.303	1	0	C	2025-04-03 18:57:05.927	2025-04-08 09:04:25.307
3	237E34B0-4B0F-F011-A23F-E08F4CC457EC	2025-04-01 16:50:21.143	2025-04-03 18:50:20.783	1	2	C	2025-04-01 16:50:21.167	2025-04-03 18:50:20.790
4	813178D8-100F-F011-A23E-E08F4CC457EC	2025-04-01 09:49:08.387	2025-04-01 16:45:15.350	1	4	C	2025-04-01 09:49:08.410	2025-04-01 16:45:15.360
5	BA2EE4C3-950E-F011-A23E-E08F4CC457EC	2025-03-31 19:08:05.670	2025-04-01 09:42:46.603	1	1	C	2025-03-31 19:08:05.790	2025-04-01 09:42:46.607
	codigoTurno	fechaHoraInicioTurno	fechaHoraFinalTurno	cajaDisponi...	reutilizarTu...	estadoAct...	syncCreadoFecha	syncActualizadoFecha
1	60A6D2B9-EF10-F011-A23F-E08F4CC457EC	2025-04-03 18:56:24.903	2025-04-08 09:04:25.303	1	0	C	2025-04-03 18:57:05.927	2025-04-08 09:04:25.307
2	237E34B0-4B0F-F011-A23F-E08F4CC457EC	2025-04-01 16:50:21.143	2025-04-03 18:50:20.783	1	2	C	2025-04-01 16:50:21.167	2025-04-03 18:50:20.790
3	39484A95-5008-F011-A23D-E08F4CC457EC	2025-03-23 19:37:45.097	2025-04-01 16:49:37.717	1	1	C	2025-03-23 19:37:45.357	2025-04-01 16:49:37.717
4	F53344EA-B4D7-ED11-92A8-862E52002284	2023-04-10 08:32:34.993	2025-04-01 16:49:13.467	1	1	C	2023-04-10 10:33:27.977	2025-04-01 16:49:13.470
5	813178D8-100F-F011-A23E-E08F4CC457EC	2025-04-01 09:49:08.387	2025-04-01 16:45:15.350	1	4	C	2025-04-01 09:49:08.410	2025-04-01 16:45:15.360

	<h1 style="text-align: center;">Dclose generado con ventas pendientes</h1>		
Autor	Versión	Fecha de elaboración	Última edición
Eduardo Ochoa	1.2	07-04-2025	03-06-2025

Si no se cuenta con ella se procede a eliminar el turno para no dejarlo abierto:

```

select
    codigoTurno,fechaHoraInicioTurno, fechaHoraFinalTurno,cajaDisponible, reutilizarTurno, estadoActual, syncCreadoFecha, syncActualizadoFecha
    from [FARMAX_CAJA].[dbo].venTurnosCajaReg order by syncActualizadoFecha desc
select
    codigoTurno,fechaHoraInicioTurno, fechaHoraFinalTurno,cajaDisponible, reutilizarTurno, estadoActual, syncCreadoFecha, syncActualizadoFecha
    from [FARMAX_localIT].[dbo].venTurnosCajaReg order by syncActualizadoFecha desc

```

100%

Results Messages

	codigoTurno	fechaHoraInicioTurno	fechaHoraFinalTurno	cajaDisponi...	reutilizarTu...	estadoAct...	syncCreadoFecha	syncActualizadoFecha
1	60A6D2B9-EF10-F011-A23F-E08F4CC457EC	2025-04-03 18:56:24.903	2025-04-08 09:04:25.303	1	0	C	2025-04-03 18:57:05.927	2025-04-08 09:04:25.307
2	237E34B0-4B0F-F011-A23F-E08F4CC457EC	2025-04-01 16:50:21.143	2025-04-03 18:50:20.783	1	2	C	2025-04-01 16:50:21.167	2025-04-03 18:50:20.790
3	813178D8-100F-F011-A23E-E08F4CC457EC	2025-04-01 09:49:08.387	2025-04-01 16:45:15.350	1	4	C	2025-04-01 09:49:08.410	2025-04-01 16:45:15.360
4	BA2EE4C3-950E-F011-A23E-E08F4CC457EC	2025-03-31 19:08:05.670	2025-04-01 09:42:46.603	1	1	C	2025-03-31 19:08:05.790	2025-04-01 09:42:46.607
5	818D8DA2-640E-F011-A23E-E08F4CC457EC	2025-03-31 13:16:24.483	2025-03-31 16:44:47.177	1	2	C	2025-03-31 13:16:24.517	2025-03-31 16:44:47.180
	codigoTurno	fechaHoraInicioTurno	fechaHoraFinalTurno	cajaDisponi...	reutilizarTu...	estadoAct...	syncCreadoFecha	syncActualizadoFecha
1	60A6D2B9-EF10-F011-A23F-E08F4CC457EC	2025-04-03 18:56:24.903	2025-04-08 09:04:25.303	1	0	C	2025-04-03 18:57:05.927	2025-04-08 09:04:25.307
2	237E34B0-4B0F-F011-A23F-E08F4CC457EC	2025-04-01 16:50:21.143	2025-04-03 18:50:20.783	1	2	C	2025-04-01 16:50:21.167	2025-04-03 18:50:20.790
3	39484A95-5008-F011-A23D-E08F4CC457EC	2025-03-23 19:37:45.097	2025-04-01 16:49:37.717	1	1	C	2025-03-23 19:37:45.357	2025-04-01 16:49:37.717
4	F53344EA-B4D7-ED11-92A8-862E52002284	2023-04-10 08:32:34.993	2025-04-01 16:49:13.467	1	1	C	2023-04-10 10:33:27.977	2025-04-01 16:49:13.470
5	813178D8-100F-F011-A23E-E08F4CC457EC	2025-04-01 09:49:08.387	2025-04-01 16:45:15.350	1	4	C	2025-04-01 09:49:08.410	2025-04-01 16:45:15.360

Estado de tablas sincronizadas de forma inmediata en caja y store al abrir el turno:


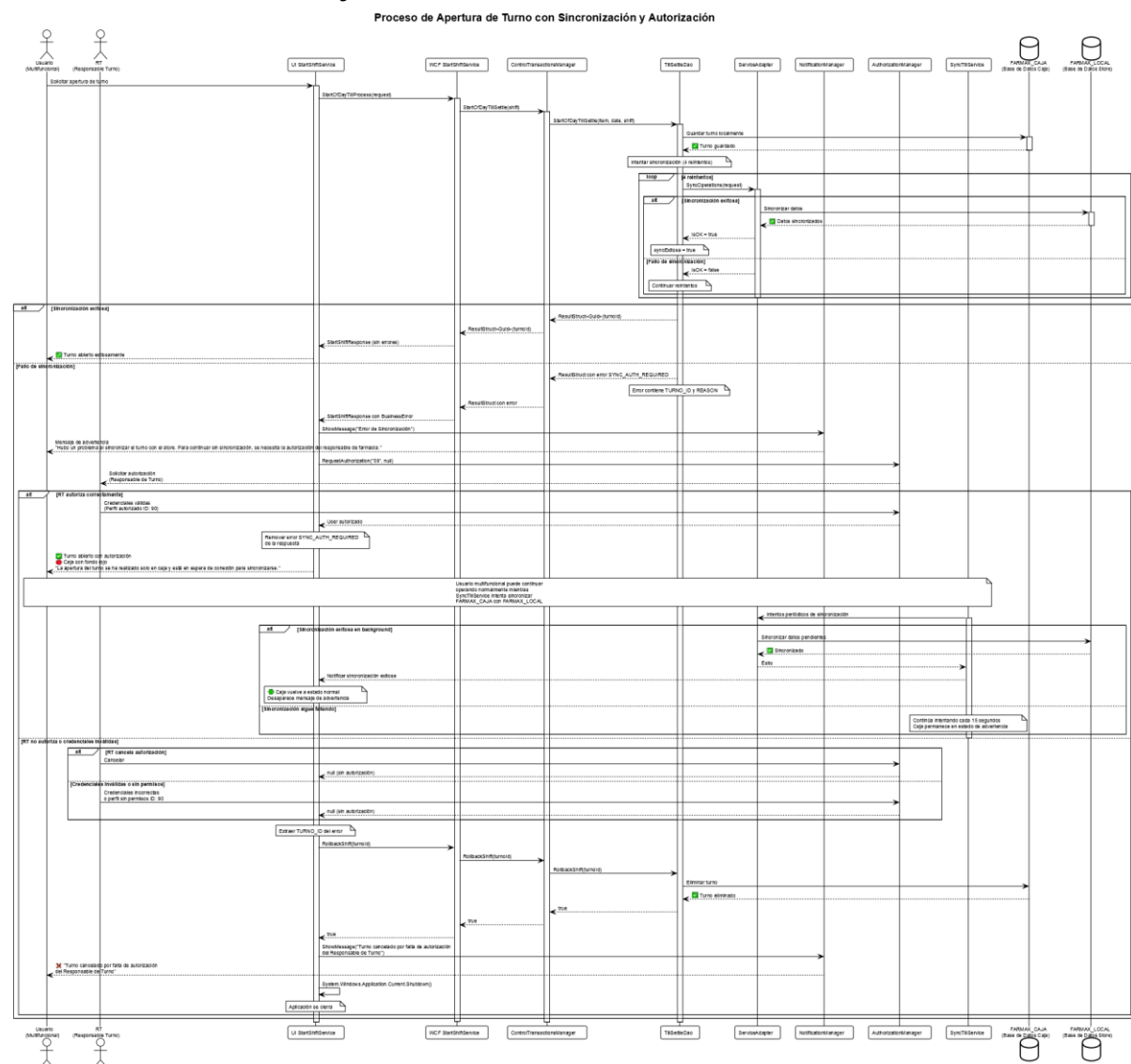

	<h1 style="text-align: center;">Dclose generado con ventas pendientes</h1>		
Autor	Versión	Fecha de elaboración	Última edición
Eduardo Ochoa	1.2	07-04-2025	03-06-2025

Diagrama de proceso de apertura con sincronización y autorización



[Ver diagrama completo](#)

	Dclose generado con ventas pendientes		
Autor	Versión	Fecha de elaboración	Última edición
Eduardo Ochoa	1.2	07-04-2025	03-06-2025

Implementación de la validación de comunicación en el DClose.

Para garantizar que no se realice un dclose sin contemplar cajas en funcionamiento con transacciones del día pero no sincronizadas se implementó una validación adicional en el método CloseWorkingDay(). Para su funcionamiento se agregaron métodos nuevos a continuación explicados:

Se agrego el método UpdateCommunicationStatus en la clase SyncStoreOp para registrar la comunicación exitosa entre la caja y el store con una frecuencia de ejecución de 60 segundos

```

/// <summary>
/// Actualiza el estado de comunicación de la caja
/// Registra la última vez que se comunicó
/// </summary>
/// <param name="storeId">ID de la sucursal</param>
/// <param name="tillId">ID de la caja</param>
/// <param name="tools">Herramientas de base de datos</param>

public static void UpdateCommunicationStatus(Int32 storeId, Int32 tillId, DatabaseTools tools) {
    GENSYNCSEGUIMIENTO segdb = tools.ExecuteReportObject(new GENSYNCSEGUIMIENTO())
    {
        CODIGOSUCURSAL = storeId,
        NUMEROCAJA = tillId
    });


    DateTime ahora = DateTime.Now;

    if (segdb == null)
    {
        segdb = new
GENSYNCSEGUIMIENTO();
segdb.CODIGOSUCURSAL = storeId;
segdb.NUMEROCAJA = tillId;
segdb.ESTADOSISTEMA = String.Empty;
segdb.ESTADOCATALOGOS = String.Empty;
segdb.SYNC_TRANSFERIDO = "N";
        segdb.ULTIMA_COMUNICACION_FECHA = ahora;
    }
    else
    {
        segdb.ULTIMA_COMUNICACION_FECHA = ahora;
segdb.SYNC_TRANSFERIDO = "N";
    }

    Session s = tools.CreateSession(SUBMODULE);

    if (segdb.InDatabase)
    {

```

	Dclose generado con ventas pendientes		
Autor	Versión	Fecha de elaboración	Última edición
Eduardo Ochoa	1.2	07-04-2025	03-06-2025

```

        s.UpdateObjectChanges(segdb);
    }
    else
    {
        s.InsertObjectWithID(segdb);
    }
}

try
{
    s.Execute();
}
catch (Exception ex)
{
    Logger.AddErrorSubModule(SUBMODULE, AxNLS.Data(1308390750, 2284, "Excepción al
    actualizar el estado de comunicación. Excepción:{0}", ex.Message));
}

```


Para lo cual se necesito agregar dos nuevas columnas **cajaActiva** (la cual contiene el valor "S" si la caja esta activa) y **ultimaComunicacionFecha** (la cual contiene la ultima fecha con la cual la caja tuvo comunicación con la store que se actualiza con el método UpdateCommunicationStatus) en la tabla **genSyncSeguimiento** para poder tener un control de las cajas de una store.

```

, [numeroCaja]
, [estadoSistema]
, [estadoCatalogos]
, [syncCreadoFecha]
, [syncActualizadoFecha]
, [syncActualizadoPor]
, [syncVersionRegistro]
, [syncTransferido]
, [cajaActiva]
, [ultimaComunicacionFecha]
FROM [FARMAX_local].[dbo].[genSyncSeguimiento]

```

	codigoSucur...	numeroC...	estadoSiste...	estadoCatalo...	syncCreadoFecha	syncActualizadoFecha	syncActualizadoPor	syncVersionRegi...	syncTransfer...	cajaActi...	ultimaComunicacionFecha
1	1553	10	DISPONIBLE	DISPONIBLE	2022-05-05 17:49:07.813	2025-06-02 16:34:20.047	FARMAX SYNC.UPDATE	7	I	NULL	NULL
2	1825	0	DISPONIBLE	DISPONIBLE	2025-04-25 16:04:54.320	2025-06-02 17:39:43.967	FARMAX SYNC.UPDATE	10275	I	NULL	NULL
3	1825	10	DISPONIBLE	DISPONIBLE	2025-04-25 16:04:54.320	2025-06-02 16:34:20.047	FARMAX SYNC.UPDATE	7	I	NULL	NULL
4	1825	20	DISPONIBLE	DISPONIBLE	2025-04-25 16:04:54.320	2025-06-02 17:40:12.530	FARMAX SYNC.TRIGGER	4361	N	S	2025-06-02 16:29:39.383
5	1825	21	DISPONIBLE	DISPONIBLE	2025-06-02 17:37:15.667	2025-06-02 17:37:15.667	FARMAX SYNC.UPDATE	3932	I	S	2025-06-01 16:29:39.383
6	1825	22	DISPONIBLE	DISPONIBLE	2025-06-02 17:37:15.667	2025-06-02 17:37:15.667	FARMAX SYNC.UPDATE	2666	I	N	2025-06-01 16:29:39.383
7	1825	23	DISPONIBLE	DISPONIBLE	2025-03-31 10:09:40.390	2025-04-19 06:58:57.960	FARMAX SYNC.UPDATE	89	I	NULL	NULL

	Dclose generado con ventas pendientes		
Autor	Versión	Fecha de elaboración	Última edición
Eduardo Ochoa	1.2	07-04-2025	03-06-2025

Nueva validación en CloseWorkingDay

Se agregó una validación que verifica si las cajas marcadas como activas se han comunicado en el día actual, si tienen más de un día sin sincronizar y están activas se bloquea el dclose, si no están activas no se toman en cuenta.

```

try
{
    IList<GENSYNCSEGUIMIENTO> tillsTracking = StoreDataCenter.StoreTools.ExecuteQueryObjects<GENSYNCSEGUIMIENTO>("
SELECT * FROM GENSYNCSEGUIMIENTO
WHERE CODIGOSUCURSAL = @PARAM1
AND cajaActiva = 'S'
AND (ultimaComunicacionFecha IS NULL OR CAST(ultimaComunicacionFecha AS DATE) < CAST(GETDATE() AS DATE))
", StoreConfig.StoreNumber);

    if (tillsTracking != null && tillsTracking.Count > 0)
    {
        tillsValidForClosing = false;

        foreach (GENSYNCSEGUIMIENTO till in tillsTracking)
        {
            if (!String.IsNullOrEmpty(tillsWithoutCommunication))
            {
                tillsWithoutCommunication += ", ";
            }
            tillsWithoutCommunication += till.NUMEROCAJA.ToString();
        }
    }
}
catch (Exception ex)
{
    Logger.AddErrorSubModule(ClosingDayService.SubModule, message: AxNLS.Data(assemblycode: 1427185856, code: 2200, message: "Error al verificar comunicación de cajas.
");

    if (!tillsValidForClosing)
    {
        rc.Set( @errorCode: 2201, message: $"El día de operación no puede ser cerrado porque las siguientes cajas activas no han comunicado hoy: {tillsWithoutCommunication}");
        rc.LogSubModule(ClosingDayService.SubModule);
        return rc;
    }
}

```


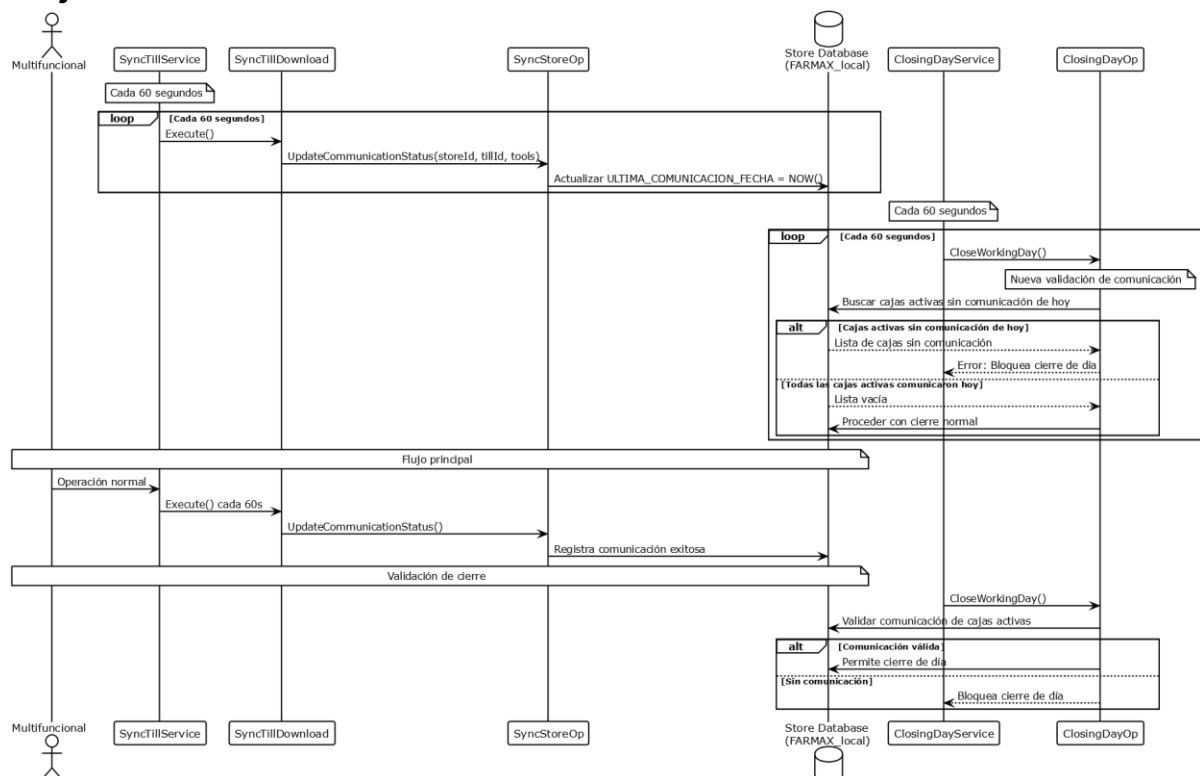
	Dclose generado con ventas pendientes		
Autor	Versión	Fecha de elaboración	Última edición
Eduardo Ochoa	1.2	07-04-2025	03-06-2025

Diagrama del proceso del dclose y validación de cajas activas



Conclusión

La implementación de la validación de comunicación de cajas en el proceso de cierre de día, junto con la sincronización inmediata en la apertura de turnos y autorización reduce significativamente el riesgo de errores en cierres de día.

Estas mejoras aseguran que:

- Los datos entre las cajas y el store estén sincronizados
- Se identifiquen rápidamente las cajas con problemas de conexión
- Las operaciones puedan continuar aunque haya fallas temporales de red
- El sistema sea más estable y confiable