

Ticket 99999999999 Datalogic

Autor: Huilver Nolasco Aguilar

Fecha: Noviembre del 2024

Contenido

Descripción general.....	3
Implementación y pruebas.....	4
TimeOut.....	4
Pruebas.....	7
Caso TIMEOUT por datalogic para todas las operaciones.....	8
Caso TIMEOUT por datalogic solo para operación Execute.....	14
Manejo de logs funcionales.....	19
Pruebas.....	20
Caso error por bloqueo en BD central.....	21
Cola de reversos centrales.....	27
Pruebas.....	28
TimeOut para operaciones Execute, Status y Reverse de broker.....	28
Manejo de no disponibilidad.....	29
Pruebas.....	29
Caso, operación EXECUTE de broker no disponible.....	29
HappyPath.....	33

Descripción general

Actualmente existe un incidente con el manejo de los tickets con autorización 0, en particular con aquellos atendidos por Datalogic. La causas son varias entre las que se encuentran:

- El desfase del tiempo de timeout que se presenta entre el POS, Central y broker de servicios
- Manejo incorrecto de logs como parte del proceso funcional en las que por intermitencia de BD ocasionan errores que no tienen que ver con el broker
- Falta de manejo de reversos interno ante varios escenarios por respuestas no manejadas.
- Evitar llamadas innecesarias ante caída del broker

Por lo anterior se ha se realizaron una serie de ajustes que permitan solventar dichas deficiencias:

- Aumentar en en 133 segundos la espera del cliente en el pos, este cálculo, es el mayor de los tiempos en que puede tardar una transacción, y corresponde a 20 segs de revisar el estatus de la operación, 45 segs en mandar invocar la transacción, 20 segs en caso de que la invocación se presentará un error, 45 segs del mandar llamar al reverso por no saber el estado de la operación y un tiempo 3 segs de holgura para las demás operaciones, dando la suma de $20+45+20+45+3=133$ segs
- Para mitigar el error que se produce debido a que se consultan o registran logs en bd y esta última puede no contestar por algún error, se agrego el manejo de errores en DAOS incluyendo una cola de solicitudes de logs no procesados

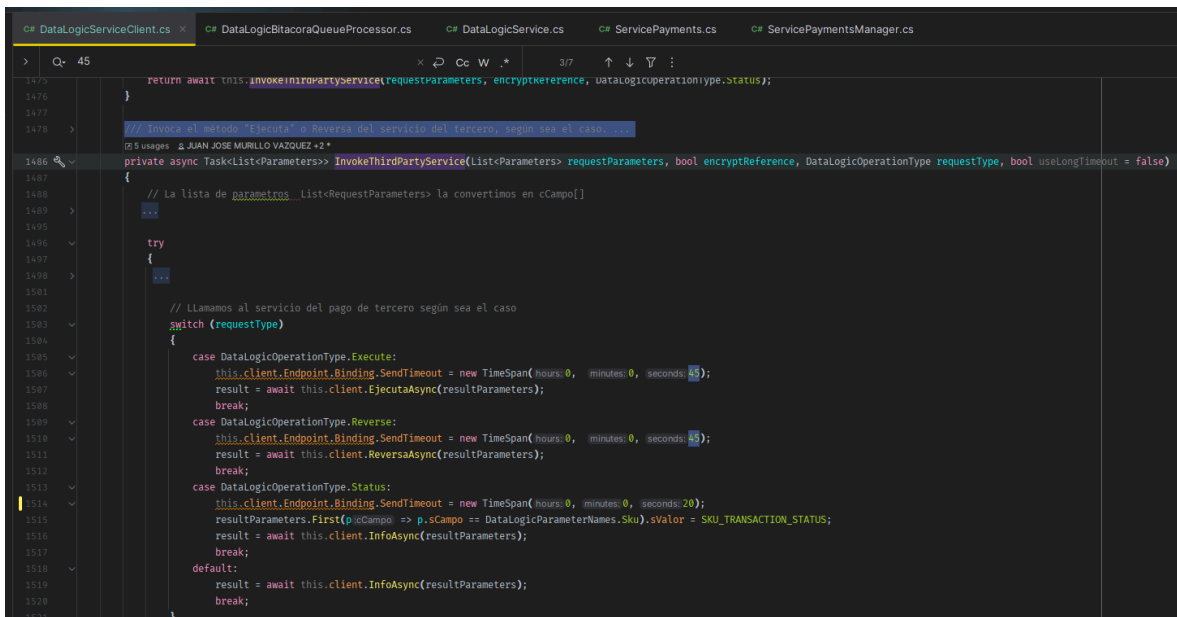
- Se incorpora una cola de reversos a nivel central para aquellos reversos que no tuvieron respuesta del integrador ya sea por falta de disponibilidad del broker, una respuesta desconocida o incluso timeout del broker.
- Se determina el tipo de excepción recibida y al invocar al integrador y si se determina que el error es por `EndpointNotFoundException`, no se intenta realizar ninguna operación debido a que el servidor no está respondiendo

Implementación y pruebas

Timeout

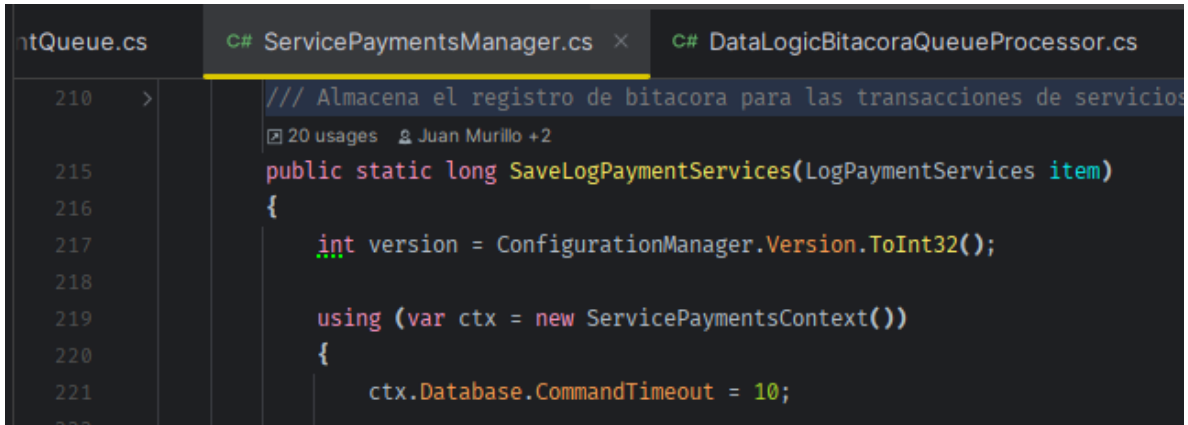
Los timeouts que intervienen son a nivel central y a nivel POS. A nivel central se debe ajustar en el cliente de datalogic el tiempo de **sendTimeout**, dado que actualmente está con tiempo 1 min 45 seg y se debe colocar a 45 seg para las operaciones de *execute* y *cancel*, pero para *status* se debe dejar en 20.

A nivel central los timeout quedan de la siguiente forma:



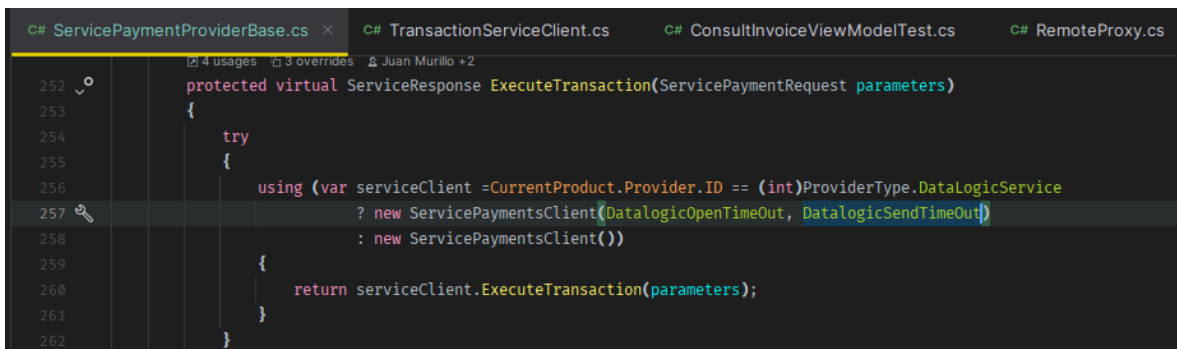
```
1475 return await this.InvokeThirdPartyService(requestParameters, encryptReference, dataLogicOperationType.status);
1476 }
1477
1478 // Invoca el método "Ejecuta" o "Reversa" del servicio del tercero, según sea el caso...
1479 // 5 usages JUAN JOSE MURILLO VAZQUEZ +2
1480 private async Task<List<Parameters>> InvokeThirdPartyService(List<Parameters> requestParameters, bool encryptReference, DataLogicOperationType requestType, bool useLongTimeout = false)
1481 {
1482     // La lista de parametros List<RequestParameters> la convertimos en cCampo[]
1483     ...
1484     try
1485     {
1486         // Llamamos al servicio del pago de tercero según sea el caso
1487         switch (requestType)
1488         {
1489             case DataLogicOperationType.Execute:
1490                 this.client.Endpoint.Binding.SendTimeout = new TimeSpan(hours:0, minutes:0, seconds:45);
1491                 result = await this.client.EjecutaAsync(resultParameters);
1492                 break;
1493             case DataLogicOperationType.Reverse:
1494                 this.client.Endpoint.Binding.SendTimeout = new TimeSpan(hours:0, minutes:0, seconds:45);
1495                 result = await this.client.ReversaAsync(resultParameters);
1496                 break;
1497             case DataLogicOperationType.Status:
1498                 this.client.Endpoint.Binding.SendTimeout = new TimeSpan(hours:0, minutes:0, seconds:20);
1499                 resultParameters.First(p => p.sCampo == DataLogicParameterNames.Sku).sValor = SKU_TRANSACTION_STATUS;
1500                 result = await this.client.InfoAsync(resultParameters);
1501                 break;
1502             default:
1503                 result = await this.client.InfoAsync(resultParameters);
1504                 break;
1505         }
1506     }
1507 }
```

Los timeouts para acceso a la bitácora se establecieron en 10 Segs debido a que por defecto son 30 segs, que para este tipo de operación es un tiempo excesivo.



```
ntQueue.cs  C# ServicePaymentsManager.cs x  C# DataLogicBitacoraQueueProcessor.cs
210 >      /// Almacena el registro de bitacora para las transacciones de servicios
211      [20 usages] [Juan Murillo +2]
215      public static long SaveLogPaymentServices(LogPaymentServices item)
216      {
217          int version = ConfigurationManager.Version.ToInt32();
218
219          using (var ctx = new ServicePaymentsContext())
220          {
221              ctx.Database.CommandTimeout = 10;
222          }
223      }
```

A nivel POS los timeouts quedaron establecidos en **220** segs esto debido a que se debe colocar el mayor tiempo posible en que una operación puede tardar con sus respectivos timeouts a nivel central.



```
C# ServicePaymentProviderBase.cs x  C# TransactionServiceClient.cs  C# ConsultInvoiceViewModelTest.cs  C# RemoteProxy.cs
252 ✓ [4 usages] [3 overrides] [Juan Murillo +2]
253      protected virtual ServiceResponse ExecuteTransaction(ServicePaymentRequest parameters)
254      {
255          try
256          {
257              using (var serviceClient = CurrentProduct.Provider.ID == (int)ProviderType.DataLogicService
258                  ? new ServicePaymentsClient(DatalogicOpenTimeOut, DatalogicSendTimeOut)
259                  : new ServicePaymentsClient())
260              {
261                  return serviceClient.ExecuteTransaction(parameters);
262              }
263          }
264          catch { }
265      }
```

```

c# ServicePaymentProviderBase.cs x c# ServicePaymentsClient.cs c# ApplyPaymentResult.cs c# PaymentMethodViewModel.cs c# TenderManager.cs c# TenderProviderFa
1 // .....
6
7 using Newtonsoft.Json;
8
9 namespace FAhorro.POS.Win.ServicePayments.Providers
10 {
11     using ...
12
13     /// Clase base para implementar la comunicación con proveedores de servicios. ...
14     [InheritedExport(typeof(ServicePaymentProviderBase))]
15     public abstract class ServicePaymentProviderBase
16     {
17         /// Tiempo en segs en el que se esperara para abrir el canal de comunicacion del servicio ...
18         private const int DatalogicOpenTimeout = 10;
19
20         /// <summary>
21         /// Tiempo en segs en el que se esperara una respuesta en el consumo del servicio
22         /// //20=consulta estatus + 45=execute + 20 = consulta estatus(error en execute) + 45=reverso + 90 = por timeouts de bd
23         /// </summary>
24         private const int DatalogicSendTimeout = 220;
25     }
26 }

```

Para entender este tiempo debemos describir lo que termina sucediendo en el peor de los casos.

Primero debemos tomar en cuenta el tiempo en que puede tardar el broker de servicios, así como del tiempo en que la bd puede tardar al intentar consultar o guardar en bitácora. Los tiempos timeouts del brokers para execute y reverse son de 45 segs y el de consulta es de 20 segs; los timeouts de BD se establecieron en 10 segs. Así que el peor de los casos es el siguiente:

1. Manejo de casos previos con de un request recibido(30 segs= 3 accesos bd)
2. Guardar en bitácora la operación execute(10 segs = 1 acceso bd)
3. obtención del estatus de la operación(40 segs = 2 accesos bd y operación status con broker)
4. Ejecución del método execute(45 segs = operación execute del broker)
5. Si se detecta un error, se obtiene el estatus de la operación(40 segs = 2 accesos bd y operación status con broker)
6. Si no se puede obtener el estado de la operación se intenta un reverso(55 segs = 1 acceso bd y operación reverse del broker)

```

public async Task<ServicePaymentResponse> ProcessPaymentServiceStateless(ServicePaymentRequest request)
{
    //TOTAL TIMEOUTS: 3 accesos BD = 3 * 10 = 30Segs
    if(!ManagePreviousRequest(request, out ServicePaymentResponse servicePaymentResponse, out bool treatAsPending)) {...}
    Log.WriteInfo(request.TypeServicePayments.TypeService, message: $"{request.TypeServicePayments.FolioTransaction} treatAsPending={treatAsPending}");
    try
    {
        ...
        var idLog (long?) = LogRequestExecute(request, generalInformation); //TOTAL TIMEOUTS: 1 acceso BD = 1 * 10 = 10Segs
        try
        {
            if (treatAsPending){
                List<Parameters> statusResponse = await GetTransactionStatus(request, requestParameters); //TOTAL TIMEOUTS: 2 accesos BD + 1 consulta broker = (2*10) + (1*20) = 40Segs
                if (statusResponse == null){...}
                if (IsSuccessfulResponse(statusResponse)){...}
                else{
                    // no successful response: retry ejecuta
                    respuestaIntegrador = await InvokeExecuteTransactionLong(requestParameters, request); //TOTAL TIMEOUTS: 1 execute de broker = 1*45 = 45Segs
                }
            }
            else{...}
        }
        catch (Exception ex){
            //Log.WriteError(request.TypeServicePayments.TypeService, $"{request.TypeServicePayments.FolioTransaction} excepcion ejecuta. Causa:{ex}");
            WriteError(request.TypeServicePayments.TypeService, msg: $"{request.TypeServicePayments.FolioTransaction} excepcion ejecuta. ", ex);
            if (ContainsServiceException(ex, typeof(EndpointNotFoundException)) || ContainsServiceException(ex, typeof(TimeoutException))){
                if (ContainsServiceException(ex, typeof(EndpointNotFoundException))){treatAsPending}{...}
                List<Parameters> statusResponse = await GetTransactionStatus(request, requestParameters); //TOTAL TIMEOUTS: 3 accesos BD + 1 consulta broker = (2*10) + (1*20) = 40Segs
                if (statusResponse == null){
                    return await TryReverseIfApply(request); //TOTAL TIMEOUTS: 1 acceso BD + 1 reverso de broker = 10 + 45 = 55 Segs
                }
            }
        }
    }
}

```

Si se hace la suma de todos estos tiempos se llega a los **220 Segs** antes de intentar realizar la operación de nueva cuenta desde el POS.

Pruebas

Se establecen tres escenarios posibles:

1. el primero es que datalogic se tarde el tiempo máximo establecido antes de que el cliente corte la comunicación, pero la bd responda en tiempo y forma,
2. el segundo es que la bd consuma todo el tiempo posible pero que datalogic responda en tiempo y forma
3. el tercero es que tanto la bd como el datalogic den timeout.

Caso TIMEOUT por datalogic para todas las operaciones

Para lograr este escenario de manera controlada, se genera un proxy hacia datalogic, que permite generar timeouts controlados de manera programática y controlado por configuración.

```

processor.cs  C# DataLogicServiceClientHelper.cs  C# PxUniversalSoapClientFake.cs  C# StringBuilder.cs  C# Exception.cs  C# ServicePayments
26
27 7 usages
28 private PxUniversalSoapClient client { get; }
29
30 1 usage 2 hultver
31 public PxUniversalSoapClientFake(string endpointConfigurationName)
32 {
33     client = new PxUniversalSoapClient(endpointConfigurationName);
34 }
35
36 0+1 usages 2 hultver
37 public async Task<ccampo[]> EjecutaAsync(ccampo[] cArrayCampos)
38 {
39     try{
40         return await Task.Run(async () =>
41         {
42             ...
43             int appliedDelay = await ManageWaitEjecuta(maxtries);
44             token.ThrowIfCancellationRequested();
45             ccampo[] result = null;
46             try{
47                 result = client.EjecutaAsync(cArrayCampos).Result;
48                 token.ThrowIfCancellationRequested();
49             }
50             finally{...}
51         }
52         return result;
53     }, token);
54 }
55
56 catch (OperationCanceledException oce){
57     throw new TimeoutException($"The request was aborted: The request was canceled. After {cancelTimeOut}ms",oce);
58 }
59
60 }

```



```

Web.config x C# POSServicePaymentsModel.Context.cs C# CancellationToken.cs C# PxUnive
403
404
405 <!--=====DATALOGIC TESTING=====-->
406 <!--Para controlar los timeouts de prueba de datalogic-->
407 <!-- <add key="TimeOutExecute" value="44600" /> --><!--valor para timeout-->
408 <add key="TimeOutExecute" value="10" />
409 <add key="TriesToCompleteExecute" value="2" />
410 <!--<add key="EndpointNotFoundExceptionExecute" value="true" />-->
411 <add key="EndpointNotFoundExceptionExecute" value="false" /><!--Obliga a un error de no
412
413
414 <!--<add key="TimeOutStatus" value="19980" />--><!--valor para timeout-->
415 <add key="TimeOutStatus" value="10" />
416 <add key="TriesToCompleteStatus" value="1" />
417 <add key="EndpointNotFoundExceptionStatus" value="false" />
418
419 <!--<add key="TimeOutReverse" value="44600" />--><!--valor para timeout-->
420 <add key="TimeOutReverse" value="10" />
421 <!--<add key="TriesToCompleteReverse" value="2" />--><!--en conjunti con timeout obliga
422 <add key="TriesToCompleteReverse" value="1" />
423 <add key="EndpointNotFoundExceptionReverse" value="false" />
424
425
426 <add key="TimeOutInfo" value="10" />
427 <add key="TriesToCompleteInfo" value="1" />
428 <add key="EndpointNotFoundExceptionInfo" value="false" />
429 <add key="DatalogicTesting" value="true" />
430 <!--end timeouts datalogic-->
431 <!--=====-->

```

```

C# DataLogicServiceClientHelper.cs x C# PxUniversalSoapClientFake.cs C# StringBuilder.cs C# Exception.cs C# ServicePayments
396 }
397
398 2 usages hultver
398 internal static IPxUniversalSoapClientWrapper BuildPxUniversalSoapClientWrapper(string endpointConfigurationName)
399 {
400     bool.TryParse(ConfigurationManager.AppSettings["DatalogicTesting"], out bool datalogicTesting);
401     IPxUniversalSoapClientWrapper clientWrapper;
402     if (datalogicTesting){
403         clientWrapper = new PxUniversalSoapClientFake(endpointConfigurationName);
404     }
405     else{
406         clientWrapper = new PxUniversalSoapClientProxy(endpointConfigurationName);
407     }
408     return clientWrapper;
409 }

```

```
<!--=====DATALOGIC TESTING=====-->

<add key="TimeOutExecute" value="44600" />
<add key="TriesToCompleteExecute" value="2" />
<add key="EndpointNotFoundExceptionExecute" value="false" />

<add key="TimeOutStatus" value="19980" />
<add key="TriesToCompleteStatus" value="1" />
<add key="EndpointNotFoundExceptionHandlerStatus" value="false" />


<add key="TimeOutReverse" value="44600" />
<add key="TriesToCompleteReverse" value="2" />
<add key="EndpointNotFoundExceptionHandlerReverse" value="false" />

<!--=====-->
```

Se configura el proxy para que nos genere timeouts y despues de los intentos configurados nos responderá de manera normal.

Ejecutamos un servicio de datalogic desde el POS:

10
05:30 PM 25/11/2024


Bienvenido a Farmacias del Ahorro
Te Atiende FABIOLA

Servicio a Domicilio
PENDIENTES: 73
POR DESPACHAR: 18
POR ENTREGAR: 36
POR DEVOLVER: 0

Pago de Servicios

SERVICIOS

> SERVICIOS

> VARIOS

> ENTRETENIMIENTO

> SERVICIOS VARIOS

> TRANSPORTE

> TIEMPO AIRE

> SERVICIOS FINANCIEROS

> PAGO DE SERVICIOS MP

> VENTAS POR CATALOGO

> PAGO DE SERVICIOS

> COMPRAS POR INTERNET

> LUZ

RECARGA UNEFON \$ 20

USACELL
SIN MARCA
MOVISTAR
UNEFON
VIRION
USACELUATAT

Monto
RECARGA UNEFON \$ 20

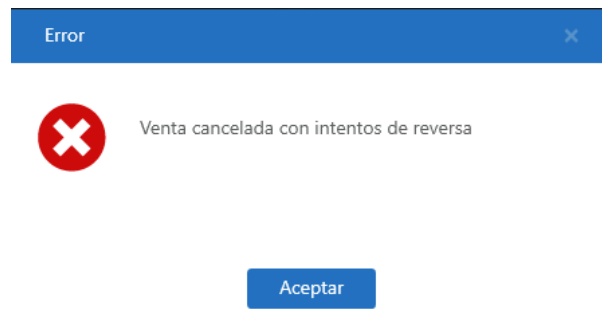
Número
5555555555

Confirmar
5555555555

Formas de Pago

E	EFFECTIVO (CA)	20.00
T	TARJETA BANCARIA	0.00

Subtotal: \$17.24	I.V.A: \$2.76	Total: \$20.00
Tu Donativo \$0.00	Tu Ahorro \$0.00	Por Pagar \$0.00



El resultado esperado en este escenario es que la venta termine cancelada e independientemente de estado de la transacción en Datalogic, se aplique la reversa incluso si en ese momento no pueda aplicarse, se debe encolar la reversa en el central y además decirle al POS que también termine mandando la reversa para asegurar que no existan transacciones canceladas que Datalogic las tenga como aplicadas.

Para sustentar esto se revisa el log del POS y central:

El siguiente es el log del POS, se ve que se ejecuta la transacción de la venta V15531000002993, el resultado de dicha transacción es el código 1002, que significa que hubo un error en reversa, dado ese error, el POS encola la reversa para su posterior ejecución y por último se dispara en POS el evento de cancelación de venta.

```
[2024-11-25 17:32:58:186], FARMAX_POS, DATALOGICSERVICE, INFO, [LOG], ##### ExecuteServiceTransaction V15531000002993 request={{"Type": "ServicePayment", "Brand": "UNEFON", "TypeService": "DataLogicService", "BusinessUnit": 1553, "UserCode": 3, "EncryptReference": true, "FolioTransaction": "V15531000002993", "OperationDate": "2023-04-13T00:00:00", "TransactionDate": "2024-11-25T17:32:57", "ShiftId": "45a04ffa-84ab-ef11-9692-94bb432cf0d5", "SupportsReverse": true, "PrintsTicketNoAuth": true}, {"ProductCode": 758778193, "RelatedCode": "8578690100208", "ExternalCode": "N/A", "ProductServiceCode": 758778193, "TillId": 10, "BusinessUnitId": 1553, "MacAddress": "94-BB-43-2C-F0-D4", "IPAddress": "192.168.1.6", "OperationDate": "2023-04-13T00:00:00", "ListParameters": {"Parameters": [{"Label": "COMISION", "Type": 3, "InputType": 1, "Length": 8, "Action": 1, "Value": 0.00, "Field": "ComissionAmount", "FieldClass": 1}, {"Label": "REFERENCIA", "Type": 1, "InputType": 1, "Length": 10, "Action": 1, "Value": "5555555555", "Encrypted": true, "Field": "Reference", "FieldClass": 1}, {"Label": "MONTO", "Type": 3, "InputType": 1, "Length": 8, "Action": 1, "Value": 20.0, "Field": "Amount", "FieldClass": 1}]}}
[2024-11-25 17:34:49:789], FARMAX_POS, DATALOGICSERVICE, INFO, [LOG], ##### ExecuteServiceTransaction V15531000002993 result={"DateCancel": "", "DateTransaction": "", "Folio": "", "CodeError": "1002", "DescriptionError": "Venta cancelada con intentos de reversa", "Store": "", "ServicePaid": "", "TicketMessage": []}
[2024-11-25 17:34:49:804], FARMAX_POS, DATALOGICSERVICE, INFO, [LOG], ##### QueueReverse V15531000002993 queueing reverse, response == null->False response.IsErrorOrCodeWaitingOrUnknown()->False !printsTicketNoAuth False response?.CodeError->1002
[2024-11-25 17:35:22:768], FARMAX_POS, DATALOGICSERVICE, INFO, [LOG], ##### ExecuteServiceTransaction V15531000002993 action CancelSale
```

Revisamos los datos de la venta y se observa que tanto el cabecero como el detalle del movimiento se encuentran como inactivo y cancelado respectivamente.

```
select idTransaccion, folioTransaccion, subTotal, codigoMovimientoCaja, fechaTransaccion, estaActivo from trnTransaccionesCab with (nolock) where folioTransaccion = 'V15531000002993';
idTransaccion  folioTransaccion  subTotal  codigoMovimientoCaja  fechaTransaccion  estaActivo
DC39AC5C-85A8-EF11-9692-94BB432CF0D5  V15531000002993  17.2413  64829999-85A8-EF11-9692-94BB432CF0D5  2024-11-25 17:35:22.817  false

select codigoMovimientoCaja, importePago, estatusFormaPago from genMovimientosCajasDet with (nolock)
where codigoMovimientoCaja in (select codigoMovimientoCaja from trnTransaccionesCab with (nolock) where folioTransaccion = 'V15531000002993');
codigoMovimientoCaja  importePago  estatusFormaPago
64829999-85A8-EF11-9692-94BB432CF0D5  20.0000  C
```

Además se revisa que se haya encolada la reversa:

```
select syncActualizadoFecha, codigoSucursal, numeroCaja, numeroIntentos, tipoOperacion, objetoSerializado from opColaOperaciones order by syncActualizadoFecha desc;
syncActualizadoFecha  codigoSucursal  numeroCaja  numeroIntentos  tipoOperacion  objetoSerializado
2024-11-25 17:34:50.027  1553  10  0  11  /EncryptReference><FolioTransaction>V15531000002993;/FolioTransaction>
```

Se revisan los logs del central, en el se puede observar que no se encontró ninguna solicitud pendiente, despues se *ejecuta* la solicitud al broker , sin embargo despues de 45 segs la solicitud fue cancelada por timeout(justo lo esperado), despues se procede a *consultar* el estado de la operación para saber en qué estado se encuentra la operación pero la solicitud es cancelada por timeout despues de 20 segs, dada la situación anterior, se procede a intentar el *reverso* pero la solicitud se cancela por timeout despues de 45 segs, como no se logró realizar la reversa, se *encola* la reversa en la cola de reversos centrales como consecuencia se responde al POS el código 1002 que significa que el POS debe *encolar* el reverso para asegurar que se aplique si o si.

```
[*] <> tail -f -n 200 20241125_FARMAX_WSC_DATALOGICSERVICE.XLOG | egrep "\w{4}-\w{2}-\w{2} \w{2}:\w{2}:\w{2}:\w{3}|V[0-9]+|\\"
[2024-11-25 17:32:58:281],FARMAX_WSC_DATALOGICSERVICE,INFO,,V15531000002993 treatAsPending=False
[2024-11-25 17:32:58:420],FARMAX_WSC_DATALOGICSERVICE,INFO,,V15531000002993 (FAhorro.POS.Services.DataLogic.helper.PdUniversalSoap
ClientFake) ejecuta...
[2024-11-25 17:33:43:950],FARMAX_WSC_DATALOGICSERVICE,ERROR,[WRITEERROR],(thread=14) V15531000002993 excepcion ejecuta. - System.
TimeoutException: The request was aborted: The request was canceled. After 45000ms
[2024-11-25 17:33:43:952],FARMAX_WSC_DATALOGICSERVICE,INFO,,V15531000002993 consulta...
[2024-11-25 17:34:04:282],FARMAX_WSC_DATALOGICSERVICE,ERROR,[WRITEERROR],(thread=15) V15531000002993 excepci3n consulta. - System.
TimeoutException: The request was aborted: The request was canceled. After 20000ms
[2024-11-25 17:34:04:302],FARMAX_WSC_DATALOGICSERVICE,INFO,,V15531000002993 consulta... result es null
[2024-11-25 17:34:04:303],FARMAX_WSC_DATALOGICSERVICE,INFO,,V15531000002993 reversa...
[2024-11-25 17:34:04:305],FARMAX_WSC_DATALOGICSERVICE,INFO,,#####Start -> CancelPaymentService(V15531000002993,reasonCode=TimeOut
Emisor)
[2024-11-25 17:34:49:729],FARMAX_WSC_DATALOGICSERVICE,ERROR,[WRITEERROR],##### CancelPaymentService V15531000002993. Error:The re
quest was aborted: The request was canceled. After 45000ms
[2024-11-25 17:34:49:756],FARMAX_WSC_DATALOGICSERVICE,WARNING,,#####End CancelPaymentService -> (V15531000002993,isReverseSuccess
ful=False,errorCode=8)
[2024-11-25 17:34:49:756],FARMAX_WSC_DATALOGICSERVICE,ERROR,[WRITEERROR],V15531000002993 Reversa no exitosa, se encolara reverso d
ebido a CodeError: 8
[2024-11-25 17:34:49:774],FARMAX_WSC_DATALOGICSERVICE,INFO,,##### DoTransactionProcessing V15531000002993 result={"PaymentError"
:{"CodeError":"1002","DescriptionError":"Venta cancelada con intentos de reversa"}}
```

Se revisa el log de las colas de reversos en el central y por lo que se puede observar es que el reverso pudo aplicarse solo hasta el intento 2 que es como se tenía prevista la prueba

```
[*] <> tail -f -n 200 20241125_FARMAX_WSC_DATALOGIC_REVERSE_PROCESSOR.XLOG | egrep "\w{4}-\w{2}-\w{2} \w{2}:\w{2}:\w{2}:\w{3}|V[
0-9]+|\\"
[2024-11-25 17:35:08:987],FARMAX_WSC_DATALOGIC_REVERSE_PROCESSOR,INFO,,#### procesando '1' reversos
[2024-11-25 17:35:08:999],FARMAX_WSC_DATALOGIC_REVERSE_PROCESSOR,INFO,,#### procesando reverso de 'V15531000002993', intento '1'
[2024-11-25 17:35:09:001],FARMAX_WSC_DATALOGIC_REVERSE_PROCESSOR,INFO,,#####Start -> CancelPaymentService(V15531000002993,reasonC
ode=TimeOutEmisor)
[2024-11-25 17:35:54:373],FARMAX_WSC_DATALOGIC_REVERSE_PROCESSOR,ERROR,[WRITEERROR],##### CancelPaymentService V15531000002993. E
rror:The request was aborted: The request was canceled. After 45000ms
[2024-11-25 17:35:54:377],FARMAX_WSC_DATALOGIC_REVERSE_PROCESSOR,WARNING,,#####End CancelPaymentService -> (V15531000002993,isRev
erseSuccessful=False,errorCode=8)
[2024-11-25 17:35:54:377],FARMAX_WSC_DATALOGIC_REVERSE_PROCESSOR,WARNING,,Reverso no exitoso de la transaccion 'V15531000002993'.
El integrador respondio con el codigo: '8'
[2024-11-25 17:35:54:377],FARMAX_WSC_DATALOGIC_REVERSE_PROCESSOR,WARNING,,V15531000002993'. Se reencola el reverso debido a que f
allo el intento
[2024-11-25 17:36:09:001],FARMAX_WSC_DATALOGIC_REVERSE_PROCESSOR,INFO,,#### procesando '1' reversos
[2024-11-25 17:36:09:002],FARMAX_WSC_DATALOGIC_REVERSE_PROCESSOR,INFO,,#### procesando reverso de 'V15531000002993', intento '2'
[2024-11-25 17:36:09:002],FARMAX_WSC_DATALOGIC_REVERSE_PROCESSOR,INFO,,#####Start -> CancelPaymentService(V15531000002993,reasonC
ode=TimeOutEmisor)
[2024-11-25 17:36:09:772],FARMAX_WSC_DATALOGIC_REVERSE_PROCESSOR,INFO,,#####End CancelPaymentService -> (V15531000002993,isRevers
eSuccessful=True')
```

Por último revisamos la bitácora de servicios en donde se observa que prácticamente todas las operaciones se marcaron con estatus pendiente o con error, salvo el último intento de reverso que se registró como completada y la respuesta fue exitosa

```
select idBitacoraServicios,folioTransaccion,numeroAutorizacion,fechaSolicitud,fechaRespuesta,tipoOperacion,status
from venPagoServiciosBitacoraReg order by idBitacoraServicios desc
```

idBitacoraServicios	folioTransaccion	numeroAutorizacion	fechaSolicitud	fechaRespuesta	tipoOperacion	status
117	V15531000002993	enull	2024-11-25 17:36:09.003	2024-11-25 17:36:09.773	R	C
116	V15531000002993	enull	2024-11-25 17:35:09.003	enull	R	P
115	V15531000002993	enull	2024-11-25 17:34:04.313	enull	R	P
114	V15531000002993	enull	2024-11-25 17:33:43.953	2024-11-25 17:34:04.283	I	CE
113	V15531000002993	enull	2024-11-25 17:32:58.280	enull	I	P

3	117	V15531000002993	<null>	2024-11-25 17:36:09.003	2024-11-25 17:36:09.773	R	C	"Unspecified" Encrypted="false">transaccion Exitosa</Param></ResponseParameters>
---	-----	-----------------	--------	-------------------------	-------------------------	---	---	--

Caso TIMEOUT por datalogic solo para operación Execute

Para este caso es necesario configurar el proxy para que el timeout se presente solo en el caso de la operación Execute con el broker y que la operación Status conteste de manera correcta. Para esto usaremos la siguiente configuración del proxy.

```
<!--=====DATALOGIC TESTING=====-->
<!--Para controlar los timeouts de prueba de datalogic-->
<add key="TimeOutExecute" value="44600" />
<add key="TriesToCompleteExecute" value="2" />
<add key="EndpointNotFoundExceptionExecute" value="false" />

<add key="TimeOutStatus" value="10" />
<add key="TriesToCompleteStatus" value="1" />
<add key="EndpointNotFoundExceptionStatus" value="false" />

<add key="TimeOutReverse" value="10" />
<add key="TriesToCompleteReverse" value="2" />
<add key="EndpointNotFoundExceptionReverse" value="false" />

<add key="TimeOutInfo" value="10" />
<add key="TriesToCompleteInfo" value="1" />
<add key="EndpointNotFoundExceptionInfo" value="false" />
<add key="DatalogicTesting" value="true" />
<!--end timeouts datalogic-->
<!--=====-->
```

Realizamos una transacción de datalogic en el POS.

PENDIENTES: 73
 POR DESPACHAR: 18
 POR ENTREGAR: 36
 POR DEVOLVER: 0

Pago de Servicios

- > SERVICIOS
- > VARIOS
- > ENTRETENIMIENTO
- > SERVICIOS VARIOS
- > TRANSPORTE
- > TIEMPO AIRE
- > SERVICIOS FINANCIEROS
- > PAGO DE SERVICIOS MP
- > VENTAS POR CATALOGO
- > PAGO DE SERVICIOS
- > COMPRAS POR INTERNET
- > LUZ
- > TELECOMUNICACIONES
- > EDITORIAL
- > DEPOSITO EN EFECTIVO
- > HOGAR
- > CREDITO
- > ALARMAS
- > RECIBO DE CELULAR

←

RECARGA UNEFON \$ 20

IUSACELL
 SIN MARCA
 MOVISTAR
UNEFON
 VIRGIN
 IUSACELL/AT&T

Monto

RECARGA UNEFON \$ 20

Número

5555555555

Confirmar

5555555555

Formas de Pago

E	EFFECTIVO (CA)	20.00
T	TARJETA BANCARIA	0.00

Procesando

Transacción en proceso. Favor de esperar.

Tu Cambio
\$0.00

Gracias por tu preferencia


Farmacias del Ahorro

COMERCIALIZADORA FARMACEUTICA DE CHIAPAS SAPI DE CV
 TLAXCALA, 151 PISO 4 OFNA 407 HIPODROMO, CUAUHTEMOC
 CP.06100 RFC: CFC110121742
 F A TIJUANA, PLUTARCO ELIAS CALLES, 11 PLUTARCO
 ELIAS CALLES, 9026
 ZONA ESTE, TIJUANA, TIJUANA, BAJA CALIFORNIA
 CP.22000
 Fecha OP: 13/04/2023
 V15531000002994 25/11/2024 22:26
 Multifuncional: FABIOLA VALERIO MORENO
 Le Atendió: FABIOLA VALERIO MORENO
 RECARGA UNEFON \$ 20 \$17.24
 IVA: \$2.76
 TOTAL: \$20.00
 PAGO: \$20.00
 CAMBIO: \$0.00
 VEINTE PESOS 00/100 M.N.
 FORMA DE PAGO:
 EFECTIVO (CA): \$20.00
 REFERENCIA: 5555555555
 AUTORIZACION: 102245
 TRANSACCION: 1553102994
 FECHA: 25/11/2024 10:26:31 p. m.
 TRANSACCION: 1553102994
 Dudas tiempo aire marca *050 o 018003330050
 ITU: 73b736da-adab-ef11-9692-94bb432cf0d5

V15531000002994

Lo que se espera en este caso es que, a pesar de que el broker de un Timeout en la operación Execute, al consultar el estatus de la operación hacia el broker este último responda lo ocurrido con la operación, en esta caso particular responde con un folio de transacción de éxito

Corroboramos lo anterior revisando los logs de POS, en donde se ve que se realizó la transacción, y esta última recibe un folio de transacción, lo cual indica que la operación fue exitosa y la acción realizada por el POS es completar la venta.

```
[2024-11-25 22:25:49:076],FARMAX_POS,DATALOGICSERVICE,INFO,[LOG],##### ExecuteServiceTransaction V15531000002994 request={"TypeS
ervicePayments":{"Brand":"UNEFON","TypeService":"DataLogicService","BusinessUnit":1553,"UserCode":3,"EncryptReference":true,"Folio
Transaction":"V15531000002994","OperationDate":"2023-04-13T00:00:00","TransactionDate":"2024-11-25T22:25:48","ShiftId":"45a04ffa-8
4ab-ef11-9692-94bb432cf0d5","SupportsReverse":true,"PrintsTicketNoAuth":true,"ProductCode":758778193,"RelatedCode":"8578690100208
","ExternalCode":"N/A","ProductServiceCode":758778193,"TillId":10,"BusinessUnitId":1553,"MacAddress":"94-BB-43-2C-F0-D4","IPAdres
s":"192.168.1.6","OperationDate":"2023-04-13T00:00:00","ListParameters":{"Parameters":[{"Label":"COMISION","Type":3,"InputType":1,
"Length":8,"Action":1,"Value":0.00,"Field":"ComissionAmount","FieldClass":1},{Label":"REFERENCIA","Type":1,"InputType":1,"Length"
:10,"Action":1,"Value":"5555555555","Encrypted":true,"Field":"Reference","FieldClass":1},{Label":"MONTO","Type":3,"InputType":1,"
Length":8,"Action":1,"Value":20.0,"Field":"Amount","FieldClass":1}]}}
[2024-11-25 22:26:35:242],FARMAX_POS,DATALOGICSERVICE,INFO,[LOG],##### ExecuteServiceTransaction V15531000002994 result={"Amount
":20.00,"DateCancel":"","DateTransaction":"","Folio":"102245","Store":"","ServicePayed":"","TicketMessage":[{"Label":"Transacción
","Value":"1553102994","Type":2},{Label":"FECHA","Value":"25/11/2024 10:26:31 p. m.","Type":1},{Label":"TRANSACCION","Value":"155
3102994","Type":1},{Label":"","Value":"","Type":1},{Label":"","Value":"Dudas tiempo aire marca *050 o 018003330050","Type":1}]}
[2024-11-25 22:26:35:246],FARMAX_POS,DATALOGICSERVICE,INFO,[LOG],##### ExecuteServiceTransaction V15531000002994 action CompleteP
ayment
```

Se revisan los logs del central, en donde se observa que la operación Execute del broker, se cancela por Timeout, sin embargo al consultar el estatus de esta transacción ante ese fallo, se obtiene un folio de transacción, lo que indica que la operación se aplicó con éxito en el broker, por lo que se responde al POS dicho folio, terminando en una operación exitosa

```
[2024-11-25 22:25:49:219],FARMAX_WSC,DATALOGICSERVICE,INFO,,V15531000002994 treatAsPending=False
[2024-11-25 22:25:49:361],FARMAX_WSC,DATALOGICSERVICE,INFO,,V15531000002994 (FAhorro.POS.Services.DataLogic.helper.PdUniversalSoap
ClientFake) ejecuta...
[2024-11-25 22:26:34:914],FARMAX_WSC,DATALOGICSERVICE,ERROR,[WRITEERROR],(thread=188) V15531000002994 excepcion ejecuta. - System
.TimeoutException: The request was aborted: The request was canceled. After 45000ms
[2024-11-25 22:26:34:921],FARMAX_WSC,DATALOGICSERVICE,INFO,,V15531000002994 consulta...
[2024-11-25 22:26:35:221],FARMAX_WSC,DATALOGICSERVICE,INFO,,V15531000002994 consulta... result=[{"Label":"FECHA","Type":1,"InputTy
pe":1,"Length":25,"Action":1,"Value":"25/11/2024 10:26:31 p. m.", "Field":"FECHA","FieldClass":6}, {"Label":"TRANSAccion","Type":1,"
InputType":1,"Length":8,"Action":1,"Value":"1553102994","Field":"TRANSAccion","FieldClass":6}, {"Label":"REFERENCIA","Type":1,"InputT
ype":1,"Length":10,"Action":1,"Value":"5555555555","Field":"REFERENCIA","FieldClass":6}, {"Label":"MONTO","Type":3,"InputType":1,"L
ength":5,"Action":1,"Value":"20.00","Field":"MONTO","FieldClass":6}, {"Label":"COMISION","Type":3,"InputType":1,"Length":4,"Action":1
,"Value":"0.00","Field":"COMISION","FieldClass":6}, {"Label":"AUTORIZACION","Type":1,"InputType":1,"Length":8,"Action":1,"Value":"1022
45","Field":"AUTORIZACION","FieldClass":6}, {"Label":"LEYENDA2","Type":1,"InputType":1,"Length":0,"Action":1,"Value":"","Field":"LEY
ENDA2","FieldClass":7}, {"Label":"LEYENDA","Type":1,"InputType":1,"Length":43,"Action":1,"Value":"Dudas tiempo aire marca *050 o 01
8003330050","Field":"LEYENDA","FieldClass":7}]
[2024-11-25 22:26:35:228],FARMAX_WSC,DATALOGICSERVICE,INFO,,V15531000002994 ::: Datos de la transaccion exitosa datalogic IDGRUPO=
4|IDCADENA=1|IDTIENDA=1553|IDPOS=10|IDCAJERO=3|FECHALOCAL=25/11/2024|HORALOCAL=22:25:48|TRANSAccion=1553102994|SKU=8578690100208|C
OMISION=0.00|REFERENCIA=D8301C1D70FAF4E6CDB05|MONTO=20|FECHA=25/11/2024 10:26:31 p. m.|AUTORIZACION=102245|LEYENDA2=|LEYENDA=Duda
s tiempo aire marca *050 o 018003330050|
[2024-11-25 22:26:35:237],FARMAX_WSC,DATALOGICSERVICE,INFO,,##### DoTransactionProcessing V15531000002994 result={"Folio":"10224
5","Amount":20.00,"TicketMessage":[{"Label":"Transacción","Value":"1553102994","Type":2}, {"Label":"FECHA","Value":"25/11/2024 10:2
6:31 p. m.", "Type":1}, {"Label":"TRANSAccion","Value":"1553102994","Type":1}, {"Label":"","Value":"","Type":1}, {"Label":"","Value":"
Dudas tiempo aire marca *050 o 018003330050","Type":1}]}
```

Revisamos la bd para ver el estado de la transacción, en que se puede observar que la venta está activa y el pago aplicado:

```
select idTransaccion, folioTransaccion, subTotal, codigoMovimientoCaja, fechaTransaccion, estaActivo from trnTransaccionesCab with (nolock) where folioTransaccion = 'V15531000002994';
```

idTransaccion	folioTransaccion	subTotal	codigoMovimientoCaja	fechaTransaccion	estaActivo
738736DA-AEAB-EF11-9692-94B8A32CF805	V15531000002994	17.2413	217BC382-AEAB-EF11-9692-94B8A32CF805	2024-11-25 22:25:48.903	true

```
select codigoMovimientoCaja, importePago, estatusFormaPago from genMovimientosCajasDet with (nolock)
where codigoMovimientoCaja in (select codigoMovimientoCaja from trnTransaccionesCab with (nolock) where folioTransaccion = 'V15531000002994');
```

codigoMovimientoCaja	importePago	estatusFormaPago
217BC382-AEAB-EF11-9692-94B8A32CF805	20.0000	A

Se revisa la bitácora de operaciones

```
select idBitacoraServicios, folioTransaccion, numeroAutorizacion, fechaSolicitud, fechaRespuesta, tipoOperacion, status from venPagoServiciosBitacoraReg order by idBitacoraServicios desc
```

idBitacoraServicios	folioTransaccion	numeroAutorizacion	fechaSolicitud	fechaRespuesta	tipoOperacion	status
119	V15531000002994	102245	2024-11-25 22:26:34.917	2024-11-25 22:26:35.210	I	C
118	V15531000002994	102245	2024-11-25 22:25:49.223	2024-11-25 22:26:35.227	E	C

En el cual se ve una operación de Execute y despues una de consulta, la operación Execute se registró correctamente aunque haya dado error, debido a que los datos de respuesta se toman del resultado de la operación Estatus

Manejo de logs funcionales

Existe una bitácora en BD, en la que se guardan las solicitudes que se realizan hacia el integrador, una de sus funciones es la auditoría de lo sucedido en una transacción, sin embargo otra de sus funciones es que sirve como una suerte de caché de transacciones para evitar ir hacia el integrador y así no realizar solicitudes innecesarias. Sin embargo hay escenarios que por errores de BD en bitácora, se lanza de manera incorrecta un error aún y habiendo habido éxito con el integrador, teniendo una transacción incongruente, debido a esto ahora se debe manejar los errores de manejo de bitácora.

```
C# DataLogicServiceClient.cs x C# DataLogicBitacoraQueueProcessor.cs C# DataLogicService.cs C# ServicePayments.cs C# ServicePaymentsManager.cs

1613         return null;
1614     }
1615 }
1616
1617 > /// Registra consulta de status en el service log ...
1618 [3 usages] [hover +2]
1619 private long? LogRequestStatus(ServicePaymentRequest request, DataLogicServiceInformation informationLog)
1620 {
1621     if (GetfolioTransaccionFromRequest(request, out string folioTransaccion)){
1622         try{
1623             return LogInDataBase(request, informationLog, DataLogicOperationType.Status);
1624         }
1625         catch (Exception ex){
1626             Log.WriteException(request.TypeServicePayments.TypeService, ex,
1627                 methodNames:"#### Ocurrio un error al guardar en bitacora la operacion de 'Status' para el folio '{folioTransaccion}' ");
1628         }
1629     }
1630     return null;
1631 }
1632
1633
1634
1635
1636 }
```

```
C# DataLogicServiceClient.cs x C# DataLogicBitacoraQueueProcessor.cs C# DataLogicService.cs C# ServicePayments.cs C# ServicePaymentsManager.cs

> Q- 45 x Cc W .* 3/7 ↑ ↓ ▾
1595
1596 > /// Registra execute en el service log ...
1597 [3 usages] [hover +2]
1602 public long? LogRequestExecute(ServicePaymentRequest request, DataLogicServiceInformation informationLog)
1603 {
1604     if (GetfolioTransaccionFromRequest(request, out string folioTransaccion)){
1605         try{
1606             return LogInDataBase(request, informationLog, DataLogicOperationType.Execute);
1607         }
1608         catch (Exception ex){
1609             Log.WriteException(request.TypeServicePayments.TypeService, ex,
1610                 methodNames:"#### Ocurrio un error al guardar en bitacora la operacion de 'Execute' para el folio '{folioTransaccion}' ");
1611         }
1612     }
1613     return null;
1614 }
1615 }
```

```
C# DataLogicServiceClient.cs x C# DataLogicBitacoraQueueProcessor.cs C# DataLogicService.cs C# ServicePaymentsManager.cs
> Qc 45
420
421 > /// New logic for processing the service payments that does not keep transaction state in the central WS ...
422 [1 usage] holver +2
423 public async Task<ServicePaymentResponse> ProcessPaymentServiceStateless(ServicePaymentRequest request)
424 {
425     var response = new ServicePaymentResponse();
426
427     // Verificamos que todos los datos que serán enviados al servicio del tercero, tengan un valor
428     this.ValidateRequest(request);
429     ...
430
431     try
432     {
433         var requestParameters = PrepareExecuteRequestParameters(request);
434         List<Parameters> respuestaIntegrador = new List<Parameters>();
435
436         // Guardamos en bd la petición que se enviará al servicio.
437         var generalInformation = MapDataContractToLocalModel(request.TypeServicePayments, requestParameters);
438         var idLog = LogRequestExecute(request, generalInformation);
439
440         try{...}
441         catch (Exception ex){...}
442         // default to success
443         var responseCode = ObtainResponseCodeFromList(respuestaIntegrador, defaultValue: (int)DataLogicResponseCode.SuccessfulProcess);
444
445         if (responseCode == (int)DataLogicResponseCode.SuccessfulProcess){...}
446         else{...}
447
448         LogRequestExecuteForced(idLog, responseCode, request, generalInformation, respuestaIntegrador);
449     }
450 }
```

```
private void LogRequestExecuteForced(long? idLog, int responseCode, ServicePaymentRequest request, DataLogicServiceInformation generalInformation, List<Parameter> respuestaIntegrador)
{
    generalInformation.ErrorCode = responseCode;
    generalInformation.LogId = idLog;
    generalInformation.ResponseParameters = MapDataContractToLocalModel(request.TypeServicePayments, respuestaIntegrador).RequestParameters;
    long? idBitacora = LogRequestExecute(request, generalInformation);
    if (idBitacora == null)
    {
        _ = DataLogicQueueManager.GetInstance().AddElementInQueue(DataLogicBitacoraQueue, new DataLogicBitacoraQueueData(request, generalInformation), maxSizeQueue: 2000);
    }
}
```

20

Caso error por bloqueo en BD central

Para poder tener este caso es necesario provocar un bloqueo en la base de datos central para la bitácora de servicios

```
1 BEGIN TRAN
2 SELECT TOP (1) 1 FROM venPagoServiciosBitacoraReg WITH (TABLOCKX)
3 WAITFOR DELAY '00:02:00'
4 ROLLBACK TRAN
5 GO
6
7
```

Además del bloqueo es necesario configurar el proxy para que no simule ningún TIMEOUT

```
<!--=====DATALOGIC TESTING=====-->
<add key="TimeOutExecute" value="10" />
<add key="TriesToCompleteExecute" value="2" />
<add key="EndpointNotFoundExceptionExecute" value="false" />

<add key="TimeOutStatus" value="10" />
<add key="TriesToCompleteStatus" value="1" />
<add key="EndpointNotFoundExceptionStatus" value="false" />

<add key="TimeOutReverse" value="10" />
<add key="TriesToCompleteReverse" value="2" />
<add key="EndpointNotFoundExceptionReverse" value="false" />

<add key="TimeOutInfo" value="10" />
<add key="TriesToCompleteInfo" value="1" />
<add key="EndpointNotFoundExceptionInfo" value="false" />
<add key="DataLogicTesting" value="true" />
<!--end timeouts datalogic-->
<!--=====-->
```

Ejecutamos la transacción datalogic

←

RECARGA UNEFON \$ 20

IUSACELL

SIN MARCA

MOVISTAR

UNEFON

VIRGIN

IUSACELL/AT&T

Monto

RECARGA UNEFON \$ 20

Número

555555555

Confirmar

555555555

Formas de Pago

E EFECTIVO (CA)

20.00

T TARJETA BANCARIA

0.00

Subtotal: \$17.24

I.V.A: \$2.76

Total: \$20.00

Tu Donativo \$0.00

Tu Ahorro \$0.00

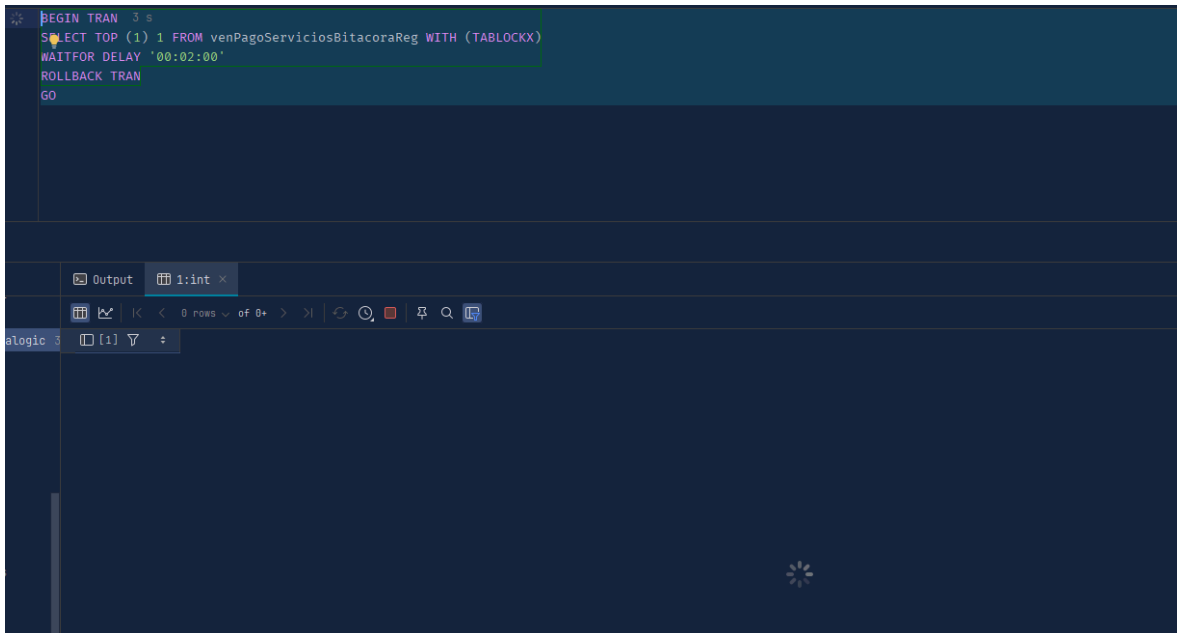
Por Pagar \$0.00

Antes de proceder a realizar la transacción, generamos un bloqueo manual

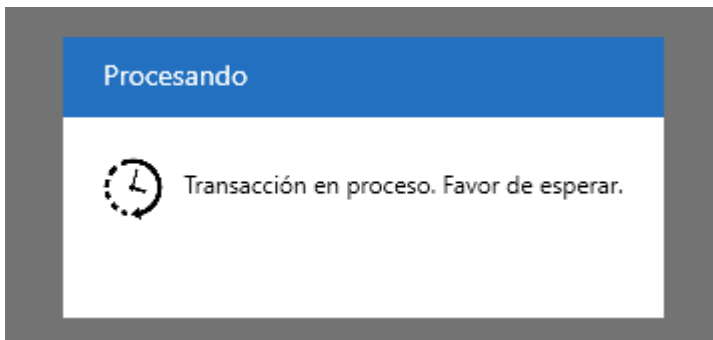
```

BEGIN TRAN 3 s
SELECT TOP (1) 1 FROM venPagoServiciosBitacoraReg WITH (TABLOCKX)
WAITFOR DELAY '00:02:00'
ROLLBACK TRAN
GO

```



Aplicamos la transacción, y se puede observar que se tiene una operación exitosa





COMERCIALIZADORA FARMACEUTICA DE CHIAPAS SAPI DE CV
 TLAXCALA, 151 PISO 4 OFNA 407 HIPODROMO, CUAUHTEMOC
 CP.06100 RFC: CFC110121742
 F A TIJUANA, PLUTARCO ELIAS CALLES, 11 PLUTARCO
 ELIAS CALLES, 9026
 ZONA ESTE, TIJUANA, TIJUANA, BAJA CALIFORNIA
 CP.22000

Fecha OP: 13/04/2023
 V15531000002995 25/11/2024 23:45
 Multifuncional: FABIOLA VALERIO MORENO
 Le Atendió: FABIOLA VALERIO MORENO

RECARGA UNEFON \$ 20	\$17.24
IVA:	\$2.76
TOTAL:	\$20.00
PAGO:	\$20.00
CAMBIO:	\$0.00

VEINTE PESOS 00/100 M.N.

FORMA DE PAGO:
 EFECTIVO (CA): \$20.00

REFERENCIA: 5555555555
 AUTORIZACIÓN: 102245
 TRANSACCIÓN: 1553102995
 UNEFON

Dudas tiempo aire marca *050 o 018003330050
 ITU: f5dfa45c-b9ab-ef11-9692-94bb432cf0d5


V15531000002995

Revisamos el log del POS, en el que se puede ver que se lanzó la transacción y se recibió un respuesta exitosa a pesar de los bloqueados en la bitácora de servicios

```
[2024-11-25 23:44:20:518],FARMAX_POS,DATALOGICSERVICE,INFO,[LOG],##### ExecuteServiceTransaction V15531000002995 request={"TypeS
ervicePayments":{"Brand":"UNEFON","TypeService":"DataLogicService","BusinessUnit":1553,"UserCode":3,"EncryptReference":true,"Folio
Transaction":"V15531000002995","OperationDate":"2023-04-13T00:00:00","TransactionDate":"2024-11-25T23:44:20","ShiftId":"45a04ffa-8
4ab-ef11-9692-94bb432cf0d5","SupportsReverse":true,"PrintsTicketNoAuth":true},"ProductCode":758778193,"RelatedCode":"8578690100208
","ExternalCode":"N/A","ProductServiceCode":758778193,"TillId":10,"BusinessUnitId":1553,"MacAddress":"94-BB-43-2C-F0-D4","IPAdres
s":"192.168.1.6","OperationDate":"2023-04-13T00:00:00","ListParameters":{"Parameters":[{"Label":"COMISION","Type":3,"InputType":1,
"Length":8,"Action":1,"Value":0.00,"Field":"ComissionAmount","FieldClass":1},{Label":"REFERENCIA","Type":1,"InputType":1,"Length"
:10,"Action":1,"Value":"5555555555","Encrypted":true,"Field":"Reference","FieldClass":1},{Label":"MONTO","Type":3,"InputType":1,"
Length":8,"Action":1,"Value":20.0,"Field":"Amount","FieldClass":1}]}}
[2024-11-25 23:45:32:293],FARMAX_POS,DATALOGICSERVICE,INFO,[LOG],##### ExecuteServiceTransaction V15531000002995 result={"Amount
":20.0,"DateCancel":"","DateTransaction":"","Folio":"102245","Store":"","ServicePayed":"","TicketMessage":{"Label":"Transacción",
"Value":"1553102995","Type":2},{Label":"","Value":"UNEFON","Type":1},{Label":"","Value":"Dudas tiempo aire marca *050 o 01800333
0050 ","Type":1}}
[2024-11-25 23:45:32:294],FARMAX_POS,DATALOGICSERVICE,INFO,[LOG],##### ExecuteServiceTransaction V15531000002995 action CompleteP
ayment
24
```


Tanto la venta como el pago se muestran como activo y aplicado respectivamente

```

select idTransaccion, folioTransaccion, subTotal, codigoMovimientoCaja, fechaTransaccion, estaActivo from trnTransaccionesCab with (nolock) where folioTransaccion = 'V15531000002995';

```

idTransaccion	folioTransaccion	subTotal	codigoMovimientoCaja	fechaTransaccion	estaActivo
F5DFA45C-B9A8-EF11-9692-948B432CF8D5	V15531000002995	17.2413	3B18FD7A-B9A8-EF11-9692-948B432CF8D5	2024-11-25 23:44:20.313	true


```

select codigoMovimientoCaja, importePago, estatusFormaPago from genMovimientosCajasDet with (nolock)
where codigoMovimientoCaja in (select codigoMovimientoCaja from trnTransaccionesCab with (nolock) where folioTransaccion = 'V15531000002995');

```

codigoMovimientoCaja	importePago	estatusFormaPago
3B18FD7A-B9A8-EF11-9692-948B432CF8D5	20.0000	A

Para entender lo que paso revisamos el log de los servicios centrales, donde se puede apreciar que al revisar si el request tiene una operación pendiente da un error que indica “Se perdió la conexión con la base de datos.”, debido a esto la solicitud se toma como pendiente por lo que se intenta realizar el guardado en bitácora de la operación **Execute**, pero no lo logra debido al bloqueo, despues intenta consultar el estatus de la solicitud con el broker, pero previo a eso intenta guardar en bitácora la operación de **Status**, pero no lo logra, con todo y eso se puede consultar al broker obteniendo como respuesta “La transaccion consultada no existe”, porlo que se procede a invocar la operación **Execute** teniendo un folio de autorización y por lo tanto una operacion exitosa, sin embargo no se logra persistir en bitácora de servicios, pero se maneja el error y responde al POS con Éxito debido a que eso fue lo que el broker respondió.

```
[2024-11-25 23:44:30:732],FARMAX_WSC,DATALOGICSERVICE,ERROR,[WRITEERROR],(thread=130) #####Ocurrio un error al revisar si el folio
'V15531000002995' tiene autorizacion previa. - FAhorro.POS.Application.Core.DbConnectionException: Se perdió la conexión con la
base de datos.
[2024-11-25 23:44:40:749],FARMAX_WSC,DATALOGICSERVICE,ERROR,[WRITEERROR],(thread=130) ##### Ocurrio un error al revisar si el foli
o 'V15531000002995' ha sido reversado previamente. - FAhorro.POS.Application.Core.DbConnectionException: Se perdió la conexión co
n la base de datos.
[2024-11-25 23:44:50:756],FARMAX_WSC,DATALOGICSERVICE,ERROR,[WRITEERROR],(thread=130) ##### Ocurrio un error al revisar si el foli
o 'V15531000002995' tiene una solicitud pendiente previa, con mas de 50 segs. - FAhorro.POS.Application.Core.DbConnectionExcepti
on: Se perdió la conexión con la base de datos.
[2024-11-25 23:44:50:757],FARMAX_WSC,DATALOGICSERVICE,INFO,,V15531000002995 treatAsPending=True
[2024-11-25 23:45:00:921],FARMAX_WSC,DATALOGICSERVICE,ERROR,[WRITEERROR],(thread=130) ##### Ocurrio un error al guardar en bitacor
a la operacion de 'Execute' para el folio 'V15531000002995' - FAhorro.POS.Application.Core.DbConnectionException: Se perdió la con
exión con la base de datos.
[2024-11-25 23:45:10:941],FARMAX_WSC,DATALOGICSERVICE,ERROR,[WRITEERROR],(thread=130) ##### Ocurrio un error al guardar en bitacor
a la opetacion de 'Status' para el folio 'V15531000002995' - FAhorro.POS.Application.Core.DbConnectionException: Se perdió la con
exión con la base de datos.
[2024-11-25 23:45:10:942],FARMAX_WSC,DATALOGICSERVICE,INFO,,V15531000002995 consulta...
[2024-11-25 23:45:21:460],FARMAX_WSC,DATALOGICSERVICE,ERROR,[WRITEERROR],(thread=40) ##### Ocurrio un error al guardar en bitacora
la opetacion de 'Status' para el folio 'V15531000002995' - FAhorro.POS.Application.Core.DbConnectionException: Se perdió la cone
xión con la base de datos.
[2024-11-25 23:45:21:468],FARMAX_WSC,DATALOGICSERVICE,INFO,,V15531000002995 consulta... result=[{"Label":"CODIGORESPUESTA","Type":
2,"InputType":1,"Length":4,"Action":1,"Value":38,"Field":"CODIGORESPUESTA","FieldClass":0}, {"Label":"CODIGORESPUESTADESCR","Type":
1,"InputType":1,"Length":36,"Action":1,"Value":"La transaccion consultada no existe ","Field":"CODIGORESPUESTADESCR","FieldClass":
0}]
[2024-11-25 23:45:21:469],FARMAX_WSC,DATALOGICSERVICE,INFO,,V15531000002995 (FAhorro.POS.Services.DataLogic.helper.PxUniversalSoap
ClientFake) ejecuta...
[2024-11-25 23:45:22:229],FARMAX_WSC,DATALOGICSERVICE,INFO,,V15531000002995 ::: Datos de la transaccion exitosa datalogic IDGRUPO=
4|IDCADENA=1|IDTIENDA=1553|IDPOS=10|IDCAJERO=3|FECHALOCAL=25/11/2024|HORALOCAL=23:44:20|TRANSACCION=1553102995|SKU=8578690100208|C
OMISION=0.00|REFERENCIA=DB301C1D70FAF4E6CDB05|MONTO=20|PROVEEDOR=UNEFON|AUTORIZACION=102245|LEYENDA=Dudas tiempo aire marca *050
o 018003330050 |
[2024-11-25 23:45:32:244],FARMAX_WSC,DATALOGICSERVICE,ERROR,[WRITEERROR],(thread=146) ##### Ocurrio un error al guardar en bitacor
a la operacion de 'Execute' para el folio 'V15531000002995' - FAhorro.POS.Application.Core.DbConnectionException: Se perdió la con
exión con la base de datos.
[2024-11-25 23:45:32:285],FARMAX_WSC,DATALOGICSERVICE,INFO,,##### DoTransactionProcessing V15531000002995 result={"Folio":"10224
5","Amount":20.0,"TicketMessage":[{"Label":"Transacción","Value":"1553102995","Type":2}, {"Label":"","Value":"UNE FON","Type":1}, {"L
abel":"","Value":"Dudas tiempo aire marca *050 o 018003330050 ","Type":1}]}
```

Debido a la imposibilidad de guardar en bitácora en ese momento, se encola la bitácora que sirve como consulta de validación respecto si una transacción se ejecutó correctamente incluso si esta tuvo autorización. Como se observa en el primer intento de persistir la bitácora fallo debido al bloqueo, pero en el segundo intento logra persistir la bitácora que es relevante para consultas respecto a la operación

```
[-]$ >> tail -f -n 200 20241125 FARMAX_WSC DATALOGIC_LOGS_PROCESSOR.XLOG | egrep "\w{4}-\w{2}-\w{2} \w{2}:\w{2}:\w{2}:\w{3}|\\"
[2024-11-25 23:45:41:829],FARMAX_WSC,DATALOGIC_LOGS_PROCESSOR,INFO,,##### procesando '1' bitacoras
[2024-11-25 23:45:41:849],FARMAX_WSC,DATALOGIC_LOGS_PROCESSOR,INFO,,##### procesando bitacora de 'V15531000002995'
[2024-11-25 23:45:51:866],FARMAX_WSC,DATALOGIC_LOGS_PROCESSOR,ERROR,[WRITEERROR],(thread=64) ##### Ocurrio un error al guardar en
bitacora la operacion de 'Execute' para el folio 'V15531000002995' - FAhorro.POS.Application.Core.DbConnectionException: Se perdió
la conexión con la base de datos.
[2024-11-25 23:45:51:867],FARMAX_WSC,DATALOGIC_LOGS_PROCESSOR,WARNING,,V15531000002995'. Se reencola la bitacora debido a que fal
lo el intento 1
[2024-11-25 23:46:01:870],FARMAX_WSC,DATALOGIC_LOGS_PROCESSOR,INFO,,##### procesando '1' bitacoras
[2024-11-25 23:46:01:871],FARMAX_WSC,DATALOGIC_LOGS_PROCESSOR,INFO,,##### procesando bitacora de 'V15531000002995'
```

```
select idBitacoraServicios,folioTransaccion,numeroAutorizacion,fechaSolicitud,fechaRespuesta,tipoOperacion,status, datosRespuesta
from venPagoServiciosBitacoraReg order by idBitacoraServicios desc
```

id	numeroAutorizacion	fechaSolicitud	fechaRespuesta	tipoOperacion	status	datosRespuesta
1	182245	2024-11-25 23:46:01.870	<null>	E	P	Type="Unspecified" Encrypted="false">182245</Param><Param Name="MONTO" FieldClas

Cola de reversos centrales

Dado que no necesariamente, la respuesta del integrador ante una reversa está garantizada, ya que pueden presentarse fallos, ante una situación se agregó una cola de reversos para intentar posteriormente su reproceso

```

424 /// <param name=" request "></param>
425 /// <returns></returns>
426 [Usage: <hulver +2>
427 public async Task<ServicePaymentResponse> ProcessPaymentServiceStateless(ServicePaymentRequest request)
428 {
429     if(ManagePreviousRequest(request,out ServicePaymentResponse servicePaymentResponse,out bool hasPendingRequest))...
430
431     try
432     {
433         var requestParameters List<Parameters> = PrepareExecuteRequestParameters(request);
434         List<Parameters> respuestaIntegrador=new List<Parameters>();
435
436         // Guardamos en bd la petición que se enviará al servicio.
437         var generalInformation = MapDataContractToLocalModel(request.TypeServicePayments, requestParameters);
438         var idLog long? = LogRequestExecute(request, generalInformation);
439
440         try
441         {
442             if (hasPendingRequest){
443                 List<Parameters> statusResponse = await GetTransactionStatus(request, requestParameters);
444
445                 if (statusResponse == null){
446                     return await TryReverseIfApply(request);//HNA:SI hay una request pendiente pero no se puede determinar su estado, se intentara la reversa
447                 }
448
449                 if (IsSuccessfulResponse(statusResponse))...
450                 else...
451             }
452             else{
453                 // No pending request. call ejecuta. This the normal stuff
454                 respuestaIntegrador = await InvokeExecuteTransactionLong(requestParameters, request);
455             }
456         }
457         catch (Exception ex){
458             Log.WriteException(request.TypeServicePayments.TypeService, ex, methodName: $"{request.TypeServicePayments.FolioTransaccion} excepción ejecuta", resourceName: string.Empty, resource: null);
459
460             if (ContainsServiceException(ex, typeof(EndpointNotFoundException)) || ContainsServiceException(ex, typeof(TimeoutException)) ||
461                 if (ContainsServiceException(ex, typeof(EndpointNotFoundException)) && hasPendingRequest)...
462
463             List<Parameters> statusResponse = await GetTransactionStatus(request,requestParameters);
464             if (statusResponse == null){//HNA:no se puede determinar su estado, se intentara la reversa y de ser el caso se encolara un reverso,
465                 //pero al tiempo indicaremos que el pos tambien lo debe realizar para garantizar que se realice si o si;
466                 return await TryReverseIfApply(request);
467             }
468         }
469     }

```

```

546 //Intenta aplicar el reverso correspondiente al request, siempre y cuando el producto asociado soporte reversos
547 [Usage: <hulver +1>
548 private async Task<ServicePaymentResponse> TryReverseIfApply(ServicePaymentRequest request) {
549     if (request.TypeServicePayments.SupportsReverse){
550         Log.WriteLine(request.TypeServicePayments.TypeService, message: $"{request.TypeServicePayments.FolioTransaccion} reversa...");
551         // attempt reverse
552         try{
553             var reverseResponse = await this.CancelPaymentService(request, DataLogicResponseCode.TimeoutEmisor, isCancelSue: false);
554             if(IsReversedAccordingToResponse(reverseResponse))
555             {
556                 return ServiceResponseFactory.GetErrorResponseCancelledAndReversed();
557             }
558         }
559         catch (Exception rex){
560             Log.WriteException(request.TypeServicePayments.TypeService, rex,
561                 methodName: $"{request.TypeServicePayments.FolioTransaccion} excepción reversa", resourceName: string.Empty, resource: null);
562
563             Log.WriteLine(request.TypeServicePayments.TypeService, message: $"{request.TypeServicePayments.FolioTransaccion} Encolando reverso");
564             await DataLogicQueueManager.GetInstance().AddElementInQueue(DataLogicReverseQueue,new DataLogicReverseQueueData(request, DataLogicResponseCode.TimeoutEmisor, isCancelSue: false,maxSizeQueue:1000);

```

```

47 Log.WriteInfo(Submodule,
48     message: $"### Un proceso anterior aun no termina su ejecucion, por lo que se intenta a la proxima ejecucion");
49 }
50 }
51
52 [Usage & holver
53 private void ProcesaReversos()
54 {
55     int actualSize = _manager.Count(DataLogicServiceClient.DataLogicReverseQueue);
56     if (actualSize > 0){
57         Log.WriteInfo(Submodule, message: $"### procesando '{actualSize}' reversos");
58     }
59
60     for (int i = 0; i < actualSize; i++){
61         var task = Task.Run(async () =>
62             await _manager.DequeueElementOfQueue<DataLogicReverseQueueData>(DataLogicServiceClient.DataLogicReverseQueue));
63
64         var reverseToProcess = task.Result;
65
66         if (reverseToProcess != null){
67             if (reverseToProcess.NumIntentos < MaxNumIntentos){
68                 reverseToProcess.NumIntentos += 1;
69
70                 var request = reverseToProcess.Request;
71                 string folioTransaction = request.TypeServicePayments.FolioTransaction;
72                 Log.WriteInfo(Submodule, message: $"### procesando reverso de '{folioTransaction}'");
73
74                 Task.Run(async () =>
75                 {
76                     try{
77                         using (var client = new DataLogicServiceClient()){
78                             var resultReverseServicePaymentResponse = await client.CancelPaymentService(request, reverseToProcess.ReasonCode, reverseToProcess.IsCancelSale);

```

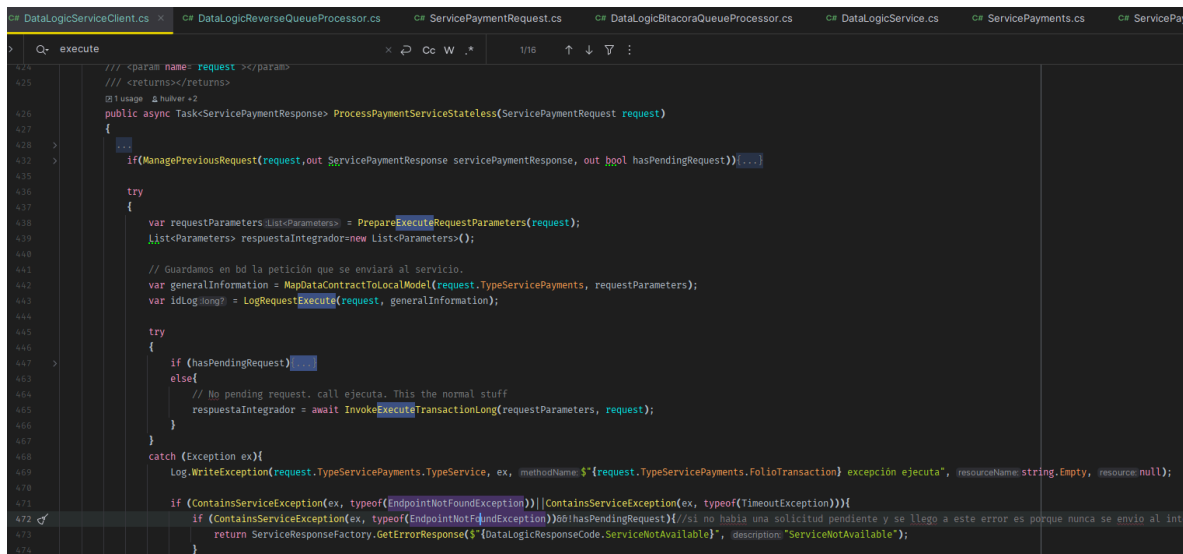
Pruebas

TimeOut para operaciones Execute, Status y Reverse de broker

La pruebas de esta implementación pueden observarse en las pruebas por TimeOut del broker en la sección [Caso TIMEOUT por datalogic para todas las operaciones](#)

Manejo de no disponibilidad

Para evitar llamadas innecesarias cuando no esté disponible el servicio se maneja la excepción `EndpointNotFoundException` que indica la no disponibilidad del servicio.



```
624 /// <param name="request"></param>
625 /// <returns></returns>
626 [Usage: huiyer + 2]
627 public async Task<ServicePaymentResponse> ProcessPaymentServiceStateless(ServicePaymentRequest request)
628 {
629     if(ManagePreviousRequest(request,out ServicePaymentResponse servicePaymentResponse, out bool hasPendingRequest)){}
630
631     try
632     {
633         var requestParameters:List<Parameters> = PrepareExecuteRequestParameters(request);
634         List<Parameters> respuestaIntegrador=new List<Parameters>();
635
636         // Guardamos en bd la petición que se enviará al servicio.
637         var generalInformation = MapDataContractToLocalModel(request.TypeServicePayments, requestParameters);
638         var idLog:long? = LogRequestExecute(request, generalInformation);
639
640         try
641         {
642             if (hasPendingRequest)
643             else
644             // No pending request. call ejecuta. This the normal stuff
645             respuestaIntegrador = await InvokeExecuteTransactionLong(requestParameters, request);
646         }
647     }
648     catch (Exception ex){
649         Log.WriteException(request.TypeServicePayments.TypeService, ex, methodName:${request.TypeServicePayments.FolioTransaction} excepción ejecuta", resourceName:String.Empty, resource:null);
650
651         if (ContainsServiceException(ex, typeof(EndpointNotFoundException))||ContainsServiceException(ex, typeof(TimeoutException))){
652             if (ContainsServiceException(ex, typeof(EndpointNotFoundException))&&hasPendingRequest){//si no habia una solicitud pendiente y se lleo a este error es porque nunca se envío al int
653                 return ServiceResponseFactory.GetErrorResponse("${DataLogicResponseCode.ServiceNotAvailable}", description:"ServiceNotAvailable");
654             }
655         }
656     }
```

Pruebas

Caso, operación EXECUTE de broker no disponible

Para poder garantizar este comportamiento, se configura el proxy para que se dispare la excepción de `EndpointNotFoundException` de tal suerte que se logre manejar dicho escenario.


```
<!--=====DATALOGIC TESTING=====-->
<!--Para controlar los timeouts de prueba de datalogic-->
<add key="TimeOutExecute" value="10" />
<add key="TriesToCompleteExecute" value="2" />
<add key="EndpointNotFoundExceptionExecute" value="true" />

<add key="TimeOutStatus" value="10" />
<add key="TriesToCompleteStatus" value="1" />
<add key="EndpointNotFoundExceptionStatus" value="false" />

<add key="TimeOutReverse" value="10" />
<add key="TriesToCompleteReverse" value="2" />
<add key="EndpointNotFoundExceptionReverse" value="false" />
```

```
<add key="TimeOutInfo" value="10" />
<add key="TriesToCompleteInfo" value="1" />
<add key="EndpointNotFoundExceptionInfo" value="false" />
<add key="DatalogicTesting" value="true" />
<!--end timeouts datalogic-->
<!--=====-->
```

Se ejecuta la operación datalogic:


RECARGA UNEFON \$ 20

IUSACELL
SIN MARCA
MOVISTAR
UNEFON
VIRGIN
IUSACELL/AT&T

Monto

RECARGA UNEFON \$ 20

Número

555555555

Confirmar

555555555

orro

Servicio a Domicilio

PENDIENTES: 73
POR DESPACHAR: 18
POR ENTREGAR: 36
POR DEVOLVER: 0

Formas de Pago

E

EFFECTIVO (CA)

20.00

T


TARJETA BANCARIA


0.00

Subtotal: \$17.24



I.V.A: \$2.76

Total: \$20.00

 Tu Donativo
\$0.00


 Tu Ahorro
\$0.00

Por Pagar
\$0.00



Lo que se obtiene es un mensaje que el servicio no se encuentra en línea, lo cual es un comportamiento esperado

12:59 AM 26/11/2024



Bienvenido a Farmacias del Ahorro

Te Atiende FABIOLA

Servicio a Domicilio

PENDIENTES: 73
POR DESPACHAR: 18
POR ENTREGAR: 36
POR DEVOLVER: 0

Formas de Pago

E

EFFECTIVO (CA)


20.00

T

TARJETA BANCARIA

0.00

Error



Código 72: ServiceNotAvailable

Aceptar

Revisamos los logs del POS, en donde se observa que se obtiene una respuesta 1001(reversa exitosa), que indica que se debe cancelar la venta, el motivo es por **ServiceNotAvailable**

```
[L[$] <> tail -f -n 200 20241126_FARMAX_POS_DATALOGICSERVICE.XLOG | egrep "\w{4}-\w{2}-\w{2} \w{2}:\w{2}:\w{2}:\w{3}|V[0-9]+|\\"
[2024-11-26 01:00:10:154],FARMAX_POS,DATALOGICSERVICE,INFO,[LOG],##### ExecuteServiceTransaction V15531000002996 request={"TypeS
ervicePayments":{"Brand":"UNEFON","TypeService":"DataLogicService","BusinessUnit":1553,"UserCode":3,"EncryptReference":true,"Folio
Transaction":"V15531000002996","OperationDate":"2023-04-13T00:00:00","TransactionDate":"2024-11-26T01:00:09","ShiftId":"b66c518c-c
2ab-ef11-9692-94bb432cf0d5","SupportsReverse":true,"PrintsTicketNoAuth":true},"ProductCode":758778193,"RelatedCode":"8578690100208
","ExternalCode":"N/A","ProductServiceCode":758778193,"TillId":10,"BusinessUnitId":1553,"MacAddress":"94-BB-43-2C-F0-D4","IPAddres
s":"192.168.1.6","OperationDate":"2023-04-13T00:00:00","ListParameters":{"Parameters":[{"Label":"COMISION","Type":3,"InputType":1,
"Length":8,"Action":1,"Value":0.00,"Field":"ComissionAmount","FieldClass":1},{Label":"REFERENCIA","Type":1,"InputType":1,"Length"
:10,"Action":1,"Value":"5555555555","Encrypted":true,"Field":"Reference","FieldClass":1},{Label":"MONTO","Type":3,"InputType":1,"
Length":8,"Action":1,"Value":20.0,"Field":"Amount","FieldClass":1}]}}
[2024-11-26 01:00:11:242],FARMAX_POS,DATALOGICSERVICE,INFO,[LOG],##### ExecuteServiceTransaction V15531000002996 result={"DateCa
ancel":"","DateTransaction":"","Folio":"","CodeError":"1001","DescriptionError":"Código 72: ServiceNotAvailable","Store":"","Servic
ePaid":"","TicketMessage":[]}
[2024-11-26 01:00:20:917],FARMAX_POS,DATALOGICSERVICE,INFO,[LOG],##### ExecuteServiceTransaction V15531000002996 action CancelSal
e
```

Se observa que tanto la venta como el pago se encuentran como inactivo y cancelado respectivamente.

```
select idTransaccion, folioTransaccion, subTotal, codigoMovimientoCaja, fechaTransaccion, estaActivo from trnTransaccionesCab with (nolock) where folioTransaccion = 'V15531000002996';
+-----+-----+-----+-----+-----+
| idTransaccion | folioTransaccion | subTotal | codigoMovimientoCaja | fechaTransaccion | estaActivo |
+-----+-----+-----+-----+-----+
| 1890A604-CAAB-EF11-9692-94BB432CF0D5 | V15531000002996 | 17.2413 | DB03BC12-CAAB-EF11-9692-94BB432CF0D5 | 2024-11-26 01:00:20,970 | false |
+-----+-----+-----+-----+-----+

select codigoMovimientoCaja, importePago, estatusFormaPago from genMovimientosCajasDet with (nolock)
where codigoMovimientoCaja in (select codigoMovimientoCaja from trnTransaccionesCab with (nolock) where folioTransaccion = 'V15531000002996');
+-----+-----+-----+
| codigoMovimientoCaja | importePago | estatusFormaPago |
+-----+-----+-----+
| DB03BC12-CAAB-EF11-9692-94BB432CF0D5 | 20.0000 | C |
+-----+-----+-----+
```

Se revisan los logs de los servicios centrales, y lo que se observa es el invocar la operación Execute del broker, este lanza una excepción de `EndpointNotFoundException`, dado este error, el servicio va a intentar un reverso, lo cual logra aplicar y manda como respuesta al POS el código de error 1001

```
[L[$] <> tail -f -n 200 20241126_FARMAX_WSC_DATALOGICSERVICE.XLOG | egrep "\w{4}-\w{2}-\w{2} \w{2}:\w{2}:\w{2}:\w{3}|V[0-9]+|\\"
2024-11-26 01:00:10:213],FARMAX_WSC,DATALOGICSERVICE,INFO,V15531000002996 treatAsPending=False
2024-11-26 01:00:10:357],FARMAX_WSC,DATALOGICSERVICE,INFO,V15531000002996 (FAhorro.POS.Services.DataLogic.helper.PxUniversalSoap
lientFake) ejecuta...
2024-11-26 01:00:10:360],FARMAX_WSC,DATALOGICSERVICE,ERROR,[WRITEERROR],(thread=176) V15531000002996 excepcion ejecuta. - System
ServiceModel.EndpointNotFoundException: Unable to connect to the remote server
2024-11-26 01:00:10:381],FARMAX_WSC,DATALOGICSERVICE,INFO,#####Start -> CancelPaymentService(V15531000002996,reasonCode=Service
otAvailable)
2024-11-26 01:00:11:227],FARMAX_WSC,DATALOGICSERVICE,INFO,#####End CancelPaymentService -> (V15531000002996,isReverseSuccessful
True')
2024-11-26 01:00:11:236],FARMAX_WSC,DATALOGICSERVICE,INFO,##### DoTransactionProcessing V15531000002996 result={"PaymentError"
{"CodeError":"1001","DescriptionError":"Código 72: ServiceNotAvailable"}}
```


Consultamos la bitácora de servicios, en la cual se ve que se ejecutó la operación Execute la cual no tuvo respuesta y despues una operación de reversa la cual fue exitosa

```
select idBitacoraServicios,folioTransaccion,numeroAutorizacion,fechaSolicitud,fechaRespuesta,tipoOperacion,status, datosRespuesta
from venPagoServiciosBitacoraReg order by idBitacoraServicios desc
```

Lo anterior demuestra que Farmax POS está manejando bien el escenario de no disponibilidad del servicio de broker.

HappyPath

En esta prueba no se requiere la simulación de algún caso, más bien solo garantizar que el funcionamiento siga funcionando de manera correcta, por lo que se elimina toda la configuración de testing y sea un pase directo al broker de servicios.

Es decir eliminamos la configuración de datalogic testing:

```
<!--=====DATALOGIC TESTING=====-->
<!--Para controlar los timeouts de prueba de datalogic-->
...
<!--end timeouts datalogic-->
<!--=====-->
```

Y ejecutamos una transacción de datalogic

10

03:55 PM 26/11/2024

Bienvenido a Farmacias del Ahorro

Te Atiende FABIOLA

Servicio al Cliente
PENDIENTES: 73
POR DESGRACIAR: 18
POR ENTREGAR: 36
POR DEVOLVER: 0

Pago de Servicios

> SERVICIOS

> VARIOS

> ENTRETENIMIENTO

> SERVICIOS VARIOS

> TRANSPORTE

> TIEMPO AIRE

> SERVICIOS FINANCIEROS

> PAGO DE SERVICIOS MP

> VENTAS POR CATALOGO

> PAGO DE SERVICIOS

> COMPRAS POR INTERNET

> LUZ

> TELECOMUNICACIONES

← RECARGA UNEFON \$ 20

RUSACELL

SIN MARCA

MOVISTAR

UNEFON

VIRGIN

RUSACELL/AT&T

Monto

RECARGA UNEFON \$ 20

Número

5555555555

Confirmar

5555555555

Tu Cambio
\$0.00

Gracias por tu preferencia



COMERCIALIZADORA FARMACEUTICA DE CHIAPAS SAPI DE CV
TLAXCALA, 151 PISO 4 OFNA 407 HIPODROMO, CUAUHEMOC
CP.06100 RFC: CFC110121742

F A TIJUANA, PLUTARCO ELIAS CALLES, 11 PLUTARCO
ELIAS CALLES, 9026
ZONA ESTE, TIJUANA, TIJUANA, BAJA CALIFORNIA
CP.22000

Fecha OP: 13/04/2023

V15531000003004 26/11/2024 15:57

Multifuncional: FABIOLA VALERIO MORENO

Le Atendió: FABIOLA VALERIO MORENO

RECARGA UNEFON \$ 20 \$17.24

IVA: \$2.76

TOTAL: \$20.00

PAGO: \$20.00

CAMBIO: \$0.00

VEINTE PESOS 00/100 M.N.

FORMA DE PAGO:

EFFECTIVO (CA): \$20.00

REFERENCIA: 5555555555

AUTORIZACIÓN: 102245

TRANSACCIÓN: 1553103004

UNEFON

Dudas tiempo aire marca *050 o 018003330050

ITU: c987ec62-41ac-ef11-9692-94bb432cf0d5



v15531000003004

Se revisa el log del POS, en que se observa que la transacción corrió correctamente en 1 segundo y con la acción de completar pago

```
[2024-11-26 15:57:40:063],FARMAX_POS,DATALOGICSERVICE,INFO,[LOG],##### ExecuteServiceTransaction V15531000003004 request={"TypeS
ervicePayments":{"Brand":"UNE FON","TypeService":"DataLogicService","BusinessUnit":1553,"UserCode":3,"EncryptReference":true,"Folio
Transaction":"V15531000003004","OperationDate":"2023-04-13T00:00:00","TransactionDate":"2024-11-26T15:57:39","ShiftId":"b66c518c-c
2ab-ef11-9692-94bb432cf8d5","SupportsReverse":true,"PrintsTicketNoAuth":true,"ProductCode":758778193,"RelatedCode":"8578690100208
","ExternalCode":"N/A","ProductServiceCode":758778193,"TillId":10,"BusinessUnitId":1553,"MacAddress":"94-BB-43-2C-F0-D4","IPAdres
s":"192.168.1.6","OperationDate":"2023-04-13T00:00:00","ListParameters":{"Parameters":[{"Label":"COMISION","Type":3,"InputType":1,
"Length":8,"Action":1,"Value":0.00,"Field":"ComissionAmount","FieldClass":1},{Label":"REFERENCIA","Type":1,"InputType":1,"Length
":10,"Action":1,"Value":"555555555","Encrypted":true,"Field":"Reference","FieldClass":1},{Label":"MONTO","Type":3,"InputType":1,"
Length":8,"Action":1,"Value":20.0,"Field":"Amount","FieldClass":1}]}}
[2024-11-26 15:57:41:201],FARMAX_POS,DATALOGICSERVICE,INFO,[LOG],##### ExecuteServiceTransaction V15531000003004 result={"Amount
":20.0,"DateCancel":"","DateTransaction":"","Folio":"102245","Store":"","ServicePayed":"","TicketMessage":[{"Label":"Transacción",
"Value":"1553103004","Type":2},{Label":"","Value":"UNE FON","Type":1},{Label":"","Value":"Dudas tiempo aire marca *050 o 01800333
0050 ","Type":1}]}
[2024-11-26 15:57:41:207],FARMAX_POS,DATALOGICSERVICE,INFO,[LOG],##### ExecuteServiceTransaction V15531000003004 action CompleteP
ayment
```

Se revisa el estado de la venta y del pago, em ambos se ven como activos o aplicados según sea el caso

```
select idTransaccion, folioTransaccion, subTotal, codigoMovimientoCaja, fechaTransaccion, estaActivo from trnTransaccionesCab with (nolock) where folioTransaccion = 'V15531000003004';
```

idTransaccion	folioTransaccion	subTotal	codigoMovimientoCaja	fechaTransaccion	estaActivo
C987EC62-41AC-EF11-9692-94BB432CF8D5	V15531000003004	17.2413	CEC1D973-41AC-EF11-9692-94BB432CF8D5	2024-11-26 15:57:39.890	true

```
select codigoMovimientoCaja, importePago, estatusFormaPago from genMovimientosCajasDet with (nolock)
where codigoMovimientoCaja in (select codigoMovimientoCaja from trnTransaccionesCab with (nolock) where folioTransaccion = 'V15531000003004');
```

codigoMovimientoCaja	importePago	estatusFormaPago
CEC1D973-41AC-EF11-9692-94BB432CF8D5	20.0000	A

Despues se revisa el log del central, en que se observa que la transacción ocurrió de manera exitosa, regresado al POS un folio de autorización.

```
[2024-11-26 15:57:40:139],FARMAX_WSC,DATALOGICSERVICE,INFO,,V15531000003004 treatAsPending=False
[2024-11-26 15:57:40:317],FARMAX_WSC,DATALOGICSERVICE,INFO,,V15531000003004 (FAhorro.POS.Services.DataLogic.helper.PxUniversalSoap
ClientProxy) ejecuta...
[2024-11-26 15:57:41:034],FARMAX_WSC,DATALOGICSERVICE,INFO,,V15531000003004 :: Datos de la transaccion exitosa datalogic IDGRUPO=
4|IDCADENA=1|IDIENDA=1553|IDPOS=10|IDCAJERO=3|FECHALOCAL=26/11/2024|HORALOCAL=15:57:39|TRANSACCION=1553103004|SKU=8578690100208|C
OMISION=0.00|REFERENCIA=DB301C1D70FAF4E6CDB05|MONTO=20|PROVEEDOR=UNE FON|AUTORIZACION=102245|LEYENDA=Dudas tiempo aire marca *050
o 018003330050 |
[2024-11-26 15:57:41:201],FARMAX_WSC,DATALOGICSERVICE,INFO,,##### DoTransactionProcessing V15531000003004 result={"Folio":"10224
5","Amount":20.0,"TicketMessage":[{"Label":"Transacción","Value":"1553103004","Type":2},{Label":"","Value":"UNE FON","Type":1},{L
abel":"","Value":"Dudas tiempo aire marca *050 o 018003330050 ","Type":1}]}
```

Por último se revisa la bitácora de servicios en la que se observa que hubo una llamada al método Execute y se completo la llamada con datos de respuesta

```
from venPagoServiciosBitacoraReg order by idBitacoraServicios desc
```

idBitacoraServicios	folioTransaccion	numeroAutorizacion	fechaSolicitud	fechaRespuesta	tipoOperacion	status	datosRespuesta
110	V15531000003004	102245	2024-11-26 15:57:40.310	2024-11-26 15:57:41.037	C		DATALOGICServiceInformation: {\"Label\":\"Transacción\", \"Value\":\"1553103004\", \"Type\":2}, {\"Label\":\"\", \"Value\":\"UNE FON\", \"Type\":1}, {\"Label\":\"\", \"Value\":\"Dudas tiempo aire marca *050 o 018003330050 \", \"Type\":1}}