



Ticket 9999999999 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 han realizado una serie de ajustes que permitan solventar dichas deficiencias:

- Aumentar 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

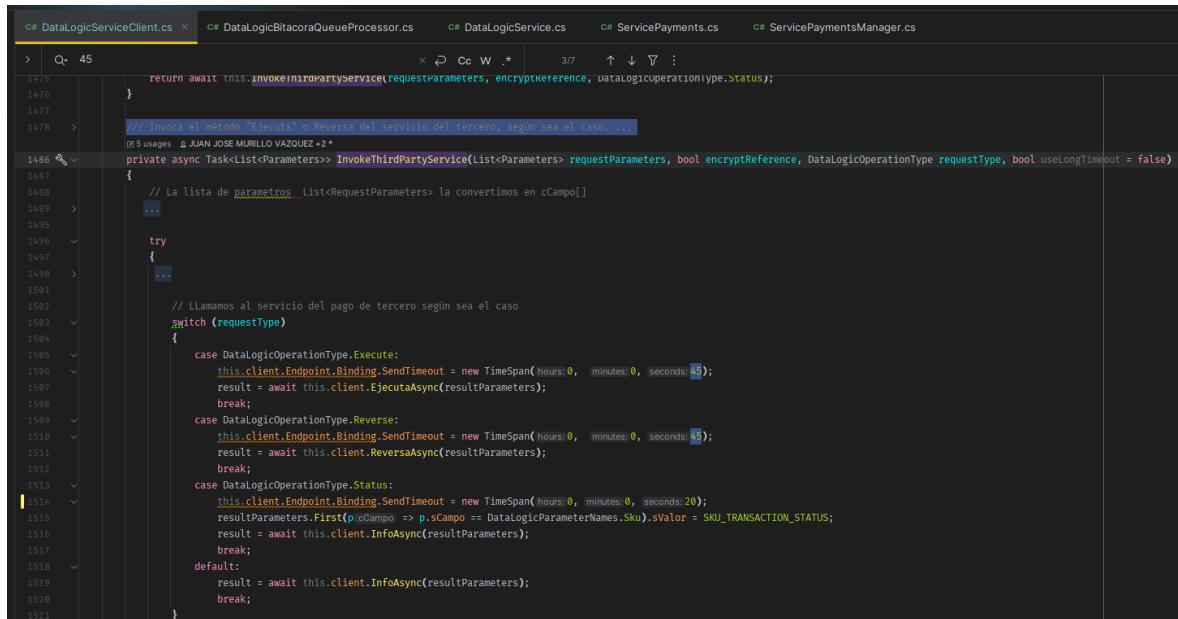
- Se incorpora una cola de reversos a nivel central para aquellos reversos que no tuvieron respuesta del integrador ya sea por falta de disponibilidad del broker, una respuesta desconocida o incluso timeout del broker.
- Se determina el tipo de excepción recibida y al invocar al integrador y si se determina 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:



```

C# DataLogicServiceClient.cs X C# DataLogicBitacoraQueueProcessor.cs C# DataLogicService.cs C# ServicePayments.cs C# ServicePaymentsManager.cs
> Q- 45
1475     return await this.InvokeInThirdPartyService(requestParameters, encryptReference, DataLogicOperationType.Status);
1476 }
1477
1478 > // Invoca el método "Ejecuta" o Reversa del servicio del tercero, según sea el caso. ....
1479 @ 5 usages Δ 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
1485     try
1486     {
1487         ...
1488
1489         // Llamamos al servicio del pago de tercero según sea el caso
1490         switch (requestType)
1491         {
1492             case DataLogicOperationType.Execute:
1493                 this.client.Endpoint.Binding.SendTimeout = new TimeSpan(hours: 0, minutes: 0, seconds: 45);
1494                 result = await this.client.EjecutaAsync(resultParameters);
1495                 break;
1496             case DataLogicOperationType.Reverse:
1497                 this.client.Endpoint.Binding.SendTimeout = new TimeSpan(hours: 0, minutes: 0, seconds: 45);
1498                 result = await this.client.ReversaAsync(resultParameters);
1499                 break;
1500             case DataLogicOperationType.Status:
1501                 this.client.Endpoint.Binding.SendTimeout = new TimeSpan(hours: 0, minutes: 0, seconds: 20);
1502                 resultParameters.First(p => p.cCampo == DataLogicParameterNames.Sku).sValor = SKU_TRANSACTION_STATUS;
1503                 result = await this.client.InfoAsync(resultParameters);
1504                 break;
1505             default:
1506                 result = await this.client.InfoAsync(resultParameters);
1507                 break;
1508         }
1509     }
1510 }
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521

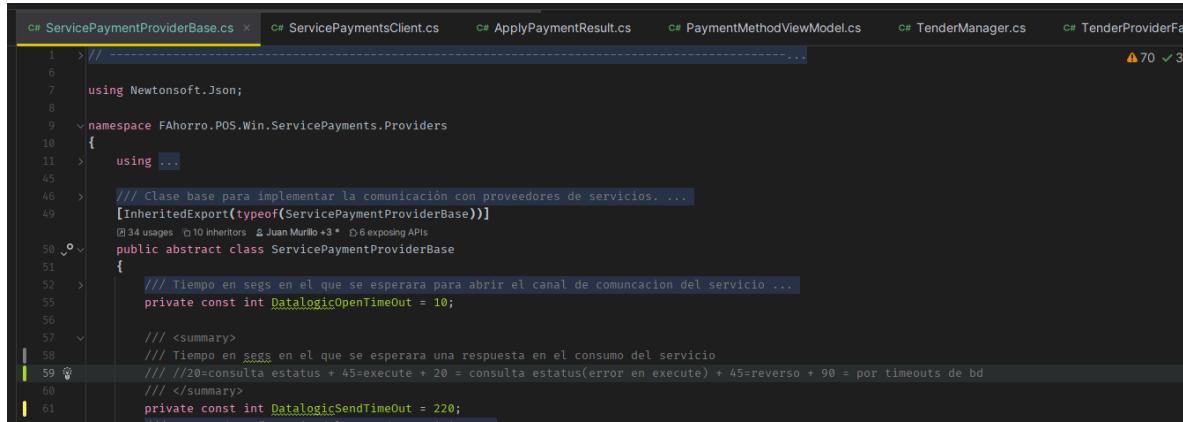
```

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
212  |    public static long SaveLogPaymentServices(LogPaymentServices item)
213  |    {
214  |        int version = ConfigurationManager.Version.ToInt32();
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
```

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     }
265 }
```



```

1  > // ...
6
7  using Newtonsoft.Json;
8
9  <namespace FAhorro.POS.Win.ServicePayments.Providers
10 {
11  >     using ...
14
15  >     /// Clase base para implementar la comunicación con proveedores de servicios. ...
16  >     [InheritedExport(typeof(ServicePaymentProviderBase))]
17  >     [34 usages 10 inheritors Juan Murillo +3* 6 exposing APIs]
18  >     public abstract class ServicePaymentProviderBase
19  {
20  >         /// Tiempo en segs en el que se esperara para abrir el canal de comunicación del servicio ...
21  >         private const int DatalogicOpenTimeOut = 10;
22
23  >         /// <summary>
24  >         /// Tiempo en segs en el que se esperara una respuesta en el consumo del servicio
25  >         /// 20=consulta estatus + 45=execute + 20 = consulta estatus(error en execute) + 45=reverso + 90 = por timeouts de bd
26  >         /// </summary>
27  >         private const int DatalogicSendTimeOut = 220;
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61

```

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 reversa 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.WriteLine(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:2 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.

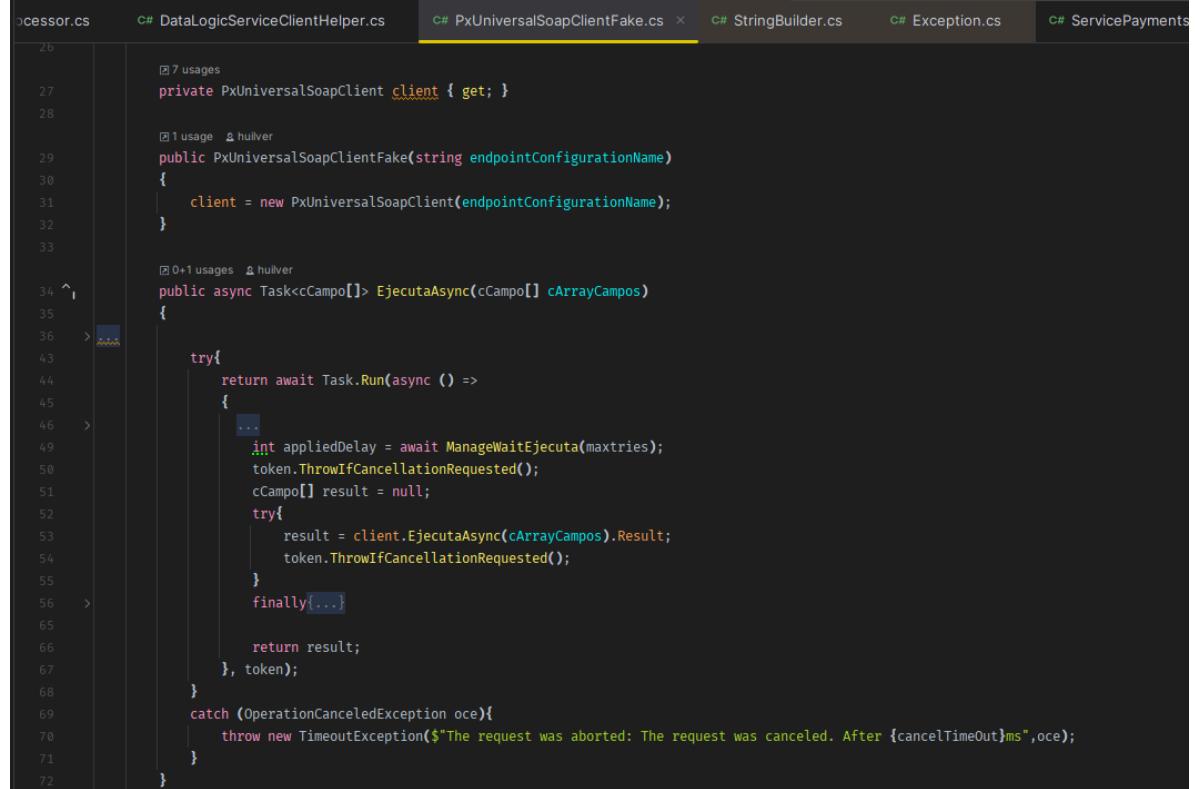
Pruebas

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.



```
26
27     [7 usages]
28     private PxUniversalSoapClient client { get; }
29
30     [1 usage  ↳ hulver]
31     public PxUniversalSoapClientFake(string endpointConfigurationName)
32     {
33         client = new PxUniversalSoapClient(endpointConfigurationName);
34     }
35
36     [0+1 usages  ↳ hulver]
37     public async Task<ccampo[]> EjecutaAsync(ccampo[] carrayCampos)
38     {
39
40         try{
41             return await Task.Run(async () =>
42             {
43                 ...
44                 int appliedDelay = await ManageWaitEjecuta(maxtries);
45                 token.ThrowIfCancellationRequested();
46                 ccampo[] result = null;
47                 try{
48                     result = client.EjecutaAsync(carrayCampos).Result;
49                     token.ThrowIfCancellationRequested();
50                 }
51                 finally{...}
52
53                 return result;
54             }, token);
55         }
56
57         catch (OperationCanceledException oce){
58             throw new TimeoutException($"The request was aborted: The request was canceled. After {cancelTimeOut}ms",oce);
59         }
60     }
61
62 }
```

Web.config

```

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
432

```

DataLogicServiceClientHelper.cs

```

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

```

```

<!-----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="EndpointNotFoundExceptionStatus" value="false" />

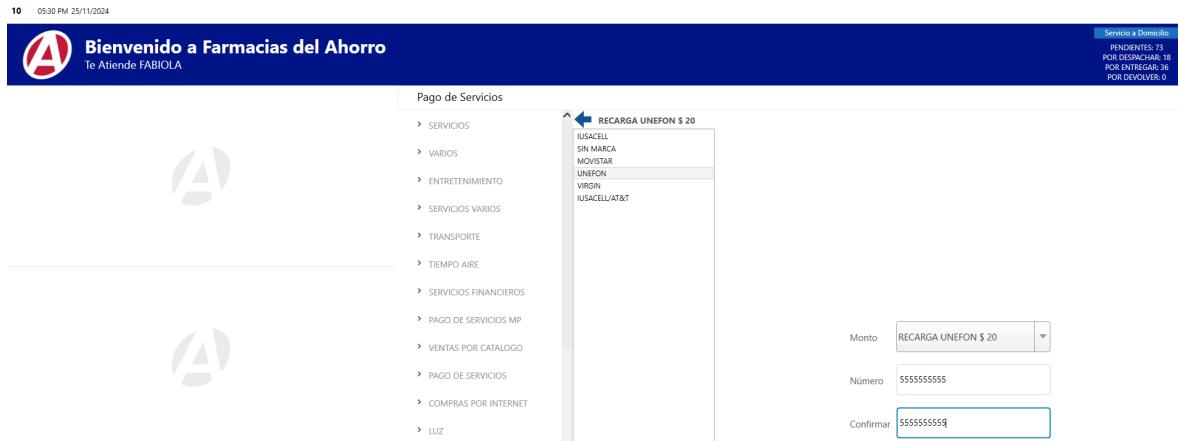
<add key="TimeOutReverse" value="44600" />
<add key="TriesToCompleteReverse" value="2" />
<add key="EndpointNotFoundExceptionReverse" value="false" />

<!----->

```

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:



The screenshot shows a POS interface for Farmacias del Ahorro. At the top, there is a header with the logo and the text "Bienvenido a Farmacias del Ahorro" and "Te Atiende FABIOLA". On the right, there is a "Servicio a Domicilio" section showing pending deliveries: 73 pending, 36 to deliver, and 0 to return. The main area is titled "Pago de Servicios" and shows a list of service categories. A dropdown menu is open under "RECARGA UNEFON \$ 20", listing providers: IUSACELL, SIN MARCA, MOVISTAR, UNEFON, VIRGIN, and IUSACELL/AT&T. To the right, there are fields for "Monto" (set to \$20), "Número" (55555555), and "Confirmar" (555555559).

10 05:31 PM 25/11/2024

Bienvenido a Farmacias del Ahorro
Te Atiende FABIOLA

Formas de Pago

E	Efectivo (CA)	20.00
T	Tarjeta Bancaria	0.00

Subtotal: \$17.24 **I.V.A: \$2.76** **Total: \$20.00**

Donativo: \$0.00 **Tu Ahorro: \$0.00** **Por Pagar: \$0.00**

Error

 Venta cancelada con intentos de reversa

Aceptar

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 tambien 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.

```
[\$] <> tail -f -n 200 20241125 FARMAX_POS.DATALOGICSERVICE.XLOG | egrep "\w{4}-\w{2}-\w{2} \w{2}:\w{2}:\w{2}:\w{3}|\w{0-9}+|\\""
[2024-11-25 17:32:58:186],FARMAX_POS.DATALOGICSERVICE,INFO,[LOG],##### ExecuteServiceTransaction V15531000002993 request={"Type
ServicePayments":{"Brand": "UNEFON", "TypeService": "DataLogicService", "BusinessUnit": 1553, "UserCode": 3, "EncryptReference": true, "Foli
oTransaction": "V15531000002993", "OperationDate": "2023-04-13T00:00:00", "TransactionDate": "2024-11-25T17:32:57", "ShiftId": "45a04ff
-84ab-ef11-9692-94bb432cf0d5", "SupportsReverse": true, "PrintsTicketNoAuth": true}, "ProductCode": 758778193, "RelatedCode": "857869010020
8", "ExternalCode": "N/A", "ProductServiceCode": 758778193, "TillId": 10, "BusinessUnitId": 1553, "MacAddress": "94-BB-43-2C-F0-D4", "IPAddre
ss": "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": "ComisionAmount", "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={"DateCa
ncel": "", "DateTransaction": "", "Folio": "", "CodeError": "1002", "DescriptionError": "Venta cancelada con intentos de reversa", "Store": " "
, "ServicePayed": "", "TicketMessage": []}
[2024-11-25 17:34:49:804],FARMAX_POS.DATALOGICSERVICE,INFO,[LOG],##### QueueReverse V15531000002993 queueing reverse, response ==
null->False response.IsError CodeWaitingOrUnknown()->False !printsTicketNoAuth False response?.CodeError->1002
[2024-11-25 17:35:22:768],FARMAX_POS.DATALOGICSERVICE,INFO,[LOG],##### ExecuteServiceTransaction V15531000002993 action CancelSal
e
```

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

select idTransaccion, folioTransaccion, subtotal, codigoMovimientoCaja, fechaTransaccion, estaActivo from tmnTransaccionesCab with (nolock) where folioTransaccion = 'V15531000002993';					
idTransaccion	folioTransaccion	subtotal	codigoMovimientoCaja	fechaTransaccion	estaActivo
DC39AC5C-85AB-EF11-9692-94BB432CF0D5	V15531000002993	17.2413	64829999-85AB-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 tmnTransaccionesCab with (nolock) where folioTransaccion = 'V15531000002993');					
codigoMovimientoCaja	importePago	estatusFormaPago			
64829999-85AB-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, después se *ejecuta* la solicitud al broker, sin embargo después de **45** segs la solicitud fue cancelada por timeout(justo lo esperado), después 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 después de **20** segs, dada la situación anterior, se procede a intentar el *reverso* pero la solicitud se cancela por timeout después 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}|\w{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.PxUniversalSoap
ClientFake) ejecuta...
[2024-11-25 17:33:43:950],FARMAX_WSC,DATALOGICSERVICE,ERROR,[WRITERERROR],(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,[WRITERERROR],(thread=15) V15531000002993 excepción consulta. - System.
TimeoutException: The request was aborted: The request was canceled. After 20000ms - por timeout de respuesta de 20 segs
[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,[WRITERERROR],##### 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,[WRITERERROR],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}|\w[
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,[WRITERERROR],##### 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,isRev
erseSuccessful='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							
7	FARMAX_CENTRAL.dbo.v_oServiciosBitacoraReg	FARMAX_CENTRAL.dbo.v_oServiciosBitacoraReg 2	FARMAX_CENTRAL.dbo.v_oServiciosBitacoraReg 3	FARMAX_CENTRAL.dbo.v_oServiciosBitacoraReg 4			
	idBitacoraServicios	folioTransaccion	numeroAutorizacion	fechaSolicitud	fechaRespuesta	tipoOperacion	status
3	117	V15531000002993	<null>	2024-11-25 17:36:09.003	2024-11-25 17:36:09.773	P	C
4	116	V15531000002993	<null>	2024-11-25 17:36:09.003	<null>	P	P
5	115	V15531000002993	<null>	2024-11-25 17:34:04.213	<null>	P	P
6	114	V15531000002993	<null>	2024-11-25 17:34:43.953	2024-11-25 17:34:04.283	I	CE
7	113	V15531000002993	<null>	2024-11-25 17:32:58.280	<null>	E	P

1	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" />

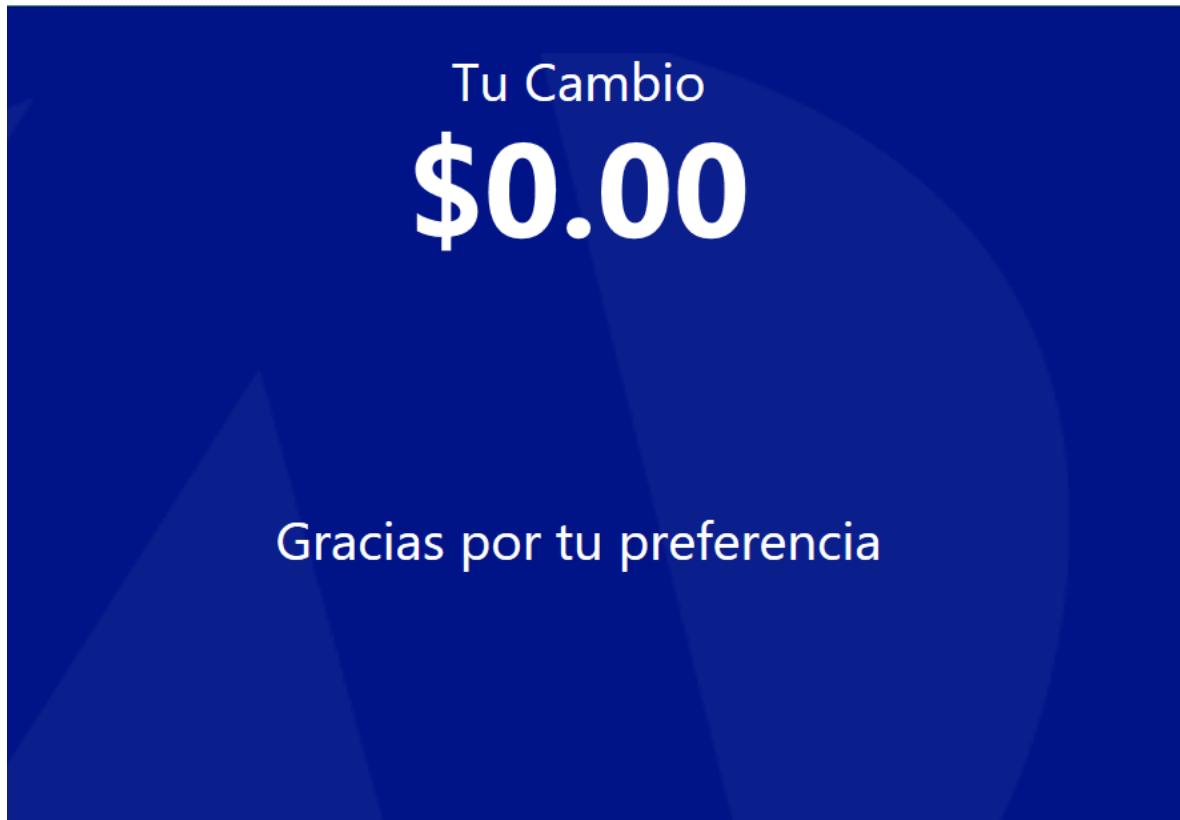
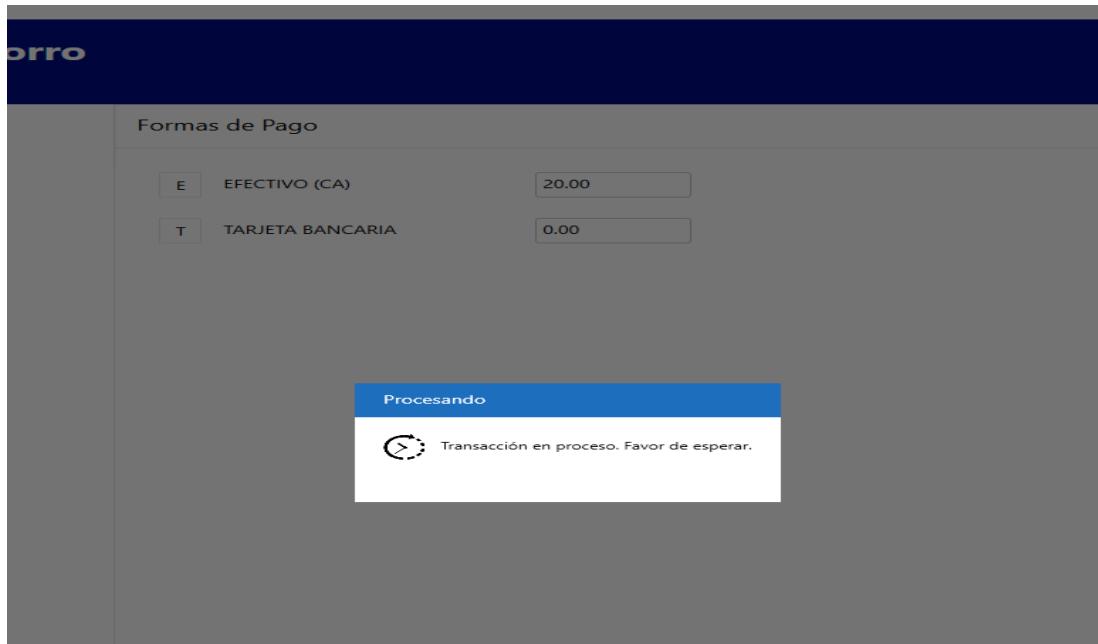
<add key="TimeOutReverse" value="10" />
<add key="TriesToCompleteReverse" value="2" />
<add key="EndpointNotFoundExceptionReverse" value="false" />

<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: 72
POR DESPACHAR: 18
POR ENTREGAR: 36
POR DEVOLVER: 0

Pago de Servicios	
<ul style="list-style-type: none"> > 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 	<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> ^ RECARGA UNEFON \$ 20 </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> IUSACELL SIN MARCA MOVISTAR UNEFON VIRGIN IUSACELL/AT&T </div> <div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> Monto <input style="width: 100%; border: 1px solid #ccc;" type="text" value="RECARGA UNEFON \$ 20"/> </div> <div style="width: 45%;"> Número <input style="width: 100%; border: 1px solid #ccc;" type="text" value="5555555555"/> </div> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> Confirmar <input style="width: 100%; border: 1px solid #0070C0;" type="text" value="5555555555"/> </div>





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={"TypeServicePayments":[{"Brand":"UNEFON","TypeService":"DataLogicService","BusinessUnit":1553,"UserCode":3,"EncryptReference":true,"FolioTransaction":"V15531000002994","OperationDate":"2023-04-13T00:00:00","TransactionDate":"2024-11-25T22:25:48","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,"IPAddresses":"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":"ComisionAmount","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":20.0,"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": "1553102994", "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 CompletePayment
```

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.PxUniversalSoap
ClientFake) ejecuta...
[2024-11-25 22:26:34:914],FARMAX_WSC,DATALOGICSERVICE,ERROR,[WRITERERROR],(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, "InputType": 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, "InputType": 1, "Length": 10, "Action": 1, "Value": "5555555555", "Field": "REFERENCIA", "FieldClass": 6}, {"Label": "MONTO", "Type": 3, "InputType": 1, "Length": 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": 4, "Action": 1, "Value": "102245", "Field": "AUTORIZACION", "FieldClass": 6}, {"Label": "LEYENDA2", "Type": 1, "InputType": 1, "Length": 0, "Action": 1, "Value": "", "Field": "LEYENDA2"}, {"Label": "LEYENDA", "Type": 1, "InputType": 1, "Length": 43, "Action": 1, "Value": "Dudas tiempo aire marca *050 o 018003330050", "Field": "LEYENDA"}, {"Label": "LEYENDA2", "Type": 1, "InputType": 1, "Length": 7, "Action": 1, "Value": "LEYENDA", "Field": "LEYENDA"}, {"Label": "LEYENDA", "Type": 1, "InputType": 1, "Length": 7, "Action": 1, "Value": "LEYENDA", "Field": "LEYENDA"}]
[2024-11-25 22:26:35:228],FARMAX_WSC,DATALOGICSERVICE,INFO,,V15531000002994 :: Datos de la transacción exitosa datalogic IDGRUPO=4|IDCAJERA=1|IDTIENDA=1553|IDPOS=10|IDCAJERO=3|FECHALOCAL=25/11/2024|HORALOCAL=22:25:48|TRANSACCION=1553102994|SKU=8578690100208|COMISION=0.00|REFERENCIA=DB381C1D700FAF4E6CDB05|MONTO=20|FECHA=25/11/2024 10:26:31 p. m.|AUTORIZACION=102245|LEYENDA2=Dudas tiempo aire marca *050 o 018003330050|LEYENDA=Duda s tiempo aire marca *050 o 018003330050|transacción de éxito
[2024-11-25 22:26:35:237],FARMAX_WSC,DATALOGICSERVICE,INFO,,##### DoTransactionProcessing V15531000002994 result={"Folio": "102245", "Amount": 20.00, "TicketMessage": [{"Label": "Transacción", "Value": "1553102994", "Type": 2}, {"Label": "FECHA", "Value": "25/11/2024 10:26:31 p. m.", "Type": 1}, {"Label": "TRANSACCION", "Value": "1553102994", "Type": 1}, {"Label": "", "Value": "", "Type": 1}, {"Label": "", "Value": "", "Type": 1}, {"Label": "", "Value": "", "Type": 1}], "Message": "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, FechTransaccion, estaActivo from trnTransaccionesCab with (nolock) where FolioTransaccion = 'V15531000002994';					
1 idTransaccion	folioTransaccion	subTotal	codigoMovimientoCaja	fechTransaccion	estaActivo
1 7387360A-ADAB-EF11-9692-94BB432CF0D5	V15531000002994	17.2413	217BC382-AEAB-EF11-9692-94BB432CF0D5	2024-11-25 22:25:48.983	true
select codigoMovimientoCaja, importePago, estatusFormaPago from genMovimientosCajasDet with (nolock)					
where codigoMovimientoCaja in (select codigoMovimientoCaja from trnTransaccionesCab with (nolock) where folioTransaccion = 'V15531000002994');					
1 codigoMovimientoCaja	importePago	estatusFormaPago			
1 217BC382-AEAB-EF11-9692-94BB432CF0D5	20.0000 A				

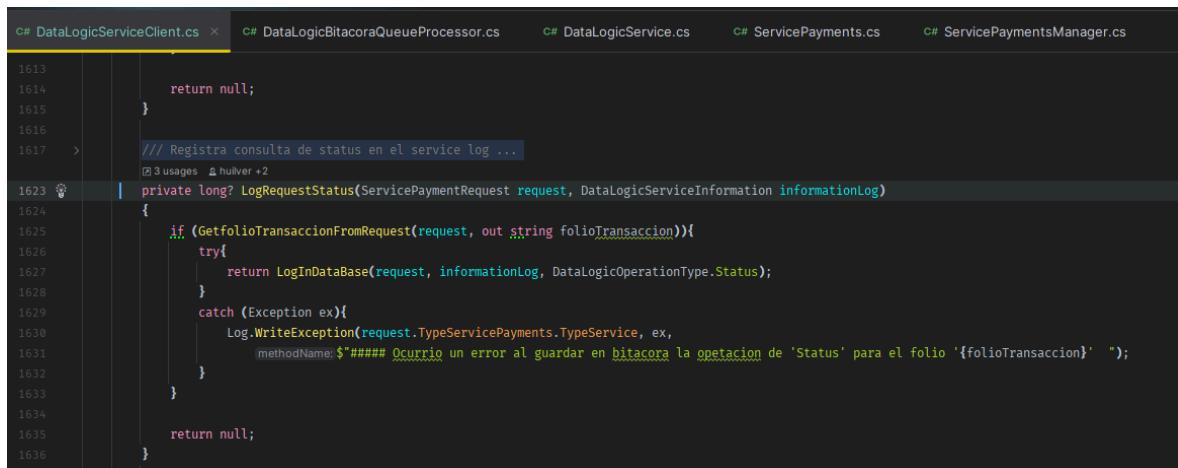
Se revisa la bitácora de operaciones

select idBitacoraServicios, folioTransaccion, numeroAutorizacion, fechaSolicitud, fechaRespuesta, tipoOperacion, status from venPagoServiciosBitacoraReg order by idBitacoraServicios desc						
1 idBitacoraServicios	folioTransaccion	numeroAutorizacion	fechaSolicitud	fechaRespuesta	tipoOperacion	status
1 119 V15531000002994	102245	2024-11-25 22:26:34.917	2024-11-25 22:26:35.210	I	C	
2 118 V15531000002994	102245	2024-11-25 22:26:34.223	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.



```

1613
1614     return null;
1615 }
1616 >     /// Registra consulta de status en el service log ...
1617     private long? LogRequestStatus(ServicePaymentRequest request, DataLogicServiceInformation informationLog)
1618     {
1619         if (GetfolioTransaccionFromRequest(request, out string folioTransaccion))
1620         {
1621             try
1622             {
1623                 return LogInDataBase(request, informationLog, DataLogicOperationType.Status);
1624             }
1625             catch (Exception ex)
1626             {
1627                 Log.WriteLine(request.TypeServicePayments.TypeService, ex,
1628                               methodName: $"#### Ocurrio un error al guardar en bitacora la operacion de 'Status' para el folio '{folioTransaccion}' ");
1629             }
1630         }
1631         return null;
1632     }
1633 }
1634
1635 }
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2279
2280
2281
2282
2283
2284
2285
2286
2287
2287
2288
2289
2289
2290
2291
2292
2293
2294
2295
2296
2297
2297
2298
2299
2299
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2379
2380
2381
2382
2383
2384
2385
2386
2387
2387
2388
2389
2389
2390
2391
2392
2393
2394
2395
2396
2397
2397
2398
2399
2399
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2489
2490
2491
2492
2493
2494
2495
2496
2497
2497
2498
2499
2499
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2589
2590
2591
2592
2593
2594
2595
2596
2597
2597
2598
2599
2599
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2689
2690
2691
2692
2693
2694
2695
2696
2697
2697
2698
2699
2699
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2789
2790
2791
2792
2793
2794
2795
2796
2797
2797
2798
2799
2799
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2889
2890
2891
2892
2893
2894
2895
2896
2897
2897
2898
2899
2899
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2989
2990
2991
2992
2993
2994
2995
2996
2997
2997
2998
2999
2999
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3069
3070
3071
3072
3073
3074
3075
3076
3077
3078
3079
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3089
3089
3090
3091
3092
3093
3094
3095
3096
3097
3097
3098
3099
3099
3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
3109
3109
3110
3111
3112
3113
3114
3115
3116
3117
3118
3119
3119
3120
3121
3122
3123
3124
3125
3126
3127
3128
3129
3129
3130
3131
3132
3133
3134
3135
3136
3137
3138
3139
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148
3149
3149
3150
3151
3152
3153
3154
3155
3156
3157
3158
3159
3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3179
3180
3181
3182
3183
3184
3185
3186
3187
3188
3189
3189
3190
3191
3192
3193
3194
3195
3196
3197
3197
3198
3199
3199
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3209
3210
3211
3212
3213
3214
3215
3216
3217
3218
3219
3219
3220
3221
3222
3223
3224
3225
3226
3227
3228
3229
3229
3230
3231
3232
3233
3234
3235
3236
3237
3238
3239
3239
3240
3241
3242
3243
3244
3245
3246
3247
3248
3249
3249
3250
3251
3252
3253
3254
3255
3256
3257
3258
3259
3259
3260
3261
3262
3263
3264
3265
3266
3267
3268
3269
3269
3270
3271
3272
3273
3274
3275
3276
3277
3278
3279
3279
3280
3281
3282
3283
3284
3285
3286
3287
3288
3289
3289
3290
3291
3292
3293
3294
3295
3296
3297
3297
3298
3299
3299
3299
3300
3301
3302
3303
3304
3305
3306
3307
3308
3309
3309
3310
3311
3312
3313
3314
3315
3316
3317
3318
3319
3319
3320
3321
3322
3323
3324
3325
3326
3327
3328
3329
3329
3330
3331
3332
3333
3334
3335
3336
3337
3338
3339
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3379
3379
3380
3381
3382
3383
3384
3385
3386
3387
3388
3389
3389
3390
3391
3392
3393
3394
3395
3396
3397
3397
3398
3399
3399
3399
3400
3401
3402
3403
3404
3405
3406
3407
3408
3409
3409
3410
3411
3412
3413
3414
3415
3416
3417
3418
3419
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429
3429
3430
3431
3432
3433
3434
3435
3436
3437
3438
3439
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3469
3470
3471
3472
3473
3474
3475
3476
3477
3478
3479
3479
3480
3481
3482
3483
3484
3485
3486
3487
3488
3489
3489
3490
3491
3492
3493
3494
3495
3496
3497
3497
3498
3499
3499
3499
3500
3501
3502
3503
3504
3505
3506
3507
3508
3509
3509
3510
3511
3512
3513
3514
3515
3516
3517
3518
3519
3519
3520
3521
3522
3523
3524
3525
3526
3527
3528
3529
3529
3530
3531
3532
3533
3534
3535
3536
3537
3538
3539
3539
3540
3541
3542
3543
3544
3545
3546
3547
3548
3549
3549
3550
3551
3552
3553
3554
3555
3556
3557
3558
3559
3559
3560
3561
3562
3563
3564
3565
3566
3567
3568
3569
3569
3570
3571
3572
3573
3574
3575
3576
3577
3578
3579
3579
3580
3581
3582
3583
3584
3585
3586
3587
3588
3589
3589
3590
3591
3592
3593
3594
3595
3596
3597
3597
3598
3599
3599
3599
3600
3601
3602
3603
3604
36
```

Además, dado que la bitácora sirve para consultar el estatus de una transacción, se agrega una cola de logs a fin poder registrar la información posteriormente, en caso que en el momento se no se haya logrado

```
420
421 >     /// New logic for processing the service payments that does not keep transaction state in the central WS ...
422     □ 1 usage  □ halver +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
432 >
433
434         try
435         {
436             var requestParameters List<Parameters> = PrepareExecuteRequestParameters(request);
437             List<Parameters> respuestaIntegrador=new List<Parameters>();
438
439             // Guardamos en bd la petición que se enviará al servicio.
440             var generalInformation = MapDataContractToLocalModel(request.TypeServicePayments, requestParameters);
441             var idLog long? = LogRequestExecute(request, generalInformation);
442
443             try{...}
444             catch (Exception ex){...}
445             // default to success
446             var responseCode int = ObtainResponseCodeFromList(respuestaIntegrador, defaultValue: (int)DataLogicResponseCode.SuccessfulProcess);
447
448             if (responseCode == (int)DataLogicResponseCode.SuccessfulProcess){...}
449             else{...}
450
451             LogRequestExecuteForced(idLog, responseCode, request, generalInformation, respuestaIntegrador);
452
453
454     // RESTORE AQUÍ LOS DATOS DE LA TRANSACCIÓN DE SERVICIO AL SERVICIO DE LOS RESPUESTAS
455     □ 1 usage  □ halver +1
456     private void LogRequestExecuteForced(long? idLog, int responseCode, ServicePaymentRequest request, DataLogicServiceInformation generalInformation, List<Parameters> respuestaIntegrador)
457     {
458         generalInformation.ErrorCode = responseCode;
459         generalInformation.LogId = idLog;
460         generalInformation.ResponseParameters = MapDataContractToLocalModel(request.TypeServicePayments, respuestaIntegrador).RequestParameters;
461         long? idBitacora=LogRequestExecute(request, generalInformation);
462         if (idBitacora == null){
463             _ = DataLogicQueueManager.GetInstance().AddElementInQueue(DataLogicBitacoraQueue, new DataLogicBitacoraQueueData(request,generalInformation),maxSizeQueue:2000);
464         }
465     }
466 }
```

Pruebas

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

Número

Confirmar

Formas de Pago

<input type="checkbox"/> E	EFFECTIVO (CA)	<input type="text" value="20.00"/>
<input type="checkbox"/> T	TARJETA BANCARIA	<input type="text" value="0.00"/>

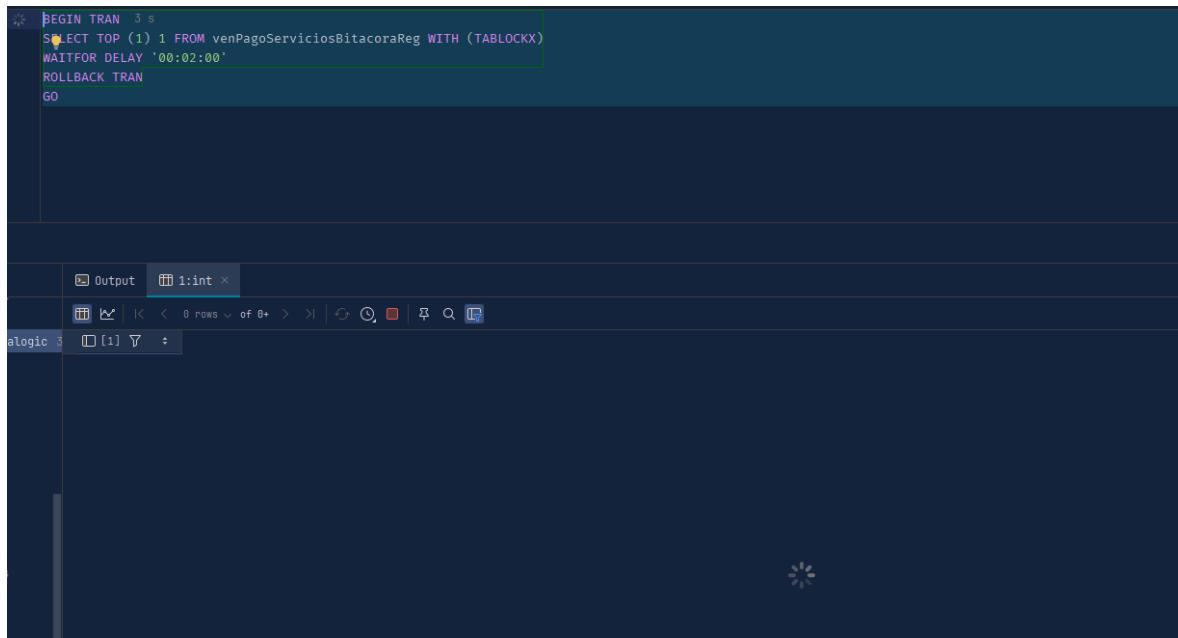
Subtotal: **\$17.24**

I.V.A: **\$2.76**

Total: **\$20.00**

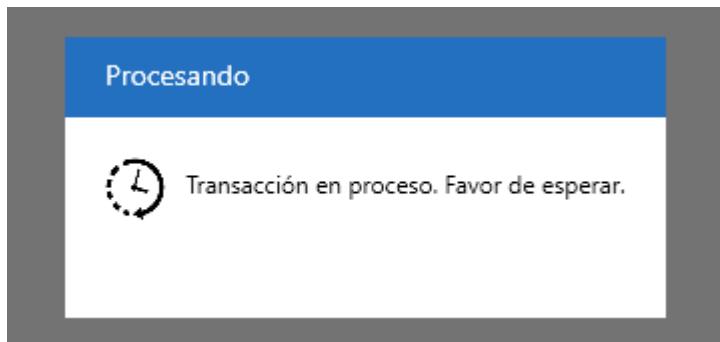
 Tu Donativo \$0.00	 Tu Ahorro \$0.00	Por Pagar \$0.00
--	--	----------------------------

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





Revisamos el log del POS, en el que se puede ver que se lanzó la transacción y se recibió una 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={"Type5  
servicePayments":{"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-ef11-9692-94b8432cf0d5","SupportsReverse":true,"PrintsTicketNoAuth":true,"ProductCode":758778193,"RelatedCode":8578690100208  
,"ExternalCode":"N/A","ProductServiceCode":758778193,"FillId":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":"ComisionAmount","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}]}}  
[2024-11-25 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 :
| F50FA45C-B9AB-EF11-9692-94BB432CF805 V15531000002995 17.2413 3B18FD7A-B9AB-EF11-9692-94BB432CF805 2024-11-25 23:44:28.313 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-B9AB-EF11-9692-94BB432CF805 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 'V1553100002995' 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 folio 'V1553100002995' ha sido reversado previamente. - FAhorro.POS.Application.Core.DbConnectionException: Se perdió la conexión con 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 folio 'V1553100002995' tiene una solicitud pendiente previa, con mas de 50 segs. - FAhorro.POS.Application.Core.DbConnectionException: Se perdió la conexión con la base de datos.
[2024-11-25 23:44:50:757],FARMAX_WSC,DATALOGICSERVICE,INFO,,V1553100002995 treatAsPending=True
[2024-11-25 23:45:00:921],FARMAX_WSC,DATALOGICSERVICE,ERROR,[WRITEERROR],,(thread=130) ##### Ocurrio un error al guardar en bitacora la operacion de 'Execute' para el folio 'V1553100002995' - FAhorro.POS.Application.Core.DbConnectionException: Se perdió la conexión 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 bitacora la operacion de 'Status' para el folio 'V1553100002995' - FAhorro.POS.Application.Core.DbConnectionException: Se perdió la conexión con la base de datos.
[2024-11-25 23:45:10:942],FARMAX_WSC,DATALOGICSERVICE,INFO,,V1553100002995 consulta...
[2024-11-25 23:45:21:460],FARMAX_WSC,DATALOGICSERVICE,ERROR,[WRITEERROR],,(thread=40) ##### Ocurrio un error al guardar en bitacora la operacion de 'Status' para el folio 'V1553100002995' - FAhorro.POS.Application.Core.DbConnectionException: Se perdió la conexión con la base de datos.
[2024-11-25 23:45:21:468],FARMAX_WSC,DATALOGICSERVICE,INFO,,V1553100002995 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,,V1553100002995 (FAhorro.POS.Services.DataLogic.helper.PxUniversalSoapClientFake) ejecuta...
[2024-11-25 23:45:22:229],FARMAX_WSC,DATALOGICSERVICE,INFO,,V1553100002995 ::: 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|COMISION=0.00|REFERENCIA=DB301C1D70FAF4E6CDB85|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 bitacora la operacion de 'Execute' para el folio 'V1553100002995' - FAhorro.POS.Application.Core.DbConnectionException: Se perdió la conexión con la base de datos.
[2024-11-25 23:45:32:285],FARMAX_WSC,DATALOGICSERVICE,INFO,,##### DoTransactionProcessing V1553100002995 result={"Folio":"102245","Amount":20.0,"TicketMessage":[{"Label":"Transacción","Value":"1553102995","Type":2}, {"Label":"","Value":"UNEFON","Type":1}, {"Label":"","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 falló 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 'V1553100002995'
[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 'V1553100002995' - 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,, 'V1553100002995'. Se reencola la bitacora debido a que falló 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 'V1553100002995'
```

FARMAX_CENTRAL.dbo.v_oServiciosBitacoraReg						
	numeroAutorizacion	fechaSolicitud	fechaRespuesta	tipoOperacion	status	datosRespuesta
1	102245	2024-11-25 23:46:01.870	<null>	E	P	Type="Unspecified" Encrypted="false">>102245</Param><Param Name="MONTO" FieldClass="Text">100000</Param>

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 reprocesso

```

    public Task<ServicePaymentResponse> ProcessPaymentServiceStateless(ServicePaymentRequest request)
    {
        ...
        if(ManagePreviousRequest(request,out ServicePaymentResponse servicePaymentResponse, out bool hasPendingRequest)){...}

        try
        {
            var requestParameters:List<Parameters> = PrepareExecuteRequestParameters(request);
            List<Parameters> respuestaIntegrador=new List<Parameters>();

            // Guardamos en bd la petición que se enviará al servicio.
            var generalInformation = MapDataContractToLocalModel(request.TypeServicePayments, requestParameters);
            var idLog:long = LogRequestExecute(request, generalInformation);

            try
            {
                if (hasPendingRequest){
                    List<Parameters> statusResponse = await GetTransactionStatus(request, requestParameters);

                    if (statusResponse == null){
                        return await TryReverseIfApply(request); //HNA:Si hay una request pendiente pero no se puede determinar su estado, se intenta la reversa
                    }

                    if (IsSuccessfulResponse(statusResponse)){...}
                    else{...}
                }
                else{
                    // No pending request. call ejecuta. This the normal stuff
                    respuestaIntegrador = await InvokeExecuteTransactionLong(requestParameters, request);
                }
            }
            catch (Exception ex){
                Log.WriteLine(request.TypeServicePayments.TypeService, ex, methodName: $"{request.TypeServicePayments.FolioTransaction} excepción ejecuta", resourceName:string.Empty, resource:null);

                if (ContainsServiceException(ex, typeof{EndpointNotFoundException})||ContainsServiceException(ex, typeof{TimeoutException})){
                    if (ContainsServiceException(ex, typeof{EndpointNotFoundException})&&hasPendingRequest){...}

                    List<Parameters> statusResponse = await GetTransactionStatus(request, requestParameters);
                    if (statusResponse == null){//HNA:No se puede determinar su estado, se intentará la reversa y de ser el caso se encolará un reverso,
                        //pero al tiempo indicaremos que el pos también lo debe realizar para garantizar que se realice si o si;
                    }
                }
            }
        }
        return await TryReverseIfApply(request);
    }

```

```

    private async Task<ServicePaymentResponse> TryReverseIfApply(ServicePaymentRequest request)
    {
        if (request.TypeServicePayments.SupportsReverse){
            Log.WriteLine(request.TypeServicePayments.TypeService, message: $"{request.TypeServicePayments.FolioTransaction} reversa...");

            try{
                var reverseResponse = await this.CancelPaymentService(request, DataLogicResponseCode.TimeOutEmisor, isCancelSale:false);
                if(IsReversedAccordingToResponse(reverseResponse))
                {
                    return ServiceResponseFactory.GetErrorResponseCancelledAndReversed();
                }
            }
            catch (Exception rex){
                Log.WriteLine(request.TypeServicePayments.TypeService, rex,
                methodName: $"{request.TypeServicePayments.FolioTransaction} excepción reversa", resourceName:string.Empty, resource:null);

                Log.WriteLine(request.TypeServicePayments.TypeService, message: $"{request.TypeServicePayments.FolioTransaction} Encolando reverso");
                await DataLogicQueueManager.GetInstance().AddElementInQueue(DataLogicReverseQueue,new DataLogicReverseQueueData(request, DataLogicResponseCode.TimeOutEmisor, isCancelSale:false),maxSizeQueue:1000);
            }
        }
    }

```

```
# DataLogicServiceClient.cs # DataLogicReverseQueueProcessor.cs # ServicePaymentRequest.cs # DataLogicBitacoraQueueProcessor.cs # DataLogicService.cs # ServicePayment
47 Log.WriteLine(Submodule, message: $"#### Un proceso anterior aun no termina su ejecucion, por lo que se intenta a la proxima ejecucion");
48 }
49 }
50 }
51
52 TI usage & halver
53
54 private void ProcesaReversos()
55 {
56     int actualSize = _manager.Count(DataLogicServiceClient.DataLogicReverseQueue);
57     if (actualSize > 0){
58         Log.WriteLine(Submodule, message: $"#### procesando '{actualSize}' reversos");
59     }
60
61     for (int i = 0; i < actualSize; i++){
62         var task = Task.Run(async () =>
63             await _manager.DequeueElementOfQueue<DataLogicReverseQueueData>(DataLogicServiceClient.DataLogicReverseQueue);
64
65         var reverseToProcess = task.Result;
66
67         if (reverseToProcess != null){
68             if (reverseToProcess.NumIntentos < MaxNumIntentos){
69                 reverseToProcess.NumIntentos += 1;
70
71                 var request = reverseToProcess.Request;
72                 string folioTransaction = request.TypeServicePayments.FolioTransaction;
73                 Log.WriteLine(Submodule, message: $"#### procesando reverso de {folioTransaction}");
74
75                 Task.Run(async () =>
76                 {
77                     try{
78                         using (var client = new DataLogicServiceClient()){
79                             var resultReverse :ServicePaymentResponse = await client.CancelPaymentService(request, reverseToProcess.ReasonCode, reverseToProcess.IsCancelSale);
80                         if (resultReverse != null && resultReverse.Result)
81                             Log.WriteLine(Submodule, message: $"#### Reverso exitoso de {folioTransaction} con resultado {resultReverse.Result}");
82                         else
83                             Log.WriteLine(Submodule, message: $"#### Reverso fallido de {folioTransaction} con resultado {resultReverse.Result}");
84                     }
85                 }
86             }
87         }
88     }
89 }
90 }
```

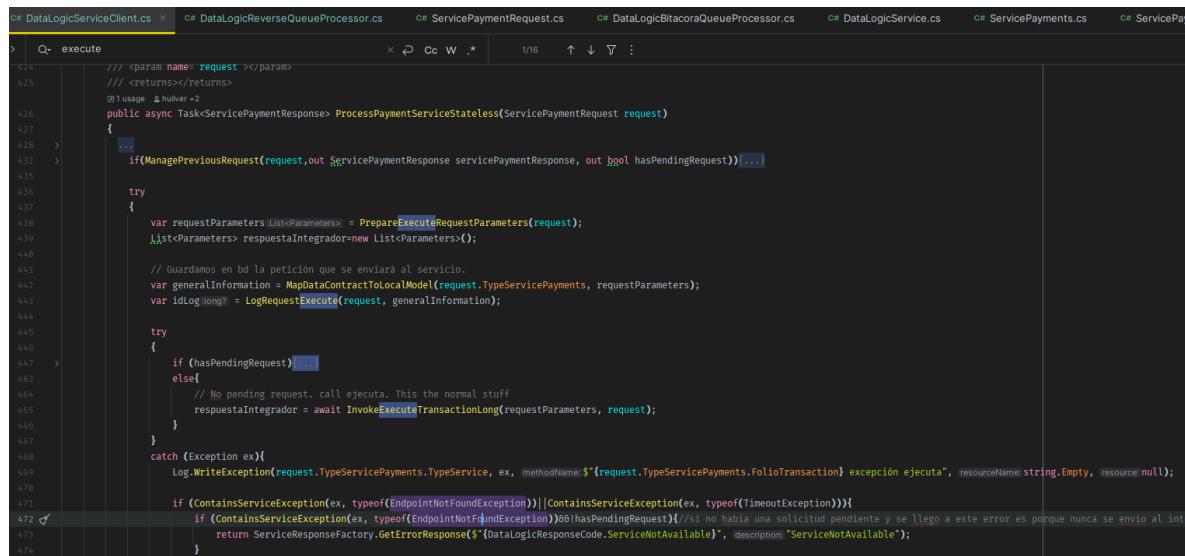
Pruebas

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.



```

424     /// <param name="request" />
425     /// <returns></returns>
426     [Usage("Δ hulver +2")]
427     public async Task<ServicePaymentResponse> ProcessPaymentServiceStateless(ServicePaymentRequest request)
428     {
429         ...
430         if(ManagePreviousRequest(request,out ServicePaymentResponse servicePaymentResponse, out bool hasPendingRequest)){...}
431         ...
432         try
433         {
434             var requestParametersList<Parameters> = PrepareExecuteRequestParameters(request);
435             List<Parameters> respuestaIntegrador=new List<Parameters>();
436             // Guardamos en bd la petición que se enviará al servicio.
437             var generalInformation = MapDataContractToLocalModel(request.TypeServicePayments, requestParameters);
438             var idLogLong = LogRequestExecute(request, generalInformation);
439             try
440             {
441                 if (hasPendingRequest){...}
442                 else
443                     // No pending request. call ejecuta. This the normal stuff
444                     respuestaIntegrador = await InvokeExecuteTransactionLong(requestParameters, request);
445             }
446             catch (Exception ex){
447                 Log.WriteException(request.TypeServicePayments.TypeService, ex, methodName: $"{request.TypeServicePayments.FolioTransaction} excepción ejecuta", resourceName: string.Empty, resource: null);
448                 if (ContainsServiceException(ex, typeof(EndpointNotFoundException))||ContainsServiceException(ex, typeof(TimeoutException))){
449                     if (ContainsServiceException(ex, typeof(EndpointNotFoundException))&&hasPendingRequest){//si no había una solicitud pendiente y se llegó a este error es porque nunca se envió al integrador
450                         return ServiceResponseFactory.GetErrorResponse($"{DataLogicResponseCode.ServiceNotAvailable}", description: "ServiceNotAvailable");
451                     }
452                 }
453             }
454         }
455     }

```

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" />

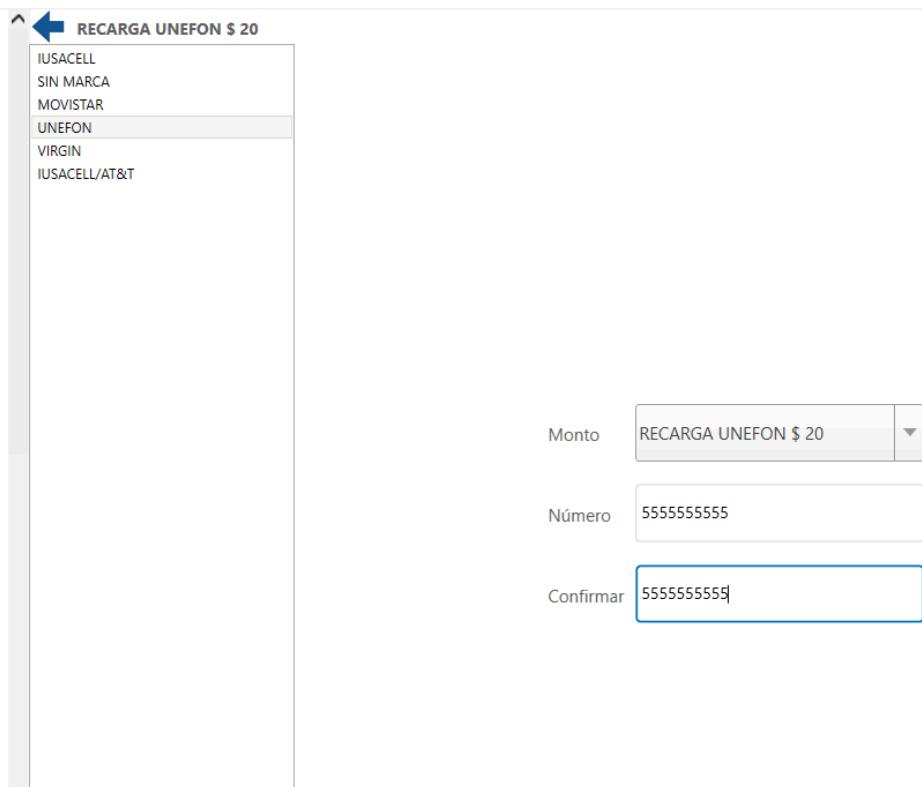
<add key="TimeOutStatus" value="10" />
<add key="TriesToCompleteStatus" value="1" />
<add key="EndpointNotFoundExceptionStatus" value="false" />

<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: 5555555555

Confirmar: 5555555555

orro

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

Formas de Pago

<input type="checkbox"/> EFECTIVO (CA) 20.00	<input type="checkbox"/> 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

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

Bienvenido a Farmacias del Ahorro
Te Atiende FABIOLA

Formas de Pago

<input type="checkbox"/> EFECTIVO (CA) 20.00	<input type="checkbox"/> TARJETA BANCARIA 0.00	
--	--	--

× 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**

```
[-$] > tail -f -n 200 20241126_FARMAX_POS_DATALOGICSERVICE.XLOG | egrep "\w{4}-\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={"TypeServicePayments":{"Brand":"UNEFON","TypeService":"DataLogicService","BusinessUnit":1553,"UserCode":3,"EncryptReference":true,"FolioTransaction":"V15531000002996","OperationDate":"2023-04-13T00:00:00","TransactionDate":"2024-11-26T01:00:09","ShiftId":"b66c518c-c2ab-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","IPAddresses":"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":"ComisionAmount","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 01:00:11:242],FARMAX_POS,DATALOGICSERVICE,INFO,[LOG],##### ExecuteServiceTransaction V15531000002996 result={"DateCancel": "", "DateTransaction": "", "Folio": "", "CodeError": "1001", "DescriptionError": "Código 72: ServiceNotAvailable", "Store": "", "ServicePayed": "", "TicketMessage": []}
[2024-11-26 01:00:20:917],FARMAX_POS,DATALOGICSERVICE,INFO,[LOG],##### ExecuteServiceTransaction V15531000002996 action CancelSale
```

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
1898A604-C4AB-EF11-9692-94BB432CF0D5	V15531000002996	17.2413	DB03BC12-C4AB-EF11-9692-94BB432CF0D5	2024-11-26 01:00:20.970
select codigoMovimientoCaja, importePago, estatusFormaPago from genMovimientosCajasDet with (nolock) where codigoMovimientoCaja in (select codigoMovimientoCaja from trnTransaccionesCab with (nolock) where folioTransaccion = 'V15531000002996');				
codigoMovimientoCaja	importePago	estatusFormaPago		
DB03BC12-C4AB-EF11-9692-94BB432CF0D5	20.0000 C			

Se revisan los 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

```
[-$] > tail -f -n 200 20241126_FARMAX_WSC_DATALOGICSERVICE.XLOG | egrep "\w{4}-\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.PxUniversalSoapClientFake) ejecuta...
2024-11-26 01:00:10:360],FARMAX_WSC,DATALOGICSERVICE,ERROR,[WRITERERROR],(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=ServiceNotAvailable)
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 después una operación de reversa la cual fue exitosa

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

FARMAX_CENTRAL.dbo.v_oServiciosBitacoraReg 2						
idBitacoraServicios	folioTransaccion	numeroAutorizacion	FechaSolicitud	FechaRespuesta	tipOperacion	status
1	122	V15531000002996	<null>	2024-11-26 01:00:10.390	2024-11-26 01:00:11.217	R
2	121	V15531000002996	<null>	2024-11-26 01:00:10.213	<null>	E

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

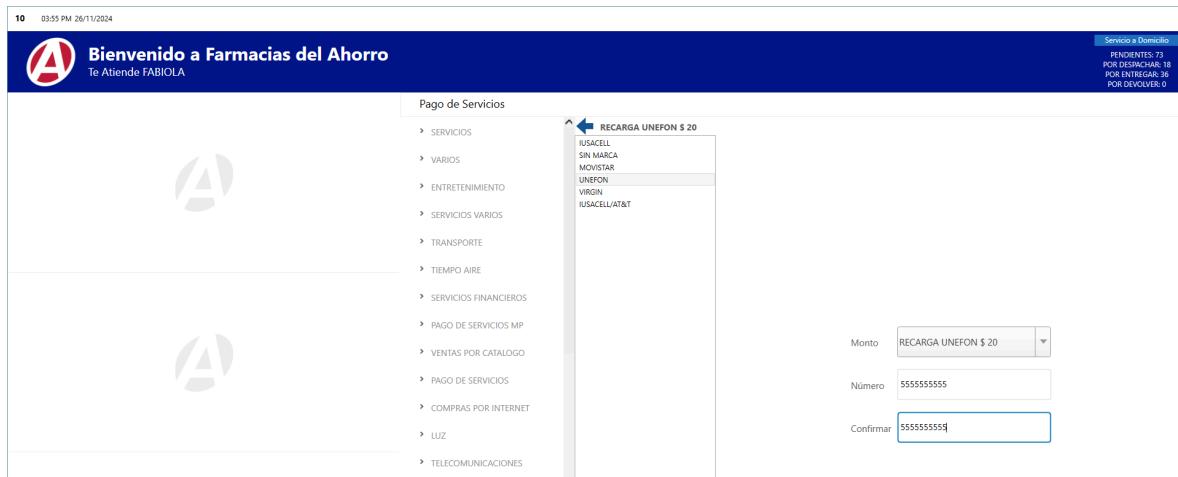
HappyPath

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





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

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:
EFECTIVO (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={"TypeServicePayments":{"Brand":"UNEFON","TypeService":"DataLogicService","BusinessUnit":1553,"UserCode":3,"EncryptReference":true,"FolioTransaction":"V15531000003004","OperationDate":"2023-04-13T00:00:00","TransactionDate":"2024-11-26T15:57:39","ShiftId":b66c518c-c2ab-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,"IPAddresses":"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":"ComisionAmount","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 15:57:41:201],FARMAX_POS,DATALOGICSERVICE,INFO,[LOG],##### ExecuteServiceTransaction V15531000003004 result={"Amount":20.0,"DateCancel":null,"DateTransaction":null,"Folio":102245,"Store":null,"ServicePayed":null,"TicketMessage":[{"Label":"Transacción","Value":1553103004,"Type":2},{"Label":null,"Value":UNEFON,"Type":1}, {"Label":null,"Value":Dudas tiempo aire marca *050 o 018003330050,"Type":1}]}[2024-11-26 15:57:41:207],FARMAX_POS,DATALOGICSERVICE,INFO,[LOG],##### ExecuteServiceTransaction V15531000003004 action CompletePayment
```

Se revisa el estado de la venta y del pago, en 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';												
✗	<table border="1"> <tr> <td>idTransaccion</td><td>: folioTransaccion</td><td>: subTotal</td><td>: codigoMovimientoCaja</td><td>: fechaTransaccion</td><td>: estaActivo</td></tr> <tr> <td>C987EC62-41AC-EF11-9692-94B8432CF0D5</td><td>V15531000003004</td><td>17.2413</td><td>CEC1D973-41AC-EF11-9692-94B8432CF0D5</td><td>2024-11-26 15:57:39.890</td><td>* true</td></tr> </table>	idTransaccion	: folioTransaccion	: subTotal	: codigoMovimientoCaja	: fechaTransaccion	: estaActivo	C987EC62-41AC-EF11-9692-94B8432CF0D5	V15531000003004	17.2413	CEC1D973-41AC-EF11-9692-94B8432CF0D5	2024-11-26 15:57:39.890	* true
idTransaccion	: folioTransaccion	: subTotal	: codigoMovimientoCaja	: fechaTransaccion	: estaActivo								
C987EC62-41AC-EF11-9692-94B8432CF0D5	V15531000003004	17.2413	CEC1D973-41AC-EF11-9692-94B8432CF0D5	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');													
✗	<table border="1"> <tr> <td>codigoMovimientoCaja</td><td>: importePago</td><td>: estatusFormaPago</td></tr> <tr> <td>CEC1D973-41AC-EF11-9692-94B8432CF0D5</td><td>20.0000 A</td><td></td></tr> </table>	codigoMovimientoCaja	: importePago	: estatusFormaPago	CEC1D973-41AC-EF11-9692-94B8432CF0D5	20.0000 A							
codigoMovimientoCaja	: importePago	: estatusFormaPago											
CEC1D973-41AC-EF11-9692-94B8432CF0D5	20.0000 A												

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

```
[2024-11-26 15:57:40:139],FARMAX_WSC,DATALOGICSERVICE,INFO,,V1553100003004 treatAsPending=False
[2024-11-26 15:57:40:317],FARMAX_WSC,DATALOGICSERVICE,INFO,,V1553100003004 (FAhorro.POS.Services.DataLogic.helper.PxUniversalSoapClientProxy) ejecuta...
[2024-11-26 15:57:41:034],FARMAX_WSC,DATALOGICSERVICE,INFO,,V1553100003004 ::: Datos de la transaccion exitosa datalogic IDGRUPO=4|IDCADENA=1|IDTIENDA=1553|IDPOS=10|IDCAJERO=3|FECHALOCAL=26/11/2024|HORALOCAL=15:57:39|TRANSACCION=1553103004|SKU=8578690100208|COMISION=0.00|REFERENCIA=DB301C1D70DFA4E6CDB05|MONTO=20|PROVEEDOR=UNEFON|AUTORIZACION=102245|LEYENDA=Dudas tiempo aire marca *050 o 018003330050 |
[2024-11-26 15:57:41:201],FARMAX_WSC,DATALOGICSERVICE,INFO,,##### DoTransactionProcessing V1553100003004 result={"Folio":"102245","Amount":20.0,"TicketMessage":[{"Label":"Transacción","Value":"1553103004","Type":2}, {"Label":"","Value":"UNEFON","Type":1}, {"Label":"","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 completa la llamada con datos de respuesta