**Proyecto Route Lite**

Especificación del Caso de Uso: Seleccionar Actividades para la Visita- CUROLMOV11

VERSIÓN 1.28

**HISTÓRICO DE REVISIÓN**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
| 05/03/2012 | 1.0 | Elaboración de la especificación del caso de uso Seleccionar Actividades para la Visita para el Proyecto Route Lite de acuerdo al Estándar de Casos de Uso 1.3  Folio CAI 000 2019 | Nancy Villalobos |
| 28/11/2012  17/01/2013 | 1.1 | Crear nueva actividad para Cambios y Cambios Automáticos de Producto.  Folio CAI 0002443  Crear nueva actividad para Devoluciones de Producto del Cliente.  Folio CAI 0002477  (Route Lite, 1.0.0.0) | Belem Jiménez  Belem Jiménez |
| 10/05/2014 | 1.2 | Crear actividad de Venta a Consignación para permitir crear, modificar, eliminar o liquidar una venta a consignación.  Folio CAI 0003016  (Route Lite, 1.1.0.0) | Belem Lizeth Jiménez Arévalo |
| 21/05/2014 | 1.3 | Folio CAI 0002774  Se validará el Límite de Crédito del cliente, mandando el mensaje en caso de que esté excedido, pero permitiendo realizar la venta de cualquier manera.  (Route Lite, 1.1.0.0) | Belem Lizeth Jiménez Arévalo |
| 13/06/2014 | 1.4 | Folio CAI 0003108  Validar Cuadre de Envase al Terminar la Visita en Reparto:  - Si se le puede realizar préstamo de envase al cliente <Cliente.Prestamo = 1> y si el parámetro de configuración indica que se debe validar el cuadre de envase <MOTConfiguracion.CuadreDeEnvase = 1>, validar que el saldo de envase sea igual o menor al que se tenía al iniciar la visita.  - Presentar mensaje si es mayor, y no permitir salir de la Visita.  (Route Lite, 1.1.0.0) | Belem Lizeth Jiménez Arévalo |
| 23/06/2014 | 1.5 | Folio CAI 0003117  Al momento de finalizar la Visita, solicitar la lectura del código de barras del cliente a fin de poder continuar con las validaciones y acciones correspondientes a la Terminación de la Visita, no permitir continuar si no se lee.  (Route Lite, 1.1.0.0) | Belem Lizeth Jiménez Arévalo |
| 22/07/2014 | 1.6 | Folio CAI 0003138  Crear la actividad de Facturación Electrónica incluyendo únicamente la funcionalidad correspondiente a la versión 3.2 y a la Prefactura (versión 1.0).  No incluir la sección del Complemento para la generación de Addendas y validar que exista por lo menos un método de pago configurado al cliente a fin de permitir crear la factura, por lo que la captura del método de pago al momento de crear la factura (cuando el cliente no cuenta con uno configurado) no será incluida en Android.  (Route Lite, 1.1.0.0) | Belem Lizeth Jiménez Arévalo |
| 22/07/2014 | 1.7 | Folio CAI 0003136  Se realizará una nueva actividad de Toma de Inventario y Mercadeo, de acuerdo a la actividad ya realizada para WM  (Route Lite, 1.1.0.0) | Nancy Elizabeth Villalobos Plascencia |
| 10/12/2014 | 1.8 | Si al terminar la visita, esta tuvo movimientos y se tiene una improductividad de Visita capturada, se le da update para poner tipoMotivo = null y comentarios = null.  Por otro lado, si se hicieron ventas y se tenía capturada una improductividad de venta, ésta se elimina.  Folio CAI 0003360  (La Costeña, 1.2.0.0) | Belem Lizeth Jiménez Arévalo |
| 24/04/2015 | 1.9 | Se deberá validar el parámetro 'Forzar Captura Improductividad' de MOTConfiguracion para forzar la captura de Improductividad de venta o Visita tanto al terminar la visita para los clientes a los que no se les haya realizad ningún movimiento durante ésta, como al cerrar la jornada para los clientes no visitados que se encuentran en agenda.  Folio CAI 0003513  (Disposur, 1.3.0.0) | Belem Lizeth Jiménez Arévalo |
| 29/02/2016 | 1.10 | Folio CAI:0003829  \*Se agrega Parámetro General en la tabla ConfigParametro para verificar si se realiza la validación de Actualización de Precio o no. \*Se realizará la validación siempre y cuando el parámetro este activo y el día de trabajo no sea igual al día actual, para ello tomar en el XML de la tabla PrecioProductoVig la fecha de la última actualización, dicha validación será al momento de levantar cada Pedido.  **Se realiza en la versión 4.9.3.0 y se replica en la 4.10.0.0 y la 4.11.0.0 Se realiza en la versión 1.4.3.0 y se replica en la 1.5.0.0 y 1.6.0.0 (Android)** | Nancy Elizabeth Villalobos Plascencia |
| 13/05/2016 | 1.11 | 1. Si se tiene configurado el aseguramiento de visita para solicitar la lectura del código de barras del cliente al inicio y final de la visita <AseguramientoVisita.TipoAseguramiento = 4 ó 5> y si no fue posible leer el código de barras para iniciar la visita <LecturaCodigo = Falso>, al terminar la visita ya no solicitar el código del cliente para permitir salir.  2. Si se tiene configurado el aseguramiento de visita para solicitar la lectura del código de barras del cliente tanto al inicio como al final de la visita <AseguramientoVisita.TipoAseguramiento = 1, 4 ó 5> y también se tiene configurado algún tipo de contraseña para el inicio de la misma <AseguramientoVisita.TipoContraseña <> 0>, inhabilitar el campo de texto que permite la captura manual del código de barras del cliente al iniciar y finalizar la visita.  Folio CAI 0003901  (Lechera GDL, 1.7.0.0 – 4.12.0.0) | Belem Lizeth Jiménez Arévalo |
| 03/06/2016 | 1.12 | Folio CAI 0003896  Agregar terminar e iniciar visita con lector de NFC, además este debe validarse con checks en actividades de la terminal (Aseguramiento de visita).  (Lechera GDL, 1.7.1.0 – 4.12.1.0) | Belem Lizeth Jiménez Arévalo |
| 20/07/2016 | 1.13 | Folio CAI: 0003951  Replicar actividad de Mercadeo que se encuentra en Windows Mobile con los requerimientos solicitados por Quesería Dos Lagunas, conservando la actividad para poder utilizarla de manera general para cualquier cliente  **Se realiza en la versión 1.8.0.0** | Nancy Elizabeth Villalobos Plascencia |
| 13/12/2016 | 1.14 | Folio CAI: 0004068  Se replicará la actividad existente en WM de Solicitudes, utilizando la misma Tabla y la misma lógica.  **Se realiza en la versión 1.9.0.0** | Nancy Elizabeth Villalobos Plascencia |
| 11/05/2017 | 1.15 | - Realizar el Formato de Nota de Venta correspondiente al Grupo: TERMINAR VISITA, REPORTEM = 24, en WM.  - Se agrega funcionalidad al momento de finalizar la visita para que en caso de que existan reportes (tickets) configurados en el grupo “TerminarVisita” del valor por referencia REPORTEA, se manden a imprimir.  Folio CAI 0004172  (eRoute, 1.11.0.0) | Belem Lizeth Jiménez Arévalo |
| 17/11/2017 | 1.16 | Folio CAI 0004432  Se registran en la nueva tabla “VisitaModificada” las visitas que fueron enviadas al servidor y se eliminan al final porque no se realizaron movimientos dentro de ellas. Esto debido a que dichas visitas se crearon porque se entró a realizar el envío y la recepción del timbrado para las facturas desde la Facturación Electrónica, por ello fueron enviadas y por lo tanto es necesario eliminar dichas visitas del servidor al realizar el envío de información. También se registran las visitas que fueron enviadas y no tienen una fecha y hora de fin de visita.  (ByFa, 1.14.0.0 - 4.19.0.0) | Belem Lizeth Jiménez Arévalo |
| 15/06/2018 | 1.17 | Folio CAI: 0004893  Hacer el envío parcial en automático una vez que finalice la visita del cliente el vendedor  Folio CAI: 0004891  Si al entrar a los clientes, existe alguno con una visita cuya fecha inicial y fecha final sea igual (significa que se salieron con otro método), ingrese forzosamente a esa visita a que estén los 5 minutos y la terminen correctamente  (FLI, 1.16.0.0 – 4.21.0.0) | Nancy Elizabeth Villalobos Plascencia  Nancy Elizabeth Villalobos Plascencia |
| 20/06/2018 | 1.18 | Folio CAI 0004896  Limitación a nivel de Cliente de la Visualización de Módulos en el Móvil:  - Crear la nueva tabla ClienteMMDExcepcion. La tabla contendrá los registros de los módulos que NO se mostraran al cliente. - Se crea un grid o pestaña nueva en clientes para llenar la nueva tabla.  - Se crea una nueva interfaz para el llenado de ClienteMMDExcepcion. - En Android al entrar al cliente, se lee la nueva tabla para saber si hay que ocultar una actividad que esté configurada como excepción.  (BYDSA, 1.16.0.0 - 4.21.0.0) | Belem Lizeth Jiménez Arévalo |
| 22/06/2018 | 1.19 | Folio CAI: 0004895  Se mostrará un MsgBox por cada mensaje al entrar al cliente y una vez mostrado ya no volverá a aparecer nuevamente, por lo que se marcará como Leído  (PAN, 1.16.0.0 – 4.21.0.0) | Nancy Elizabeth Villalobos Plascencia |
| 11/07/2018 | 1.20 | Folio CAI: 0004086  Módulo de Histórico de Ventas  **Ser realiza en la versión 4.15.0.0 (Route) y 1.10.0.0** | Nancy Elizabeth Villalobos Plascencia |
| 07/09/2018 | 1.21 | Folio CAI 0005047  Al finalizar la visita, en caso de existir el parámetro especificado en el folio 0005046 y éste tenga esquemas de clientes asignados como valor, validar que se haya proporcionado el Folio de Recepción (PedidoAdicional) para todas las ventas generadas dentro de la visita (esto siempre y cuando el cliente visitado pertenezca a un esquema de clientes incluido como valor para el parámetro), de no ser así, mostrar mensaje informativo y no permitir terminar la visita.  (Lechera Guadalajara, 4.20.0.2 – 1.15.0.2) | Belem Lizeth Jiménez Arévalo |
| 05/10/2018 | 1.22 | Folio CAI 0004481  Al finalizar la visita, validar que se hayan realizado las encuestas obligatorias, de no ser así, mostrar mensaje informativo y no permitir terminar la visita.  (1.14.0.0 - 4.19.0.0) | Erik Alejandro Amador Serrano |
| 23/11/2018 | 1.23 | Folio CAI 0005130  Se requiere que en la pantalla de selección de las actividades en la parte del menú (acceder a tareas), esté la opción de la lista de precio y que se pueda mostrar o imprimir.  (NIC, 4.22.0.0 – 1.17.0.0). | Erik Alejandro Amador Serrano |
| 08/03/2019 | 1.24 | Folio CAI 0005312  Mostrar información de crédito en una vista dentro del cliente  (Luberoil, 4.23.0.0) | Erik Alejandro Amador Serrano |
| 16/05/2019 | 1.25 | Folio CAI 0005421  Al finalizar la visita, agregar funcionalidad para que cuando se tenga configurado el aseguramiento de visita por Lectura de Código de Barras y el tipo de contraseña Variable, si se solicitó la contraseña dinámica o variable al iniciar la visita, que también se solicite para finalizarla.  **Se realiza en la versión 4.22.1.0 con replica a partir de la 4.23.0.0**  (PDR, 4.22.1.0) | Erik Alejandro Amador Serrano |
| 29/10/2019 | 1.26 | Se requiere replicar la validación de no poder salir de una visita con cliente hasta que no esten realizadas la(s) prefactura(s) de las ventas realizadas en la presente visita.  (La Florida, 4.21.0.3, 4.23.0.1, 4.24.0.0) | Belem Lizeth Jiménez Arévalo |
| 30/12/2019 | 1.27 | - Crear nueva tabla temporal en eRoute para que nos envíen disponibilidad de Backorder. Dicha tabla contendrá el folio del Backorder, productos y cantidades. \*\*NOTA: En caso de que el pedido de Backorder no se confirme completo, ya sea en número de partidas o en cantidades de producto, el flujo de confirmación de Backorder llegaría hasta aquí, es decir, no habría una segunda confirmación de lo faltante para el mismo Backorder.  - La información de la nueva tabla temporal generará un nuevo pedido que se estará enviando al móvil mediante la opción “Enviar/Recibir" información. La recepción de confirmaciones de Backorder mostrarán una alerta al vendedor mediante un mensaje.  - Crear nueva Vista mediante la que se puedan visualizar los productos y número de Backorder del que hay alguna actualización de existencias. En esa vista se prodrá Confirmar o Cancelar el Backorder.  Folio CAI 0005730  (Prisa, 1.19.0.0 - 4.24.0.0) | Belem Lizeth Jiménez Arévalo |
| 27/05/2020 | 1.28 | - El cliente Moroleón presenta problemas al facturar y se detecta que es por que las ventas están mal generadas. Se requiere encontrar y corregir el escenario por el que se generan incorrectamente. Son repartos surtidos, y al parecer el error esta desde que se generó la preventa, ya que no se ve que los detalles hayan sido modificados, aunque hay que probar tanto en preventa como en reparto. Los detalles con error son productos a los que se le aplico promoción de descuento y aparentemente se aplica doble en TransProdDetalle.Descuento, pero en TRPPRP se aplica correctamente. El campo TPDImpuesto.ImpDesGlb queda mal calculado asi como los totales de TransProd.  Folio CAI 0004978  (Prisa, 1.19.1.0 - 4.24.1.0) | Belem Lizeth Jiménez Arévalo |

Tabla de Contenido

[Especificación de Casos de Uso: Seleccionar Actividades para la Visita - CUROLMOV11 11](#_Toc42580227)

[1 Introducción 11](#_Toc42580228)

[2 Caso de uso: Seleccionar Actividades para la Visita - CUROLMOV11 11](#_Toc42580229)

[2.1 Descripción 11](#_Toc42580230)

[3 Diagrama de Casos de Uso 11](#_Toc42580231)

[4 Precondiciones 11](#_Toc42580232)

[4.1 Actores 11](#_Toc42580233)

[4.2 Generales 11](#_Toc42580234)

[4.3 Parámetros 12](#_Toc42580235)

[5 Flujo de eventos 12](#_Toc42580236)

[5.1 Flujo básico 12](#_Toc42580237)

[5.2 Flujos alternos 15](#_Toc42580238)

[5.2.1 Opcionales 15](#_Toc42580239)

[5.2.2 Opcionales 15](#_Toc42580240)

[5.2.3 Generales 15](#_Toc42580241)

[5.2.4 Extraordinarios 28](#_Toc42580242)

[5.2.5 De excepción 28](#_Toc42580243)

[5.2.6 De Validación 28](#_Toc42580244)

[6 Poscondiciones 31](#_Toc42580245)

[7 Anexos 31](#_Toc42580246)

[8 Diagramas 31](#_Toc42580247)

[9 Propuesta de Pantallas 31](#_Toc42580248)

[10 Firmas de Aceptación 31](#_Toc42580249)

# Especificación de Casos de Uso: Seleccionar Actividades para la Visita - CUROLMOV11

# Introducción

Este documento contiene la descripción de una secuencia de interacciones entre un actor y el sistema de tal manera que se puede obtener un resultado con valor particular para el actor dentro del flujo básico. También incluye el listado de las precondiciones y poscondiciones que se deben cumplir al inicio y al final del flujo básico además de los posibles flujos alternos y excepciones*.*

# Caso de uso: Seleccionar Actividades para la Visita - CUROLMOV11

## Descripción

Permite al actor visualizar las distintas actividades que puede realizar durante la visita de un cliente, dependiendo de su perfil y la configuración de las mismas.

# Diagrama de Casos de Uso

**

# Precondiciones

## Actores

1. Vendedor

## Generales

1. El actor debe de tener un usuario registrado en el sistema.
2. El actor debe de tener un perfil asignado para acceder al sistema.
3. El actor debe de estar activo.
4. Por lo menos un cliente debe de estar registrado en el catálogo de clientes y estar activo.
5. Debe de existir un día de trabajo asignado al vendedor
6. Debe de existir una agenda de trabajo asignada al vendedor
7. Debe de existir por lo menos una ruta asignada al vendedor.
8. Debe de existir una jornada de trabajo iniciada.
9. Debe de existir una visita iniciada a un cliente.
10. Debe de existir por lo menos un módulo configurado al vendedor.
11. Debe de existir por lo menos una actividad asociada a un módulo.

## Parámetros

1. ContinuarVisita

# Flujo de eventos

## Flujo básico

1. El caso de uso inicia cuando es invocado por los casos de uso
   1. [**Registrar Visita e Improductividad del Cliente – CUROLMOV10**](file:///C:\Amesol\Productos\RouteLite\Trunk\Analisis\EspecificacionRequerimientos\ECU_MOV\CUROLMOV10_RegVisitaImproductividad.docx)
   2. [**Mantenimiento de Visitas en la Terminal – CUROLMOV12**](file:///C:\Amesol\Productos\RouteLite\Trunk\Analisis\EspecificacionRequerimientos\ECU_MOV\CUROLMOV12_ManttoVisitasTerminal.docx)
2. El sistema obtiene información de los diferentes catálogos de acuerdo a las reglas de negocio [**RNGEN092 Tipos de Módulos de la Terminal**](file:///C:\Amesol\Productos\Route\Trunk\Analisis\EspecificacionRequerimientos\General\ERM_Reglas_de_Negocio.docx#RNGEN092) **y** [**RNGEN005 Valor Tipo de Estado de un Registro**](file:///C:\Amesol\Productos\Route\Trunk\Analisis\EspecificacionRequerimientos\General\ERM_Reglas_de_Negocio.docx#RNGEN005):
   1. **ValorReferencia**
      1. VARCodigo
      2. **VAVDescripcion**
         1. VARCodigo
         2. VARDescripcion
3. El sistema obtiene la siguiente información:
   1. **Vendedor**
      1. VendedorId
      2. Para cada módulo asociado al vendedor
         1. ModuloClave
   2. Para cada módulo
      1. **ModuloTerm**
         1. ModuloClave de acuerdo con la regla de negocio [**RNROLMOV068 Módulos asociados al vendedor**](file:///C:\Amesol\Productos\Route\Trunk\Analisis\EspecificacionRequerimientos\General\ERM_Reglas_de_Negocio.docx#RNROLMOV068)
         2. Nombre
         3. TipoIndice
         4. Tipo de acuerdo con la regla de negocio [**RNROLMOV69 Módulos asociados a las visitas**](file:///C:\Amesol\Productos\Route\Trunk\Analisis\EspecificacionRequerimientos\General\ERM_Reglas_de_Negocio.docx#RNROLMOV069)
         5. TipoEstado de acuerdo con la regla de negocio [**RNROLMOV070 Módulos Activos**](file:///C:\Amesol\Productos\Route\Trunk\Analisis\EspecificacionRequerimientos\General\ERM_Reglas_de_Negocio.docx#RNROLMOV070)
      2. **ModuloMov**
         1. ModuloClave de acuerdo con la regla de negocio [**RNROLMOV071 Módulos Correspondientes a la Terminal**](file:///C:\Amesol\Productos\Route\Trunk\Analisis\EspecificacionRequerimientos\General\ERM_Reglas_de_Negocio.docx#RNROLMOV071)
         2. ModuloMovClave
         3. Nombre
         4. TipoIndice
         5. TipoEstado de acuerdo con la regla de negocio [**RNROLMOV070 Módulos Activos**](file:///C:\Amesol\Productos\Route\Trunk\Analisis\EspecificacionRequerimientos\General\ERM_Reglas_de_Negocio.docx#RNROLMOV070)
         6. Baja de acuerdo con la regla de negocio [**RNROLMOV072 Módulos no dados de baja**](file:///C:\Amesol\Productos\Route\Trunk\Analisis\EspecificacionRequerimientos\General\ERM_Reglas_de_Negocio.docx#RNROLMOV072)
      3. **ModuloMovDetalle**
         1. ModuloMovDetalleClave
         2. ModuloClave de acuerdo con la regla de negocio [**RNROLMOV073 Módulos de Movimientos**](file:///C:\Amesol\Productos\Route\Trunk\Analisis\EspecificacionRequerimientos\General\ERM_Reglas_de_Negocio.docx#RNROLMOV073)
         3. ModuloMovClave de acuerdo con la regla de negocio [**RNROLMOV073 Módulos de Movimientos**](file:///C:\Amesol\Productos\Route\Trunk\Analisis\EspecificacionRequerimientos\General\ERM_Reglas_de_Negocio.docx#RNROLMOV073)
         4. Nombre
         5. TipoIndice
            1. **ClienteMMDExcepcion**

ModuloMovDetalleClave

ClienteClave = ClienteClave registrado en sesión

* + 1. **MOTConfiguracion**
       1. MCNClave
       2. ModuloClave de acuerdo con la regla de negocio [**RNROLMOV068 Módulos asociados al vendedor**](file:///C:\Amesol\Productos\Route\Trunk\Analisis\EspecificacionRequerimientos\General\ERM_Reglas_de_Negocio.docx#RNROLMOV068)
    2. **MmdMcn**
       1. MCNClave
    3. **AseguramientoVisita**
       1. MCNClave
       2. TipoAseguramiento
       3. TipoContraseña

1. El sistema obtiene la siguiente información:
   1. **ClienteMensaje**
      1. ClienteMensajeId
      2. ClienteClave = ClienteClave que se encuentra en sesión
      3. MDBMensajeID
      4. **MDBMensaje**
         1. MDBMensajeID
         2. Descripcion
      5. TipoEstado = 1 o 2
2. Si <Existe por lo menos un mensaje dentro de la tabla <ClienteMensaje> para el cliente y que se encuentre activo>
   1. Para cada mensaje encontrado que le corresponda al cliente
      1. El sistema presenta en una pantalla emergente la siguiente información:
         1. **Título**: Mensajes del Cliente
         2. **Pantalla**: <MDBMensaje.Descripcion>
      2. Si < el actor selecciona la opción “Aceptar” o “Salir”>
         1. El sistema registra la siguiente información:
            1. **ClienteMensaje**

ClienteMensajeID

TipoEstado = 0

1. El sistema presenta la siguiente información de acuerdo con la regla de negocio [**RNROLMOV10 Icono de Actividades o Módulo**](file:///C:\Amesol\Productos\Route\Trunk\Analisis\EspecificacionRequerimientos\General\ERM_Reglas_de_Negocio.docx#RNROLMOV010)
   1. Para cada módulo correspondiente al vendedor
      1. Módulo de acuerdo con la regla de negocio [**RNROLMOV074 Módulos para presentar en la Visita**](file:///C:\Amesol\Productos\Route\Trunk\Analisis\EspecificacionRequerimientos\General\ERM_Reglas_de_Negocio.docx#RNROLMOV074)
      2. El sistema presenta la siguiente información:
         1. Para cada actividad relacionada al módulo que no se encuentre en <ClienteMMDExcepcion.ModuloMovDetalleClave> donde <ClienteMMDExcepcion.ClienteClave = ClienteClave registrado en sesión>
            1. Nombre de la actividad de acuerdo con las reglas de negocio [**RNROLMOV075 Actividades Correspondientes de los Módulos**](file:///C:\Amesol\Productos\Route\Trunk\Analisis\EspecificacionRequerimientos\General\ERM_Reglas_de_Negocio.docx#RNROLMOV075) y [**RNROLMOV10 Icono de Actividades o Módulo**](file:///C:\Amesol\Productos\Route\Trunk\Analisis\EspecificacionRequerimientos\General\ERM_Reglas_de_Negocio.docx#RNROLMOV010)
            2. Si <el actor selecciona la actividad **Improductividad de Venta**>

El sistema extiende su funcionalidad con el caso de uso [**CUROLMOV22 Registrar Improductividad de Venta**](file:///C:\Amesol\Productos\RouteLite\Trunk\Analisis\EspecificacionRequerimientos\ECU_MOV\CUROLMOV22_RegImproductividadVta.docx)

* + - * 1. Si <el actor selecciona la actividad **Cobranza**>

El sistema extiende su funcionalidad con el caso de uso **[CUROLMOV23 –](C:\\Duxstar\\Productos\\RouteLite\\Trunk\\Analisis\\EspecificacionRequerimientos\\ECU_MOV\\CUROLMOV23_RealizarCobranza.docx)**

**[Realizar Cobranza](C:\\Duxstar\\Productos\\RouteLite\\Trunk\\Analisis\\EspecificacionRequerimientos\\ECU_MOV\\CUROLMOV23_RealizarCobranza.docx)**

* + - * 1. Si <el actor selecciona la actividad **Venta**>

El sistema valida que la fecha del día de trabajo que se encuentra en sesión sea igual a la del día actual del sistema

<Si FechaCaptura en sesión que corresponde al objeto Día es diferente al día actual obtenido del sistema>

El sistema obtiene la siguiente información:

**ConfigParametro**

Parametro = ValidarActPrecio

Valor

Si <ConfigParametro.Valor = 1>

El sistema valida en el XML relacionado a la tabla PrecioProductoVig la fecha de la última actualización

Si <la Fecha de la última actualización **ES MENOR** a la fecha del día actual del sistema>

El sistema presenta el mensaje: “E0954 - La fecha de la última actualización de precios es menor al día actual, para poder continuar se deberá realizar una nueva actualización de los mismos”

El sistema continúa en el [paso 4](#Paso_4) del flujo básico

Si <la fecha de la última actualización **NO** **ES MENOR** a la fecha del día actual del Sistema>

El sistema extiende su funcionalidad con el caso de uso [**CUROLMOV18 – Capturar Pedido**](file:///C:\Duxstar\Productos\RouteLite\Trunk\Analisis\EspecificacionRequerimientos\ECU_MOV\CUROLMOV18_CapturarPedido.docx)

Si <ConfigParametro.Valor = 0>

El sistema extiende su funcionalidad con el caso de uso [**CUROLMOV18 – Capturar Pedido**](file:///C:\Duxstar\Productos\RouteLite\Trunk\Analisis\EspecificacionRequerimientos\ECU_MOV\CUROLMOV18_CapturarPedido.docx)

<Si FechaCaptura en sesión que corresponde al objeto Día es igual al día actual obtenido del sistema>

El sistema extiende su funcionalidad con el caso de uso [**CUROLMOV18 – Capturar Pedido**](file:///C:\Duxstar\Productos\RouteLite\Trunk\Analisis\EspecificacionRequerimientos\ECU_MOV\CUROLMOV18_CapturarPedido.docx)

* + - * 1. Si <el actor selecciona la actividad **Cambios de Producto**>

El caso de uso extiende su funcionalidad con el caso de uso [**CUROLMOV26 - Administrar Cambios de Producto**](file:///C:\Duxstar\Productos\RouteLite\Trunk\Analisis\EspecificacionRequerimientos\ECU_MOV\CUROLMOV26_AdministrarCambiosProducto.docx)

* + - * 1. Si <el actor selecciona la actividad **Devolución de Clientes**>

El caso de uso extiende su funcionalidad con el caso de uso [**CUROLMOV27 - Administrar Devoluciones de Producto del Cliente**](file:///C:\Duxstar\Productos\RouteLite\Trunk\Analisis\EspecificacionRequerimientos\ECU_MOV\CUROLMOV27_AdministrarDevProdCliente.docx)

* + - * 1. Si <el actor selecciona la actividad **Venta a Consignación**>

El caso de uso extiende su funcionalidad con el caso de uso [**CUROLMOV44 – Capturar Ventas a Consignación**](file:///C:\Duxstar\Productos\RouteLite\Trunk\Analisis\EspecificacionRequerimientos\ECU_MOV\CUROLMOV44_CapturarVentasConsignación.docx)

* + - * 1. Si <el actor selecciona la actividad **Facturación Electrónica**>

El caso de uso extiende su funcionalidad con el caso de uso [**CUROLMOV45 – Generar Factura Electrónica**](file:///C:\Duxstar\Productos\RouteLite\Trunk\Analisis\EspecificacionRequerimientos\ECU_MOV\CUROLMOV45_GenerarFacturaElectrónica.docx)

* + - * 1. Si <el actor selecciona la actividad **Toma de Inventario y Mercadeo**>

El caso de uso extiende su funcionalidad con el caso de uso [**CUERMMOV92\_CapturarInventarioMercadeo**](file:///C:\Duxstar\Productos\Route\Trunk\Analisis\EspecificacionRequerimientos\ECU_MOV\CUERMMOV92_CapturarInventarioMercadeo.docx).(Se utilizará este caso de uso que corresponde a WM, hasta que se realice el correspondiente a Route Lite)

* + - * 1. Si<el actor selecciona la actividad Mercadeo de Productos>

El caso de uso extiende su funcionalidad con el caso de uso [**CUROLMOV81\_CapturarMercadeoProductos**](CUROLMOV81_CapturarMercadeoProductos.docx)

* + - * 1. Si<el actor selecciona la actividad Solicitudes>

El caso de uso extiende su funcionalidad con el caso [**CUROLMOV84\_CapturarSolicitudes**](CUROLMOV84_CapturarSolicitudes.docx)

* + - * 1. Si <el actor selecciona la actividad Histórico de Venta>

El caso de uso extiende su funcionalidad con el caso de uso [**CUROLMOV113\_HistoricoVentas**](CUROLMOV113_HistoricoVentas.docx)

1. Si <el actor selecciona la opción **Regresar**>
   1. El sistema continúa con el flujo alterno general [**AG01 Regresar/Terminar Visita**](#_AG01_Regresar/Terminar_Visita)
2. Si <el actor selecciona la opción **Acceder a Tareas**>
   1. El sistema presenta las siguientes opciones:
      1. Lista de Precio
      2. Vista Crédito
      3. Si <existe una configuración de parámetros Activa para Mostrar Pedidos de Backorder <ConfigParametro.Parametro = ‘MostrarPedidosBckOrder’> donde <ConfigParametro.Valor = 1>>
         1. Pedidos Backorder Disponible
      4. Terminar Visita
   2. Si <el actor selecciona Lista de Precio>
      1. El sistema presenta el siguiente mensaje “¿Que desea realizar?” enseguida muestra dos opciones:
         1. Mostrar
         2. Imprimir
         3. Si <el actor selecciona Mostrar>
            1. Se muestra la lista de precio ordenado por la clave del producto de manera ascendente, se muestra el identificador del producto <Producto.ProductoClave> enseguida el nombre <Producto.Nombre> junto con la unidad de venta <PrecioProductoVig.PRUTipoUnidad> según el valor por referencia “UNIDADV” y al final el precio anteponiendo el signo de pesos “$” <PrecioProductoVig.Precio> donde <Producto.ProductoClave> = <PrecioProductoVig.ProductoClave> and <PrecioProductoVig.PPVFechaInicio <= a la fecha actual> <PrecioProductoVig.FechaFinal >= a la fecha actual> and <Cliente.ClienteClave = Cliente seleccionado> and <PrecioClienteEsquema.EsquemaId> = <ClienteEsquema.EsquemaId> and <Precio.PrecioClave> = <PrecioClienteEsquema.PrecioClave> and <PrecioClienteEsquema.PrecioClave> = <PrecioProductoVig.PrecioClave>.
         4. Si <el actor selecciona Imprimir>
            1. Se manda la vista de la consulta anterior al dispositivo de impresión.
   3. Si <el actor selecciona la opción **Vista Crédito**>
      1. El sistema continúa con el flujo alterno general **[AG02 Vista de Crédito](#_AG01_Vista_Crédito)**
   4. Si <el actor selecciona la opción **Pedidos Backorder Disponible**>
      1. El caso de uso extiende su funcionalidad con el caso de uso **[CUROLMOV125 –](CUROLMOV125_AdministrarPedidosBackorderDisponible.docx)**

**[Administrar Pedidos de Backorder Disponible](CUROLMOV125_AdministrarPedidosBackorderDisponible.docx)**

* 1. Si <el actor selecciona la opción **Terminar Visita**>
     1. El sistema continúa con el flujo alterno general **[AG01 Regresar/Terminar Visita](#AG01)**

1. Finaliza el caso de uso

## Flujos alternos

### Opcionales

N/A

### Opcionales

N/A

### Generales

#### [AG01 Regresar/Terminar Visita](#AG01_r)

1. El sistema obtiene de sesión la siguiente información:
   1. LecturaCodigo
2. Si <AseguramientoVisita.TipoAseguramiento = 4 “Lectura código barras Inicio y Fin” ó 5 “Lectura código barras Inicio y Fin, GPS”> y Si <se logró leer el código de barras del cliente para iniciar la visita (LecturaCodigo <> 0) de acuerdo a la información registrada en sesión>
   1. El sistema presenta y solicita la siguiente información:
      1. **Código de Barras del Cliente (\*):** Se presenta deshabilitado el campo de texto que permite la captura manual del código de barras del cliente, de manera que el código sólo pueda ser leído.
      2. Si <AseguramientoVisita.TipoContrasena = 1 “Fija” o 2 “Variable”>
         1. Se muestra la opción “Ingresar contraseña” el cual permite ingresar la contraseña correspondiente.
            1. Si <AseguramientoVisita.TipoContrasena = 1 “Fija”>

El sistema solicita la contraseña fija

El actor ingresa la contraseña, si <AseguramientoVisita.ContrasenaFija = a la contraseña ingresada> donde <Vendedor.MCNClave = AseguramientoVisita.MCNClave>

El sistema continúa en el [**paso 4**](#paso5_AG01)del flujo alterno general [**AG01 Regresar/Terminar Visita**](#AG01)

* + - * 1. Si <AseguramientoVisita.TipoContrasena = 2 “Variable”>

El sistema incluye en su funcionalidad el caso de uso [**Obtención de Contraseña Dinámica – CUROLMOV013**](file:///C:\Duxstar\Productos\RouteLite\Trunk\Analisis\EspecificacionRequerimientos\ECU_MOV\CUROLMOV13_ObtencionContraDinamica.docx)

Si la autenticación de la contraseña fue exitosa.

El sistema registra el Token ingresado por el vendedor, en el campo <Visita.Token>.

El sistema continúa en el [**paso 4**](#paso5_AG01)del flujo alterno general [**AG01 Regresar/Terminar Visita**](#AG01)

* 1. El actor proporciona la siguiente información:
     1. Código de Barras del Cliente (\*)
  2. El sistema obtiene la siguiente información, donde <Cliente.ClienteClave = Cliente Actual>:
     1. **Cliente**
        1. IdElectronico
  3. Si <el código de barras proporcionado NO corresponde al código de barras del cliente (Cliente.IdElectronico <> Codigo Proporcionado)>
     1. El sistema presenta el mensaje “[E0489] El Código de Barras introducido no corresponde al $0$”, sustituir el valor de $0$ por “Cliente”
     2. El sistema continúa en el [**paso 2.1**](#paso2_1_AG01) del flujo alterno general [**AG01 Regresar/Terminar Visita**](#AG01)
  4. Si <el código de barras proporcionado SI corresponde al código de barras del cliente (Cliente.IdElectronico = Codigo Proporcionado)>
     1. El sistema continúa en el [**paso 4**](#paso5_AG01)del flujo alterno general [**AG01 Regresar/Terminar Visita**](#AG01)

1. Si <AseguramientoVisita.TipoAseguramiento = 6 “Lectura NFC Inicio y Fin”>
   1. Si <el sistema NO detecta el Tag del cliente actual por medio de NFC>
      1. El sistema presenta el mensaje “[I0298] No se puede terminar la visita debido a que no se detecta el NFC del cliente”
      2. El sistema continúa en el paso donde fue invocado (menú de actividades de la visita)
   2. Si <el sistema SI detecta el Tag del cliente actual por medio de NFC (donde Tag Cliente = ‘@’+Cliente.ClienteClave del Cliente actual)>
      1. El sistema continúa en el [**paso 4**](#paso5_AG01)del flujo alterno general [**AG01 Regresar/Terminar Visita**](#AG01)
2. El sistema obtiene la siguiente información de parámetros:
   1. ContinuarVisita
   2. VisitaClave
   3. VisitaHistId
   4. PuntoGPSID
3. Si <ContinuarVisita = Falso>
   1. Si <el parámetro de configuración indica que se debe Forzar la Captura de Improductividad de venta para la visita <MOTConfiguracion.ForzarCapturaImpro = 1>>
      1. El sistema presenta el mensaje “[I0269] No se puede terminar la visita, debe capturar una improductividad de venta ya que no se le realizaron movimientos al cliente”
      2. El sistema continúa en el paso donde fue invocado (menú de actividades de la visita)
   2. Si <existen encuestas activas <ConfigEncuesta.Fase = 2>, que sean obligatorias <ConfigEncuesta.Obligatoria = 1> y que no se hayan aplicado al tratar de finalizar la visita <Encuesta.CENClave = ConfigEncuesta.CENClave> y <Encuesta.Fase = 3> y <Encuesta.VisitaClave=Identificador de la visita actual> y <Encuesta.DiaClave = DiaClave registrado en sesión>>
      1. El sistema presenta el mensaje “E0983 - La visita no puede ser terminada, debido a que existen encuestas obligatorias que no se han aplicado”.
      2. El sistema continúa en el paso donde fue invocado (menú de actividades de la visita)
   3. Si <existe una configuración de parámetros para Capturar Pedido Adicional Sin Modificar Venta <ConfigParametro.Parametro = ‘CapturarPedAdSinModVenta’> cuyo valor(es) (separados mediante un pipe “|”) corresponda(n) a uno o varios esquemas de cliente a los que pertenezca el cliente actual <ConfigParametro.Valor = Esquema.Clave> donde <Esquema.Tipo = 1> y <Esquema.EsquemaId = ClienteEsquema.EsquemaId> o bien donde <Esquema.EsquemaIdPadre = ClienteEsquema.EsquemaId > y <ClienteEsquema.ClienteClave = Cliente actual>>
      1. El sistema obtiene la siguiente información:
         1. **TransProd**
            1. TransProdId
            2. VisitaClave = VisitaClave registrada en sesión
            3. Tipo = 1
            4. TipoFase <> 0
            5. **TRPVtaAcreditada**

TransProdId

PedidoAdicional

* + 1. Si <se obtuvieron movimientos de tipo Pedido no cancelados, realizados dentro de la visita actual, a los que no se les haya capturado el campo Pedido Adicional <TRPVtaAcreditada.PedidoAdicional = Null o ‘ ‘> o bien para los que no exista un registro en la tabla <TRPVtaAcreditada>>
       1. El sistema presenta el mensaje “[I0340] No se puede terminar la visita, debe capturar el $0$ para los movimientos de venta generados dentro de la visita actual”, sustituir el valor de $0$ por el mensaje XPedidoAdicional.
    2. El sistema continúa en el paso donde fue invocado (menú de actividades de la visita)
  1. Si <existe una configuración de parámetros Activa para Validar Promociones al Final de la Visita <ConfigParametro.Parametro = ‘ValidarPromoFinVisita’> donde <ConfigParametro.Valor = 1>>
     1. Si <existen ventas realizadas durante la visita actual <TransProd.TransProdId> y <TransProd.TipoFase <> 3> donde <TransProd.VisitaClave = Visita.VisitaClave de la visita actual> cuyos detalles cuentan con información de promociones aplicadas relacionadas <TransProdDetalle.TransProdId = TrpPrp.TransProdId> y <TransProdDetalle.TransProdDetalleId = TrpPrp.TransProdDetalleId> pero no cuentan con un importe de descuentos a nivel partida <TransProdDetalle.DescuentoImp = 0, Null o ‘ ‘>>
        1. Para cada una de estas ventas <TransProdId>
           1. Para cada detalle incluido en la venta,

El sistema actualiza la siguiente información:

**TransProdDetalle**

DescuentoImp = (∑<TrpPrp.PromocionImp> donde <TransProdDetalle.TransProdId = TrpPrp.TransProdId> y <TransProdDetalle.TransProdDetalleId = TrpPrp.TransProdDetalleId>). Si el resultado es null asignar 0.

Subtotal = <TransProdDetalle.Precio \* TransProdDetalle.Cantidad> -

TransProdDetalle.DescuentoImp.

Impuesto = <Método: CalcularImpuestoProducto (ProductoClave, MFechaHora, Subtotal)>.

Total = <TransProdDetalle.Subtotal + TransProdDetalle.Impuesto>.

El sistema obtiene información de los impuestos duplicados registrados para la partida actual, agrupándolos por ImpuestoClave (en caso de existir).

**TPDImpuesto**

TransProdID

TransProdDetalleID

ImpuestoClave

El sistema registra dentro de la tabla temporal #Duplicados la siguiente información de los impuestos previamente obtenidos:

TransProdID = TPDImpuesto.TransProdId

TransProdDetalleID = TPDImpuesto.TransProdDetalleId

ImpuestoClave = TPDImpuesto.ImpuestoClave

NumeroImpuestos = Contador del número de registros de impuestos duplicados

Para cada impuesto registrado en la tabla #Duplicados cuyo NumeroImpuestos > 1

El sistema obtiene el registro más reciente de impuesto aplicado a la partida actual

**TPDImpuesto**

TransProdID

TransProdDetalleID

TPDImpuestoID

ImpuestoClave

MFechaHora = Top 1 ordenando este campo de manera descendente.

El sistema registra la siguiente información de acuerdo a la información anterior obtenida de <TPDImpuesto>:

**TPDImpuestoEliminado**

TransProdID

TransProdDetalleID

TPDImpuestoID

ImpuestoClave

ImpuestoPor

ImpuestoImp

ImpuestoPU

ImpDesGlb

MFechaHora

MUsuarioID

El sistema elimina la siguiente información de acuerdo a la información anterior obtenida de <TPDImpuesto>

**TPDImpuesto**

TransProdID

TransProdDetalleID

TPDImpuestoID = TPDImpuestoID actual

ImpuestoClave

ImpuestoPor

ImpuestoImp

ImpuestoPU

ImpDesGlb

MFechaHora

MUsuarioID

El sistema elimina la tabla temporal #Duplicados

El sistema registra dentro de la tabla temporal #RecalculoImp la siguiente información del recálculo de impuestos para la partida actual:

TransProdID

TransProdDetalleID

ImpuestoClave

Impuesto = <Método: CalcularTPDImpuestoProducto (TransProdId)>

El sistema actualiza la siguiente información para la partida actual donde <TPDImpuesto.TransProdId = RecalculoImp.TransProdId> y <TPDImpuesto.TransProdDetalleId = RecalculoImp.TransProdDetalleId>:

**TPDImpuesto**

ImpuestoImp = @RecalculoImp.Impuesto

ImpDesGlb = @RecalculoImp.Impuesto

* + - * 1. El sistema actualiza la siguiente información del cabecero de la venta actual:

**TransProd**

Subtotal = ∑<TransProdDetalle.Subtotal> donde TransProdDetalle.TransProdId = TransProd.TransProdId actual>.

Impuesto = ∑<TPDImpuesto.ImpDesGlb> donde TPDImpuesto.TransProdID = TransProd.TransProdId actual>.

Total = <TransProd.Subtotal + TransProd.Impuesto>.

Notas = Se concatena al contenido ya existente del campo, la leyenda “Promoción Recalculada” separada mediante un pipe “|”.

* + - 1. El sistema presenta el siguiente mensaje a manera de mensaje toast “[I0356] Se encontraron errores en el cálculo de las promociones aplicadas y estas fueron re-calculadas”.
  1. Si <la visita actual ya fue enviada el servidor <Visita.Enviado = 1>>
     1. El sistema registra la siguiente información:
        1. **VisitaModificada**
           1. VisitaClave = <Visita.VisitaClave> de la visita actual
           2. DiaClave = <Visita.DiaClave> de la visita actual
           3. TipoEstado = 0
           4. Enviado = 0
  2. El sistema elimina la información contenida en los siguientes registros para la visita actual:
     1. **Visita**
     2. **VisitaHist**
  3. Si <PuntoGPSID contiene información>
     1. **PuntoGPSID**

1. Si <ContinuarVisita = Verdadero>
   1. Si <se le capturó una improductividad de visita al cliente actual, es decir, si <Agenda.TipoMotivo <> Null> donde <Agenda.DiaClave = DiaClave registrado en sesión y Agenda.ClienteClave = Cliente actual>>
      1. El sistema actualiza la siguiente información, donde <Agenda.DiaClave = DiaClave registrado en sesión y Agenda.ClienteClave = Cliente actual>:
         1. **Agenda**
            1. TipoMotivo = Null
            2. Comentario = Null
   2. Si <se le realizó una venta o pedido al cliente en el día de trabajo actual, es decir, si cuenta con un registro en <TransProd>, donde <TransProd.Tipo = 1 ó 21 y TransProd.TipoFase <> 0 y TransProd.VisitaClave = Visita.VisitaClave> donde <Visita.ClienteClave = Clienta actual y Visita.DiaClave = DiaClave registrado en sesión> y si <se capturó una improductividad de venta para la visita actual, es decir, si existe un registro en <ImproductividadVenta.VisitaClave = Visita actual>>
      1. El sistema elimina la siguiente información, donde <ImproductividadVenta.VisitaClave = Visita actual>:
         1. **ImproductividadVenta**
            1. VisitaClave
            2. DiaClave
            3. TipoMotivo
            4. Comentario
            5. MFechaHora
            6. MUsuarioID
            7. Enviado
   3. El sistema valida el límite de crédito del cliente visitado de acuerdo con la validación [**VA01 Validar Límite de Crédito del Cliente**](#_VA01_Validar_Límite)
   4. El sistema valida el cuadre de envase del cliente visitado de acuerdo con la validación **[VA02 Validar Cuadre de Envase del Cliente](#_VA01_Validar_Cuadre)**
   5. Si <la visita actual ya fue enviada el servidor <Visita.Enviado = 1>
      1. El sistema registra la siguiente información:
         1. **VisitaModificada**
            1. VisitaClave = <Visita.VisitaClave> de la visita actual
            2. DiaClave = <Visita.DiaClave> de la visita actual
            3. FechaHoraFinal = Fecha y hora actual del sistema
            4. TipoEstado = 1
            5. Enviado = 0
   6. Si <la visita actual NO ha sido enviada el servidor <Visita.Enviado = 0>>
      1. El sistema actualiza la siguiente información para la visita actual
         1. **Visita**
            1. FechaHoraFinal = Fecha y hora actual del sistema
            2. MFechaHora = Fecha y hora actual del sistema
            3. MUsuarioID = USUId registrado en sesión
            4. Enviado = 0
   7. El sistema obtiene la siguiente información:
      1. **VARValor**
         1. VARCodigo = ‘REPORTEA’
         2. Grupo = ‘TerminarVisita”
         3. VAVClave
         4. Estado = 1
   8. Si <se obtuvo por lo menos un valor (reporte) asociado al grupo “TerminarVisita” del valor por referencia ‘REPORTEA’>
      1. Para cada reporte incluido en el grupo “TerminarVisita”
         1. El sistema presenta el mensaje “[P0800] Desea imprimir $0$“, sustituir el valor de $0$ por la descripción del valor por referencia correspondiente <VAVDescripcion.Descripcion>.
            1. Si <el actor selecciona la opción **No**>

El sistema continúa en el [**paso 7**](#FIN) del flujo básico.

* + - * 1. Si <el actor selecciona la opción **Si**>

Si <se trata de la **Nota de Venta (Modelo de Oriente) (VAVClave = 29)**>

El caso de uso extiende su funcionalidad con el caso de uso [**Imprimir Ticket de Nota de Venta (Modelo de Oriente) – CUROLMOV86**](CUROLMOV86_ImprimirTicketNotaVenta%20(MOO).docx)

* 1. El sistema obtiene la siguiente información de la Configuración de Parametros:
     1. ConfigParametro
        1. Parametro = EnvioAtmFinVisita / TiempoMinVisita
        2. Identificador = Modulo registrado en sesión
        3. Valor
     2. Si <ConfigParametro.Valor = 1 donde ConfigParametro.Parametro = EnvioAtmFinVisita>
        1. El sistema incluye la funcionalidad del caso de uso [CUROLSIN03\_Servidor de Comunicaciones](../ECU_SIN/CUROLSIN03_ServidorDeComunicaciones.doc) mandando llamar el siguiente método
           1. EnviarInformacion

EnvioParcial = Verdadero

* + - 1. El sistema recibe como parámetros la siguiente información:
         1. EnvioExitoso = Verdadero
    1. Si <ConfigParametro.Valor > 0 donde ConfigParametro.Parametro = TiempoMinVisita>
       1. El sistema realiza la siguiente validación:
          1. TiempoVisita = <Visita.FechaHoraFinal – Visita.FechaHoraInicial>
       2. Si<TiempoVisita < ConfigParametro.Valor (donde ConfigParametro.Parametro = TiempoMinVisita) >
          1. El sistema presenta el siguiente mensaje: “E1010 - No es posible finalizar la visita, se requiere un tiempo mínimo de $0 minuto (s) con el cliente”, donde $0 es igual a <ConfigParametro.Valor (donde ConfigParametro.Parametro = TiempoMinVisita).
          2. El sistema continúa en el paso en donde fue invocado, NO permitiendo terminar la visita
       3. Si <TiempoVisita < ConfigParametro.Valor (donde ConfigParametro.Parametro = TiempoMinVisita) >
          1. El sistema continúa en el paso en donde fue invocado, permitiendo terminar la visita
  1. Si <el módulo actual es el de Venta o Reparto <ModuloTerm.TipoIndice = 1 ó 3>>
     1. Si <el parámetro de configuración indica que se deben facturar todas las ventas <MOTConfiguracion.VentasFacturas = 1>>
        1. Si <existen ventas realizadas al cliente en el día de trabajo actual, que no han sido facturadas, es decir, si cuenta con registros en <TransProd>, donde <TransProd.Tipo = 1 y TransProd.TipoFase = 1 o 2 y TransProd.VisitaClave/VisitaClave1 = Visita.VisitaClave> donde <Visita.ClienteClave = Clienta actual y Visita.DiaClave = DiaClave registrado en sesión>>
           1. El sistema presenta el mensaje “[I0265] No se puede terminar la visita, la(s) venta(s) $0$, aún no han sido facturada(s)”, sustituyendo el valor de $0$ por la cadena conformada por el folio <TransProd.Folio> de las ventas no facturadas obtenidas, separadas por una coma en caso de ser más de una.
  2. El sistema continúa en el [**paso 7**](#FIN) del flujo básico.

#### [AG02 Vista Crédito](#AG02)

1. Se obtiene el saldo de las ventas:
   1. **TransProd**
      1. VisitaClave
      2. DiaClave
      3. ClienteClave
      4. **Visita**
         1. VisitaClave
         2. DiaClave
         3. ClienteClave = Cliente actual.
      5. Tipo
      6. CFVTipo = 2
      7. TipoFase <> 0
      8. Saldo
2. Se obtiene el saldo de las ventas vencidas:
   1. **TransProd**
      1. VisitaClave
      2. DiaClave
      3. ClienteClave
      4. **Visita**
         1. VisitaClave
         2. DiaClave
         3. ClienteClave = Cliente actual.
      5. Tipo
      6. CFVTipo = 2
      7. TipoFase <> 0
      8. Saldo
      9. FechaCobranza < a la fecha actual del sistema.
3. Se obtiene el saldo de las ventas con cheque:
   1. **TransProd**
      1. TransProdID
      2. VisitaClave
      3. DiaClave
      4. ClienteClave
      5. **Visita**
         1. VisitaClave
         2. DiaClave
         3. ClienteClave = Cliente actual.
      6. **TRPCheque**
         1. TransProdID
         2. ABNId
         3. **Abono**
            1. ABNCheque
            2. ABNId
      7. Tipo
      8. CFVTipo = 2
      9. TipoFase <> 0
      10. Saldo
4. Se obtiene el saldo de las ventas vencidas con cheque:
   1. **TransProd**
      1. TransProdID
      2. VisitaClave
      3. DiaClave
      4. ClienteClave
      5. **Visita**
         1. VisitaClave
         2. DiaClave
         3. ClienteClave = Cliente actual.
      6. **TRPCheque**
         1. TransProdID
         2. ABNId
         3. **Abono**
            1. ABNCheque
            2. ABNId
      7. Tipo
      8. CFVTipo = 2
      9. TipoFase <> 0
      10. Saldo
      11. FechaCobranza < a la fecha actual del sistema.
5. Se obtiene el saldo de las facturas:
   1. **TransProd**
      1. VisitaClave
      2. DiaClave
      3. ClienteClave
      4. **Visita**
         1. VisitaClave
         2. DiaClave
         3. ClienteClave = Cliente actual.
      5. Tipo
      6. CFVTipo = 2
      7. TipoFase <> 0
      8. Saldo
6. Se obtiene el saldo de las facturas vencidas:
   1. **TransProd**
      1. VisitaClave
      2. DiaClave
      3. ClienteClave
      4. **Visita**
         1. VisitaClave
         2. DiaClave
         3. ClienteClave = Cliente actual.
      5. Tipo
      6. CFVTipo = 2
      7. TipoFase <> 0
      8. Saldo
      9. FechaCobranza < a la fecha actual del sistema.
7. Se obtiene el saldo de las facturas con cheque:
   1. **TransProd**
      1. TransProdID
      2. VisitaClave
      3. DiaClave
      4. ClienteClave
      5. **Visita**
         1. VisitaClave
         2. DiaClave
         3. ClienteClave = Cliente actual.
      6. **TRPCheque**
         1. TransProdID
         2. ABNId
         3. **Abono**
            1. ABNCheque
            2. ABNId
      7. Tipo
      8. CFVTipo = 2
      9. TipoFase <> 0
      10. Saldo
8. Se obtiene el saldo de las facturas vencidas con cheque:
   1. **TransProd**
      1. TransProdID
      2. VisitaClave
      3. DiaClave
      4. ClienteClave
      5. **Visita**
         1. VisitaClave
         2. DiaClave
         3. ClienteClave = Cliente actual.
      6. **TRPCheque**
         1. TransProdID
         2. ABNId
         3. **Abono**
            1. ABNCheque
            2. ABNId
      7. Tipo
      8. CFVTipo = 2
      9. TipoFase <> 0
      10. Saldo
      11. FechaCobranza < a la fecha actual del sistema.
9. Se obtiene el saldo de las ventas no facturadas:
   1. **TransProd**
      1. VisitaClave
      2. DiaClave
      3. ClienteClave
      4. **Visita**
         1. VisitaClave
         2. DiaClave
         3. ClienteClave = Cliente actual.
      5. Tipo = 1
      6. CFVTipo = 2
      7. TipoFase <> 0
      8. Saldo
      9. FacturaID = Null o vacío
10. Se obtiene el saldo de las ventas vencidas no facturadas:
    1. **TransProd**
       1. VisitaClave
       2. DiaClave
       3. ClienteClave
       4. **Visita**
          1. VisitaClave
          2. DiaClave
          3. ClienteClave = Cliente actual.
       5. Tipo = 1
       6. CFVTipo = 2
       7. TipoFase <> 0
       8. Saldo
       9. FacturaID = Null o vacío
       10. FechaCobranza < a la fecha actual del sistema.
11. Se obtiene el saldo del cliente:
    1. **TransProd**
       1. VisitaClave
       2. DiaClave
       3. **Visita**
          1. VisitaClave
          2. DiaClave
          3. ClienteClave = Cliente actual
       4. PCEModuloMovDetClave
       5. **ModuloMovDetalle**
          1. PCEModuloMovDetClave
          2. TipoIndice = 9
       6. CFVTipo = 2
       7. TipoFase = 1
       8. Total
12. Se obtiene el saldo vencido del cliente:
    1. **TransProd**
       1. VisitaClave
       2. DiaClave
       3. **Visita**
          1. VisitaClave
          2. DiaClave
          3. ClienteClave = Cliente actual
       4. PCEModuloMovDetClave
       5. **ModuloMovDetalle**
          1. PCEModuloMovDetClave
          2. TipoIndice = 9
       6. CFVTipo = 2
       7. TipoFase = 1
       8. Total
       9. FechaCobranza < a la fecha actual del sistema.
13. Se obtiene la siguiente información sobre el historial de configuración:
    1. **CONHist**
       1. TipoCobranza
14. Se obtiene la siguiente información referente al cliente actual:
    1. **Cliente**
       1. ActualizaSaldoCheque
       2. SaldoEfectivo
       3. TipoFiscal
    2. **CLIFormaVenta**
       1. LimiteCredito
       2. DiasCredito
       3. CFVTipo
15. Si <CLIFormaVenta.CFVTipo = 2> es decir, si hay ventas a crédito:
    1. **Límite de Crédito**: se obtiene el límite de crédito del cliente actual <CLIFormaVenta.LimiteCredito>.
    2. **Días de Crédito**: se obtiene los días de crédito del cliente actual <CLIFormaVenta.DiasCredito>.
    3. Si es de tipo Venta o Reparto
       1. Si <CONHist.TipoCobranza = 1 o 2 y además <Cliente.TipoFiscal = 1>, es decir si el tipo de cobranza es a venta o a venta y factura, y además el documento fiscal es nota de venta, entonces se obtiene lo siguiente:
          1. Si <Cliente.ActualizaSaldoCheque>, es decir, los pagos hechos con cheque no serán tomados en cuenta, se obtiene lo siguiente:
             1. **Saldo Documentos:** se obtiene la información del punto [1 de la vista crédito](#AG02_1), de la sumatoria del saldo de las ventas ∑<TransProd.Saldo> donde ((<TransProd.Tipo = 1> y <TransProd.Tipo=2>) o <TransProd.CFVTipo = 24>) y <TransProd.TipoFase <> 0> y <Visita.ClienteClave = al cliente actual del sistema>.
             2. **Saldo Vencido:** se obtiene la información del punto [2 de la vista crédito](#AG02_2), de la sumatoria del saldo de las ventas vencidas ∑<TransProd.Saldo> donde ((<TransProd.Tipo = 1> y <TransProd.Tipo=2>) o <TransProd.CFVTipo = 24>) y <TransProd.TipoFase <> 0> y <Visita.ClienteClave = al cliente actual del sistema> y <TransProd.FechaCobranza < Fecha actual del sistema>.
          2. Si no, se obtiene lo siguiente:
             1. **Saldo Documentos:** se obtiene la información del punto [3 de la vista crédito](#AG02_3), de la sumatoria del saldo de las ventas, presentar la sumatoria de: ∑<TransProd.Saldo + (si es Null o vacío TRPCheque.AbnCheque entonces presentar un 0, si no, presentar TRPCheque.AbnCheque)> si es Null o vacío entonces presentar un 0, si no, se realiza la sumatoria de: ∑<TransProd.Saldo + (si es Null o vacío TRPCheque.AbnCheque entonces presentar un 0, si no, presentar TRPCheque.AbnCheque)>, donde ((<TransProd.Tipo = 1> y <TransProd.Tipo=2>) o <TransProd.CFVTipo = 24>) y <TransProd.TipoFase <> 0> y <Visita.ClienteClave = al cliente actual del sistema>.
             2. **Saldo Vencido:** se obtiene la información del punto [4 de la vista crédito](#AG02_4), de la sumatoria del saldo de las ventas vencidas, presentar la sumatoria de: ∑<TransProd.Saldo + (si es Null o vacío TRPCheque.AbnCheque entonces presentar un 0, si no, presentar TRPCheque.AbnCheque)> si es Null o vacío entonces presentar un 0, si no, se realiza la sumatoria de: ∑<TransProd.Saldo + (si es Null o vacío TRPCheque.AbnCheque entonces presentar un 0, si no, presentar TRPCheque.AbnCheque)>, donde ((<TransProd.Tipo = 1> y <TransProd.Tipo=2>) o <TransProd.CFVTipo = 24>) y <TransProd.TipoFase <> 0> y <Visita.ClienteClave = al cliente actual del sistema> y <TransProd.FechaCobranza < Fecha actual del sistema>.
       2. Si <CONHist.TipoCobranza = 0 o 2 y además <Cliente.TipoFiscal = 1>, es decir si el tipo de cobranza es a factura o a venta y factura, y además el documento fiscal es tipo factura, entonces se obtiene lo siguiente:
          1. Si <Cliente.ActualizaSaldoCheque>, es decir, los pagos hechos con cheque no serán tomados en cuenta, se obtiene lo siguiente:
             1. **Saldo Documentos:** se obtiene la información del punto [5 de la vista crédito](#AG02_5), de la sumatoria del saldo de las facturas (∑<TransProd.Saldo> donde ((<TransProd.Tipo = 8> y <TransProd.Tipo=2>) o <TransProd.CFVTipo = 24>) y <TransProd.TipoFase <> 0> y <Visita.ClienteClave = al cliente actual del sistema>) + el saldo de las ventas no facturadas, obteniendo la información en el punto [9 de la vista de crédito](#AG02_9), (∑<TransProd.Saldo> donde <TransProd.Tipo = 1> y <TransProd.CFVTipo=2> y <TransProd.TipoFase <> 0> y <TransProd.FacturaID = Null o vacío> y <Visita.ClienteClave = al cliente actual del sistema>).
             2. **Saldo Vencido:** se obtiene la información del punto [6 de la vista crédito](#AG02_6), de la sumatoria del saldo de las facturas vencidas (∑<TransProd.Saldo> donde ((<TransProd.Tipo = 8> y <TransProd.Tipo=2>) o <TransProd.CFVTipo = 24>) y <TransProd.TipoFase <> 0> y <Visita.ClienteClave = al cliente actual del sistema> y <TransProd.FechaCobranza < Fecha actual del sistema>) + el saldo de las ventas vencidas no facturadas, obteniendo la información en el punto [10 de la vista de crédito](#AG02_10), (∑<TransProd.Saldo> donde <TransProd.Tipo = 1> y <TransProd.CFVTipo = 2> y <TransProd.TipoFase <> 0> y <TransProd.FacturaID = Null o vacío> y <Visita.ClienteClave = al cliente actual del sistema> y <TransProd.FechaCobranza < a la fecha actual del sistema>).
          2. Si no, se obtiene lo siguiente:
             1. **Saldo Documentos:** se obtiene la información del punto [7 de la vista crédito](#AG02_7), de la sumatoria del saldo de las facturas, (Presentar la sumatoria de: ∑<TransProd.Saldo + (si es Null o vacío TRPCheque.AbnCheque entonces presentar un 0, si no, presentar TRPCheque.AbnCheque)> si es Null o vacío entonces presentar un 0, si no, se realiza la sumatoria de: ∑<TransProd.Saldo + (si es Null o vacío TRPCheque.AbnCheque entonces presentar un 0, si no, presentar TRPCheque.AbnCheque)>, donde ((<TransProd.Tipo = 8> y <TransProd.Tipo=2>) o <TransProd.CFVTipo = 24>) y <TransProd.TipoFase <> 0> y <Visita.ClienteClave = al cliente actual del sistema>) + el saldo de las ventas no facturadas, obteniendo la información en el punto [9 de la vista de crédito](#AG02_10), (∑<TransProd.Saldo> donde <TransProd.Tipo = 1> y <TransProd.CFVTipo=2> y <TransProd.TipoFase <> 0> y <TransProd.FacturaID = Null o vacío> y <Visita.ClienteClave = al cliente actual del sistema>).
             2. **Saldo Vencido:** se obtiene la información del punto [8 de la vista crédito](#AG02_8), de la sumatoria del saldo de las facturas vencidas, (presentar la sumatoria de: ∑<TransProd.Saldo + (si es Null o vacío TRPCheque.AbnCheque entonces presentar un 0, si no, presentar TRPCheque.AbnCheque)> si es Null o vacío entonces presentar un 0, si no, se realiza la sumatoria de: ∑<TransProd.Saldo + (si es Null o vacío TRPCheque.AbnCheque entonces presentar un 0, si no, presentar TRPCheque.AbnCheque)>, donde ((<TransProd.Tipo = 8> y <TransProd.Tipo=2>) o <TransProd.CFVTipo = 24>) y <TransProd.TipoFase <> 0> y <Visita.ClienteClave = al cliente actual del sistema> y <TransProd.FechaCobranza < Fecha actual del sistema>) + el saldo de las ventas vencidas no facturadas, obteniendo la información en el punto [10 de la vista de crédito](#AG02_10), (∑<TransProd.Saldo> donde <TransProd.Tipo = 1> y <TransProd.CFVTipo = 2> y <TransProd.TipoFase <> 0> y <TransProd.FacturaID = Null o vacío> y <Visita.ClienteClave = al cliente actual del sistema> y <TransProd.FechaCobranza < a la fecha actual del sistema>).
    4. Si es de tipo Preventa
       1. **Saldo Documentos:** se obtiene la información del punto [11 de la vista crédito](#AG02_11), de la sumatoria del total del cliente (∑ (<TransProd.Total>) si es igual a Null o Vacío, entonces presentar un 0 (cero), donde <Visita.ClienteClave = Cliente actual del sistema> y <ModuloMovDetalle.TipoIndice = 9> y <TransProd.CFVTipo = 2> y <TransProd.TipoFase = 1>) + el saldo del cliente <Cliente.SaldoEfectivo>.
       2. **Saldo Vencido:** se obtiene la información del punto [12 de la vista crédito](#AG02_12), de la sumatoria del total vencido del cliente ∑ (<TransProd.Total>) si es igual a Null o Vacío, entonces presentar un 0 (cero), donde <Visita.ClienteClave = Cliente actual del sistema> y <ModuloMovDetalle.TipoIndice = 9> y <TransProd.CFVTipo = 2> y <TransProd.TipoFase = 1> y <TransProd.FechaCobranza = Fecha actual del sistema>.
    5. **Crédito Disponible:** se obtiene el crédito disponible haciendo una resta de <CLIFormaVenta.LimiteCredito> - <**Saldo Documentos**> obtenido anteriormente.

### Extraordinarios

N/A

### De excepción

N/A

### De Validación

#### [VA01 Validar Límite de Crédito del Cliente](#VA01_r)

1. El sistema valida el límite de crédito de acuerdo a la configuración actual:
   1. Si <el parámetro de configuración indica que se debe realizar la validación del límite de crédito al Terminar la Visita <CONHist.TipoLimiteCredito = 1 “Terminar Visita” ó 3 “Terminar Visita(sólo mensaje)”>>
      1. El sistema obtiene y registra en sesión la siguiente información, donde <CLIFormaVenta.ClienteClave = Cliente Actual y CLIFormaVenta.CFVTipo = 2>:
         1. **CLIFormaVenta**
            1. ValidaLimite
            2. LimiteCredito
            3. DiasCredito
      2. Si <la configuración de la forma de venta del cliente indica que se debe validar el límite de crédito <CLIFormaVenta.ValidaLimite = 1>>
         1. Si <el módulo actual es el de Venta o Reparto <ModuloTerm.TipoIndice = 1 ó 3>>
            1. Si <el parámetro de configuración indica que la cobranza se realiza sobre Ventas <CONHist.TipoCobranza = 1>> o si <el parámetro de configuración indica que la cobranza se realiza sobre Ambas <CONHist.TipoCobranza = 2> y la configuración del cliente indica que la cobranza se le realiza sobre Ventas <Cliente.TipoFiscal = 1 donde Cliente.ClienteClave = Cliente Actual>>

El sistema obtiene el importe total del saldo de las ventas a crédito y ventas a consignación realizadas al Cliente Actual:

***SaldoCliente*** = <∑TransProd.Saldo> donde <TransProd.VisitaClave = Visita.VisitaClave> y <TransProd.DiaClave = Visita.DiaClave> y <TransProd.ClienteClave = Visita.ClienteClave> y <(TransProd.Tipo = 1 y TransProd.CFVTipo = 2) o (TransProd.Tipo = 24)> y <TransProd.TipoFase <> 0>.

* + - * 1. Si <el parámetro de configuración indica que la cobranza se realiza sobre Facturas <CONHist.TipoCobranza = 0>> o si <el parámetro de configuración indica que la cobranza se realiza sobre Ambas <CONHist.TipoCobranza = 2> y la configuración del cliente indica que la cobranza se le realiza sobre Facturas <Cliente.TipoFiscal = 2 donde Cliente.ClienteClave = Cliente Actual>>

El sistema obtiene el importe total del saldo de las facturas y ventas a crédito, y de las ventas a consignación realizadas al Cliente Actual:

***SaldoCliente*** = <∑TransProd.Saldo> donde <TransProd.VisitaClave = Visita.VisitaClave> y <TransProd.DiaClave = Visita.DiaClave> y <TransProd.ClienteClave = Visita.ClienteClave> y <(TransProd.Tipo = 8 y TransProd.CFVTipo = 2) o (TransProd.Tipo = 24)> y <TransProd.TipoFase <> 0>.

***SaldoCliente*** = *SaldoCliente* + <∑TransProd.Saldo> donde <TransProd.VisitaClave = Visita.VisitaClave> y <TransProd.DiaClave = Visita.DiaClave> y <TransProd.ClienteClave = Visita.ClienteClave> y <TransProd.Tipo = 1> y <TransProd.CFVTipo = 2> y <TransProd.TipoFase <> 0> y <TransProd.FacturaId = Null>.

* + - 1. Si <el módulo actual es el de Preventa <ModuloTerm.TipoIndice = 2> >
         1. Si <el parámetro de configuración indica que la cobranza se realiza sobre Ventas <CONHist.TipoCobranza = 1> o si <el parámetro de configuración indica que la cobranza se realiza sobre Ambas <CONHist.TipoCobranza = 2> y la configuración del cliente indica que la cobranza se le realiza sobre Ventas <Cliente.TipoFiscal = 1 donde Cliente.ClienteClave = Cliente Actual>>

El sistema obtiene el importe total del saldo de las ventas a crédito y ventas a consignación realizadas al Cliente Actual, más el saldo del cliente:

***SaldoCliente*** = <∑TransProd.Total> donde <TransProd.VisitaClave = Visita Actual> y <TransProd.DiaClave = Visita.DiaClave> y <TransProd.ClienteClave = Visita.ClienteClave> y <(TransProd.Tipo = 1 y TransProd.CFVTipo = 2) o (TransProd.Tipo = 24)> y <TransProd.TipoFase = 1>.

***SaldoCliente*** = *SaldoCliente* + <Cliente.SaldoEfectivo donde Cliente.ClienteClave = Cliente Actual>

* + - 1. Si <la configuración del cliente indica que el pago con cheque NO afecta el saldo <Cliente.ActualizaSaldoCheque = 0 donde Cliente.ClienteClave = Cliente Actual>>
         1. El sistema registra en sesión la siguiente información:

TipoTrp = Cadena vacía

* + - * 1. Si <el parámetro de configuración indica que la cobranza se realiza sobre Ventas <CONHist.TipoCobranza = 1> o si <el parámetro de configuración indica que la cobranza se realiza sobre Ambas <CONHist.TipoCobranza = 2> y la configuración del cliente indica que la cobranza se le realiza sobre Ventas <Cliente.TipoFiscal = 1 donde Cliente.ClienteClave = Cliente Actual>>

TipoTrp = 1

* + - * 1. Si <el parámetro de configuración indica que la cobranza se realiza sobre Facturas <CONHist.TipoCobranza = 0> o si <el parámetro de configuración indica que la cobranza se realiza sobre Ambas <CONHist.TipoCobranza = 2> y la configuración del cliente indica que la cobranza se le realiza sobre Facturas <Cliente.TipoFiscal = 2 donde Cliente.ClienteClave = Cliente Actual>>

TipoTrp = 8

* + - * 1. El sistema obtiene el importe total de los abonos del cliente realizados con cheque sobre las ventas o facturas a crédito (según corresponda al tipo de cobranza) y sobre las ventas a consignación realizadas:

***AbonosCheque*** = <∑ABNDetalle.Importe> donde <ABNDetalle.ABNId = AbnTrp.ABNId> y <AbnTrp.TransProdID = TransProd.TransProdId> y <TransProd.VisitaClave = Visita.VisitaClave> y <TransProd.DiaClave = Visita.DiaClave> y <Visita.ClienteClave = Cliente Actual> donde <ABNDetalle.TipoPago = 2 “Cheque”> y <TransProd.CFVTipo = 2 “Crédito”> y <(TransProd.Tipo = TipoTrp registrado en sesión) o (TransProd.Tipo = 24)> y <TransProd.TipoFase <> 0>.

* + - * 1. El sistema obtiene el importe total del saldo deudor del cliente, incluyendo el importe de los abonos realizados con cheque:

***SaldoCliente*** = *SaldoCliente* +*AbonosCheque.*

* + - * 1. El sistema obtiene el importe total del saldo de los abonos del cliente NO realizados con cheque (saldo a favor):

***SaldoAbonos*** = <∑Abono.Saldo> donde <Abono.VisitaClave = Visita.VisitaClave> y <Abono.DiaClave = Visita.DiaClave> donde <Visita.ClienteClave = Cliente Actual> y <Abono.ABNId <> Abono.ABNId (donde Abono.ABNId = ABNDetalle.ABNId y Abono.VisitaClave = Visita.VisitaClave y Abono.DiaClave = Visita.DiaClave y Visita.ClienteClave = Cliente Actual y ABNDetalle.TipoPago = 2 “Cheque”)>.

* + - * 1. El sistema obtiene el importe total del saldo deudor del cliente, excluyendo el importe del saldo de los abonos (saldo a favor) NO realizados con cheque:

***SaldoCliente*** = *SaldoCliente* – *SaldoAbonos.*

* + - 1. Si <la configuración del cliente indica que el pago con cheque SI afecta el saldo <Cliente.ActualizaSaldoCheque = 1 donde Cliente.ClienteClave = Cliente Actual>>
         1. El sistema obtiene el importe total del saldo de los abonos del cliente (saldo a favor):

***SaldoAbonos*** = <∑Abono.Saldo> donde <Abono.VisitaClave = Visita.VisitaClave> y <Abono.DiaClave = Visita.DiaClave> donde <Visita.ClienteClave = Cliente Actual>.

* + - * 1. El sistema obtiene el importe total del saldo deudor del cliente, excluyendo el importe del saldo de los abonos realizados (saldo a favor):

***SaldoCliente*** = *SaldoCliente* – *SaldoAbonos.*

* + - 1. El sistema obtiene la siguiente información:
         1. ***SaldoCliente*** = LimiteCredito (registrado en sesión) – *SaldoCliente*
      2. Si <el saldo del cliente excede su límite de crédito (*SaldoCliente* < 0)>
         1. El sistema obtiene y registra en sesión el porcentaje de riesgo de la empresa:

**CONHist**

PorcentajeRiesgo

* + - * 1. El sistema obtiene el importe por el que podría excederse el cliente, de acuerdo a su límite de crédito y al porcentaje de riesgo de la empresa:

***Riesgo***  = [LimiteCredito (registrado en sesión) \* PorcentajeRiesgo (registrado en sesión)] / 100.

* + - * 1. Si <el saldo del cliente excede el riesgo (*SaldoCliente* (valor absoluto)> *Riesgo*)>

Si <el parámetro de configuración indica que se debe realizar la validación del límite de crédito al Terminar la Visita <CONHist.TipoLimiteCredito = 1 “Terminar Visita”> de acuerdo a la información registrada en sesión>

El sistema presenta el mensaje “[E0530] La visita no puede ser terminada, debido a que ha Excedido el Límite de Crédito del Cliente en $0$”, sustituir el valor de $0$ por el valor absoluto de SaldoCliente.

El sistema continúa en el paso en donde fue invocado, NO permitiendo terminar la visita.

Si <el parámetro de configuración indica que se debe realizar la validación del límite de crédito al Terminar la Visita <CONHist.TipoLimiteCredito = 3 “Terminar Visita(sólo mensaje)”> de acuerdo a la información registrada en sesión>

El sistema presenta el mensaje “[I0267] Los movimientos de pedidos o consignación están sujetos a aprobación debido a que el cliente ha excedido su límite de crédito en $0$”, sustituir el valor de $0$ por el valor absoluto de *SaldoCliente*.

El sistema continúa en el siguiente paso de donde fue invocado, permitiendo terminar la visita.

* 1. El sistema continúa en el siguiente paso de donde invocado

#### [VA02 Validar Cuadre de Envase del Cliente](#VA02_r)

1. El sistema valida el cuadre de envase de acuerdo a la configuración actual:
   1. Si <el módulo actual es el de REPARTO <ModuloTerm.TipoInidice = 3> y Si <al cliente se le debe realizar el manejo del préstamo de envase <Cliente.Prestamo = 1>> y Si <el parámetro de configuración indica que se debe validar el cuadre de envase <MOTConfiguracion.CuadreDeEnvase = 1>>
      1. El sistema obtiene la siguiente información, donde <ProductoPrestamoCli.ClienteClave = Cliente Actual>
         1. **ProductoPrestamoCli**
            1. ProductoClave
            2. Saldo
            3. SaldoCarga
      2. Si <el Saldo de alguno de los productos prestados al cliente es mayor a la cantidad que tenía prestada el cliente de ese producto al momento de subir a la terminal la agenda <ProductoPrestamoCli.Saldo > ProductoPrestamoCli.SaldoCarga>>
         1. El sistema presenta el mensaje “[E0912] La visita no puede ser terminada, debido a que queda envase pendiente por recolectar o vender”
         2. El sistema continúa en el paso en donde fue invocado, NO permitiendo terminar la visita.
   2. El sistema continúa en el siguiente paso de donde fue invocado, permitiendo terminar la visita.

# Poscondiciones

1. Eliminación de Registros Visita, VisitaHist, PuntoGPSID

# Anexos

N/A

# Diagramas

Los Diagramas se encuentran en la estructura de \Productos\RouteLite\Trunk\Analisis\EspecificacionRequerimientos\Modelo\ROL\

# Propuesta de Pantallas

La propuesta de pantallas se encuentran en la estructura de \Productos\RouteLite\Trunk\Analisis\EspecificacionRequerimientos\Pantallas\ESC\_MOV\

# Firmas de Aceptación

|  |
| --- |
|  |
| José María Alcalá |
| Líder de Proyecto / Gerente de Ingeniería |
| 14/04/2012 |

|  |
| --- |
|  |
| Gilberto Ochoa |
| Gerente de Implementación |
| 14/04/12 |

|  |
| --- |
|  |
| Lizza Pasindo |
| Analista / Coordinador de Análisis |
| 14/04/12 |