**Proyecto Route Lite (ROL)**

Especificación del Caso de Uso: Registrar Inicio y Fin de Jornada – CUROLMOV43

VERSIÓN 1.11

**HISTÓRICO DE REVISIÓN**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
| 14/02/2014 | 1.0 | Elaboración de la especificación del caso de uso Registrar Inicio y Fin de Jornada para el Proyecto Route Lite de acuerdo al Estándar de Casos de Uso 1.4.  Folio CAI 0002968  (Route Lite, 1.0.0.0) | Nancy Elizabeth Villalobos Plascencia |
| 26/05/2014 | 1.1 | Se necesita un mensaje de que ya se leyó el código de barras de inicio de jornada. Esto resulta porque sin querer se volvía a capturar la clave pensando que era el inicio pero me cerro la jornada. En Windows Mobile si manda una advertencia de que si se desea cerrar jornada.  Folio CAI 0003076  (Route Lite, 1.0.0.0 y 1.1.0.0) | Belem Lizeth Jiménez Arévalo |
| 23/06/2014 | 1.2 | Se modificó la descripción del mensaje [E0489] para que en lugar de decir “El Código de Barras introducido no corresponde al Centro de Distribución” diga “El Código de Barras introducido no corresponde al $0$” de manera que sea genérico y pueda ser utilizado para otras validaciones.  Realizar los ajustes necesarios en los lugares donde es utilizado el mensaje de acuerdo a su nueva descripción.  Folio CAI 0003117  (eRoute 4.6.0.0, Route Lite 1.1.0.0) | Belem Lizeth Jiménez Arévalo |
| 18/09/2014 | 1.3 | - Ajustar el Inicio de Jornada para que recupere el Vendedor a partir del Usuario principal registrado en el archivo de configuración, y no a partir del usuario registrado en sesión.  - Agregar validación para que al momento de iniciar la jornada, se valide si existe un usuario sustituto logueado en el sistema, de ser así y si el sustituto registrado en sesión cuenta con un registro de sustitución vigente donde Enviado = 1, actualizar el campo Enviado = 0 para dicho registro.  Folio CAI 0003172  (La Costeña, 1.1.0.0) | Belem Lizeth Jiménez Arévalo |
| 24/04/2015 | 1.4 | 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 |
| 04/07/2017 | 1.5 | - Se requiere agregar una validación al finalizar jornada la cual deberá hacer referencia a la actividad de "Depósitos Vinculados", la cual consista en que no se pueda cerrar jornada si existe saldo por vincular dentro de esta actividad. Dicho de otra manera, NO deben existir abonos recibidos por vincular para poder cerrar jornada.  - Dicha validación deberá ser regida por un parámetro dentro de "ConfigParametro".  Folio CAI 0004257  (Nutec, 1.11.0.1) | Belem Lizeth Jiménez Arévalo |
| 06/07/2017 | 1.6 | Se requiere hacer un parámetro a nivel actividad de la terminal, para registrar tripulación al momento de fin de jornada. Si el parámetro está activo, al momento de finalizar jornada deberá solicitar la captura del registro de tripulación, en la cual deberá capturar la clave de vendedor o usuario para los siguientes roles: Chofer, Ayudante o Sobreayudante.  Al enviar deberá validar que si está el parámetro de registro de tripulación activo no permita envío de información si no está registrada la tripulación.  Folio CAI 0004260  (Lechera Gdl, 1.13.0.0 – 4.18.0.0) | Belem Lizeth Jiménez Arévalo |
| 20/04/2018 | 1.7 | Al finalizar la jornada de trabajo, se requiere que todos los abonos que no fueron vinculados a un Depósito se sumen y se guarden en la tabla SaldoDeposito y el Saldo será en positivo para indicar que es un Saldo que debe pagar el vendedor.  Folio CAI 0004779  (BYDSA, 1.16.0.0 - 4.21.0.0) | Belem Lizeth Jiménez Arévalo |
| 11/05/2018 | 1.8 | Agregar validación para no permitir finalizar jornada si no se ha registrado un conteo de inventario para el día de trabajo y éste corresponde al Día de Auditoría de Inventario configurado en ConfigParametro.  Folio CAI 0004825  (BYDSA, 1.16.0.0 - 4.21.0.0) | Belem Lizeth Jiménez Arévalo |
| 17/05/2018 | 1.9 | 1. Después de realizada la auditoria (conteo de inventario), se determinan las diferencias y en caso de que estas sean mayores a la cantidad máxima configurada en el sistema, se generara un registro de bloqueo de carga para el vendedor.  2. Al siguiente día del conteo, se subirá el registro de bloqueo de carga a la hand held. Cuando el vendedor entre a la actividad de cargas se validará si existe un registro de bloqueo de carga, en cuyo caso no le permitirá realizar la carga y solicitará autorización del supervisor.  3. El supervisor deberá introducir usuario y contraseña para quitar al vendedor del registro de bloqueo de carga y de esta forma pueda realizar su carga.  Folio CAI 0004858  (BYDSA, 1.16.0.0 - 4.21.0.0) | Belem Lizeth Jiménez Arévalo |
| 02/10/2018 | 1.10 | Folio CAI: 0005068  al cerrar jornada se valide que si hay una descarga o devolución en fase captura que no deje cerrar jornada  Folio CAI: 0005069  Al cerrar jornada se valide que si hay una descarga o devolución en fase captura que no deje cerrar jornada  (JER, 1.17.0.0 – 4.22.0.0) | Nancy Elizabeth Villalobos Plascencia |
| 21/04/2020 | 1.11 | Folio CAI 0005854  Se revisó en la base de datos para revisar los productos y realizar la validación del movimiento 23, el último movimiento 23 del mes de noviembre, el día 29/11/2019 se sacó la información del inventario final de ese día así mismo la primera recarga del día dan el inventario inicial, el cual coincide con el inventario inicial que ellos muestran.  (Corona Alteña, 4.23.1.6, 4.24.0.4, 4.25.0.0) | Belem Lizeth Jiménez Arévalo |
| 19/02/2021 | 1.12 | Folio CAI 0006613  Si se tiene activo el parámetro de control de inventario <CONHist.HabilitaInventario>, no permitir finalizar jornada si hay aún producto en mal estado en el inventario, para ello considerar que está activo por default el parámetro <CONHist.ValidaInv = 1> (no importa que en BD se encuentre inactivo).  (Duxstar, 4.27.0.0) | Belem Lizeth Jiménez Arévalo |

Tabla de Contenido

[Especificación de Casos de Uso: Registrar Inicio y Fin de Jornada – CUROLMOV43 6](#_Toc64666968)

[1 Introducción 6](#_Toc64666969)

[2 Caso de uso: Registrar Inicio y Fin de Jornada – CUROLMOV43 6](#_Toc64666970)

[2.1 Descripción 6](#_Toc64666973)

[3 Diagrama de Casos de Uso 6](#_Toc64666974)

[4 Precondiciones 6](#_Toc64666975)

[4.1 Actores 6](#_Toc64666978)

[4.2 Generales 6](#_Toc64666979)

[4.3 Parámetros 7](#_Toc64666980)

[5 Flujo de eventos 7](#_Toc64666981)

[5.1 Flujo básico 7](#_Toc64666983)

[5.2 Flujos alternos 17](#_Toc64666984)

[5.2.1 Opcionales 18](#_Toc64666985)

[5.2.2 Generales 18](#_Toc64666986)

[5.2.3 Validación 18](#_Toc64666987)

[6 Poscondiciones 18](#_Toc64666988)

[7 Firmas de Aceptación 18](#_Toc64666989)

# Especificación de Casos de Uso: Registrar Inicio y Fin de Jornada – CUROLMOV43

# 1 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*.*

# 2 Caso de uso: Registrar Inicio y Fin de Jornada – CUROLMOV43



## Descripción

Permite al usuario capturar el registro de Inicio y Fin de Jornada de Trabajo

# 3 Diagrama de Casos de Uso

**

# 4 Precondiciones



## Actores

1. Administrador

## Generales

1. El vendedor debe de tener un usuario registrado en el sistema.
2. El vendedor debe de estar activo.
3. El vendedor debe tener activo el inicio de Jornada de Trabajo

## Parámetros

N/A

# 5 Flujo de eventos



## Flujo básico

1. El caso de uso inicia cuando es invocado por el caso de uso:
   1. **[Seleccionar Actividades del Vendedor](C:\\Duxstar\\Productos\\RouteLite\\Trunk\\Analisis\\EspecificacionRequerimientos\\ECU_MOV\\CUROLMOV32_SeleccionarDíaDeTrabajo.docx)**
2. El sistema obtiene de sesión la siguiente información:
   1. sUSUId del usuario logeado en sistema.
   2. sDiaClave
3. El sistema obtiene del archivo de configuración XML la siguiente información:
   1. Usuario
   2. MostrarInfoCB
4. El sistema obtiene la siguiente información:
   1. **Vendedor** (donde Vendedor.USUId = Usuario.USUId y Usuario.Clave = Usuario registrado en el archivo de configuración (Usuario))
      1. VendedorID
      2. JornadaTrabajo
      3. USUId
      4. CodigoBarras
   2. **VendedorJornada** (donde VendedorJornada.VendedorID = Vendedor.VendedorID)
      1. VendedorID
      2. VEJFechaInicial
      3. DiaClave
      4. FechaFinal
   3. **ModuloTerm**
      1. TipoIndice
   4. **CONHist**
      1. VentaSinSurtir
5. Si <La Jornada ya fue iniciada (es decir si ya existe un registro en VendedorJornada que esté relacionado al Vendedor y que FechaFinal = NULL)>
   1. Si <el módulo es REPARTO y no se puede finalizar la jornada de trabajo hasta que se hayan surtido todas las ventas <ModuloTerm.TipoIndice = 3 y CONHist.VentaSinSurtir = 0>>
      1. El sistema obtiene la siguiente información donde (<TransProd.Tipo = 1 “Pedido” y TransProd.TipoFase = 1 “Captura” ó 7 “Captura Escritorio”>):
         1. **TransProd**
            1. Tipo
            2. TipoFase
            3. ClienteClave
            4. **Cliente**

ClienteClave

Clave

* + 1. Si <se obtuvo por lo menos un registro>
       1. El sistema presenta el mensaje “[E0751] – Los clientes $0$ tienen ventas por surtir”, sustituir la variable $0$ por las claves de los diferentes clientes <Cliente.Clave> de las ventas que no se han surtido <TransProd.ClienteClave>
       2. El sistema continúa en el [**paso 10**](#Paso_FIN) del flujo básico.
  1. Si <existe una configuración de parámetros activa para Validar Inventario al Finalizar Jornada <ConfigParametro.Parametro = ‘ValidarInvFinJornada> donde <ConfigParametro.Valor = 1>>
     1. El sistema obtiene los productos existentes en el catálogo de productos <Producto> cuya existencia lógica registrada en inventario obtenida a partir de lo siguiente, para ***PRODUCTO TERMINADO:*** (<Inventario.Disponible> donde <Inventario.ProductoClave = Producto.ProductoClave actual>), para ***PRODUCTO TIPO CONTENIDO O ENVASE:*** (<Inventario.Disponible – (<∑TransProdDetalle.PrestamoVendido \* ProductoDetalle.Factor> donde <TransProdDetalle.ProductoClave = Producto.ProductoClave y Producto.ProductoClave = ProductoDetalle.ProductoClave y ProductoDetalle.PRUTipoUnidad = TransProdDetalle.TipoUnidad actual y ProductoDetalle.ProductoDetClave = TransProdDetalle.ProductoClave donde ProductoDetalle.ProductoDetClave = Producto.ProductoClave> y <TransProdDetalle.TransProdId = TransProd.TransProdId> y <TransProd.Tipo = 1 ó 24> y <TransProd.TipoFase = 2 ó 3>)), *SEA DIFERENTE* de la existencia física de dicho producto, calculada a partir de las entrada y salidas efectuadas del mismo, de acuerdo a lo siguiente: ***PRODUCTO TERMINADO*** [(<∑TransProdDetalle.Cantidad \* ProductoDetalle.Factor> donde <TransProdDetalle.ProductoClave = Producto.ProductoClave y Producto.ProductoClave = ProductoDetalle.ProductoClave y ProductoDetalle.PRUTipoUnidad = TransProdDetalle.TipoUnidad actual y ProductoDetalle.ProductoDetClave = Producto.ProductoClave> y <TransProdDetalle.TransProdId = TransProd.TransProdId> y <TransProd.Tipo = 23> y <TransProd.TipoFase = 1>) + (<∑TransProdDetalle.Cantidad \* ProductoDetalle.Factor> donde <TransProdDetalle.ProductoClave = Producto.ProductoClave y Producto.ProductoClave = ProductoDetalle.ProductoClave y ProductoDetalle.PRUTipoUnidad = TransProdDetalle.TipoUnidad = actual y ProductoDetalle.ProductoDetClave = Producto.ProductoClave> y <TransProdDetalle.TransProdId = TransProd.TransProdId> y <TransProd.Tipo = 2> y <TransProd.TipoFase = 1>) + (<∑TransProdDetalle.Cantidad \* ProductoDetalle.Factor> donde <TransProdDetalle.ProductoClave = Producto.ProductoClave y Producto.ProductoClave = ProductoDetalle.ProductoClave y ProductoDetalle.PRUTipoUnidad = TransProdDetalle.TipoUnidad actual y ProductoDetalle.ProductoDetClave = Producto.ProductoClave> y <TransProdDetalle.TransProdId = TransProd.TransProdId> y <TransProd.Tipo = 3> y <TransProd.TipoFase = 1>) + (<∑TransProdDetalle.Cantidad \* ProductoDetalle.Factor> donde <TransProdDetalle.ProductoClave = Producto.ProductoClave y Producto.ProductoClave = ProductoDetalle.ProductoClave y ProductoDetalle.PRUTipoUnidad = TransProdDetalle.TipoUnidad actual y ProductoDetalle.ProductoDetClave = Producto.ProductoClave> y <TransProdDetalle.TransProdId = TransProd.TransProdId> y <TransProd.Tipo = 9> y <TransProd.TipoFase = 1> y <TransProd.TipoMovimiento = 1>) – (<∑TransProdDetalle.Cantidad \* ProductoDetalle.Factor> donde <TransProdDetalle.ProductoClave = Producto.ProductoClave y Producto.ProductoClave = ProductoDetalle.ProductoClave y ProductoDetalle.PRUTipoUnidad = TransProdDetalle.TipoUnidad actual y ProductoDetalle.ProductoDetClave = Producto.ProductoClave> y <TransProdDetalle.TransProdId = TransProd.TransProdId> y <TransProd.Tipo = 1> y <TransProd.TipoFase = 2 ó 3>) – (<∑TransProdDetalle.Cantidad \* ProductoDetalle.Factor> donde <TransProdDetalle.ProductoClave = Producto.ProductoClave y Producto.ProductoClave = ProductoDetalle.ProductoClave y ProductoDetalle.PRUTipoUnidad = TransProdDetalle.TipoUnidad actual y ProductoDetalle.ProductoDetClave = Producto.ProductoClave> y <TransProdDetalle.TransProdId = TransProd.TransProdId> y <TransProd.Tipo = 24> y <TransProd.TipoFase = 2>) – (<∑TransProdDetalle.Cantidad \* ProductoDetalle.Factor> donde <TransProdDetalle.ProductoClave = Producto.ProductoClave y Producto.ProductoClave = ProductoDetalle.ProductoClave y ProductoDetalle.PRUTipoUnidad = TransProdDetalle.TipoUnidad actual y ProductoDetalle.ProductoDetClave = Producto.ProductoClave> y <TransProdDetalle.TransProdId = TransProd.TransProdId> y <TransProd.Tipo = 4> y <TransProd.TipoFase = 1>) – (<∑TransProdDetalle.Cantidad \* ProductoDetalle.Factor> donde <TransProdDetalle.ProductoClave = Producto.ProductoClave y Producto.ProductoClave = ProductoDetalle.ProductoClave y ProductoDetalle.PRUTipoUnidad = TransProdDetalle.TipoUnidad actual y ProductoDetalle.ProductoDetClave = Producto.ProductoClave> y <TransProdDetalle.TransProdId = TransProd.TransProdId> y <TransProd.Tipo = 6> y <TransProd.TipoFase = 1>) – (<∑TransProdDetalle.Cantidad \* ProductoDetalle.Factor> donde <TransProdDetalle.ProductoClave = Producto.ProductoClave y Producto.ProductoClave = ProductoDetalle.ProductoClave y ProductoDetalle.PRUTipoUnidad = TransProdDetalle.TipoUnidad actual y ProductoDetalle.ProductoDetClave = Producto.ProductoClave> y <TransProdDetalle.TransProdId = TransProd.TransProdId> y <TransProd.Tipo = 7> y <TransProd.TipoFase = 1>) – (<∑TransProdDetalle.Cantidad \* ProductoDetalle.Factor> donde <TransProdDetalle.ProductoClave = Producto.ProductoClave y Producto.ProductoClave = ProductoDetalle.ProductoClave y ProductoDetalle.PRUTipoUnidad = TransProdDetalle.TipoUnidad actual y ProductoDetalle.ProductoDetClave = Producto.ProductoClave> y <TransProdDetalle.TransProdId = TransProd.TransProdId> y <TransProd.Tipo = 9> y <TransProd.TipoFase = 1> y <TransProd.TipoMovimiento = 2>)]. ***PRODUCTO TIPO CONTENIDO O ENVASE*** [(<∑TransProdDetalle.Cantidad \* ProductoDetalle.Factor> donde <TransProdDetalle.ProductoClave = Producto.ProductoClave y Producto.ProductoClave = ProductoDetalle.ProductoClave y ProductoDetalle.PRUTipoUnidad = TransProdDetalle.TipoUnidad actual y ProductoDetalle.ProductoDetClave <> TransProdDetalle.ProductoClave donde ProductoDetalle.ProductoDetClave = Producto.ProductoClave> y <TransProdDetalle.TransProdId = TransProd.TransProdId> y <TransProd.Tipo = 2> y <TransProd.TipoFase = 1>) + (<∑TransProdDetalle.Cantidad \* ProductoDetalle.Factor> donde <TransProdDetalle.ProductoClave = Producto.ProductoClave y Producto.ProductoClave = ProductoDetalle.ProductoClave y ProductoDetalle.PRUTipoUnidad = TransProdDetalle.TipoUnidad actual y ProductoDetalle.ProductoDetClave <> TransProdDetalle.ProductoClave donde ProductoDetalle.ProductoDetClave = Producto.ProductoClave> y <TransProdDetalle.TransProdId = TransProd.TransProdId> y <TransProd.Tipo = 3> y <TransProd.TipoFase = 1>) + (<∑TransProdDetalle.Cantidad \* ProductoDetalle.Factor> donde <TransProdDetalle.ProductoClave = Producto.ProductoClave y Producto.ProductoClave = ProductoDetalle.ProductoClave y ProductoDetalle.PRUTipoUnidad = TransProdDetalle.TipoUnidad actual y y ProductoDetalle.ProductoDetClave <> TransProdDetalle.ProductoClave donde ProductoDetalle.ProductoDetClave = Producto.ProductoClave> y <TransProdDetalle.TransProdId = TransProd.TransProdId> y <TransProd.Tipo = 9> y <TransProd.TipoFase = 1> y <TransProd.TipoMovimiento = 1>) – (<∑TransProdDetalle.Cantidad \* ProductoDetalle.Factor> donde <TransProdDetalle.ProductoClave = Producto.ProductoClave y Producto.ProductoClave = ProductoDetalle.ProductoClave y ProductoDetalle.PRUTipoUnidad = TransProdDetalle.TipoUnidad actual y y ProductoDetalle.ProductoDetClave <> TransProdDetalle.ProductoClave donde ProductoDetalle.ProductoDetClave = Producto.ProductoClave> y <TransProdDetalle.TransProdId = TransProd.TransProdId> y <TransProd.Tipo = 1> y <TransProd.TipoFase = 2 ó 3>) – (<∑TransProdDetalle.Cantidad \* ProductoDetalle.Factor> donde <TransProdDetalle.ProductoClave = Producto.ProductoClave y Producto.ProductoClave = ProductoDetalle.ProductoClave y ProductoDetalle.PRUTipoUnidad = TransProdDetalle.TipoUnidad actual y y ProductoDetalle.ProductoDetClave <> TransProdDetalle.ProductoClave donde ProductoDetalle.ProductoDetClave = Producto.ProductoClave> y <TransProdDetalle.TransProdId = TransProd.TransProdId> y <TransProd.Tipo = 24> y <TransProd.TipoFase = 2>) – (<∑TransProdDetalle.Cantidad \* ProductoDetalle.Factor> donde <TransProdDetalle.ProductoClave = Producto.ProductoClave y Producto.ProductoClave = ProductoDetalle.ProductoClave y ProductoDetalle.PRUTipoUnidad = TransProdDetalle.TipoUnidad actual y y ProductoDetalle.ProductoDetClave <> TransProdDetalle.ProductoClave donde ProductoDetalle.ProductoDetClave = Producto.ProductoClave> y <TransProdDetalle.TransProdId = TransProd.TransProdId> y <TransProd.Tipo = 4> y <TransProd.TipoFase = 1>) – (<∑TransProdDetalle.Cantidad \* ProductoDetalle.Factor> donde <TransProdDetalle.ProductoClave = Producto.ProductoClave y Producto.ProductoClave = ProductoDetalle.ProductoClave y ProductoDetalle.PRUTipoUnidad = TransProdDetalle.TipoUnidad actual y y ProductoDetalle.ProductoDetClave <> TransProdDetalle.ProductoClave donde ProductoDetalle.ProductoDetClave = Producto.ProductoClave> y <TransProdDetalle.TransProdId = TransProd.TransProdId> y <TransProd.Tipo = 6> y <TransProd.TipoFase = 1>) – (<∑TransProdDetalle.Cantidad \* ProductoDetalle.Factor> donde <TransProdDetalle.ProductoClave = Producto.ProductoClave y Producto.ProductoClave = ProductoDetalle.ProductoClave y ProductoDetalle.PRUTipoUnidad = TransProdDetalle.TipoUnidad actual y ProductoDetalle.ProductoDetClave <> TransProdDetalle.ProductoClave donde ProductoDetalle.ProductoDetClave = Producto.ProductoClave> y <TransProdDetalle.TransProdId = TransProd.TransProdId> y <TransProd.Tipo = 7> y <TransProd.TipoFase = 1>) – (<∑TransProdDetalle.Cantidad \* ProductoDetalle.Factor> donde <TransProdDetalle.ProductoClave = Producto.ProductoClave y Producto.ProductoClave = ProductoDetalle.ProductoClave y ProductoDetalle.PRUTipoUnidad = TransProdDetalle.TipoUnidad actual y ProductoDetalle.ProductoDetClave <> TransProdDetalle.ProductoClave donde ProductoDetalle.ProductoDetClave = Producto.ProductoClave> y <TransProdDetalle.TransProdId = TransProd.TransProdId> y <TransProd.Tipo = 9> y <TransProd.TipoFase = 1> y <TransProd.TipoMovimiento = 2>)]
     2. Si <se obtuvieron productos con existencia lógica diferente a la existencia física calculada por el sistema en el punto anterior>
        1. El sistema presenta el mensaje “[I0355] No se puede finalizar la jornada de trabajo debido a que los siguientes productos presentan diferencias en el inventario: $0$. Contacte a Soporte.”, sustituyendo el valor de $0$ por la concatenación de las claves de los productos con diferencias obtenidos <Producto.ProductoClave>, separados mediante comas.
        2. El sistema continúa en el [**paso 10**](#Paso_FIN) del flujo básico.
  2. El sistema obtiene la siguiente información:
     1. **MOTConfiguracion**
        1. ForzarCapturaImpro
        2. RegistrarTripulacion
     2. **Agenda** (donde Agenda.VendedorId = Vendedor.VendedorID)
        1. VendedorID
        2. ClienteClave
     3. Para cada cliente que se encuentre en la Agenda
        1. **Visita**
           1. VisitaClave
           2. VendedorID
           3. ClienteClave
  3. Si <por lo menos un cliente de la Agenda no tiene una Visita relacionada y si el parámetro de Forzar Captura de Improductividad está activo <MOTConfiguracion.ForzarCapturaImpro = 1>>
     1. El sistema presenta el mensaje “[I0270] – No se puede finalizar la jornada de trabajo, debe capturar una improductividad de venta para los clientes no visitados”
     2. El sistema continúa en el [**paso 10**](#Paso_FIN) del flujo básico.
  4. El sistema obtiene la siguiente información:
     1. **Agenda** (Donde Agenda.VendedorID = Vendedor.VendedorID y Dia.FueraFrecuencia = 0 (donde Dia.DiaClave = Agenda.DiaClave))
        1. VendedorID
        2. DiaClave
        3. **Dia**
           1. DiaClave
           2. FueraFrecuencia
     2. **VendedorJornada** (Donde VendedorJornada.VendedorId = Vendedor.VendedorID)
        1. VendedorID
        2. DiaClave
        3. VEJFechaInicial
        4. FechaFinal
     3. **CONHist**
        1. Inventario
        2. ValidaInv
  5. Si <existe un Día <Agenda.DiaClave> que se encuentre relacionado a la Agenda del Vendedor y que no esté finalizado <VendedorJornada.FechaFinal = NULL>, pero si existe un inicio de jornada <VendedorJornada.VEJFechaInicial != NULL.> y los demás días ya están finalizados y Si <no se debe conservar el inventario a bordo <CONHist.Inventario = 0>> y Si <se requiere que los inventarios estén en cero <CONHist.ValidaInv=1> y/o si está activo el parámetro de control de inventario <CONHist.HabilitaInventario = 1>>
     1. El sistema obtiene la siguiente información:
        1. **Inventario**
           1. Disponible
           2. NoDisponible
           3. Apartado
           4. Contenido
     2. Si <El Inventario Disponible, No Disponible, Apartado y Contenido son diferente de cero <Inventario.Disponible != 0>, <Inventario.NoDisponible != 0>, <Inventario.Apartado != 0> y <Inventario.Contenido !=0>>
        1. El sistema presenta el mensaje “[E0686] – La jornada solo podrá cerrarse cuando no haya inventario a bordo por vender o entregar al almacén”
        2. El sistema continúa en el [**paso 10**](#Paso_FIN) del flujo básico.
  6. El sistema obtiene la siguiente información:
     1. **ConfigParametro**
        1. Parametro = ‘ValidarAbnVinDep’
        2. Valor
     2. **ConfigParametro**
        1. Parametro = ‘ToleranciaAbnDepVin’
        2. Valor
  7. Si <se obtuvo una configuración de parámetros Activa para Validar Abonos Vinculados a Depósitos <ConfigParametro.Parametro = ‘ValidarAbnVinDep’> donde <ConfigParametro.Valor = 1>>
     1. El sistema obtiene la siguiente información:
        1. **Deposito**
           1. DEPId
           2. Total
        2. **ABNTrp**
           1. ABNId
           2. Importe
     2. Si <la sumatoria del importe total de los abonos obtenidos <∑ABNTrp.Importe> MENOS la sumatoria del importe total de los depósitos obtenidos <∑Deposito.Total> NO ES MENOR O IGUAL a <ConfigParametro.Valor> donde <ConfigParametro.Parametro = ‘ToleranciaAbnDepVin’>>
        1. El sistema presenta el mensaje “[I0312] – No se puede finalizar la jornada de trabajo ya que existe cobranza sin vincular a un depósito”
        2. El sistema continúa en el [**paso 10**](#Paso_FIN) del flujo básico.
  8. El sistema obtiene la siguiente información de la Configuración de Parámetros:
     1. **ConfigParametro**
        1. Parametro = ‘DiaAuditoriaInventario’
        2. Valor
  9. El sistema obtiene la siguiente información:
     1. **ConteoInventario**
        1. ContID
        2. FechaHoraAlta = Fecha actual del sistema (usar sólo la parte de la Fecha para comparar)
  10. Si <se obtuvo una configuración de parámetros para Día de Auditoría de Inventario <ConfigParametro.Parametro = ‘DiaAuditoriaInventario’> donde <ConfigParametro.Valor = Dia.Frecuencia> y <Dia.FechaCaptura = Fecha actual del sistema>> y si <no se obtuvo un conteo de inventario para la fecha actual>.
      1. El sistema presenta el mensaje “[I0334] – No se puede finalizar la jornada de trabajo, debe capturar el conteo de inventario ya que es día de auditoría”
      2. El sistema continúa en el [**paso 10**](#Paso_FIN) del flujo básico.
  11. El sistema obtiene la siguiente información de la Configuración de Parámetros:
      1. **ConfigParametro**
         1. Parametro = ‘ConfirmarDevAlmacen’
         2. Valor
      2. Si <la Confirmación de la Devolución al Almacen se encuentra activa <ConfigParametro.Valor = 1> donde <ConfigParametro.Parametro = ConfirmarDevAlmacen> >
         1. El sistema obtiene la siguiente información:
            1. **TransProd**

TransProdId

Tipo = 4 (Devolución al Almacén)

TipoFase = 15 (“Por Confirmar”)

* + - 1. Si <se obtuvo por lo menos un registro>
         1. El sistema presenta el mensaje “[E1015] – No se puede finalizar la jornada de trabajo, debe confirmar los movimientos pendientes de $0$”, sustituir $0$ por “Devolución al Almacén”.
    1. El sistema continúa en el [**paso 10**](#Paso_FIN) del flujo básico.
  1. El sistema obtiene la siguiente información de la Configuración de Parámetros:
     1. **ConfigParametro**
        1. Parametro = ‘ConfirmarDescarga’
        2. Valor
     2. Si <la Confirmación de la Descarga al Almacén se encuentra activa <ConfigParametro.Valor = 1> donde <ConfigParametro.Parametro = ConfirmarDescarga> >
        1. El sistema obtiene la siguiente información:
           1. **TransProd**

TransProdId

Tipo = 7 (Descarga al Almacén)

TipoFase = 15 (“Por Confirmar”)

* + - 1. Si <se obtuvo por lo menos un registro>
         1. El sistema presenta el mensaje “[E1015] – No se puede finalizar la jornada de trabajo, debe confirmar los movimientos pendientes de $0$”, sustituir $0$ por “Descarga al Almacén”.
  1. El sistema continúa en el [**paso 10**](#Paso_FIN) del flujo básico

1. El sistema presenta como título
   1. **Jornada de Trabajo**
2. Si <Vendedor.JornadaTrabajo =1 >
   1. Si <La Jornada no ha sido iniciada (es decir si no existe un registro en VendedorJornada que esté relacionado al Vendedor y que FechaFinal != NULL)>
      1. El sistema presenta la siguiente información:
         1. Iniciar Jornada de Trabajo
         2. El sistema solicita la siguiente información:
            1. Código Barras CEDI (\*)
      2. El sistema presenta la siguiente información ***de sólo lectura***:
         1. Finalizar Jornada de Trabajo
         2. El sistema solicita la siguiente información:
            1. Código Barras CEDI (\*)
      3. Si <existe un usuario sustituto logueado en el sistema, es decir, si el usuario registrado en sesión (sUSUId) es diferente del usuario registrado en el archivo de configuración (Usuario)>
         1. El sistema obtiene la siguiente información:
            1. **UsuarioSustituto**

SustitucionId

USUIdSustituto = sUSUId

FechaHoraInicio <= Fecha y hora actual del sistema

FechaHoraFin > Fecha y hora actual del sistema

Enviado = 1

* + - 1. Si <se obtuvo un registro de sustitución vigente para el usuario sustituto registrado en sesión donde Enviado = 1>
         1. El sistema actualiza la siguiente información para el registro de sustitución obtenido:

**UsuarioSustituto**

Enviado = 0

* 1. Si <La Jornada ya fue iniciada (es decir si ya existe un registro en VendedorJornada que esté relacionado al Vendedor y que FechaFinal = NULL)>
     1. El sistema presenta la siguiente información ***de sólo lectura***:
        1. Iniciar Jornada de Trabajo
        2. El sistema presenta y solicita la siguiente información:
           1. Código Barras CEDI (\*)
           2. FechaInicial = <VendedorJornada.VEJFechaInicial> (mostrar en formato dd/mm/aaaa hh:mm am/pm).
     2. El sistema presenta la siguiente información:
        1. Finalizar Jornada de Trabajo
        2. El sistema solicita la siguiente información:
           1. Código Barras CEDI (\*)
  2. Si <La Jornada ya fue iniciada y finalizada (es decir si existe un registro en VendedorJornada que esté relacionado al Vendedor y cuya FechaFinal <> NULL)>
     1. El sistema presenta la siguiente información ***de sólo lectura***:
        1. Iniciar Jornada de Trabajo
        2. El sistema presenta y solicita la siguiente información:
           1. Código Barras CEDI (\*)
           2. FechaInicial = <VendedorJornada.VEJFechaInicial> (mostrar en formato dd/mm/aaaa hh:mm am/pm).
     2. El sistema presenta la siguiente información ***de sólo lectura***:
        1. Finalizar Jornada de Trabajo
        2. El sistema presenta y solicita la siguiente información:
           1. Código Barras CEDI (\*)
           2. FechaFinal = <VendedorJornada.FechaFinal> (mostrar en formato dd/mm/aaaa hh:mm am/pm).
  3. El actor proporciona la siguiente información:
     1. Código Barras CEDI (\*) (Puede ser leída por el Scanner, si MostrarInfoCB = 0 se oculta la información con asteriscos)
     2. El sistema registra en sesión la siguiente información:
        1. sCodigoLeido

1. Si <el actor selecciona la opción **Continuar**>
   1. Si <La Jornada no ha sido iniciada (es decir si no existe un registro en VendedorJornada que esté relacionado al Vendedor y que FechaFinal != NULL)> o Si <La Jornada ya fue iniciada (es decir si ya existe un registro en VendedorJornada que esté relacionado al Vendedor y que FechaFinal = NULL)>
      1. Si <el no actor no proporciona la información requerida>
         1. El sistema presenta el mensaje “[BE0001] El campo $0$ es requerido“, sustituyendo el valor $0$ por el nombre del campo que no se haya proporcionado
         2. El sistema continúa en el paso correspondiente al campo que no se haya proporcionado
      2. El sistema valida el código de barras proporcionado por el usuario de acuerdo a la validación **[VA01 Validar Código Barras CEDI](#VA01)**
      3. Si <La Jornada no ha sido iniciada (es decir si no existe un registro en VendedorJornada que esté relacionado al Vendedor y que FechaFinal != NULL)>
         1. El sistema registra la siguiente información:
            1. **VendedorJornada**

VendedorID = <Vendedor.VendedorID>

VEJFechaInicial = Fecha y Hora actual del sistema

Enviado = 0

MFechaHora = Fecha y Hora actual del sistema

MUsuarioID = sUSUId

* 1. Si <La Jornada ya fue iniciada (es decir si ya existe un registro en VendedorJornada que esté relacionado al Vendedor y que FechaFinal = NULL)>
     1. El sistema obtiene la siguiente información de la Configuración de Parámetros:
        1. **ConfigParametro**
           1. Parametro = ‘RegistrarDepositoAdicional’
           2. Valor
     2. Si <se obtuvo una configuración de parámetros Activa para Registrar Deposito Adicional <ConfigParametro.Valor = 1> donde <ConfigParametro.Parametro = ‘RegistrarDepositoAdicional’>> y <Si existen abonos que no fueron vinculados a un depósito <ABNDetalle.ABNId y ABNDetalle.ABDId *no existen en* AbdDep.ABNId y AbdDep.ABDId>>
        1. El sistema registra la siguiente información:
           1. **SaldoDeposito**

SaldoDepositoID = Utilizar <Método KeyGen>

DiaClave = <VendedorJornada.DiaClave> donde <VendedorJornada.FechaFinal = NULL>

VendedorID = <VendedorJornada.VendedorID> donde <VendedorJornada.FechaFinal = NULL>

DepositoCapturaId = Null

Saldo = Sumatoria del importe de los abonos que no fueron vinculados a un depósito <∑ABNDetalle.Importe> donde <ABNDetalle.ABNId y ABNDetalle.ABDId no existen en AbdDep.ABNId y AbdDep.ABDId>

Aplicado = 0

FechaAplicacion = Null

DepositoAplicadoId = Null

MFechaHora = Fecha y hora actual del sistema

MUsuarioID = sUSUId

Enviado = 0

* + 1. El sistema obtiene la siguiente información de la Configuración de Parámetros:
       1. **ConfigParametro**
          1. Parametro = ‘FaltanteMaxPermitidoAuditoria’
          2. Valor
    2. Si <se obtuvo una configuración de parámetros para Faltante Máximo Permitido en Auditoría <ConfigParametro.Parametro = ‘FaltanteMaxPermitidoAuditoria’>>
       1. El sistema obtiene la siguiente información de la venta generada como cargo al vendedor por concepto de faltantes del conteo de inventario:
          1. **TransProd**

TransProdId

ClienteClave = <Vendedor.ClienteModelo> donde <Vendedor.USUId = sUSUId registrado en sesión>

Tipo = 1

FechaCaptura = Fecha actual del sistema

Total

* + - 1. Si <se obtuvo una venta generada como cargo al vendedor>
         1. El sistema registra en sesión la siguiente información:

ImporteFaltante = <TransProd.Total> de la venta obtenida

* + - 1. Si <NO se obtuvo una venta generada como cargo al vendedor>
         1. El sistema registra en sesión la siguiente información:

ImporteFaltante = 0

* + - 1. El sistema obtiene la siguiente información de la devolución generada como abono al vendedor por concepto de sobrantes del conteo de inventario:
         1. **TransProd**

TransProdId

ClienteClave = <Vendedor.ClienteModelo> donde <Vendedor.USUId = sUSUId registrado en sesión>

Tipo = 3

FechaCaptura = Fecha actual del sistema

Total

* + - 1. Si <se obtuvo una devolución generada como abono al vendedor>
         1. El sistema registra en sesión la siguiente información:

ImporteSobrante = <TransProd.Total> de la devolución obtenida

* + - 1. Si <NO se obtuvo una devolución generada como abono al vendedor>
         1. El sistema registra en sesión la siguiente información:

ImporteSobrante = 0

* + - 1. El sistema registra en sesión la siguiente información:
         1. Diferencia = *ImporteFaltante* – *ImporteSobrante* de acuerdo con la información registrada en sesión
      2. Si <la Diferencia resultante entre el importe faltante y el importe sobrante es mayor al valor del parámetro para Faltante Máximo Permitido en Auditoría <*Diferencia*  > ConfigParametro.Valor> donde <ConfigParametro.Parametro = ‘FaltanteMaxPermitidoAuditoria’>>
         1. El sistema presenta el mensaje “[I0335] La diferencia resultante del conteo de inventario fue de $0$. El Vendedor quedará bloqueado para cargar producto hasta que liquide dicho importe.”, sustituir la variable $0$ por el contenido de la variable *Diferencia* registrada en sesión.
         2. El sistema registra la siguiente información:

**BloqueoCargaVendedor**

DiaClave = <VendedorJornada.DiaClave> donde <VendedorJornada.FechaFinal = NULL>

VendedorID = <VendedorJornada.VendedorID> donde <VendedorJornada.FechaFinal = NULL>

Estado = 1

MFechaHora = Fecha y hora actual del sistema

MUsuarioID = sUSUId

Enviado = 0

* + 1. Si <el parámetro de configuración indica que se debe realizar el registro de tripulación al finalizar la jornada <MOTConfiguracion.RegistrarTripulacion = 1>>
       1. El sistema presenta una nueva pantalla con la siguiente información:
          1. El sistema presenta el siguiente título:

Registro de Tripulación

* + - * 1. El sistema presenta y solicita los siguientes campos:

**Rol (\*)(c):** Se deberá mostrar la descripción de los valores del valor por referencia <TIPOTRI>, de acuerdo al tipo de lenguaje configurado para el sistema. Si el actor selecciona un rol que ya se encuentra registrado en el objeto UsuariosTrip, se deberá recuperar la clave del usuario correspondiente y mostrarla en el campo “Usuario”, de manera que pueda ser modificada.

**Usuario (\*):** Se permite capturar la clave del usuario correspondiente al rol seleccionado. Si el actor proporciona una clave de usuario que no existe en el sistema <Usuario.Clave <> Usuario capturado por el actor> y abandona el campo, el sistema presenta el mensaje “[MDB050601] Nombre de usuario o contraseña no válido”, y continua en el paso donde fue invocado.

* + - * 1. El actor proporciona la siguiente información:

Rol

Usuario

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

El sistema presenta en una nueva pantalla un listado con la siguiente información de los Usuarios activos en el sistema, donde <Usuario.Activo = 1>:

**Clave:** Se presenta la clave del usuario <Usuario.Clave>.

**Nombre:** Se presenta el nombre del usuario <Usuario.Nombre>.

Si <el actor selecciona un usuario del listado>

El sistema regresa a la pantalla de Registro de Tripulación y muestra en el campo “Usuario” la Clave del Usuario seleccionado.

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

El sistema valida la información proporcionada por el usuario:

Si <el actor no proporciona la información requerida>

El sistema presenta el mensaje “[BE0001] El campo $0$ es requerido”, sustituir el valor de $0$ por el nombre del campo

El sistema continúa en el punto correspondiente al campo que no se haya proporcionado

El sistema registra/actualiza en sesión la siguiente información dentro del objeto UsuariosTrip (si la ClaveUsuario ya se encuentra registrada en el objeto con otro Rol, actualizar éste último de acuerdo al nuevo seleccionado):

Rol

ClaveUsuario

Para cada usuario incluido en el objeto UsuariosTrip

El sistema presenta en un listado la siguiente información:

**Rol:** Se presenta la descripción del rol, para ello ir a <VAVDescripcion> y obtener <VAVDescripcion.Descripcion> donde <VAVDescripcion.VARCodigo = ‘TIPOTRI’> y <VAVDescripcion.VAVClave = UsuariosTrip.Rol>, de acuerdo al tipo de lenguaje configurado para el sistema.

**Clave Usuario:** <UsuariosTrip.ClaveUsuario>.

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

Si <no se proporcionó por lo menos un usuario para la tripulación>

El sistema presenta el mensaje “[E0867] Se debe asignar por lo menos un (a) $0$”, sustituir el valor de $0$ por “usuario a la tripulación”

El sistema continúa en el paso donde fue invocado

Para cada usuario incluido en el objeto UsuariosTrip

Si <el usuario ya fue registrado como parte de la tripulación para la jornada de trabajo actual>

El sistema actualiza la siguiente información:

**Tripulacion**

TipoTripulación = UsuariosTrip.Rol

USUId = <Usuario.USUId> donde <Usuario.Clave = UsuariosTrip.ClaveUsuario>

MFechaHora = Fecha y Hora Actual del Sistema

MUsuarioID = sUSUId

Enviado = 0

Si <el usuario NO ha sido registrado como parte de la tripulación para la jornada de trabajo actual>

El sistema registra la siguiente información:

**Tripulacion**

TripId = Generar identificador único para la tripulación, utilizar <Método KeyGen>

VendedorID = <VendedorJornada.VendedorID> de la Jornada actual

VEJFechaInicial = <VendedorJornada.VEJFechaInicial> de la Jornada actual

TipoTripulación = UsuariosTrip.Rol

USUId = <Usuario.USUId> donde <Usuario.Clave = UsuariosTrip.ClaveUsuario>

MFechaHora = Fecha y Hora Actual del Sistema

MUsuarioID = sUSUId

Enviado = 0

* + 1. El sistema actualiza la siguiente información:
       1. **VendedorJornada**
          1. FechaFinal = Fecha y Hora actual del sistema
          2. Enviado = 0
          3. MFechaHora = Fecha y Hora Actual del Sistema
          4. MUsuarioID = sUSUId
  1. El sistema continúa en el [**paso 10**](#Paso_FIN) del flujo básico

1. Si <el actor selecciona la opción **Regresar**>
   1. Si <se realizaron cambios (se capturó un código de barras)>
      1. El sistema presenta el mensaje “BP0002 – Se perderán los cambios, ¿Está seguro de regresar?”
         1. Si <el usuario selecciona la opción **NO**>
            1. El sistema continúa en el paso anterior desde donde fue llamado
         2. Si <el usuario selecciona la opción **SI**>
            1. ~~El sistema extiende su funcionalidad al caso de uso:~~
            2. [**~~Seleccionar Actividades del Vendedor~~**](file:///C:\Duxstar\Productos\RouteLite\Trunk\Analisis\EspecificacionRequerimientos\ECU_MOV\CUROLMOV32_SeleccionarDíaDeTrabajo.docx)
            3. El sistema continúa en el [**paso 10**](#Paso_FIN) del flujo básico
   2. El sistema continúa en el [**paso 10**](#Paso_FIN) del flujo básico
2. Finaliza el caso de uso

## Flujos alternos

### Opcionales

N/A

### Generales

N/A

### Validación

#### [VA01 Validar Placa Capturada](#VA01_r)

1. Si <sCodigoLeido es diferente a Vendedor.CodigoBarras >
   1. El sistema presenta el mensaje “E0489 – El código de Barras introducido no corresponde al $0$”, sustituir el valor de $0$ por “Centro de Distribución”
   2. El sistema continúa en el paso anterior donde fue invocado
2. El sistema continúa en el siguiente paso desde donde fue invocado.

# 6 Poscondiciones

1. Registro y Actualización de **VendedorJornada**

# 7 Firmas de Aceptación

|  |
| --- |
|  |
| José María Alcalá |
| Gerente de Ingeniería / Departamento de Ingeniería – Duxstar Solutions |
| 07/03/2014 |

|  |
| --- |
|  |
| Nancy Elizabeth Villalobos Plascencia |
| Analista de Requerimientos / Departamento de Análisis y Diseño – Duxstar Solutions |
| 07/03/2014 |