**Proyecto Route Lite (ROL)**

Especificación del Caso de Uso: Modificar Precio – CUROLMOV37

VERSIÓN 1.0

**HISTÓRICO DE REVISIÓN**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
| 19/03/2014 | 1.0 | Elaboración de la especificación del caso de uso Modificar Precio para el Proyecto Route Lite de acuerdo al Estándar de Casos de Uso 1.4.  Folio CAI 0002741  (Route Lite (Grupo Premier), 1.0.0.0) | Belem Lizeth Jiménez Arévalo |
| 02/05/2014 | 1.1 | Folio CAI: 0003030  Se requiere agregar un parámetro para verificar si se valida que el precio modificado se encuentra entre el Precio y PrecioMinimo, además agregar la validación en el móvil para que no sobrepase estas cantidades, ni hacía arriba, ni hacía abajo.  (Route Lite, 1.0.0.0) | Nancy Elizabeth Villalobos Plascencia |
| 18/04/2018 | 1.2 | Folio CAI: 0004729  Al modificar el Precio Mostrar y Validar el Precio Máximo y Mínimo del Producto | Nancy Elizabeth Villalobos Plascencia |
| 19/09/2018 | 1.3 | Folio CAI: 0004982  Si se tiene configurada la el valor = 1, no debería validar el precio máximo, únicamente el mínimo, es decir debería permitir poner como precio máximo la cantidad que el vendedor desee  (AYA, 1.16.0.0) | Nancy Elizabeth Villalobos Plascencia |

Tabla de Contenido

[Especificación de Casos de Uso: Modificar Precio – CUROLMOV37 4](#_Toc387047186)

[1 Introducción 4](#_Toc387047187)

[2 Caso de uso: Modificar Precio – CUROLMOV37 4](#_Toc387047188)

[2.1 Descripción 4](#_Toc387047189)

[3 Diagrama de Casos de Uso 4](#_Toc387047190)

[4 Precondiciones 4](#_Toc387047191)

[4.1 Actores 4](#_Toc387047192)

[4.2 Generales 4](#_Toc387047193)

[5 Flujo de eventos 5](#_Toc387047194)

[5.1 Flujo básico 5](#_Toc387047195)

[5.2 Flujos alternos 6](#_Toc387047196)

[5.2.1 Opcionales 6](#_Toc387047197)

[5.2.1.1. AO01 Asignar Nuevo Precio al Producto 6](#_Toc387047206)

[6 Poscondiciones 7](#_Toc387047207)

[7 Firmas de Aceptación 8](#_Toc387047208)

# 37

# 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: Modificar Precio – CUROLMOV37

## Descripción

Permite al actor modificar el precio de lista de un producto, validando que el precio proporcionado no sea menor al precio mínimo permitido para el producto, y en caso de ser así, envía al servidor la solicitud para autorización del precio especial.

# 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 estar activo.
3. El actor debe de tener un perfil asignado para acceder al sistema.
4. Debe existir un pedido capturado para el cliente actual.
5. Deben existir impuestos vigentes configurados para los productos capturados en el pedido, o bien, para los esquemas a los que pertenecen estos productos.
6. Deben existir promociones vigentes para el cliente y/o productos capturados en el pedido.

# Flujo de eventos

## Flujo básico

1. El caso de uso inicia cuando es invocado por el siguiente caso de uso:
   1. [**Capturar Pedido – CUROLMOV18**](file:///C:\Duxstar\Productos\RouteLite\Trunk\Analisis\EspecificacionRequerimientos\ECU_MOV\CUROLMOV18_CapturarPedido.docx)
2. El sistema obtiene del objeto Productos registrado en sesión la siguiente información del producto seleccionado:
   1. ProductoClave
   2. Descripcion
   3. DecimalProducto
   4. Unidad
   5. Cantidad
   6. Precio
   7. Existencia
   8. Subtotal
3. El sistema obtiene de sesión la siguiente información del producto:
   1. **ArregloTransProdDetalle**
      1. TransProdDetalle
4. El caso de uso incluye la funcionalidad del caso de uso [**Determinar Lista de Precios – CUROLGEN04**](file:///C:\Amesol\Productos\RouteLite\Trunk\Analisis\EspecificacionRequerimientos\ECU_GEN\CUROLGEN04_DeterminarListaPrecio.doc)
   1. El sistema envía como parámetro la siguiente información
      1. ObtenerPrecioMinimo
         1. ListaPreciosPredeterminada = <TransProd.PCEPrecioClave> (objeto TransProd registrado en sesión correspondiente a la transacción a la que pertenece el producto (en caso de existir separación de pedidos por Subempresa))
         2. ProductoClave = ProductoClave (objeto Productos)
         3. PRUTipoUnidad = Unidad (objeto Productos)
   2. El sistema recibe como parámetro la siguiente información:
      1. PS\_PrecioProductoVig
   3. El sistema registra en sesión la siguiente información:
      1. PrecioMinimo = PS\_PrecioProductoVig
5. Si <Vendedor.PrecioMinimo = 2>
   1. El caso de uso incluye la funcionalidad del caso de uso [**Determinar Lista de Precios – CUROLGEN04**](file:///C:\Amesol\Productos\RouteLite\Trunk\Analisis\EspecificacionRequerimientos\ECU_GEN\CUROLGEN04_DeterminarListaPrecio.doc)
      1. El sistema envía como parámetro la siguiente información
         1. ObtenerPrecioSugerido
            1. ListaPreciosPredeterminada = <TransProd.PCEPrecioClave> (objeto TransProd registrado en sesión correspondiente a la transacción a la que pertenece el producto (en caso de existir separación de pedidos por Subempresa))
            2. ProductoClave = ProductoClave (objeto Productos)
            3. PRUTipoUnidad = Unidad (objeto Productos)
      2. El sistema recibe como parámetro la siguiente información:
         1. PS\_PrecioSugerido
      3. El sistema registra en sesión la siguiente información:
         1. PrecioSugerido = PS\_PrecioProductoVig
6. El sistema presenta el siguiente título de pantalla:
   1. Modificar Precio
7. Si <Vendedor.ModificarPrecio = 1>
   1. El sistema presenta la siguiente información:
      1. Precio**:** Se presenta el Precio de lista del producto seleccionado <Productos.Precio> (objeto Productos). Permitir modificar el precio de manera manual (formatearlo siempre al número de decimales que se especifica en el parámetro Decimales Importe <ConHist.DecimalesImporte>).
8. Si <Vendedor.ModificarPrecio = 2>
   1. El sistema presenta la siguiente información:
      1. **Precio Mínimo:** Se presenta el Precio Mínimo que se encuentra en sesión *\*\*Solo Lectura*
      2. **Precio:** Se presenta el Precio Sugerido que se encuentra en sesión. Permitir modificar el precio de manera manual (formatearlo siempre al número de decimales que se especifica en el parámetro Decimales Importe <ConHist.DecimalesImporte>).
      3. **Precio Máximo:** Se presenta el Precio de lista del producto seleccionado <Productos.Precio> (objeto Productos) *\*\*Solo Lectura*
9. Si <el actor selecciona la opción **Cancelar**>
   1. El sistema continúa en el [**paso**](#FIN) **9** del flujo básico
10. Si <el actor selecciona la opción **Aceptar**>
    1. Si <el actor **modificó el precio de lista** del producto>
       1. Si <Vendedor.ModificarPrecios = 1>
          1. El sistema obtiene la siguiente información:
             1. MOTConfiguracion

ValidaRangoPrecio

* + - 1. Si <ValidaRangoPrecio = 1>
         1. El sistema valida el Precio proporcionado por el actor:

Si <el Precio proporcionado por el actor es MAYOR o IGUAL al precio mínimo de lista permitido para el producto (PrecioMinimo) registrado en sesión> o Si <el Precio prorporcionado por el actor es MENOR o IGUAL al precio de lista <Productos.Precio> (objeto Productos)>

El sistema registra en sesión la siguiente información:

PrecioEspecial = Precio proporcionado por el actor

El sistema continúa con el flujo alterno opcional [**AO01 Asignar Nuevo Precio al Producto**](#_AO02_Actualizar_Precio)

Si <el Precio proporcionado por el actor es MENOR al precio mínimo de lista permitido para el producto (PrecioMinimo) registrado en sesión> ~~o Si <el Precio prorporcionado por el actor MAYOR al precio de lista <Productos.Precio> (objeto Productos)>~~

El sistema presenta el mensaje “[I0278] El precio debe ser Mayor o Igual al Precio Mínimo”

El sistema continúa en el [**paso 6.1**](#paso5_1) del flujo básico

* + - 1. Si <ValidaRangoPrecio = 0>
         1. El sistema valida el Precio proporcionado por el actor:

Si <el Precio proporcionado por el actor es MENOR a Cero>

El sistema presenta el mensaje “[E0007] La cantidad debe ser igual o mayor a cero”

El sistema continúa en el [**paso 6.1**](#paso5_1) del flujo básico

El sistema continúa con el flujo alterno opcional **[AO01 Asignar Nuevo Precio al Producto](#_AO02_Actualizar_Precio)**

* + 1. Si <Vendedor.ModificarPrecios = 2>
       1. El sistema obtiene la siguiente información:
          1. MOTConfiguracion

ValidaRangoPrecio

* + - 1. Si <ValidaRangoPrecio = 1>
         1. El sistema valida el Precio proporcionado por el actor:

Si <el Precio proporcionado por el actor es MAYOR o IGUAL al precio mínimo de lista permitido para el producto (PrecioMinimo) registrado en sesión> o Si <el Precio prorporcionado por el actor es MENOR o IGUAL al precio Máximo <Productos.Precio> (objeto Productos)>

El sistema registra en sesión la siguiente información:

PrecioEspecial = Precio proporcionado por el actor

El sistema continúa con el flujo alterno opcional [**AO01 Asignar Nuevo Precio al Producto**](#_AO02_Actualizar_Precio)

Si <el Precio proporcionado por el actor es MENOR al precio mínimo de lista permitido para el producto (PrecioMinimo) registrado en sesión> o Si <el Precio prorporcionado por el actor MAYOR al precio Máximo <Productos.Precio> (objeto Productos)>

El sistema presenta el mensaje “[I0333] El precio debe ser Mayor o Igual al Precio Mínimo y Menor o Igual al Precio Máximo”

El sistema continúa en el [**paso 6.1**](#paso5_1) del flujo básico

* + - 1. Si <ValidaRangoPrecio = 0>
         1. El sistema valida el Precio proporcionado por el actor:

Si <el Precio proporcionado por el actor es MENOR a Cero>

El sistema presenta el mensaje “[E0007] La cantidad debe ser igual o mayor a cero”

El sistema continúa en el [**paso 6.1**](#paso5_1) del flujo básico

El sistema continúa con el flujo alterno opcional [**AO01 Asignar Nuevo Precio al Producto**](#_AO02_Actualizar_Precio)

1. Finaliza el caso de uso

## Flujos alternos

### Opcionales



### [AO01 Asignar Nuevo Precio al Producto](#AO01_r)

1. El caso de uso incluye la funcionalidad del caso de uso [**Administrar Detalle de la Transacción – CUROLGEN06**](../ECU_GEN/CUROLGEN06_AdministrarDetalleTransaccion.doc)
   1. El sistema envía como parámetro la siguiente información:
      1. EliminarDetalleTransaccion
         * 1. TransProdID
           2. TransProdDetalleID
   2. El sistema recibe como parámetro la siguiente información:
      1. Verdadero
2. El sistema elimina del arreglo la información del objeto correspondiente al producto:
   1. **ArregloTransProdDetalle**
      1. TransProdDetalle
3. El caso de uso incluye la funcionalidad del caso de uso [**Administrar Detalle de la Transacción – CUROLGEN06**](../ECU_GEN/CUROLGEN06_AdministrarDetalleTransaccion.doc)
   1. El sistema envía como parámetro la siguiente información:
      1. GenerarDetallePedido
         1. TransProdID = <TransProd.TransProdID> (objeto TransProd)
         2. ProductoClave = ProductoClave (objeto Productos)
         3. TipoUnidad = Unidad (objeto Productos)
         4. Cantidad = Cantidad (objeto Productos)
         5. CantidadOriginal = 0
         6. PrecioClave = <TransProd.PCEPrecioClave> (objeto TransProd)
         7. PrecioEspecial = Precio proporcionado por el actor
   2. El sistema recibe como parámetro la siguiente información:
      1. PS\_TransProdDetalle
   3. El sistema registra la información del objeto recibido dentro del arreglo:
      1. **ArregloTransProdDetalle**
         1. TransProdDetalle
4. El sistema elimina del objeto Productos la siguiente información del producto:
   1. Productos
      1. ProductoClave
      2. Descripcion
      3. DecimalProducto
      4. Unidad
      5. Cantidad
      6. Precio
      7. Existencia
      8. Subtotal
5. El sistema obtiene la siguiente información del objeto recibido:
   1. **ArregloTransProdDetalle**
      1. ProductoClave
      2. Unidad
      3. Cantidad
      4. Precio
      5. Existencia
      6. Subtotal
6. El sistema registra la información obtenida dentro del objeto Productos
   1. Productos
      1. ProductoClave
      2. Unidad
      3. Cantidad
      4. Precio
      5. Existencia
      6. Subtotal
   2. El sistema obtiene la siguiente información del producto actual:
      1. **Producto**
         1. ProductoClave
         2. Nombre
         3. DecimalProducto
   3. El sistema actualiza el objeto Productos con la siguiente información para el producto actual:
      1. Productos
         1. Descripcion de acuerdo con la regla de negocio [**RNGEN187 Nombre Corto de un Producto**](file:///C:\Amesol\Productos\Route\Trunk\Analisis\EspecificacionRequerimientos\General\ERM_Reglas_de_Negocio.docx#RNGEN187)
         2. DecimalProducto
7. El caso de uso incluye la funcionalidad del caso de uso [**Administrar Transacción – CUROLGEN05**](file:///C:\Amesol\Productos\RouteLite\Trunk\Analisis\EspecificacionRequerimientos\ECU_GEN\CUROLGEN05_AdministrarTransaccion.doc)
   1. El sistema envía como parámetro la siguiente información:
      1. CalcularTotalesTransaccion
         1. TransProd = TransProd registrado en sesión (objeto TransProd correspondiente al Producto)
   2. El sistema recibe como parámetro la siguiente información:
      1. TransProd
   3. El sistema actualiza la información del objeto registrado en sesión:
      1. TransProd
8. El sistema continúa en el [**paso**](#FIN) **9** del flujo básico

# Poscondiciones

N/A

# Firmas de Aceptación

|  |
| --- |
|  |
| José María Alcalá |
| Gerente de Ingeniería / Departamento de Ingeniería – Duxstar Solutions |
| 19/03/2014 |

|  |
| --- |
|  |
| Belem Lizeth Jiménez Arévalo |
| Analista de Sistemas / Departamento de Análisis y Diseño – Duxstar Solutions |
| 19/03/2014 |