

ADMINISTRACIÓN CENTRAL DE MODERNIZACIÓN ADUANERA.

Manual de Referencia de la Interfase del WEB Service de Recintos Fiscalizados 2017

Histórico de revisiones

Fecha	Versión	Descripción	Autor
20-jul-2004	1.01	Contiene las operaciones de consulta de datos.	David Aguilera Cuevas
21-dic-2004	1.02	Contiene las operaciones de entradas y salidas de mercancía.	David Aguilera Cuevas
11-feb-2005	1.03	Contiene las operaciones de transferencia de mercancías y traspaleo de contenedores.	David Aguilera Cuevas
09-mar-2005	1.04	Contiene las operaciones de desconsolidación y registro de incidencias.	David Aguilera Cuevas
08-abr-2005	1.04.1	Cambios a la forma de operar transferencia de mercancía entre recintos.	David Aguilera Cuevas
18-abr-2005	1.05	Contiene las operaciones de entradas y salidas por una autoridad.	David Aguilera Cuevas
18-may-2005	1.05.1	Incorpora las consultas de mercancías en tránsito por transferencia.	David Aguilera Cuevas
16-jun-2005	1.06	Contiene las operaciones de separación y subdivisión de mercancía suelta.	David Aguilera Cuevas
07-sep-2005	1.06.1	Contiene a los métodos para consultar los eventos y las incidencias de una mercancía del recinto. Además permite obtener un manifiesto con el formato de la validación.	David Aguilera Cuevas
01-oct-2005	1.06.2	Cambia la salida de mercancía del recinto por pedimento. Ahora necesita registrar una operación de autorización	David Aguilera Cuevas

		de salida antes de ejecutar el registro de la salida.	
10-oct-2005	1.06.1.1	Contiene cambios a la búsqueda de manifiestos. Se le agregó el nuevo filtro opcional "nombre de buque".	David Aguilera Cuevas.
20-oct-2005	1.06.2.1	Conjunta las versiones 1.06.2 y 1.06.1.1 en esta versión.	David Aguilera Cuevas.
29-nov-2005	1.06.2.2	Incluye la especificación de los tipos de mercancía en la entrada al recinto. Registra el número de autorización de salida y el acuse de autorización de salida en la operación de registro de autorización de salida.	David Aguilera Cuevas.
23-dic-2005	1.06.2.3	Incluye las condiciones para permitir el uso de recitificaciones de pedimento en la autorización y registro de salidas	David Aguilera Cuevas.
17-ene-2006	1.06.2.4	Incluye las condiciones para validar pedimentos previos de consolidado, partes II y tránsitos en los métodos de autorización y registro de salidas.	David Aguilera Cuevas.
6-mar-2006	1.06.2.5	Incluye las condiciones para pedimentos rectificadas y para rectificaciones en la autorización y registro de salidas.	David Aguilera Cuevas.
29-ene-2007	1.06.2.6	Incluye el parámetro para marcar las autorizaciones de salida como preliberadas.	Santiago Pérez Bojorges.
03-dic-2007	1.06.2.7	Inlcuye las modificaciones para ajustarse al Manual de Registros SAM - M3 Versión 5.0.9.	Santiago Pérez Bojorges.
05-ago-2008	1.06.2.8	Cambios para validación de remesa y contingencia en los métodos de autorización y registro de salidas. Se	Santiago Pérez Bojorges.



		agrega método para consulta de ingresos por transferencia.	
19-mayo-2009	1.06.2.9	Incluye las consultas de Recintos Fiscalizados para pedimentos.	Edgar Malagón Calderón

MANUAL DE REFERENCIA DE LA INTERFASE DEL WEB SERVICE DE RECINTOS FISCALIZADOS VERSIÓN 1.06.2.9 ¡ERROR!
MARCADOR NO DEFINIDO.

Histórico de revisiones.....	2
Manual de referencia	16
Responsabilidades legales	17
Seguridad	18
Identificación y autenticación del usuario que hace la petición.....	19
Operaciones en mercancías del propio recinto.....	19
Registro de incidentes de seguridad y de operación.....	19
Registro de pistas de auditoría	19
1ª Etapa: Consulta de datos.....	20
Consulta manifiestos por rango de fechas	20
Método.....	20
Mensaje SOAP.....	20
Respuesta del método.....	21
Ejemplo con C#.....	26
Consulta un manifiesto específico.....	30
Método.....	30
Mensaje SOAP.....	30
Respuesta del método.....	31
Ejemplo con C#.....	35
Consulta los movimientos de un manifiesto.....	37
Método.....	37
Mensaje SOAP.....	37
Respuesta del método.....	38
Ejemplo con C#.....	40
Consulta los conocimientos de un manifiesto.....	42
Método.....	42
Mensaje SOAP.....	42
Respuesta del método.....	43

Ejemplo con C#	47
Consulta un conocimiento del recinto	52
Método	52
Mensaje SOAP	52
Respuesta del método	53
Ejemplo con C#	56
Consulta las mercancías de un conocimiento	58
Método	58
Mensaje SOAP	58
Respuesta del método	59
Ejemplo con C#	64
2ª Etapa: Administración de entradas y salidas de mercancía	68
Registrar arribo de buque	69
Método	69
Mensaje SOAP	69
Respuesta del método	70
Ejemplo con C#	71
Registra la entrada de mercancías al almacén	73
Mensaje SOAP	73
Para mercancía que está en el SAM	74
Método	75
Entrada de mercancía suelta	76
Respuesta de la entrada de mercancía suelta	77
Ejemplo con C#	78
Entrada de mercancía suelta a granel	79
Respuesta de la entrada de mercancía a granel	79
Ejemplo con C#	80
Entrada de mercancía contenerizada	82
Respuesta de la entrada de mercancía contenerizada	82

Ejemplo con C#	83
<u>Cancela la entrada de mercancías al almacén</u>	<u>85</u>
<u>Método</u>	<u>85</u>
<u>Mensaje SOAP</u>	<u>85</u>
<u>Respuesta del método</u>	<u>86</u>
<u>Ejemplo con C#</u>	<u>87</u>
<u>Registra la autorización de salida de mercancías del almacén</u>	<u>88</u>
<u>Método</u>	<u>88</u>
<u>Mensaje SOAP de autorización de salida con uno o más pedimentos</u>	<u>88</u>
<u>Respuesta del método</u>	<u>90</u>
<u>Ejemplo en C#</u>	<u>91</u>
<u>Cancela la autorización de salida de mercancías del almacén</u>	<u>93</u>
<u>Método</u>	<u>93</u>
<u>Mensaje SOAP</u>	<u>93</u>
<u>Respuesta del método</u>	<u>94</u>
<u>Ejemplo con C#</u>	<u>95</u>
<u>Registra la salida de mercancías del almacén</u>	<u>96</u>
<u>Método</u>	<u>96</u>
<u>Mensaje SOAP de salida con uno o más pedimentos</u>	<u>96</u>
<u>Respuesta del método</u>	<u>98</u>
<u>Ejemplo en C#</u>	<u>99</u>
<u>Cancela la salida de mercancías del almacén</u>	<u>101</u>
<u>Método</u>	<u>101</u>
<u>Mensaje SOAP</u>	<u>101</u>
<u>Respuesta del método</u>	<u>102</u>
<u>Ejemplo en C#</u>	<u>103</u>
<u>3ª Etapa Transferencia y Traspaleo</u>	<u>104</u>
<u>Transferencia de mercancía entre recintos</u>	<u>104</u>
<u>Primer paso. Registra la salida de transferencia de mercancías</u>	<u>105</u>
<u>Método</u>	<u>105</u>
<u>Mensaje SOAP</u>	<u>105</u>
<u>Respuesta del método</u>	<u>106</u>
<u>Ejemplo con C#</u>	<u>108</u>
<u>Cancela una salida de transferencia de mercancías</u>	<u>109</u>

Método.....	109
Mensaje SOAP.....	109
Respuesta del método.....	111
Ejemplo con C#.....	111
<u>Segundo paso. Consulta las salidas registradas por el recinto origen en transferencia</u>	113
Método.....	113
Mensaje SOAP.....	113
Respuesta del método.....	114
Ejemplo con C#.....	115
<u>Tercer paso. Registra la entrada de mercancías por transferencia al recinto destino</u>	117
Método.....	117
Mensaje SOAP.....	117
Respuesta del método.....	118
Ejemplo con C#.....	119
<u>Cancela la entrada de mercancías por transferencia al recinto destino</u>	121
Método.....	121
Mensaje SOAP.....	121
Respuesta del método.....	123
Ejemplo con C#.....	123
<u>Traspaleo de mercancía de uno o más contenedores a otro u otros contenedores</u>	125
<u>Primer paso. Registra el aviso de traspaleo de contenedores</u>	125
Método.....	125
Mensaje SOAP.....	126
Respuesta del método.....	97
Ejemplo con C#.....	98
<u>Cancela un aviso de traspaleo de contenedores</u>	133
Método.....	133
Mensaje SOAP.....	133
Respuesta del método.....	135
Ejemplo con C#.....	135
<u>Segundo paso. Registra la conclusión del traspaleo de uno o más contenedores</u>	138

Método.....	138
Mensaje SOAP.....	138
Respuesta del método.....	140
Ejemplo con C#.....	141
4ª Etapa Desconsolidación y registro de incidencias.....	145
Desconsolidación de un contenedor.....	145
Primer paso. Registra el aviso de desconsolidación de un contenedor.....	145
Método.....	145
Mensaje SOAP.....	145
Respuesta del método.....	146
Ejemplo con C#.....	147
Cancela el aviso de desconsolidación.....	149
Método.....	149
Mensaje SOAP.....	149
Respuesta del método.....	151
Ejemplo con C#.....	152
Segundo paso. Registra la conclusión de la desconsolidación del contenedor.....	153
Método.....	153
Mensaje SOAP.....	154
Respuesta del método.....	156
Ejemplo con C#.....	158
Registro de incidencias.....	161
Registra un aviso de incidencia a una mercancía.....	161
Método.....	161
Mensaje SOAP.....	161
Respuesta del método.....	162
Ejemplo con C#.....	163
Cancela una incidencia.....	165
Método.....	165
Mensaje SOAP.....	165
Respuesta del método.....	166

Ejemplo con C#	167
5ª Etapa: registro de entradas y salidas por órdenes de una autoridad.....	168
<u>Registra entradas por órdenes de una autoridad</u>	168
<u>Método</u>	168
<u>Mensaje SOAP</u>	169
<u>Respuesta del método</u>	171
<u>Ejemplo con C#</u>	173
<u>Cancela entradas por órdenes de una autoridad</u>	175
<u>Método</u>	175
<u>Mensaje SOAP</u>	175
<u>Respuesta del método</u>	177
<u>Ejemplo con C#</u>	177
<u>Registra la salida de mercancías por órdenes de una autoridad</u>	179
<u>Método</u>	179
<u>Mensaje SOAP de salida por órdenes de una autoridad</u>	179
<u>Respuesta del método</u>	180
<u>Ejemplo en C#</u>	181
<u>Cancela la salida por órdenes de una autoridad</u>	183
<u>Método</u>	183
<u>Mensaje SOAP</u>	183
<u>Respuesta del método</u>	185
<u>Ejemplo en C#</u>	185
6ª etapa Separación y Subdivisión de mercancía.....	187
<u>Separación de mercancía</u>	187
<u>Primer paso. Registra el aviso de separación</u>	187
<u>Método</u>	187
<u>Mensaje SOAP</u>	187
<u>Respuesta del método</u>	188
<u>Ejemplo con C#</u>	189
<u>Cancela un aviso de separación</u>	191
<u>Método</u>	191
<u>Mensaje SOAP</u>	191
<u>Respuesta del método</u>	193
<u>Ejemplo en C#</u>	193

<u>Segundo paso. Registra la conclusión de la separación</u>	195
<u>Método</u>	195
<u>Mensaje SOAP</u>	195
<u>Respuesta del método</u>	198
<u>Ejemplo con C#</u>	198
<u>Subdivisión de mercancía</u>	201
<u>Primer paso. Registra el aviso de subdivisión</u>	201
<u>Método</u>	201
<u>Mensaje SOAP</u>	201
<u>Respuesta del método</u>	202
<u>Ejemplo con C#</u>	203
<u>Cancela un aviso de subdivisión</u>	205
<u>Método</u>	205
<u>Mensaje SOAP</u>	205
<u>Respuesta del método</u>	207
<u>Ejemplo en C#</u>	207
<u>Segundo paso. Registra la conclusión de la subdivisión</u>	209
<u>Método</u>	209
<u>Mensaje SOAP</u>	210
<u>Respuesta del método</u>	211
<u>Ejemplo con C#</u>	213
<u>Eventos e incidencias</u>	215
<u>Eventos de una mercancía en el recinto</u>	215
<u>Método</u>	215
<u>Mensaje SOAP para obtener los eventos de una mercancía en el recinto</u>	215
<u>Respuesta del método</u>	217
<u>Ejemplo en C#</u>	219
<u>Incidencias de una mercancía en el recinto</u>	220
<u>Método</u>	220
<u>Mensaje SOAP para obtener las incidencias de una mercancía en el recinto</u>	220
<u>Respuesta del método</u>	221

Ejemplo en C#	222
Consulta de Información de un Manifiesto	223
Método.....	223
Mensaje SOAP.....	223
Respuesta del método.....	224
Ejemplo con C#	225
Consulta de Información de un Pedimento	226
Método.....	226
Mensaje SOAP.....	226
Respuesta del método.....	227
Ejemplo:.....	228
Ejemplo con C#	233
Consulta los datos de un contenedor	234
Método.....	234
Mensaje SOAP.....	234
Respuesta del método.....	235
Ejemplo en C#	236
Consulta de ingresos por transferencia	237
Método.....	237
Mensaje SOAP.....	237
Respuesta del método.....	239
Ejemplo en C#	244
Catálogo de errores.....	246
Descripción detallada de errores.....	251
-1 (MENOS 1).....	251
Descripción:.....	251
Solución:.....	251
-2 (MENOS 2).....	251
Descripción:.....	251
Solución:.....	251
2.....	251

Descripción:.....	251
Solución:.....	251
<u>3</u>	251
Descripción:.....	252
Solución:.....	252
<u>4</u>	252
Descripción:.....	252
Solución:.....	252
<u>5</u>	252
Descripción:.....	252
Solución:.....	252
<u>6</u>	252
Descripción:.....	252
Solución:.....	253
<u>7</u>	253
Descripción:.....	253
Solución:.....	253
<u>8</u>	253
Descripción:.....	253
Solución:.....	253
<u>9</u>	253
Descripción:.....	253
Solución:.....	253
<u>10</u>	253
Descripción:.....	254
Solución:.....	254
<u>11</u>	254
Descripción:.....	254
Solución:.....	254
<u>12</u>	254
Descripción:.....	254
Solución:.....	254
<u>13</u>	255
Descripción:.....	255

Solución:.....	255
<u>14</u>	255
<u>Descripción:</u>	255
<u>Solución:</u>	255
<u>15</u>	255
<u>Descripción:</u>	255
<u>Solución:</u>	255
<u>16</u>	255
<u>Descripción:</u>	256
<u>Solución:</u>	256
<u>17</u>	257
<u>Descripción:</u>	257
<u>Solución:</u>	257
<u>18</u>	257
<u>Descripción:</u>	257
<u>Solución:</u>	257
<u>19</u>	257
<u>Descripción:</u>	257
<u>Solución:</u>	257
<u>20</u>	258
<u>Descripción:</u>	258
<u>Solución:</u>	258
<u>21</u>	259
<u>Descripción:</u>	259
<u>Solución:</u>	259
<u>22</u>	259
<u>Descripción:</u>	259
<u>Solución:</u>	259
<u>23</u>	259
<u>Descripción:</u>	259
<u>Solución:</u>	259

<u>24</u>	260
<u>Descripción:</u>	260
<u>Solución:</u>	260
<u>25</u>	260
<u>Descripción:</u>	260
<u>Solución:</u>	260
<u>26</u>	260
<u>Descripción:</u>	260
<u>Solución:</u>	260

Manual de referencia

Web service de recintos fiscalizados versión 1.06.2.9

El presente manual está dirigido a todos los usuarios de los recintos fiscalizados que deben registrar las operaciones realizadas a las mercancías de importación dentro de sus almacenes.

Las operaciones cubiertas por este manual son las siguientes:

- Consulta de manifiestos y conocimientos registrados por el validador.
- Registro de entradas y salidas de mercancía en el recinto.
- Registro de transferencia de mercancía entre dos recintos fiscalizados.
- Registro de traspaleo de mercancía entre contenedores en el mismo recinto.
- Registro de desconsolidación de mercancías en el recinto.
- Registro de incidencias a las mercancías del recinto.
- Registro de entrada de mercancías por órdenes de una autoridad.
- Registro de salida de mercancías por órdenes de una autoridad.
- Registro de separación y subdivisión de mercancía suelta.
- Consulta de eventos e incidencias de mercancías en el recinto.

Cada método presenta un ejemplo en C#, para que el ejemplo funcione correctamente debes agregar una referencia web en tu proyecto .NET con la dirección <http://satcpanxch02/wsrefis/RecintosFis.asmx?WSDL>.

Los métodos del web service que atienden a cada una de estas operaciones se describe a continuación.

Responsabilidades legales

De acuerdo con las políticas de seguridad establecidas por el SAT, la información que genera en los repositorios institucionales y que entrega al recinto fiscalizado cada uno de los métodos definidos en este web service, está clasificada como confidencial y crítica por lo que es de entera responsabilidad del personal del recinto fiscalizado que la obtiene mantenerla en ese estado y cuidar que no se divulgue.

Seguridad

NOTA: Hasta esta versión del web service la característica de autenticación esta INHABILITADA por lo que no se llevará a cabo. Sin embargo, la identificación del usuario si se realiza por lo que cualquier llamada a los métodos de este web service deberán ser ejecutados por un usuario conocido, esto es, debe estar dado de alta en el catálogo de usuarios con el estado activo.

Dentro de cada petición de servicio, el usuario debe entregar los datos que le permitan al Web service identificarlo y autenticarlo, para esto, todos los métodos del Web service deben ser llamados a través de SOAP y la petición debe tener el siguiente formato:

```
POST /wsRecintos/RecintosFis.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/registraEntradaMerc"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <EncabezadoSOAP xmlns="http://tempuri.org/">
      <idUsuario>string</idUsuario>
      <firma>base64Binary</firma>
    </EncabezadoSOAP>
  </soap:Header>
  <soap:Body>
    .
    .
    .
  </soap:Body>
</soap:Envelope>
```

Identificación y autenticación del usuario que hace la petición

Esta petición resalta dos partes importantes: <soap:Header>...</soap:Header> y <soap:Body>...</soap:Body>. Dentro del encabezado (soap:Header) es donde se incluyen los datos que le permitirán al Web service identificar y autenticar al usuario que está solicitando el servicio:

- <idUsuario>...</idUsuario>. Contiene el identificador del usuario que está solicitando la ejecución de un método del Web service.
- <firma>...</firma>. Firma digital elaborada con los datos del soap:Body concatenados como un string.

En la parte del cuerpo del mensaje (soap:Body) se entregan los parámetros de ejecución para cada método.

Operaciones en mercancías del propio recinto

El recinto fiscalizado podrá usar como fuente sólo los datos que suministra el Validador de manifiestos. Esto es, podrá hacer operaciones únicamente con los manifiestos y con los conocimientos que haya registrado el Validador de manifiestos en la base de datos saai en su recinto fiscalizado excepto en el caso de transferencia de mercancía entre recintos fiscalizados.

Registro de incidentes de seguridad y de operación

Siempre que haya un error en cualquiera de los métodos, el Web service intentará registrarlo con los siguientes datos:

Fecha y hora en que sucedió el error.

Número consecutivo del error para la fecha en que sucedió el error.

Número de error (ver tabla de errores).

Descripción del error (ver tabla de errores).

En el catálogo de errores se establece la diferencia entre los incidentes de seguridad y los incidentes de operación.

Registro de pistas de auditoría

Cada método del Web service registra la pista de auditoría con los siguientes datos:

Servicio solicitado por el usuario.

Usuario que solicitó el servicio.

Fecha y hora en que se realizó el servicio.

1ª Etapa: Consulta de datos

A través de este Web service el recinto fiscalizado puede obtener datos de manifiestos de carga y conocimientos de embarque registrados por el Validador de manifiestos.

Los métodos públicos que forman parte de la interfase de recintos fiscalizados para consultar datos de manifiestos son los siguientes:

- **Consulta los datos generales de manifiestos por rango de fechas.**
- **Consulta los datos generales de un manifiesto específico.**
- **Consulta los movimientos de un manifiesto.**
- **Consulta los datos generales de los conocimientos de un manifiesto.**
- **Consulta los datos generales de un conocimiento específico de un manifiesto.**
- **Consulta las mercancías de un conocimiento específico.**

Consulta manifiestos por rango de fechas

Método

resManifiesto consultaManifPorFecha(string fechaIni, string fechaFin, string buque)

Parámetros:

fechaIni. Es la fecha inicial del rango de fechas.

fechaFin. Es la fecha final del rango de fechas.

buque. Es el nombre del buque por el que desea hacer un filtro opcional de búsqueda de manifiestos.

Las fechas de los parámetros debe estar en el formato DD-MM-AAAA.

Este método lo usa el recinto fiscalizado para obtener los manifiestos registrados por el Validador de manifiestos cuya fecha de validación esté dentro del rango de fechas indicado en los parámetros. Como un filtro opcional puede indicarse el nombre completo o parte del nombre del buque donde se piensa se declararon los manifiestos.

Mensaje SOAP

Este es el formato del mensaje SOAP para consultar los datos generales de manifiestos recibidos por el Validador de manifiestos dentro de un rango de fechas.

```
POST /wsRefis/RecintosFis.asmx HTTP/1.1
Host: satcpnpxch02
Content-Type: text/xml; charset=utf-8
Content-Length: length
```

```
SOAPAction: "http://Recintos/ServiciosWebXml/consultaManifPorFecha"

<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

  <soap:Header>
    <EncabezadoSOAP xmlns="http://Recintos/ServiciosWebXml/">
      <idUsuario>string</idUsuario>
      <firma>base64Binary</firma>
      <ipCliente>string</ipCliente>
    </EncabezadoSOAP>
  </soap:Header>
  <soap:Body>
    <consultaManifPorFecha xmlns="http://Recintos/ServiciosWebXml/">
      <fechaIni>string</fechaIni>
      <fechaFin>string</fechaFin>
      <buque>string</buque>
    </consultaManifPorFecha>
  </soap:Body>
</soap:Envelope>
```

Respuesta del método

El método regresa un objeto del tipo **resManifiesto**. Una vez que el método terminó de procesar la solicitud del usuario, regresa el resultado de la operación con el siguiente formato:

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
```

```
<soap:Body>
  <consultaManifPorFechaResponse xmlns="http://Recintos/ServiciosWebXml/">
    <consultaManifPorFechaResult>
      <error>int</error>
      <descripError>string</descripError>
      <registros>int</registros>
      <man>
        <nTrans>string</nTrans>
        <cTransp>string</cTransp>
        <idTransp>string</idTransp>
        <nomBuque>string</nomBuque>
        <numViaje>string</numViaje>
        <fecEstimada>string</fecEstimada>
        <tipoOper>string</tipoOper>
        <lineaNaviera>string</lineaNaviera>
        <fecRecibe>string</fecRecibe>
        <cuAduana>string</cuAduana>
        <cuPais>string</cuPais>
        <cuPuerto>string</cuPuerto>
        <cuEntidadMan>string</cuEntidadMan>
        <casos>

          <anyType />
          <anyType />
        </casos>
      </man>
      <manifiestos>
        <anyType />
        <anyType />
      </manifiestos>
    </consultaManifPorFechaResult>
  </consultaManifPorFechaResponse>
</soap:Body>
</soap:Envelope>
```

Donde:

error. Es el número de error que sucedió durante la consulta de manifiestos.

descripError. Es la descripción del error.

registros. Es el número de registros que resultaron de la consulta.

man. Es un objeto del tipo **objManifiesto**

manifiestos. Es un arreglo con elementos del tipo **objManifiesto**.

Estructura de **objManifiesto**:

```
<anyType xsi:type="objManifiesto">
  <nTrans>string</nTrans>
  <cTransp>string</cTransp>
  <idTransp>string</idTransp>
  <nomBuque>string</nomBuque>
  <numViaje>string</numViaje>
  <fecEstimada>string</fecEstimada>
  <tipoOper>string</tipoOper>
  <lineaNaviera>string</lineaNaviera>
  <fecRecibe>string</fecRecibe>
  <cuAduana>string</cuAduana>
  <cuPais>string</cuPais>
  <cuPuerto>string</cuPuerto>
  <cuEntidadMan>string</cuEntidadMan>
  <casos>
    <anyType />
    <anyType />
  </casos>
</anyType>
```

Donde:

nTrans. Es el número único de manifiesto.

cTransp. Es la clave del transporte.

idTransp. Es la identificación del transporte.

nomBuque. Es el nombre del buque.

numViaje. Es el número de viaje.

fecEstimada. Es la fecha estimada de arribo.
tipoOper. Es el tipo de operación.
lineaNaviera. Es el nombre de la línea naviera.
fecRecibe. Es la fecha de validación del manifiesto.
cuAduana. Es la clave de aduana donde se validó el manifiesto.
cuPais. Es la clave del país de arribo.
cuPuerto. Es la clave del puerto de arribo.
cuEntidadMan. Es la clave del agente EEEE.
casos. Son los casos relacionados al manifiesto.

El arreglo **casos** contiene a objetos del tipo **caso**:

```
<caso>  
  <idCasoNav>string</idCasoNav>  
  <cuCasoNav>string</cuCasoNav>  
</caso>
```

Donde:

idCasoNav. Es el identificador del caso.
cuCasoNav. Es la clave del caso.

Por ejemplo, una respuesta sin errores sería como ésta:

```
<?xml version="1.0" encoding="utf-8" ?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xmlns:xsd="http://www.w3.org/2001/XMLSchema">  
  <soap:Body>  
    <consultaManifPorFechaResponse xmlns="http://Recintos/ServiciosWebXml/">  
      <consultaManifPorFechaResult>  
        <error>0</error>  
        <descripError />  
        <registros>3</registros>  
        <man>  
          <nTrans>11239707000518</nTrans>
```



```
<cTransp>2509</cTransp>
<idTransp>DGNB</idTransp>
<nomBuque>BONN EXPRESS</nomBuque>
<numViaje>41W37</numViaje>
<fecEstimada>29/09/2007 12:00:00 AM</fecEstimada>
<tipoOper>1</tipoOper>
<lineaNaviera>HAPAG-LLOYD AKTIENGESELLSCHAFT, HAMBURG</lineaNaviera>
<fecRecibe>15/09/2007 2:00:59 PM</fecRecibe>
<cuAduana>43</cuAduana>
<cuPais>DEU</cuPais>

<cuPuerto>VER</cuPuerto>
<cuEntidadMan>2397</cuEntidadMan>
<casos />
</man>
<manifiestos>
  <anyType xsi:type="ObjManifiesto">
    <nTrans>11257307000015</nTrans>
    <cTransp>5069</cTransp>
    <idTransp>A8CJ5</idTransp>
    <nomBuque>MSC FLORIDA</nomBuque>
    <numViaje>15A</numViaje>
    <fecEstimada>03/09/2007 12:00:00 AM</fecEstimada>
    <tipoOper>1</tipoOper>
    <lineaNaviera>MEDITERRANEAN SHIPPING COMPANY SA</lineaNaviera>
    <fecRecibe>03/09/2007 11:47:46 AM</fecRecibe>
    <cuAduana>43</cuAduana>
    <cuPais>LBR</cuPais>
    <cuPuerto>VER</cuPuerto>
    <cuEntidadMan>2573</cuEntidadMan>
    <casos />
  </anyType>
  <anyType xsi:type="ObjManifiesto">
    <nTrans>11277907000009</nTrans>
    <cTransp>5440</cTransp>
    <idTransp>DICP</idTransp>
    <nomBuque>CAP POLONIO</nomBuque>
```

```
<numViaje>V-135-S</numViaje>
<fecEstimada>29/08/2007 12:00:00 AM</fecEstimada>
<tipoOper>2</tipoOper>
<lineaNaviera>ALIANCA NAVECAGAO E LOGISTICA LTDA.</lineaNaviera>
<fecRecibe>03/09/2007 12:37:39 PM</fecRecibe>
<cuAduana>43</cuAduana>
<cuPais>DEU</cuPais>
<cuPuerto>VER</cuPuerto>
<cuEntidadMan>2779</cuEntidadMan>
<casos />
</anyType>
<anyType xsi:type="ObjManifiesto">
  <nTrans>11239707000518</nTrans>
  <cTransp>2509</cTransp>
  <idTransp>DGNE</idTransp>

  <nomBuque>BONN EXPRESS</nomBuque>
  <numViaje>41W37</numViaje>
  <fecEstimada>29/09/2007 12:00:00 AM</fecEstimada>
  <tipoOper>1</tipoOper>
  <lineaNaviera>HAPAG-LLOYD AKTIENGESELLSCHAFT, HAMBURG</lineaNaviera>
  <fecRecibe>15/09/2007 2:00:59 PM</fecRecibe>
  <cuAduana>43</cuAduana>
  <cuPais>DEU</cuPais>
  <cuPuerto>VER</cuPuerto>
  <cuEntidadMan>2397</cuEntidadMan>
  <casos />
</anyType>
</manifiestos>
</consultaManifPorFechaResult>
</consultaManifPorFechaResponse>
</soap:Body>
</soap:Envelope>
```

EJEMPLO CON C#

```
private void manifRangoFechas ()
```

```
{  
    //Declara al proxy de recintos  
    ProxyRecintos.Recintos wsConsultas;  
  
    //Declara el resultado que regresa la consulta por fechas.  
    ProxyRecintos.resManifiesto res;  
  
    //Declara el encabezado del mensaje.  
    ProxyRecintos.EncabezadoSOAP encabezado = new ProxyRecintos.EncabezadoSOAP();  
    byte[] firma;  
  
    //En este momento no se está autenticando la firma del mensaje pero debe estar  
    //declarada como un arreglo de bytes.  
    firma = new byte[100];  
  
    //Crea una instancia del proxy.  
    wsConsultas = new ProxyRecintos.Recintos();  
  
    //Establece los valores del encabezado.  
    encabezado.idUsuario = Usuario;  
    encabezado.firma = firma;  
  
    //Asigna el encabezado a la instancia del proxy.  
    wsConsultas.EncabezadoSOAPValue = encabezado;  
  
    //Ejecuta el método de consulta de manifiestos por fecha.  
    res = (ProxyRecintos.resManifiesto) wsConsultas.consultaManifPorFecha(FechaIni, FechaFin, buque);  
  
    //Obtiene la descripción y el número del error (si lo hay).  
    TBNErrror.Text = res.error.ToString();  
    TBDescError.Text = res.descripcionError;  
  
    muestraTabla(res);  
}
```

Esta función muestra el resultado de las consultas de manifiestos.

```
private void muestraTabla(ProxyRecintos.resManifiesto res)
```

```
{  
    string[] columnas;  
  
    columnas = new string[13];  
    columnas[0] = "nTrans";  
    columnas[1] = "cTransp";
```

```
columnas[2] = "idTransp";
columnas[3] = "nomBuque";
columnas[4] = "numViaje";
columnas[5] = "fecEstimada";
columnas[6] = "tipoOper";
columnas[7] = "lineaNaviera";
columnas[8] = "fecRecibe";
columnas[9] = "cuAduana";
columnas[10] = "cuPais";
columnas[11] = "cuPuerto";
columnas[12] = "cuEntidadMan";

Response.Output.WriteLine("Error " + res.error + " " + res.descripcionError + "<br>");
Response.Output.WriteLine("Manifiestos encontrados: " + res.registros + "<BR>");

if(res.manifiestos != null)
{
    for(int i=0; i < res.manifiestos.Length;i++)
    {
        ProxyRecintos.ObjManifiesto objManif = (ProxyRecintos.ObjManifiesto)res.manifiestos[i];
        Response.Output.WriteLine("<table border=\"1\">");
        for(int j = 0; j< columnas.Length;j++)
        {
            Response.Output.WriteLine("<th bgColor=\"cyan\">" + columnas[j] + "</th>");
        }

        Response.Output.WriteLine("<tr>");
        Response.Output.WriteLine("<td>" + objManif.nTrans + "</td>");
        Response.Output.WriteLine("<td>" + objManif.cTransp + "</td>");
        Response.Output.WriteLine("<td>" + objManif.idTransp + "</td>");
        Response.Output.WriteLine("<td>" + objManif.nomBuque + "</td>");
        Response.Output.WriteLine("<td>" + objManif.numViaje + "</td>");
        Response.Output.WriteLine("<td>" + objManif.fecEstimada + "</td>");
        Response.Output.WriteLine("<td>" + objManif.tipoOper + "</td>");
        Response.Output.WriteLine("<td>" + objManif.lineaNaviera + "</td>");
        Response.Output.WriteLine("<td>" + objManif.fecRecibe + "</td>");
        Response.Output.WriteLine("<td>" + objManif.cuAduana + "</td>");
        Response.Output.WriteLine("<td>" + objManif.cuPais + "</td>");
        Response.Output.WriteLine("<td>" + objManif.cuPuerto + "</td>");
        Response.Output.WriteLine("<td>" + objManif.cuEntidadMan + "</td>");
        Response.Output.WriteLine("</tr>");
        Response.Output.WriteLine("</tr>");
        Response.Output.WriteLine("</table>");

        //Muestra casos del manifiesto
        muestraCasosManif(objManif);
    }
}
```

```
}  
}
```

```
private void muestraCasosManif(ProxyRecintos.ObjManifiesto objManif)  
{  
    ProxyRecintos.Caso objCaso;  
  
    if(objManif.casos.Length>0)  
    {  
        Response.Output.WriteLine("Casos del manifiesto: " + objManif.casos.Length + "<BR>");  
        Response.Output.WriteLine("<table border=\"1\">");  
        Response.Output.WriteLine("<th bgColor=\"yellow\">cuCaso</th>");  
        Response.Output.WriteLine("<th bgColor=\"yellow\">idCaso</th>");  
        for(int i=0; i<objManif.casos.Length;i++)  
        {  
            objCaso = (ProxyRecintos.Caso)objManif.casos[i];  
  
            Response.Output.WriteLine("<tr>");  
            Response.Output.WriteLine("<td>" + objCaso.cuCasoNav + "</td>");  
            Response.Output.WriteLine("<td>" + objCaso.idCasoNav + "</td>");  
            Response.Output.WriteLine("<tr>");  
        }  
        Response.Output.WriteLine("</table>");  
    }  
}
```

Consulta un manifiesto específico

MÉTODO

`resManifiesto consultaUnManifiesto(string manifiesto, string buque)`

Parámetros:

- **manifiesto.** Es el número único de manifiesto.
- **buque.** Es el nombre del buque por el que desea hacer un filtro opcional de búsqueda de manifiestos.

Este método lo usa el recinto fiscalizado para obtener los datos generales de un manifiesto específico registrado por el **Validador de manifiestos**. Como un filtro opcional puede indicarse el nombre completo o parte del nombre del buque donde se piensa se declaró el manifiesto.

MENSAJE SOAP

Este es el formato del mensaje SOAP para obtener un manifiesto específico registrado por el **Validador de manifiestos**.

```
POST /wsRefis/RecintosFis.asmx HTTP/1.1
Host: satcpnpxch02
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Recintos/ServiciosWebXml/consultaUnManifiesto"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <EncabezadoSOAP xmlns="http://Recintos/ServiciosWebXml/">
      <idUsuario>string</idUsuario>
      <firma>base64Binary</firma>
      <ipCliente>string</ipCliente>
    </EncabezadoSOAP>
  </soap:Header>
  <soap:Body>
    <consultaUnManifiesto xmlns="http://Recintos/ServiciosWebXml/">
      <manifiesto>string</manifiesto>
    </consultaUnManifiesto>
  </soap:Body>
</soap:Envelope>
```

```
<buque>string</buque>
</consultaUnManifiesto>
</soap:Body>
</soap:Envelope>
```

RESPUESTA DEL MÉTODO

El método regresa un objeto del tipo **resManifiesto**. Una vez que el método terminó de procesar la solicitud del usuario, regresa el resultado de la operación con el siguiente formato:

```
HTTP/1.1 200 OK
```

```
Content-Type: text/xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <consultaUnManifiestoResponse xmlns="http://Recintos/ServiciosWebXml/">
      <consultaUnManifiestoResult>
        <error>int</error>
        <descripError>string</descripError>
        <registros>int</registros>
        <man>
          <nTrans>string</nTrans>
          <cTransp>string</cTransp>
          <idTransp>string</idTransp>
          <nomBuque>string</nomBuque>
          <numViaje>string</numViaje>
          <fecEstimada>string</fecEstimada>
          <tipoOper>string</tipoOper>
          <lineaNaviera>string</lineaNaviera>
          <fecRecibe>string</fecRecibe>
          <cuAduana>string</cuAduana>
```

```
<cuPais>string</cuPais>
<cuPuerto>string</cuPuerto>
<cuEntidadMan>string</cuEntidadMan>
<casos>
  <anyType />
  <anyType />
</casos>
</man>
<manifiestos>
  <anyType />
  <anyType />
</manifiestos>
</consultaUnManifiestoResult>
</consultaUnManifiestoResponse>
</soap:Body>
</soap:Envelope>
```

Donde:

- error.** Es el número de error que sucedió durante la consulta de manifiestos.
- descripError.** Es la descripción del error.
- registros.** Es el número de registros que resultaron de la consulta.
- man.** Es un objeto del tipo **objManifiesto**
- manifiestos.** Es un arreglo con un solo elemento del tipo **objManifiesto**.

Estructura de **objManifiesto**:

```
<anyType xsi:type="objManifiesto">
  <nTrans>string</nTrans>
  <cTransp>string</cTransp>
  <idTransp>string</idTransp>
  <nomBuque>string</nomBuque>
  <numViaje>string</numViaje>
  <fecEstimada>string</fecEstimada>
  <tipoOper>string</tipoOper>
  <lineaNaviera>string</lineaNaviera>
```



```
<fecRecibe>string</fecRecibe>  
<cuAduana>string</cuAduana>  
<cuPais>string</cuPais>  
<cuPuerto>string</cuPuerto>  
<cuEntidadMan>string</cuEntidadMan>  
<casos>  
  <anyType />  
  <anyType />  
</casos>  
</anyType>
```

Donde:

- nTrans.** Es el número único de manifiesto.
- cTransp.** Es la clave del transporte.
- idTransp.** Es la identificación del transporte.
- nomBuque.** Es el nombre del buque.
- numViaje.** Es el número de viaje.
- fecEstimada.** Es la fecha estimada de arribo.
- tipoOper.** Es el tipo de operación.
- lineaNaviera.** Es el nombre de la línea naviera.
- fecRecibe.** Es la fecha de validación del manifiesto.
- cuAduana.** Es la clave de aduana donde se validó el manifiesto.
- cuPais.** Es la clave del país de arribo.
- cuPuerto.** Es la clave del puerto de arribo.
- cuEntidadMan.** Es la clave del agente EEEE.
- casos.** Son los casos relacionados al manifiesto.

El arreglo **casos** contiene a objetos del tipo **caso**:

```
<caso>  
  <idCasoNav>string</idCasoNav>  
  <cuCasoNav>string</cuCasoNav>  
</caso>
```

Donde:

idCasoNav. Es el identificador del caso.

cuCasoNav. Es la clave del caso.

Por ejemplo, una respuesta sin errores sería como ésta:

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <consultaUnManifiestoResponse xmlns="http://Recintos/ServiciosWebXml/">
      <consultaUnManifiestoResult>
        <error>0</error>
        <descripError />
        <registros>1</registros>
        <man>
          <nTrans>09239707000302</nTrans>
          <cTransp>2509</cTransp>
          <idTransp>VRCX7</idTransp>
          <nomBuque>MANILA EXPRESS</nomBuque>
          <numViaje>003E</numViaje>
          <fecEstimada>27/09/2007 12:00:00 a.m.</fecEstimada>
          <tipoOper>1</tipoOper>
          <lineaNaviera>HAPAG-LLOYD AKTIENGESELLSCHAFT, HAMBURG</lineaNaviera>
          <fecRecibe>12/11/2007 01:10:37 p.m.</fecRecibe>
          <cuAduana>43</cuAduana>
          <cuPais>HKG</cuPais>
          <cuPuerto>VER</cuPuerto>
          <cuEntidadMan>2397</cuEntidadMan>
          <casos />
        </man>
      </man>
      <manifiestos>
        <anyType xsi:type="ObjManifiesto">
          <nTrans>09239707000302</nTrans>
          <cTransp>2509</cTransp>
          <idTransp>VRCX7</idTransp>
        </anyType>
      </manifiestos>
    </consultaUnManifiestoResponse>
  </soap:Body>
</soap:Envelope>
```

```
<nomBuque>MANILA EXPRESS</nomBuque>
<numViaje>003E</numViaje>
<fecEstimada>27/09/2007 12:00:00 a.m.</fecEstimada>
<tipoOper>1</tipoOper>
<lineaNaviera>HAPAG-LLOYD AKTIENGESELLSCHAFT, HAMBURG</lineaNaviera>
<fecRecibe>12/11/2007 01:10:37 p.m.</fecRecibe>
<cuAduana>43</cuAduana>
<cuPais>HKG</cuPais>
<cuPuerto>VER</cuPuerto>
<cuEntidadMan>2397</cuEntidadMan>
```

```
                <casos />
            </anyType>
        </manifiestos>
    </consultaUnManifiestoResult>
</consultaUnManifiestoResponse>
</soap:Body>
</soap:Envelope>
```

EJEMPLO CON C#

```
private void unManifiesto()
{
    //Declara al proxy de recintos
    ProxyRecintos.Recintos wsConsultas;

    //Declara el resultado que regresa la consulta.
    ProxyRecintos.resManifiesto res;

    //Declara el encabezado del mensaje.
    ProxyRecintos.EncabezadoSOAP encabezado = new ProxyRecintos.EncabezadoSOAP();
```

```
//En este momento no se está autenticando la firma del mensaje pero debe estar
//declarada como un arreglo de bytes.
byte[] firma = new byte[64];

//Crea una instancia del proxy.
wsConsultas = new ProxyRecintos.Recintos();

//Establece lo valores del encabezado.
encabezado.idUsuario = Usuario;
encabezado.firma = firma;

//Asigna el encabezado a la instancia del proxy.
wsConsultas.EncabezadoSOAPValue = encabezado;

//Ejecuta el método de consulta de manifiestos.
res = (ProxyRecintos.resManifiesto) wsConsultas.consultaUnManifiesto(numManif, buque);
TBNErrror.Text = res.error.ToString();
TBDescError.Text = res.descripcionError;

muestraTabla(res);
}
```

CONSULTA LOS MOVIMIENTOS DE UN MANIFIESTO

MÉTODO

`resMovimiento consultaMovsDeManif(string manifiesto)`

Parámetros:

- **manifiesto**. Es el número único de manifiesto.

Este método lo usa el recinto fiscalizado para obtener los movimientos de un manifiesto específico registrado por el **Validador de manifiestos**.

MENSAJE SOAP

Este es el formato del mensaje SOAP para obtener los movimientos de un manifiesto específico registrados por el **Validador de manifiestos**:

```
POST /wsRefis/RecintosFis.asmx HTTP/1.1
Host: satcpnpxch02
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Recintos/ServiciosWebXml/consultaMovsDeManif"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <EncabezadoSOAP xmlns="http://Recintos/ServiciosWebXml/">
      <idUsuario>string</idUsuario>
      <firma>base64Binary</firma>
      <ipCliente>string</ipCliente>
    </EncabezadoSOAP>
  </soap:Header>
  <soap:Body>
    <consultaMovsDeManif xmlns="http://Recintos/ServiciosWebXml/">
      <manifiesto>string</manifiesto>
    </consultaMovsDeManif>
  </soap:Body>
</soap:Envelope>
```

```
</soap:Body>  
</soap:Envelope>
```

RESPUESTA DEL MÉTODO

Este método regresa un objeto del tipo **resMovimiento**. Una vez que el método terminó de procesar la solicitud del usuario, regresa el resultado de la operación con el siguiente formato:

```
HTTP/1.1 200 OK  
Content-Type: text/xml; charset=utf-8  
Content-Length: length  
  
<?xml version="1.0" encoding="utf-8"?>  
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
  <soap:Body>  
    <consultaMovsDeManifResponse xmlns="http://Recintos/ServiciosWebXml/">  
      <consultaMovsDeManifResult>  
        <error>int</error>  
        <descripError>string</descripError>  
        <registros>int</registros>  
        <movimientos>  
          <anyType />  
          <anyType />  
        </movimientos>  
      </consultaMovsDeManifResult>  
    </consultaMovsDeManifResponse>  
  </soap:Body>  
</soap:Envelope>
```

Donde:

error. Es el número de error que sucedió durante la consulta.

descripError. Es la descripción del error.

registros. Indica el número de registros resultado de la consulta.

movimientos. Es un arreglo con los movimientos realizados al manifiesto seleccionado.

El arreglo movimientos contiene a objetos del tipo **movimiento** que tiene la siguiente estructura:

```
<movimiento>  
  <fecha>string</fecha>  
  <cuMovimiento>string</cuMovimiento>  
</movimiento>
```

Donde:

fecha. Es la fecha en que ocurrió el movimiento.

cuMovimiento. Es la clave del movimiento.

Por ejemplo, una respuesta sin errores sería como ésta:

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <consultaMovsDeManifResponse xmlns="http://Recintos/ServiciosWebXml/">
      <consultaMovsDeManifResult>
        <error>0</error>
        <descripError />
        <registros>1</registros>
        <movimientos>
          <anyType xsi:type="Movimiento">
            <fecha>03/10/2007</fecha>
            <cuMovimiento>1</cuMovimiento>
          </anyType>
        </movimientos>
      </consultaMovsDeManifResult>
    </consultaMovsDeManifResponse>
  </soap:Body>
</soap:Envelope>
```

EJEMPLO CON C#

```
private void movsManif()
{
    ProxyRecintos.Recintos wsConsultas;
    ProxyRecintos.resMovimiento res;
    ProxyRecintos.EncabezadoSOAP encabezado = new ProxyRecintos.EncabezadoSOAP();
    byte[] firma = new byte[64];

    wsConsultas = new ProxyRecintos.Recintos();

    encabezado.idUsuario = Usuario;
    encabezado.firma = firma;
    wsConsultas.EncabezadoSOAPValue = encabezado;
    res = (ProxyRecintos.resMovimiento) wsConsultas.consultaMovsDeManif(numManif);
    TBNErrors.Text = res.error.ToString();
    TBDescError.Text = res.descripError;
```



```
Response.Output.WriteLine("Error " + res.error + " " + res.descripcion + "<br>");
Response.Output.WriteLine("Movimientos encontrados: " + res.registros + "<br>");

if(res.movimientos != null)
{
    Response.Output.WriteLine("<table border='1'>");
    Response.Output.WriteLine("<tr>");

    Response.Output.WriteLine("<th bgColor='cyan'>fecha</th>");
    Response.Output.WriteLine("<th bgColor='cyan'>cuMovimiento</th>");

    for(int i=0; i < res.movimientos.Length;i++)
    {
        ProxyRecintos.Movimiento objMov = (ProxyRecintos.Movimiento)res.movimientos[i];

        Response.Output.WriteLine("<tr>");
        Response.Output.WriteLine("<td>" + objMov.fecha + "</td>");
        Response.Output.WriteLine("<td>" + objMov.cuMovimiento + "</td>");
        Response.Output.WriteLine("</tr>");
    }
    Response.Output.WriteLine("</tr>");
    Response.Output.WriteLine("</table>");
}
}
```

CONSULTA LOS CONOCIMIENTOS DE UN MANIFIESTO

MÉTODO

resConocimiento consultaConocimDeManif(string manifiesto)

Parámetros:

- **manifiesto**. Es el número único de manifiesto.

Este método lo usa el recinto fiscalizado para obtener los datos generales de los conocimientos de un manifiesto específico registrados por el **Validador de manifiestos**.

MENSAJE SOAP

Este es el formato del mensaje SOAP para obtener los datos generales de los conocimientos de un manifiesto específico registrado por el **Validador de manifiestos**.

```
POST /wsRefis/RecintosFis.asmx HTTP/1.1
Host: satcpnpxch02
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Recintos/ServiciosWebXml/consultaConocimDeManif"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <EncabezadoSOAP xmlns="http://Recintos/ServiciosWebXml/">
      <idUsuario>string</idUsuario>
      <firma>base64Binary</firma>
      <ipCliente>string</ipCliente>
    </EncabezadoSOAP>
  </soap:Header>
  <soap:Body>
    <consultaConocimDeManif xmlns="http://Recintos/ServiciosWebXml/">
      <manifiesto>string</manifiesto>
    </consultaConocimDeManif>
  </soap:Body>
</soap:Envelope>
```

```
</soap:Body>  
</soap:Envelope>
```

RESPUESTA DEL MÉTODO

Este método regresa un objeto del tipo **resConocimiento**. Una vez que el método terminó de procesar la solicitud del usuario, regresa el resultado de la operación con el siguiente formato:

```
HTTP/1.1 200 OK  
Content-Type: text/xml; charset=utf-8  
Content-Length: length  
  
<?xml version="1.0" encoding="utf-8"?>  
  
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
  <soap:Body>  
    <consultaConocimDeManifResponse xmlns="http://Recintos/ServiciosWebXml/">  
      <consultaConocimDeManifResult>  
        <error>int</error>  
        <descripError>string</descripError>  
        <registros>int</registros>  
        <conocimientos>  
          <anyType />  
          <anyType />  
        </conocimientos>  
      </consultaConocimDeManifResult>  
    </consultaConocimDeManifResponse>  
  </soap:Body>  
</soap:Envelope>
```

Donde:

error. Es el número de error que sucedió durante la consulta.

descripError. Es la descripción del error.

registros. Indica el número de registros resultado de la consulta.

conocimientos. Es un arreglo que contiene a los conocimientos del manifiesto, con elementos del tipo **Conocimiento**.

Estructura de Conocimiento:

```
<anyType xsi:type="Conocimiento">
  <nTrans>string</nTrans>
  <cConocim>string</cConocim>
  <nConocim>string</nConocim>
  <cuPuertoCD>string</cuPuertoCD>
  <cuRefis>string</cuRefis>
  <cuAduana>string</cuAduana>
  <cuEntidadMan>string</cuEntidadMan>
  <cuPaisCD>string</cuPaisCD>
  <tipoConocim>string</tipoConocim>
  <conocimientoRef>string</conocimientoRef>
  <personas>
    <anyType />
    <anyType />
  </personas>
  <casos>
    <anyType />
    <anyType />
  </casos>
  <puertos>
    <anyType />
  </puertos>
</anyType>
```

Donde:

- nTrans.** Es el número único de manifiesto al que pertenece el conocimiento.
- cConocim.** Es el número de conocimiento asignado por el **Validador de manifiestos**.
- nConocim.** Es el número de conocimiento que le asignó el agente naviero.
- cuPuertoCD.** Es la clave del puerto de carga/descarga.
- cuRefis.** Es la clave del recinto fiscal.
- cuAduana.** Es la clave de la aduana.

- cuEntidadMan.** Es la clave del agente.
- cuPaisCD.** Es la clave del país.
- tipoConocim.** Es la clave del tipo de conocimiento (M = Master, H = House).
- conocimientoRef.** Es la identificación del conocimiento de referencia "Master".
- personas.** Contiene a las personas asociadas al conocimiento.
- casos.** Son los casos relacionados al conocimiento.
- puertos.** Contiene a los puertos relacionados con el conocimiento.

Los arreglos **personas**, **casos** y **puertos** tienen la siguiente estructura:
El arreglo **personas** contiene a objetos del tipo **persona**:

```
<persona>  
  <nTrans>string</nTrans>  
  <nombre>string</nombre>  
  <idFiscal>string</idFiscal>  
  <domicilio>string</domicilio>  
  <cuTipoPersona>string</cuTipoPersona>  
</persona>
```

Donde:

- nTrans.** Es el número único de manifiesto al que pertenece el conocimiento.
- nombre.** Tiene el nombre de la persona.
- idFiscal.** Es la identificación fiscal de la persona.
- domicilio.** Es el domicilio de la persona.
- cuTipoPersona.** Es la clave del tipo de persona.

El arreglo **casos** contiene a objetos del tipo **caso**:

```
<caso>  
  <idCasoNav>string</idCasoNav>  
  <cuCasoNav>string</cuCasoNav>  
</caso>
```

Donde:

idCasoNav. Es el identificador del caso.

cuCasoNav. Es la clave del caso.

El arreglo **puertos** contiene a objetos del tipo **puerto**:

```
<puerto>
  <cuPuerto>string</cuPuerto>
  <cuTipoPuerto>string</cuTipoPuerto>
  <cuPais>string</cuPais>
</puerto>
```

Donde:

cuPuerto. Es la clave del puerto.

cuTipoPuerto. Es la clave del tipo de puerto.

cuPais. Es la clave del país.

Por ejemplo, una respuesta sin errores sería como ésta:

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <consultaConocimDeManifResponse xmlns="http://Recintos/ServiciosWebXml/">
      <consultaConocimDeManifResult>
        <error>0</error>
        <descripError />
        <registros>2</registros>
        <conocimientos>
          <anyType xsi:type="Conocimiento">
            <nTrans>09239707000302</nTrans>
            <cConocim>1</cConocim>
          </anyType>
        </conocimientos>
      </consultaConocimDeManifResult>
    </consultaConocimDeManifResponse>
  </soap:Body>
</soap:Envelope>
```

```
<nConocim>HLCUHKG0708AXAR7</nConocim>
<cuPuertoCD>CWN</cuPuertoCD>
<cuRefis>63</cuRefis>
<cuAduana>43</cuAduana>
<cuEntidadMan>2397</cuEntidadMan>
<cuPaisCD>CHN</cuPaisCD>
<tipoConocim>M</tipoConocim>
<conocimientoRef></conocimientoRef>
<personas />
<casos />
<puertos />
</anyType>
<anyType xsi:type="Conocimiento">
  <nTrans>09239707000302</nTrans>
  <cConocim>2</cConocim>

                                <nConocim>HLCUHKG0708AXKA0</nConocim>
                                <cuPuertoCD>CWN</cuPuertoCD>
                                <cuRefis>63</cuRefis>
                                <cuAduana>43</cuAduana>
                                <cuEntidadMan>2397</cuEntidadMan>
                                <cuPaisCD>CHN</cuPaisCD>
                                <tipoConocim>M</tipoConocim>
                                <conocimientoRef></conocimientoRef>
                                <personas />
                                <casos />
                                <puertos />
                                </anyType>
                                </conocimientos>
                                </consultaConocimDeManifResult>
                                </consultaConocimDeManifResponse>
                                </soap:Body>
</soap:Envelope>
```

EJEMPLO CON C#

```
private void conocimManif()
{
    ProxyRecintos.Recintos wsConsultas;
    ProxyRecintos.resConocimiento res;
    ProxyRecintos.EncabezadoSOAP encabezado = new ProxyRecintos.EncabezadoSOAP();
    byte[] firma = new byte[64];

    wsConsultas = new ProxyRecintos.Recintos();

    encabezado.idUsuario = Usuario;
    encabezado.firma = firma;
    wsConsultas.EncabezadoSOAPValue = encabezado;

    res = (ProxyRecintos.resConocimiento) wsConsultas.consultaConocimDeManif(numManif);

    TBNErrors.Text = res.error.ToString();
    TBDescError.Text = res.descripcion;

    muestraTablaConocim(res);
}

private void muestraTablaConocim(ProxyRecintos.resConocimiento res)
{
    string[] columnas;

    columnas = new string[10];

    columnas[0] = "nTrans";
    columnas[1] = "cConocim";
    columnas[2] = "nConocim";
    columnas[3] = "cuPaisCD";
    columnas[4] = "cuPuertoCD";
    columnas[5] = "cuRefis";
    columnas[6] = "cuAduana";
    columnas[7] = "cuEntidadMan";
    columnas[8] = "tipoConocim";
    columnas[9] = "conocimientoRef";

    Response.Output.WriteLine("Error " + res.error + " " + res.descripcion + "<br>");
    Response.Output.WriteLine("Conocimientos: " + res.conocimientos.Length + "<BR>");
    if(res.conocimientos != null)
    {
        for(int i=0; i < res.conocimientos.Length;i++)
        {
            Response.Output.WriteLine("<table border=\"1\">");
        }
    }
}
```



```
Response.Output.WriteLine("<tr>");

for(int j = 0; j< columnas.Length;j++)
{
    Response.Output.WriteLine("<th bgColor=\"cyan\">" + columnas[j] + "</th>");
}

ProxyRecintos.Conocimiento objConocim = (ProxyRecintos.Conocimiento)res.conocimientos[i];
Response.Output.WriteLine("<tr>");
Response.Output.WriteLine("<td>" + objConocim.nTrans + "</td>");
Response.Output.WriteLine("<td>" + objConocim.cConocim + "</td>");
Response.Output.WriteLine("<td>" + objConocim.nConocim + "</td>");
Response.Output.WriteLine("<td>" + objConocim.cuPaisCD + "</td>");
Response.Output.WriteLine("<td>" + objConocim.cuPuertoCD + "</td>");
Response.Output.WriteLine("<td>" + objConocim.cuRefis + "</td>");
Response.Output.WriteLine("<td>" + objConocim.cuAduana + "</td>");
Response.Output.WriteLine("<td>" + objConocim.cuEntidadMan + "</td>");
Response.Output.WriteLine("<td>" + objConocim.tipoConocim + "</td>");
Response.Output.WriteLine("<td>" + objConocim.conocimientoRef + "</td>");
Response.Output.WriteLine("</tr>");
Response.Output.WriteLine("");
Response.Output.WriteLine("</tr>");
Response.Output.WriteLine("</table>");

// Muestra las personas del conocimiento
muestraPersonas(objConocim);

//Muestra los casos del conocimiento
muestraCasosConocim(objConocim);

//Muestra los puertos del conocimiento
muestraPuertosConocim(objConocim);
}
}

private void muestraPersonas(ProxyRecintos.Conocimiento objConocim)
{
    ProxyRecintos.Persona objPersona;

    Response.Output.WriteLine("Personas del conocimiento: " + objConocim.personas.Length + "<BR>");
    if(objConocim.personas.Length>0)
    {
        Response.Output.WriteLine("<table border=\"1\">");
        Response.Output.WriteLine("<th bgColor=\"yellow\">nTrans</th>");
        Response.Output.WriteLine("<th bgColor=\"yellow\">nombre</th>");
        Response.Output.WriteLine("<th bgColor=\"yellow\">idFiscal</th>");
    }
}
```

```
Response.Output.WriteLine("<th bgColor=\"yellow\">domicilio</th>");
Response.Output.WriteLine("<th bgColor=\"yellow\">cuTipoPersona</th>");
for(int i=0; i<objConocim.personas.Length; i++)
{
    objPersona = (ProxyRecintos.Persona)objConocim.personas[i];

    Response.Output.WriteLine("<tr>");
    Response.Output.WriteLine("<td>" + objPersona.nTrans + "</td>");
    Response.Output.WriteLine("<td>" + objPersona.nombre + "</td>");
    Response.Output.WriteLine("<td>" + objPersona.idFiscal + "</td>");
    Response.Output.WriteLine("<td>" + objPersona.domicilio + "</td>");
    Response.Output.WriteLine("<td>" + objPersona.cuTipoPersona + "</td>");

    Response.Output.WriteLine("</tr>");
    Response.Output.WriteLine("");
}
Response.Output.WriteLine("</table>");
}
}

private void muestraCasosConocim(ProxyRecintos.Conocimiento objConocim)
{
    ProxyRecintos.Caso objCaso;

    Response.Output.WriteLine("Casos del conocimiento: " + objConocim.casos.Length + "<BR>");
    if(objConocim.casos.Length>0)
    {
        Response.Output.WriteLine("<table border=\"1\">");
        Response.Output.WriteLine("<th bgColor=\"yellow\">idCasoNav</th>");
        Response.Output.WriteLine("<th bgColor=\"yellow\">cuCasoNav</th>");
        for(int i=0; i<objConocim.casos.Length; i++)
        {
            objCaso = (ProxyRecintos.Caso)objConocim.casos[i];

            Response.Output.WriteLine("<tr>");
            Response.Output.WriteLine("<td>" + objCaso.idCasoNav + "</td>");
            Response.Output.WriteLine("<td>" + objCaso.cuCasoNav + "</td>");

            Response.Output.WriteLine("</tr>");
            Response.Output.WriteLine("");
        }
        Response.Output.WriteLine("</table>");
    }
}
```

```
private void muestraPuertosConocim(ProxyRecintos.Conocimiento objConocim)
{
    ProxyRecintos.Puerto objPuerto;

    if(objConocim.puertos.Length>0)
    {
        Response.Output.WriteLine("Puertos del conocimiento: " + objConocim.puertos.Length + "<BR>");
        Response.Output.WriteLine("<table border=\"1\">");
        Response.Output.WriteLine("<th bgColor=\"yellow\">cuPais</th>");
        Response.Output.WriteLine("<th bgColor=\"yellow\">cuPuerto</th>");
        Response.Output.WriteLine("<th bgColor=\"yellow\">cuTipoPuerto</th>");
        for(int i=0; i<objConocim.puertos.Length; i++)
        {
            objPuerto = (ProxyRecintos.Puerto)objConocim.puertos[i];

            Response.Output.WriteLine("<tr>");
            Response.Output.WriteLine("<td>" + objPuerto.cuPais + "</td>");
            Response.Output.WriteLine("<td>" + objPuerto.cuPuerto + "</td>");
            Response.Output.WriteLine("<td>" + objPuerto.cuTipoPuerto + "</td>");

            Response.Output.WriteLine("</tr>");
            Response.Output.WriteLine("");
        }
        Response.Output.WriteLine("</table>");
    }
}
```

Consulta un conocimiento del recinto

MÉTODO

`resConocimiento consultaUnConocim(string conocimiento)`

Parámetros:

- **conocimiento**. Es el número de conocimiento asignado por el agente naviero.

Este método lo usa el recinto fiscalizado para obtener los datos generales de un conocimiento específico registrado por el **Validador de manifiestos**.

MENSAJE SOAP

Este es el formato del mensaje SOAP para obtener los datos generales de un conocimiento específico registrado por el **Validador de manifiestos**.

```
POST /wsRefis/RecintosFis.asmx HTTP/1.1
Host: satcpnpxch02
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Recintos/ServiciosWebXml/consultaUnConocim"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <EncabezadoSOAP xmlns="http://Recintos/ServiciosWebXml/">
      <idUsuario>string</idUsuario>
      <firma>base64Binary</firma>
      <ipCliente>string</ipCliente>
    </EncabezadoSOAP>
  </soap:Header>
  <soap:Body>
    <consultaUnConocim xmlns="http://Recintos/ServiciosWebXml/">
      <conocimiento>string</conocimiento>
    </consultaUnConocim>
  </soap:Body>
</soap:Envelope>
```

```
</soap:Body>  
</soap:Envelope>
```

RESPUESTA DEL MÉTODO

Este método regresa un objeto del tipo **resConocimiento**. Una vez que el método terminó de procesar la solicitud del usuario, regresa el resultado de la operación con el siguiente formato:

```
HTTP/1.1 200 OK  
Content-Type: text/xml; charset=utf-8  
Content-Length: length  
  
<?xml version="1.0" encoding="utf-8"?>  
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
  <soap:Body>  
    <consultaUnConocimResponse xmlns="http://Recintos/ServiciosWebXml/">  
      <consultaUnConocimResult>  
        <error>int</error>  
        <descripError>string</descripError>  
        <registros>int</registros>  
        <conocimientos>  
          <anyType />  
          <anyType />  
        </conocimientos>  
      </consultaUnConocimResult>  
    </consultaUnConocimResponse>  
  </soap:Body>  
</soap:Envelope>
```

Donde:

error. Es el número de error que sucedió durante la consulta.

descripError. Es la descripción del error.

registros. Indica el número de registros resultado de la consulta.

conocimientos. Es un arreglo que contiene a los conocimientos del manifiesto, con elementos del tipo **Conocimiento**.

Estructura de **Conocimiento**:

```
<anyType xsi:type="Conocimiento">
  <nTrans>string</nTrans>
  <cConocim>string</cConocim>
  <nConocim>string</nConocim>
  <cuPuertoCD>string</cuPuertoCD>
  <cuRefis>string</cuRefis>
  <cuAduana>string</cuAduana>
  <cuEntidadMan>string</cuEntidadMan>
  <cuPaisCD>string</cuPaisCD>
  <tipoConocim>string</tipoConocim>
  <conocimientoRef>string</conocimientoRef>
  <personas>
    <anyType />
    <anyType />
  </personas>
  <casos>
    <anyType />
    <anyType />
  </casos>
  <puertos>
    <anyType />
    <anyType />
  </puertos>
</anyType>
```

Donde:

- nTrans.** Es el número único de manifiesto al que pertenece el conocimiento.
- cConocim.** Es el número de conocimiento asignado por el **Validador de manifiestos**.
- nConocim.** Es el número de conocimiento que le asignó el agente naviero.
- cuPuertoCD.** Es la clave del puerto de carga/descarga.
- cuRefis.** Es la clave del recinto fiscal.
- cuAduana.** Es la clave de la aduana.
- cuEntidadMan.** Es la clave del agente.
- cuPaisCD.** Es la clave del país.
- tipoConocim.** Es la clave del tipo de conocimiento (M = Master, H = House).
- conocimientoRef.** Es la identificación del conocimiento de referencia "Master".

- personas.** Contiene a las personas asociadas al conocimiento.
- casos.** Son los casos relacionados al conocimiento.
- puertos.** Contiene a los puertos relacionados con el conocimiento.

Los arreglos **personas**, **casos** y **puertos** tienen la siguiente estructura:
El arreglo **personas** contiene a objetos del tipo **persona**:

```
<persona>  
  <nTrans>string</nTrans>  
  <nombre>string</nombre>  
  <idFiscal>string</idFiscal>  
  <domicilio>string</domicilio>  
  <cuTipoPersona>string</cuTipoPersona>  
</persona>
```

Donde:

- nTrans.** Es el número único de manifiesto al que pertenece el conocimiento.
- nombre.** Tiene el nombre de la persona.
- idFiscal.** Es la identificación fiscal de la persona.
- domicilio.** Es el domicilio de la persona.
- cuTipoPersona.** Es la clave del tipo de persona.

El arreglo **casos** contiene a objetos del tipo **caso**:

```
<caso>  
  <idCasoNav>string</idCasoNav>  
  <cuCasoNav>string</cuCasoNav>  
</caso>
```

Donde:

- idCasoNav.** Es el identificador del caso.

cuCasoNav. Es la clave del caso.

El arreglo **puertos** contiene a objetos del tipo **puerto**:

```
<puerto>
  <cuPuerto>string</cuPuerto>
  <cuTipoPuerto>string</cuTipoPuerto>
  <cuPais>string</cuPais>
</puerto>
```

Donde:

cuPuerto. Es la clave del puerto.

cuTipoPuerto. Es la clave del tipo de puerto.

cuPais. Es la clave del país.

Normalmente el resultado de esta consulta será sólo un registro pero el método puede entregar más, así que hay que revisar el número de elementos del arreglo **conocimientos** para saber cuántos conocimientos se encontraron en la consulta.

EJEMPLO CON C#

```
private void unConocim()
{
    ProxyRecintos.Recintos wsConsultas;
    ProxyRecintos.resConocimiento res;
    ProxyRecintos.EncabezadoSOAP encabezado = new ProxyRecintos.EncabezadoSOAP();
    byte[] firma = new byte[64];

    wsConsultas = new ProxyRecintos.Recintos();

    encabezado.idUsuario = Usuario;
    encabezado.firma = firma;
    wsConsultas.EncabezadoSOAPValue = encabezado;
    string numConocim = "SUDU8291";
    res = (ProxyRecintos.resConocimiento) wsConsultas.consultaUnConocim(numConocim);
    TBNErrror.Text = res.error.ToString();
    TBDescError.Text = res.descripcionError;

    muestraTablaConocim(res);
}
```


}

Consulta las mercancías de un conocimiento

MÉTODO

`resMercancia consultaMercDeConocim(string manifiesto, string conocimiento)`

Parámetros:

- **manifiesto**. Es el número único de manifiesto.
- **conocimiento**. Es el número de conocimiento asignado por **Validador de manifiestos** en la carga del conocimiento.

Este método lo usa el recinto fiscalizado para obtener los datos de las mercancías de un conocimiento específico registradas por el **Validador de manifiestos**.

MENSAJE SOAP

Este es el formato del mensaje SOAP para obtener los datos de las mercancías asociadas a un conocimiento de un manifiesto específico registrado por el **Validador de manifiestos**.

```
POST /wsRefis/RecintosFis.asmx HTTP/1.1
Host: satcpnpxch02
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Recintos/ServiciosWebXml/consultaMercDeConocim"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <EncabezadoSOAP xmlns="http://Recintos/ServiciosWebXml/">
      <idUsuario>string</idUsuario>
      <firma>base64Binary</firma>
      <ipCliente>string</ipCliente>
    </EncabezadoSOAP>
  </soap:Header>
  <soap:Body>
    <consultaMercDeConocim xmlns="http://Recintos/ServiciosWebXml/">
      <manifiesto>string</manifiesto>
      <conocimiento>string</conocimiento>
    </consultaMercDeConocim>
  </soap:Body>
</soap:Envelope>
```

```
</soap:Body>  
</soap:Envelope>
```

RESPUESTA DEL MÉTODO

Este método regresa un objeto del tipo **resMercancia**. Una vez que el método terminó de procesar la solicitud del usuario, regresa el resultado de la operación con el siguiente formato:

```
HTTP/1.1 200 OK  
Content-Type: text/xml; charset=utf-8
```

Content-Length: **length**

```
<?xml version="1.0" encoding="utf-8"?>  
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
  <soap:Body>  
    <consultaMercDeConocimResponse xmlns="http://Recintos/ServiciosWebXml/">  
      <consultaMercDeConocimResult>  
        <error>int</error>  
        <descripError>string</descripError>  
        <registros>int</registros>  
        <mercancias>  
          <anyType />  
          <anyType />  
        </mercancias>  
      </consultaMercDeConocimResult>  
    </consultaMercDeConocimResponse>  
  </soap:Body>  
</soap:Envelope>
```

Donde:

error. Es el número de error que sucedió durante la consulta.

descripError. Es la descripción del error.

registros. Indica el número de registros resultado de la consulta.
mercancias. Contiene a las mercancías relacionadas con el conocimiento.

El arreglo **mercancias** tiene una lista de objetos del tipo mercancía con la siguiente estructura:

```
<mercancia>  
  <cConocim>string</cConocim>  
  <cMercancia>string</cMercancia>  
  <marcas>string</marcas>  
  <totPiezas>string</totPiezas>  
  <pesoBruto>string</pesoBruto>  
  <cuEmbalaje>string</cuEmbalaje>  
  <cuUnidadMed>string</cuUnidadMed>  
  <nomMercancia>string</nomMercancia>  
  <tiposMercancia>  
    <anyType />  
    <anyType />  
  </tiposMercancia>  
  <contenedores>  
    <anyType />  
    <anyType />  
  </contenedores>  
</mercancia>
```

Donde:

cConocim. Es el número de conocimiento asignado por el **Validador de manifiestos**.
cMercancia. Es el número de mercancía asignado por el **Validador de manifiestos**.
marcas. Son las marcas que identifican a la mercancía.
totPiezas. Es el número de piezas de la mercancía declarado en el manifiesto.
pesoBruto. Es el peso de las piezas declarado en el manifiesto.
cuEmbalaje. Es la clave del embalaje de la mercancía.
cuUnidadMedida. Es la clave de unidad de medida del peso de la mercancía.

nomMercancia. Contiene la descripción de la mercancía.

tiposMercancia. Es un arreglo que contiene a los tipos de mercancía asignados a esta mercancía.

contenedores. Tiene la lista de los contenedores asociados a esta mercancía.

El arreglo **tiposMercancia** contiene a los tipos de mercancía que identifican a la mercancía con los siguientes datos:

```
<tiposMercancia>
  <cuTipoMerc>string</cuTipoMerc>
  <cuTipoMercPeligrosa>string</cuTipoMercPeligrosa>
  <claseMerc>string</claseMerc>
  <numONU>string</cuTipoMerc>
  <cuTipoMerc>string</numONU>
  <contactoTel>string</contactoTel>
  <contactoNombre>string</contactoNombre>
</tiposMercancia>
```

Donde:

cuTipoMerc. Es la clave del tipo de mercancía.

cuTipoMercPeligrosa. Es la clave del tipo de mercancía peligrosa.

claseMerc. Es el número único de sustancia o grupo de sustancias clasificadas en el código IMDG como lo requiere el transporte internacional de materiales de carga.

numONU. Es el código marítimo internacional de mercancías peligrosas conocido como código IMDG.

contactoTel. Es el número telefónico a llamar en caso de emergencia.

contactoNombre. Es el nombre de la persona de contacto en caso de emergencia.

El arreglo **contenedores** tiene la siguiente estructura:

```
<contenedores>
  <numContenedor>string</numContenedor>
  <cuTipoServicio>string</cuTipoServicio>
  <cuContenedor>string</cuContenedor>
  <pesoBruto>string</pesoBruto>
  <totalPiezas>string</totalPiezas>
  <sellos>
    <anyType />
    <anyType />
  </sellos>
</contenedores>
```

Donde:

numContenedor. Es el número de contenedor.

cuTipoServicio. Es la clave del tipo de servicio que da el contenedor.

cuContenedor. Es la clave del tipo de contenedor.

pesoBruto. Es el peso bruto en kilos de la mercancía en el contenedor incluyendo el peso del embalaje en el que se presenta.

totalPiezas. Es el número total de paquetes, piezas, bultos etc.

sellos. Contiene la relación de sellos del contenedor.

El arreglo **sellos** contiene a los sellos relacionados al contenedor y tiene la siguiente estructura:

```
<sellos>
  <numSello>string</numSello>
</sellos>
```

Donde:

numSello es el número de sello del contenedor.

Por ejemplo una respuesta sin errores sería como ésta:

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <consultaMercDeConocimResponse xmlns="http://Recintos/ServiciosWebXml/">
      <consultaMercDeConocimResult>
        <error>0</error>
        <descripError />
        <registros>1</registros>
        <mercancias>
          <anyType xsi:type="Mercancia">
            <cConocim>1</cConocim>
            <cMercancia>1</cMercancia>
            <marcas>LA MARCA X SIGNIFICAX</marcas>
            <totPiezas>3</totPiezas>
            <pesoBruto>400.00</pesoBruto>
            <cuEmbalaje>1</cuEmbalaje>
            <cuUnidadMed>1</cuUnidadMed>
            <nomMercancia>DESCRIPCION GENERAL DE LA MERCANCIA</nomMercancia>
            <tiposMercancia>
              <anyType xsi:type="TipoMercancia">
                <cuTipoMerc>12</cuTipoMerc>
                <cuTipoMercPeligrosa>11</cuTipoMercPeligrosa>
                <claseMerc>1.4</claseMerc>
                <numONU>UN1234</numONU>
                <contactoTel>01800123</contactoTel>
                <contactoNombre>JUAN PEREZ</contactoNombre>
              </anyType>
            </tiposMercancia>
            <contenedores>
              <anyType xsi:type="Contenedor">
                <cContenedorNav>1</cContenedorNav>
                <numContenedor>23</numContenedor>
                <cuTipoServicio>LCL</cuTipoServicio>
                <cuContenedor>1</cuContenedor>
                <pesoBruto>1400.430</pesoBruto>
              </anyType>
            </contenedores>
          </anyType>
        </mercancias>
      </consultaMercDeConocimResult>
    </consultaMercDeConocimResponse>
  </soap:Body>
</soap:Envelope>
```

```
<totalPiezas>88</totalPiezas>
<sellos>
  <anyType xsi:type="Sello">
    <numSello>1453</numSello>
  </anyType>
</sellos>
</anyType>
</contenedores>
</anyType>
</mercancias>
</consultaMercDeConocimResult>
</consultaMercDeConocimResponse>
</soap:Body>
</soap:Envelope>
```

EJEMPLO CON C#

```
private void mercancias()
{
    ProxyRecintos.Recintos wsConsultas;
    ProxyRecintos.resMercancia res;
    ProxyRecintos.EncabezadoSOAP encabezado = new ProxyRecintos.EncabezadoSOAP();
    byte[] firma = new byte[64];

    wsConsultas = new ProxyRecintos.Recintos();

    encabezado.idUsuario = Usuario;
    encabezado.firma = firma;
    wsConsultas.EncabezadoSOAPValue = encabezado;
    string numManif = "02256107000001";
    string numConocim = "2";

    res = (ProxyRecintos.resMercancia) wsConsultas.consultaMercDeConocim(numManif, numConocim);

    TBNErrors.Text = res.error.ToString();
    TBDescError.Text = res.descripcion;

    muestraTablaMercancia(res);
}

private void muestraTablaMercancia(ProxyRecintos.resMercancia res)
```



```
{
    ProxyRecintos.Mercancia objMerc;

    Response.Output.WriteLine("Error " + res.error + " " + res.descripcionError + "<br>");
    Response.Output.WriteLine("Mercancías del conocimiento: " + res.mercancias.Length + "<BR>");
    if(res.mercancias != null)
    {
        for(int i=0; i<res.mercancias.Length; i++)
        {
            objMerc = (ProxyRecintos.Mercancia)res.mercancias[i];

            Response.Output.WriteLine("<table border=\"1\">");
            Response.Output.WriteLine("<th bgColor=\"cyan\">cConocim</th>");
            Response.Output.WriteLine("<th bgColor=\"cyan\">cMercancia</th>");
            Response.Output.WriteLine("<th bgColor=\"cyan\">marcas</th>");
            Response.Output.WriteLine("<th bgColor=\"cyan\">nomMercancia</th>");
            Response.Output.WriteLine("<th bgColor=\"cyan\">totPiezas</th>");
            Response.Output.WriteLine("<th bgColor=\"cyan\">pesoBruto</th>");
            Response.Output.WriteLine("<th bgColor=\"cyan\">cuEmbalaje</th>");
            Response.Output.WriteLine("<th bgColor=\"cyan\">cuUnidadMed</th>");

            Response.Output.WriteLine("<tr>");
            Response.Output.WriteLine("<td>" + objMerc.cConocim + "</td>");
            Response.Output.WriteLine("<td>" + objMerc.cMercancia + "</td>");
            Response.Output.WriteLine("<td>" + objMerc.marcas + "</td>");
            Response.Output.WriteLine("<td>" + objMerc.nomMercancia + "</td>");
            Response.Output.WriteLine("<td>" + objMerc.totPiezas + "</td>");
            Response.Output.WriteLine("<td>" + objMerc.pesoBruto + "</td>");
            Response.Output.WriteLine("<td>" + objMerc.cuEmbalaje + "</td>");
            Response.Output.WriteLine("<td>" + objMerc.cuUnidadMed + "</td>");
            Response.Output.WriteLine("</tr>");
            Response.Output.WriteLine("</table>");

            //Muestra tipos de mercancía
            muestraTiposMercancia(objMerc);

            //Muestra los contenedores de la mercancía
            muestraContenedorMercancia(objMerc);
        }
    }
}
```

```
private void muestraTiposMercancia (ProxyRecintos.Mercancia objMerc)
{
    ProxyRecintos.TipoMercancia objTipo;

    Response.Output.WriteLine("Tipos de mercancía: " + objMerc.tiposMercancia.Length + "<BR>");
    if (objMerc.tiposMercancia.Length>0)
    {
        Response.Output.WriteLine("<table border=\"1\">");
        Response.Output.WriteLine("<th bgColor=\"yellow\">cuTipoMerc</th>");
        Response.Output.WriteLine("<th bgColor=\"yellow\">cuTipoMercPeligrosa</th>");
        Response.Output.WriteLine("<th bgColor=\"yellow\">claseMerc</th>");
        Response.Output.WriteLine("<th bgColor=\"yellow\">numONU</th>");
        Response.Output.WriteLine("<th bgColor=\"yellow\">contactoTel</th>");
        Response.Output.WriteLine("<th bgColor=\"yellow\">contactoNombre</th>");

        for(int i=0; i<objMerc.tiposMercancia.Length;i++)
        {
            objTipo = (ProxyRecintos.TipoMercancia) objMerc.tiposMercancia[i];

            Response.Output.WriteLine("<tr>");
            Response.Output.WriteLine("<td>" + objTipo.cuTipoMerc + "</td>");
            Response.Output.WriteLine("<td>" + objTipo.cuTipoMercPeligrosa + "</td>");
            Response.Output.WriteLine("<td>" + objTipo.claseMerc + "</td>");
            Response.Output.WriteLine("<td>" + objTipo.numONU + "</td>");
            Response.Output.WriteLine("<td>" + objTipo.contactoTel + "</td>");
            Response.Output.WriteLine("<td>" + objTipo.contactoNombre + "</td>");
            Response.Output.WriteLine("</tr>");

        }
        Response.Output.WriteLine("</table>");
    }
}

private void muestraContenedorMercancia (ProxyRecintos.Mercancia objMerc)
{
    ProxyRecintos.Contenedor objConten;

    Response.Output.WriteLine("Contenedores de mercancía: " + objMerc.contenedores.Length + "<BR>");
    for(int i=0; i<objMerc.contenedores.Length;i++)
    {
        objConten = (ProxyRecintos.Contenedor) objMerc.contenedores[i];

        Response.Output.WriteLine("<table border=\"1\">");
        Response.Output.WriteLine("<th bgColor=\"yellow\">numContenedor</th>");
        Response.Output.WriteLine("<th bgColor=\"yellow\">cuTipoServicio</th>");
        Response.Output.WriteLine("<th bgColor=\"yellow\">cuContenedor</th>");
        Response.Output.WriteLine("<th bgColor=\"yellow\">pesoBruto</th>");
        Response.Output.WriteLine("<th bgColor=\"yellow\">totalPiezas</th>");
    }
}
```

```
Response.Output.WriteLine("<tr>");  
Response.Output.WriteLine("<td>" + objConten.numContenedor + "</td>");  
Response.Output.WriteLine("<td>" + objConten.cuTipoServicio + "</td>");
```

```
Response.Output.WriteLine("<td>" + objConten.cuContenedor + "</td>");  
Response.Output.WriteLine("<td>" + objConten.pesoBruto + "</td>");  
Response.Output.WriteLine("<td>" + objConten.totalPiezas + "</td>");  
Response.Output.WriteLine("</tr>");
```

```
Response.Output.WriteLine("</table>");
```

```
//Muestra los sellos del contenedor  
muestraSellosContenedor(objConten);
```

```
}
```

```
}
```

```
private void muestraSellosContenedor(ProxyRecintos.Contenedor objConten)  
{  
    ProxyRecintos.Sello objSello;  
  
    Response.Output.WriteLine("Sellos del contenedor: " + objConten.sellos.Length + "<BR>");  
    if(objConten.sellos.Length>0)  
    {  
        Response.Output.WriteLine("<table border=\"1\">");  
        Response.Output.WriteLine("<th bgColor=\"yellow\">numSello</th>");  
        for(int i=0; i<objConten.sellos.Length;i++)  
        {  
            objSello = (ProxyRecintos.Sello) objConten.sellos[i];  
  
            Response.Output.WriteLine("<tr>");  
            Response.Output.WriteLine("<td>" + objSello.numSello + "</td>");  
            Response.Output.WriteLine("</tr>");  
        }  
        Response.Output.WriteLine("</table>");  
    }  
}
```

2ª Etapa: Administración de entradas y salidas de mercancía

La administración de entradas y salidas de mercancía del almacén del recinto se lleva a cabo mediante los siguientes métodos públicos del Web service:

- **Registra la entrada de mercancías al almacén.**
- **Cancela la entrada de mercancías al almacén.**
- **Registra la autorización de salida de mercancía.**
- **Cancela la autorización de salida de mercancía.**
- **Registra la salida de mercancías del almacén.**
- **Cancela la salida de mercancías al almacén.**

Registrar arribo de buque

El recinto fiscalizado puede usar este método para registrar el arribo de un buque.

MÉTODO

ResultadoArribo registraArriboBuqueSAAPA(ParametroArribo parametroArribo)

Parámetros:

ParametroArribo:

- **caat:** Código Alfanumérico Armonizado del Transportista (CAAT) del Agente Naviero
- **manifiesto:** Número Único de Manifiesto (Múltiples Registros).
- **aduana:** Aduana.
- **idTransporte:** Identificador del Transporte.
- **nombreEmbarcacion:** Nombre de la Embarcación.
- **numeroViaje:** Número de Viaje.
- **cuPais:** País de la Bandera del Transporte.
- **recinto:** Clave de Recinto Fiscalizado.

MENSAJE SOAP

Este es el formato del mensaje SOAP para registrar el arribo de buque..

```
POST /wsRecintos/RecintosFis.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Recintos/ServiciosWebXml/registraArriboBuqueSAAPA"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <EncabezadoSOAP xmlns="http://Recintos/ServiciosWebXml/">
      <idUsuario>string</idUsuario>
      <firma>base64Binary</firma>
      <ipCliente>string</ipCliente>
    </EncabezadoSOAP>
```

```
</soap:Header>
<soap:Body>
  <registraArriboBuqueSAAPA xmlns="http://Recintos/ServiciosWebXml/">
    <parametroArribo>
      <caat>string</caat>
      <manifiestos>
        <string>string</string>
        <string>string</string>
      </manifiestos>
      <aduana>int</aduana>
      <idTransporte>string</idTransporte>
      <nombreEmbarcacion>string</nombreEmbarcacion>

      <numeroViaje>string</numeroViaje>
      <cuPais>string</cuPais>
      <recinto>int</recinto>
    </parametroArribo>
  </registraArriboBuqueSAAPA>
</soap:Body>
</soap:Envelope>
```

RESPUESTA DEL MÉTODO

Este método regresa un objeto del tipo ResultadoArribo con el siguiente formato.

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <registraArriboBuqueSAAPAResponse xmlns="http://Recintos/ServiciosWebXml/">
      <registraArriboBuqueSAAPAResult>
        <error>int</error>
        <descripError>string</descripError>
```

```
<acuse>string</acuse>
  </registraArriboBuqueSAAPAResult>
</registraArriboBuqueSAAPAResponse>
</soap:Body>
</soap:Envelope>
```

Donde:

error. Es el número de error que sucedió durante la cancelación del ingreso de la mercancía al almacén.

descripcionError. Es una cadena de caracteres que tiene los errores de tipo general encontrados en el proceso. Cada error esta encerrado en los caracteres “[]”.

acuse. Acuse de que la operación se realizó exitosamente.

EJEMPLO CON C#

```
ProxyRecintos.Recintos wsRecintos = new ProxyRecintos.Recintos();
ProxyRecintos.EncabezadoSOAP encabezado = new ProxyRecintos.EncabezadoSOAP();
ProxyRecintos.ParametroArribo parametro = new ProxyRecintos.ParametroArribo();
```

```
ProxyRecintos.ResultadoArribo resultado;
```

```
int nManifiesto = 0;
if (txtManifiesto1.Text.Length > 0) nManifiesto++;
if (txtManifiesto2.Text.Length > 0) nManifiesto++;
if (txtManifiesto3.Text.Length > 0) nManifiesto++;
parametro.manifiestos = new string[nManifiesto];
```

```
nManifiesto = 0;
```

```
parametro.caat = txtCaat.Text;
if (txtManifiesto1.Text.Length > 0) {parametro.manifiestos[nManifiesto] =
txtManifiesto1.Text;nManifiesto++;}
if (txtManifiesto2.Text.Length > 0) {parametro.manifiestos[nManifiesto] =
txtManifiesto2.Text;nManifiesto++;}
```

```
if(txtManifiesto3.Text.Length > 0) {parametro.manifiestos[nManifiesto] =
txtManifiesto3.Text;nManifiesto++;}
parametro.aduana = int.Parse(txtAduana.Text);
parametro.idTransporte = txtIdTransporte.Text;
parametro.nombreEmbarcacion = txtNombreEmbarcacion.Text;
parametro.numeroViaje = txtNumeroViaje.Text;
parametro.cuPais = txtPais.Text;
parametro.recinto = txtRecinto.Text;

byte[] firma;

/* Genera una firma ficticia */
firma = new byte[64];
for(int i = 0; i < 64; i++)
    firma[i] = (byte)i;

encabezado.idUsuario = TBIdUsuario.Text;
encabezado.firma = firma;
wsRecintos.EncabezadoSOAPValue = encabezado;
resultado = wsRecintos.registraArriboBuqueSAAPA(parametro);

lblAcuse.Text = "Acuse: " + resultado.acuse;
lblError.Text = "Error: " + resultado.error;
```


Registra la entrada de mercancías al almacén

El recinto fiscalizado debe usar este método para registrar la entrada a su almacén de la mercancía declarada en un manifiesto.

MENSAJE SOAP

Este es el formato del mensaje SOAP para registrar las mercancías de un conocimiento en el almacén del recinto fiscalizado.

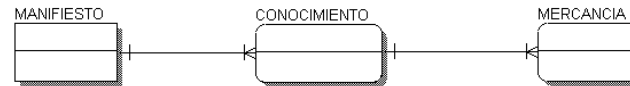
```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <EncabezadoSOAP xmlns="http://Recintos/ServiciosWebXml/">
      <idUsuario>string</idUsuario>
      <firma>base64Binary</firma>
      <ipCliente>string</ipCliente>
    </EncabezadoSOAP>
  </soap:Header>
  <soap:Body>
    <registraEntradaMerc xmlns="http://Recintos/ServiciosWebXml/">
      <paramEnt>
        <manifiesto>string</manifiesto>
        <nomConocimiento>string</nomConocimiento>
        <cuPais>string</cuPais>
        <cuPuerto>string</cuPuerto>
        <numAviso>string</numAviso>
        <autoridad>string</autoridad>
        <numDocumento>string</numDocumento>
        <tipoEntrada>int</tipoEntrada>
        <observacion>string</observacion>
        <listaMercancia>
          <ParametroSuelta>
            <entrada>int</entrada>
            <esParcial>boolean</esParcial>
            <mercancia>int</mercancia>
            <peso>double</peso>
            <piezas>int</piezas>
            <detalleMerc xsi:nil="true" />
          </ParametroSuelta>
        </listaMercancia>
      </paramEnt>
    </registraEntradaMerc>
  </soap:Body>
</soap:Envelope>
```

```
<ParametroSuelta>
  <entrada>int</entrada>
  <esParcial>boolean</esParcial>
  <mercancia>int</mercancia>
  <peso>double</peso>
  <piezas>int</piezas>
  <detalleMerc xsi:nil="true" />
</ParametroSuelta>
</listaMercancia>
<listaContenedor>

  <ParametroCont>
    <nomContenedor>string</nomContenedor>
    <peso>double</peso>
    <piezas>int</piezas>
    <tipoContenedor>string</tipoContenedor>
    <tipoServicio>string</tipoServicio>
    <listaSellos xsi:nil="true" />
    <detalleMerc xsi:nil="true" />
  </ParametroCont>
  <ParametroCont>
    <nomContenedor>string</nomContenedor>
    <peso>double</peso>
    <piezas>int</piezas>
    <tipoContenedor>string</tipoContenedor>
    <tipoServicio>string</tipoServicio>
    <listaSellos xsi:nil="true" />
    <detalleMerc xsi:nil="true" />
  </ParametroCont>
</listaContenedor>
<recintoOrigen>string</recintoOrigen>
</paramEnt>
</registraEntradaMerc>
</soap:Body>
</soap:Envelope>
```

PARA MERCANCÍA QUE ESTÁ EN EL SAM

Los datos fuente en este tipo de entradas de mercancía ya deben haber sido registrados por el validador de manifiestos en una estructura como ésta:



Donde un manifiesto puede tener uno o más conocimientos y un conocimiento pueden tener una o más mercancías.

- El manifiesto se identifica por el *numero único de manifiesto* asignado por la Asociación que transmite.
- El conocimiento se identifica por:
 - La clave del país de carga/descarga (**cuPais**).
 - La clave del puerto de carga/descarga (**cuPuerto**).
 - El nombre del conocimiento asignado por el agente naviero (**nomConocimiento**).
- La mercancía se identifica por el número consecutivo asignado en la validación (**mercancia**).

Es por esta razón que para identificar a una mercancía a ingresar al recinto, deben darse todos estos datos en cada método de entrada.

MÉTODO

EntradaMercSAM registraEntradaMerc(ParametroEntrada paramEnt)

Parámetros:

Objeto del tipo **ParametroEntrada** este objeto tiene la siguiente estructura:

- **manifiesto.**
- **nomConocimiento.**
- **cuPais.**
- **cuPuerto.**
- **autoridad.**
- **numAviso.**
- **numDocumento.**
- **tipoEntrada.**

- **observacion.**
- **recintoOrigen.**
- **listaMercancia.**
- **listaContenedor.**

Los parámetros que deben entregarse obligatoriamente dependen del tipo de entrada que se esté intentando registrar, en las siguientes páginas se explica con detalle el uso de los parámetros según el tipo de entrada.

Entrada de mercancía suelta

Con este tipo puede realizarse la entrada de mercancía que no esté contenerizada. Para hacer una entrada de mercancía suelta deben indicarse los siguientes parámetros en la estructura **ParametroEntrada**:

Identificador de la mercancía del manifiesto

- **manifiesto.** Es el número único de manifiesto.
- **nomConocimiento.** Es el identificador asignado al conocimiento por el agente naviero.
- **cuPais.** Es la clave del país de carga/descarga de la mercancía.
- **cuPuerto.** Es la clave del puerto de carga/descarga de la mercancía.
- **numAviso.** Este dato debe ser nulo.
- **numDocumento.** Este dato debe ser nulo.
- **tipoEntrada.** Este dato debe ser nulo.
- **listaMercancia** (arreglo de mercancía suelta)

El arreglo **listaMercancia** debe usarse siempre que la mercancía a ingresar no tenga como embalaje a un contenedor y debe contener a un objeto del tipo **ParametroSuelta** por cada mercancía suelta a ingresar al almacén.

- **ParametroSuelta**
 - **esParcial1.** Identifica si es una entrada parcial de mercancía. En este caso este valor es falso.
 - **mercancia.** Es el número de mercancía asignado por el validador de manifiestos.
 - **peso.** Indica el peso de la mercancía a ingresar. Este dato debe ser igual al peso declarado en el manifiesto.
 - **piezas.** Indica la cantidad de piezas de la mercancía a ingresar. Este dato debe ser igual al declarado en el manifiesto.
 - **detalleMerc.** En esta versión este dato es opcional. Contiene el detalle de la mercancía a ingresar.
 - **descripcion.** Este dato debe ser nulo.
 - **embalaje.** Este dato debe ser nulo.
 - **tipoMercancia.** Es un arreglo que contiene las claves del tipo de mercancía de acuerdo con el **Catálogo de tipos de mercancía**.
 - **cuTipoMerc.** Es la clave del tipo de mercancía de acuerdo con el **Catálogo de tipos de mercancía**.

- **listaContenedor**. Este dato debe ser nulo.

RESPUESTA DE LA ENTRADA DE MERCANCÍA SUELTA

Este método regresa un objeto del tipo **EntradaMercSAM** que tiene la siguiente estructura:

- **error**. Tiene el último número de error que sucedió al procesar al entrada de mercancías. Un cero significa que no hubo errores.
- **descripError**. Es una cadena de caracteres que tiene los errores de tipo general encontrados en el proceso. Cada error esta encerrado en los caracteres “[]”.
- **listaEntradas**. Contiene un arreglo con elementos del tipo **ResultadoEntrada** que tiene la siguiente estructura :
 - **consecutivo**. Es el número de mercancía declarado en **ParametroSuelta**.
 - **numEntrada**. Este es el número de control asignado por el web service al registrar la entrada de mercancía. De aquí en adelante la mercancía ingresada sólo podrá ser identificada por este número.
 - **acuse**. Es un dato calculado por el web service que le sirve al recinto fiscalizado como una evidencia de que el registro de la entrada de mercancía se realizó satisfactoriamente.
 - **listaErrores**. Es una cadena de caracteres que tiene los errores encontrados en el procesamiento de la entrada. Cada error esta encerrado en los caracteres “[]”.

Tendrás como resultado un elemento **ResultadoEntrada** por cada mercancía declarada en **ParametroSuelta** del arreglo **listaMercancía** de los parámetros de entrada de este método.

En caso de enviar mas de una vez el ingreso de una mercancía, la respuesta entregará un error además del número de entrada y el acuse.

EJEMPLO CON C#

```
//Crea los parámetros para la entrada
ParametroEntrada paramEnt = new ParametroEntrada();
paramEnt.listaMercancia = new ParametroSuelta[1];

ParametroSuelta suelta = new ParametroSuelta();
suelta.esParcial      = false;
suelta.mercancia      = 1;
suelta.peso            = 155920.80;
suelta.piezas         = 0;
paramEnt.listaMercancia[0] = suelta;

paramEnt.manifiesto = "09239707000302";
paramEnt.nomConocimiento = "KKLUZLO010080";
paramEnt.cuPais = "BEL";
paramEnt.cuPuerto = "ZEE";

//Genera el proxy del servicio
ProxyRecintos.Recintos recinto = new Recintos();

/* Define los datos del encabezado */
EncabezadoSOAP enc = new EncabezadoSOAP();
enc.idUsuario = "RECINTO69";
enc.firma     = new byte[60];

EntradaMercSAM res;
recinto.EncabezadoSOAPValue = enc;

res = recinto.registraEntradaMerc(paramEnt);
if (res.error == 0)
{
    ResultadoEntrada resEnt = (ResultadoEntrada)res.listaEntradas[0];
    string consec           = resEnt.consecutivo;
    int entrada           = resEnt.numEntrada;
    string acuse         = resEnt.acuse;
}
```

Entrada de mercancía suelta a granel

En este tipo de entrada, pueden realizarse entradas parciales de la mercancía siempre y cuando no se exceda del peso y las piezas declaradas en el manifiesto validado.

Para hacer una entrada de mercancía a granel deben indicarse los siguientes parámetros en la estructura **ParametroEntrada**:

- **manifiesto**. Es el número único de manifiesto.
- **nomConocimiento**. Es el identificador asignado al conocimiento por el agente naviero.
- **cuPais**. Es la clave del país de carga/descarga de la mercancía.
- **cuPuerto**. Es la clave del puerto de carga/descarga de la mercancía.
- **numAviso**. Este dato debe ser nulo.
- **numDocumento**. Este dato debe ser nulo.
- **tipoEntrada**. Este dato debe ser nulo.
- **listaMercancia** (arreglo de mercancía suelta)

El arreglo **listaMercancia** debe usarse siempre que la mercancía a ingresar no tenga como embalaje a un contenedor y debe contener a un objeto del tipo **ParametroSuelta** por cada mercancía suelta a ingresar al almacén.

- **ParametroSuelta**
 - **esParcial**. Identifica si es una entrada parcial de mercancía. En este caso este valor es verdadero.
 - **mercancia**. Es el número de mercancía asignado por el validador de manifiestos.
 - **peso**. Indica el peso de la mercancía a ingresar.
 - **piezas**. Indica la cantidad de piezas de la mercancía a ingresar.
 - **detalleMerc**. En esta versión este dato es opcional. Contiene el detalle de la mercancía a ingresar.
 - **descripcion**. Este dato debe ser nulo.
 - **embalaje**. Este dato debe ser nulo.
 - **tipoMercancia**. Es un arreglo que contiene las claves del tipo de mercancía de acuerdo con el **Catálogo de tipos de mercancía**.
 - **cuTipoMerc**. Es la clave del tipo de mercancía de acuerdo con el **Catálogo de tipos de mercancía**.
- **listaContenedor**. Este dato debe ser nulo.

RESPUESTA DE LA ENTRADA DE MERCANCÍA A GRANDEL

Este método regresa un objeto del tipo **EntradaMercSAM** que tiene la siguiente estructura:

- **error**. Tiene el último número de error que sucedió al procesar la entrada de mercancías.

- **descripError**. Es una cadena de caracteres que tiene los errores de tipo general encontrados en el proceso. Cada error esta encerrado en los caracteres “[]”.
- **listaEntradas**. Es un arreglo con objetos del tipo **ResultadoEntrada** que tiene la siguiente estructura:
 - **consecutivo**. Es el número de mercancía declarado en **ParametroSuelta**.
 - **numEntrada**. Este es el número de control asignado por el web service al registrar la entrada de mercancía. De aquí en adelante la mercancía ingresada sólo podrá ser identificada por este número.
 - **acuse**. Es un dato calculado por el web service que le sirve al recinto fiscalizado como una evidencia de que la operación de entrada de mercancía se realizó satisfactoriamente.
 - **listaErrores**. Es una cadena de caracteres que tiene los errores encontrados en el procesamiento de la entrada. Cada error esta encerrado en los caracteres “[]”.

EJEMPLO CON C#

```
//Crea los parámetros para la entrada
ParametroEntrada paramEnt = new ParametroEntrada();
paramEnt.listaMercancia = new ParametroSuelta[1];

ParametroSuelta partel = new ParametroSuelta();
partel.esParcial      = true;
partel.mercancia      = 1;
partel.peso           = 890221.43;
partel.piezas         = 3200;
paramEnt.listaMercancia[0] = partel;

paramEnt.manifiesto = "09239707000302";
paramEnt.nomConocimiento = "0261901";
paramEnt.cuPais = "USA";
paramEnt.cuPuerto = "DRR";

//Genera el proxy del servicio
ProxyRecintos.Recintos recinto = new Recintos();

/* Define los datos del encabezado */
EncabezadoSOAP enc = new EncabezadoSOAP();
enc.idUsuario      = "RECINTO69";
enc.firma          = new byte[60];

EntradaMercSAM res;
recinto.EncabezadoSOAPValue = enc;
```



```
res = recinto.registraEntradaMerc(paramEnt);

/* Obtiene el número y descripción del error */
int error = res.error;
string descrip = res.descripcionError;

/* Obtiene los datos de la entrada */
if(res.error == 0)
{
    ResultadoEntrada resEnt = (ResultadoEntrada)res.listaEntradas[0];
    int consec                = Int32.Parse(resEnt.consecutivo);
    int entrada               = resEnt.numEntrada;
    string acuse              = resEnt.acuse;
}
```

En este ejemplo se intenta ingresar al almacén mercancía a granel con un peso de 890,221.43 y 3,200 piezas, si estos datos más la existencia en el almacén de esta mercancía no excede el peso y las piezas declaradas en el manifiesto, entonces el web service procede a registrar la entrada.

Con este método los recintos fiscalizados pueden realizar el ingreso de mercancía a granel tal y como van llegando al recinto.

Entrada de mercancía contenerizada

El recinto fiscalizado debe usar este método siempre que ingrese un contenedor. Para hacer una entrada de mercancía contenerizada debe indicar los siguientes parámetros en la estructura **ParametroEntrada**:

- **manifiesto**. Es el número único de manifiesto.
- **nomConocimiento**. Es el identificador asignado al conocimiento por el agente naviero.
- **cuPais**. Es la clave del país de carga/descarga de la mercancía.
- **cuPuerto**. Es la clave del puerto de carga/descarga de la mercancía.
- **numAviso**. Este dato debe ser nulo.
- **numDocumento**. Este dato debe ser nulo.
- **tipoEntrada**. Este dato debe ser nulo.
- **listaMercancia**. Este dato debe ser nulo.
- **listaContenedor**. Esta lista indica a los contenedores a ingresar al almacén y tiene una lista de objetos **ParametroCont** con la siguiente estructura:
 - **ParametroCont**
 - **nomContenedor**. Identificador del contenedor de la mercancía.
 - **peso**. Peso del contenedor con la mercancía.
 - **piezas**. Piezas que están en el contenedor.
 - **tipoContenedor**. Este dato debe ser nulo.
 - **tipoServicio**. Este dato debe ser nulo.
 - **listaSellos**. Este dato debe ser nulo.
 - **detalleMerc**. En esta versión este dato es opcional. Contiene el detalle de la mercancía a ingresar.
 - **descripcion**. Este dato debe ser nulo.
 - **embalaje**. Este dato debe ser nulo.
 - **tipoMercancia**. Es un arreglo que contiene las claves del tipo de mercancía de acuerdo con el **Catálogo de tipos de mercancía**.
 - **cuTipoMerc**. Es la clave del tipo de mercancía de acuerdo con el **Catálogo de tipos de mercancía**.

RESPUESTA DE LA ENTRADA DE MERCANCÍA CONTENERIZADA

Este método regresa un objeto del tipo **EntradaMercSAM** que tiene la siguiente estructura:

- **error**. Tiene el último número de error que sucedió al procesar al entrada de mercancías.

- **descripError**. Es una cadena de caracteres que tiene los errores de tipo general encontrados en el proceso. Cada error esta encerrado en los caracteres “[]”.
- **listaEntradas**. Es un arreglo con objetos del tipo **ResultadoEntrada** que tiene la siguiente estructura:
 - **consecutivo**. Contiene al número de contenedor (nomContenedor) declarado en **ParametroCont**.
 - **numEntrada**. Este es el número de control asignado por el web service al registrar la entrada de mercancía. De aquí en adelante la mercancía ingresada sólo podrá ser identificada por este número.
 - **acuse**. Es un dato calculado por el web service que le sirve al recinto fiscalizado como una evidencia de que la operación de entrada de mercancía se realizó satisfactoriamente.
 - **listaErrores**. Es una cadena de caracteres que tiene los errores encontrados en el procesamiento de la entrada. Cada error esta encerrado en los caracteres “[]”.

NOTA: En caso de enviar mas de una vez el ingreso de un contenedor, la respuesta entregará un error ademas de el número de entrada y acuse.

EJEMPLO CON C#

```
ParametroEntrada paramEnt = new ParametroEntrada();
paramEnt.listaContenedor = new ParametroCont[1];
paramEnt.listaMercancia = null;

paramEnt.manifiesto = "09239707000302";
paramEnt.nomConocimiento = "IOA0988";
paramEnt.cuPais = "COL";
paramEnt.cuPuerto = "CAL";

ParametroCont conten = new ParametroCont();
conten.nomContenedor = "SUDU4498185";
paramEnt.listaContenedor[0] = conten;

//Genera el proxy del servicio
ProxyRecintos.Recintos recinto = new Recintos();

/* Define los datos del encabezado */
EncabezadoSOAP enc = new EncabezadoSOAP();
enc.idUsuario = "RECINTO69";
enc.firma = new byte[60];
```

```
EntradaMercSAM res;
recinto.EncabezadoSOAPValue = enc;

res = recinto.registraEntradaMerc(paramEnt);

/* Obtiene el número y descripción del error */
int error = res.error;
string descrip = res.descripcionError;

/* Obtiene los datos de las entradas */
if(res.error == 0)
{
    ResultadoEntrada resEnt;
    int consec;
    int entrada;
    string acuse;

    for(int i=0; i<res.listaEntradas.Count; i++)

        {
            resEnt = res.listaEntradas[i];
            consec      = Int32.Parse(resEnt.consecutivo);
            entrada     = resEnt.numEntrada;
            acuse       = resEnt.acuse;
        }
}
```

Cancela la entrada de mercancías al almacén

El recinto fiscalizado puede usar este método para cancelar una entrada de mercancía a su almacén registrada previamente.

MÉTODO

Cancelacion cancelaOperacion(ParametroCancela paramCancela)

Parámetros:

ParametroCancela:

- **tipoOperacion.** Este dato indica el tipo de operación a cancelar, en el caso de cancelar entradas este dato debe ser **1**.
- **operacion.** Indica el número de entrada a cancelar.
- **parcialidad.** Cuando sea necesario cancelar sólo una parcialidad de la entrada, aquí es donde se indica el número de parcialidad a cancelar. Cuando este parámetro se indica en 0 (cero), el web service cancelará la entrada indicada pero antes hay que cancelar todas las entradas parciales de dicha entrada.

MENSAJE SOAP

Este es el formato del mensaje SOAP para cancelar la entrada de mercancías en el almacén del recinto fiscalizado.

```
POST /wsRefis/RecintosFis.asmx HTTP/1.1
Host: satcpnpxch02
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Recintos/ServiciosWebXml/cancelaOperacion"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <EncabezadoSOAP xmlns="http://Recintos/ServiciosWebXml/">
      <idUsuario>string</idUsuario>
      <firma>base64Binary</firma>
      <ipCliente>string</ipCliente>
    </EncabezadoSOAP>
  </soap:Header>
  <soap:Body>
    <cancelaOperacion xmlns="http://Recintos/ServiciosWebXml/">
      <paramCancela>
        <tipoOperacion>int</tipoOperacion>
      </paramCancela>
    </cancelaOperacion>
  </soap:Body>
</soap:Envelope>
```

```
<operacion>string</operacion>
<parcialidad>int</parcialidad>
</paramCancela>
</cancelaOperacion>
</soap:Body>
</soap:Envelope>
```

RESPUESTA DEL MÉTODO

Este método regresa un objeto del tipo **Cancelación** con el siguiente formato.

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <cancelaOperacionResponse xmlns="http://Recintos/ServiciosWebXml/">
      <cancelaOperacionResult>
        <error>int</error>
        <descripError>string</descripError>
      </cancelaOperacionResult>
    </cancelaOperacionResponse>
  </soap:Body>
</soap:Envelope>
```

Donde:

error. Es el número de error que sucedió durante la cancelación del ingreso de la mercancía al almacén.

descripError. Es una cadena de caracteres que tiene los errores de tipo general encontrados en el proceso. Cada error esta encerrado en los caracteres “[]”.

EJEMPLO CON C#

```
ParametroCancela param = new ParametroCancela();
param.tipoOperacion = 1;           //Cancela una entrada
param.operacion = "3";            //Número de entrada a cancelar
param.parcialidad = 0;            //Número de parcialidad a cancelar. El cero indica
                                   //que desea cancelar la entrada y no una parcialidad.

//Genera el proxy del servicio
ProxyRecintos.Recintos recinto = new Recintos();

//Define los datos del encabezado
EncabezadoSOAP enc = new EncabezadoSOAP();
enc.idUsuario = "RECINTO69";
enc.firma      = new byte[60];

Cancelacion res;
recinto.EncabezadoSOAPValue = enc;

res = recinto.cancelaOperacion(paramCancela);
```

Registra la autorización de salida de mercancías del almacén

Antes de registrar la salida de una mercancía es necesario registrar su autorización para poder sacarla del almacén.

MÉTODO

SalidaMercancia autorizaSalidaPedim(ParametroSalida paramSalida)

Parámetros:

ParametroSalida:

- **numEntrada.** Número de entrada de la mercancía a la que se desea autorizar la salida.
- **pedimentos.** Pedimento(s) que justifican la autorización de salida de la mercancía. Este parámetro es un arreglo de objetos **PedimentoSalida** con la siguiente estructura:
 - **agente.** Clave de la patente del agente aduanal que elaboró el pedimento.
 - **numero.** Número de pedimento.
 - **anio.** Año en que fue validado el pedimento en 4 dígitos.
 - **remesa.** Número de remesa cuando se trata de un pedimento parte II.
 - **factura.** Número de factura cuando se trata de un pedimento previo de consolidado.
- **autoridad.** Clave de la autoridad que está solicitando la salida. En este método este parámetro no es significativo y debe ser nulo.
- **peso.** Peso de la mercancía que sale. Sólo en el caso de la mercancía a granel este dato puede ser distinto al registrado en el manifiesto, para la salida de mercancía (tanto suelta como contenerizada) de otro tipo este valor debe ser igual al registrado en el manifiesto.
- **piezas.** Piezas de la mercancía que salen del recinto. Sólo en el caso de la mercancía a granel este dato puede ser distinto al registrado en el manifiesto, para la salida de mercancía (tanto suelta como contenerizada) de otro tipo este valor debe ser igual al registrado en el manifiesto.
- **preliberacion.** Para marcar la solicitud de autorización como preliberada. El cual en caso de caer en este supuesto deberá de declararse como *true*.
- **contingencia.** Identificador para contingencia, el cual podrá recibir los valores de *true* o *false*.

MENSAJE SOAP DE AUTORIZACIÓN DE SALIDA CON UNO O MÁS PEDIMENTOS

Este es el formato del mensaje SOAP para registrar la autorización de salida de mercancías del almacén del recinto fiscalizado con uno o más pedimentos.

```
POST /wsRefis/RecintosFis.asmx HTTP/1.1
Host: satcpnpxch02
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Recintos/ServiciosWebXml/autorizaSalidaPedim"
```



```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>

    <EncabezadoSOAP xmlns="http://Recintos/ServiciosWebXml/">
      <idUsuario>string</idUsuario>
      <firma>base64Binary</firma>
      <ipCliente>string</ipCliente>
    </EncabezadoSOAP>
  </soap:Header>
  <soap:Body>
    <autorizaSalidaPedim xmlns="http://Recintos/ServiciosWebXml/">
      <paramSalida>
        <numEntrada>int</numEntrada>
        <pedimentos>
          <PedimentoSalida>
            <agente>string</agente>
            <numero>string</numero>
            <anio>string</anio>
            <factura>string</factura>
            <remesa>string</remesa>
          </PedimentoSalida>
          <PedimentoSalida>
            <agente>string</agente>
            <numero>string</numero>
            <anio>string</anio>
            <factura>string</factura>
            <remesa>string</remesa>
          </PedimentoSalida>
        </pedimentos>
        <autoridad>string</autoridad>
        <numDocumento>string</numDocumento>
        <observacion>string</observacion>
        <peso>string</peso>
      </paramSalida>
    </autorizaSalidaPedim>
  </soap:Body>
</soap:Envelope>
```

```
<piezas>string</piezas>
<preliberacion>boolean</preliberacion>
<contingencia>string</contingencia>
</paramSalida>
</autorizaSalidaPedim>
</soap:Body>
</soap:Envelope>
```

RESPUESTA DEL MÉTODO

Este método regresa un objeto del tipo **SalidaMercancia**. Una vez que el método terminó de procesar la solicitud del usuario, regresa el resultado de la operación con el siguiente formato:

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

  <soap:Body>
    <autorizaSalidaPedimResponse xmlns="http://Recintos/ServiciosWebXml/">
      <autorizaSalidaPedimResult>
        <error>int</error>
        <descripError>string</descripError>
        <numSalida>int</numSalida>
        <acuse>string</acuse>
        <autorizacion>int</autorizacion>
        <acuseAutorizacion>string</acuseAutorizacion>
      </autorizaSalidaPedimResult>
    </autorizaSalidaPedimResponse>
  </soap:Body>
</soap:Envelope>
```

Donde:

error. Es el número de error que sucedió durante el registro de la autorización de salida de la mercancía del almacén.

descripError. Es una cadena de caracteres que tiene los errores relacionados con el proceso de autorización de salida. Cada error esta encerrado en los caracteres “[]”.

numSalida. En este caso este dato es cero.

acuse. En este caso este dato es nulo.

autorizacion. Es el número de autorización de salida de esta mercancía.

acuseAutorizacion. Es el acuse de recibo de la autorización de la salida de esta mercancía.

NOTA: Cuando se ejecuta más de una vez esté método en una misma mercancía el web service va a regresar el código de error correspondiente además del número de autorización de salida y el acuse de la autorización de salida en la respuesta.

EJEMPLO EN C#

```
//Crea el objeto que tiene los parámetros de la salida
ParametroSalida paramSalida = new ParametroSalida();
paramSalida.pedimentos = new PedimentoSalida[1];

paramSalida.numEntrada = 3;
PedimentoSalida pedSalida = new PedimentoSalida();
pedSalida.agente = "0448";
pedSalida.numero = "021892";
pedSalida.anio = "2005";
pedSalida.remesa = "";
pedSalida.factura = "";
paramSalida.pedimentos[0] = pedSalida;

//Cuando la mercancía no es a granel estos datos debe ser iguales a los declarados
//en el manifiesto.
paramSalida.peso = "8192";
paramSalida.piezas = "819";

//Cuando la solicitud de autorización deba generarse como preliberada, la propiedad //deberá marcarse como "true", de lo
contrario como "false" o no definirla.
paramSalida.preliberacion = true;
```

```
//Identificador para contingencia, "true" o "false".
paramSalida.contingencia = "true";

//Genera el proxy del servicio
ProxyRecintos.Recintos recinto = new Recintos();

/* Define los datos del encabezado */
EncabezadoSOAP enc = new EncabezadoSOAP();
enc.idUsuario = "RECINTO69";
enc.firma = new byte[60];

SalidaMercancia res;
recinto.EncabezadoSOAPValue = enc;

res = recinto.autorizaSalidaPedim(paramSal);
```

Cancela la autorización de salida de mercancías del almacén

El recinto fiscalizado puede usar este método para cancelar la autorización de salida de mercancía del almacén.

MÉTODO

Cancelacion `cancelaOperacion` (`ParametroCancela paramCancela`)

Parámetros:

ParametroCancela:

- **tipoOperacion.** Este dato indica el tipo de operación a cancelar, en el caso de cancelar autorizaciones de salida este dato debe ser **13**.
- **operacion.** Indica el número de entrada que se desea cancelar su autorización para salir del recinto.
- **parcialidad.** Este dato siempre debe ser cero.

MENSAJE SOAP

Este es el formato del mensaje SOAP para cancelar la autorización de salida de mercancías en el almacén del recinto fiscalizado.

```
POST /wsRefis/RecintosFis.asmx HTTP/1.1
Host: satcpnpxch02
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Recintos/ServiciosWebXml/cancelaOperacion"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <EncabezadoSOAP xmlns="http://Recintos/ServiciosWebXml/">
      <idUsuario>string</idUsuario>
      <firma>base64Binary</firma>
      <ipCliente>string</ipCliente>
    </EncabezadoSOAP>
  </soap:Header>
  <soap:Body>
    <cancelaOperacion xmlns="http://Recintos/ServiciosWebXml/">
      <paramCancela>
        <tipoOperacion>int</tipoOperacion>
        <operacion>string</operacion>
      </paramCancela>
    </cancelaOperacion>
  </soap:Body>
</soap:Envelope>
```

```
<parcialidad>int</parcialidad>
</paramCancela>
</cancelaOperacion>
</soap:Body>
</soap:Envelope>
```

RESPUESTA DEL MÉTODO

Este método regresa un objeto del tipo **Cancelación** con el siguiente formato.

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <cancelaOperacionResponse xmlns="http://Recintos/ServiciosWebXml/">
      <cancelaOperacionResult>
        <error>int</error>
        <descripError>string</descripError>
      </cancelaOperacionResult>
    </cancelaOperacionResponse>
  </soap:Body>
</soap:Envelope>
```

Donde:

error. Es el número de error que sucedió durante la cancelación de la autorización de salida de la mercancía del almacén.

descripError. Es una cadena de caracteres que tiene los errores de tipo general encontrados en el proceso. Cada error esta encerrado en los caracteres “[]”.

EJEMPLO CON C#

```
ParametroCancela param = new ParametroCancela();
param.tipoOperacion = 13; //Cancela la autorización de salida de la entrada 3
param.operacion = "3"; //Número de entrada que se va a cancelar su
//autorización de salida.
param.parcialidad = 0; //En este caso siempre es cero.

//Genera el proxy del servicio
ProxyRecintos.Recintos recinto = new Recintos();

//Define los datos del encabezado
EncabezadoSOAP enc = new EncabezadoSOAP();
enc.idUsuario = "RECINTO69";
enc.firma = new byte[60];

Cancelacion res;
recinto.EncabezadoSOAPValue = enc;

res = recinto.cancelaOperacion(paramCancela);
```

Registra la salida de mercancías del almacén

Cada vez que el recinto necesite registrar la salida de mercancía de su almacén, debe usar este método.

NOTA: Para que el recinto pueda registrar la salida de mercancía del almacén debe haber registrado antes una operación de autorización de salida para la mercancía en cuestión, para mayor información vea [Registra la autorización de salida de mercancías del almacén](#).

MÉTODO

SalidaMercancia registraSalidaPedim(ParametroSalida paramSalida)

Parámetros:

ParametroSalida:

- **numEntrada.** Número de entrada de la mercancía a la que se desea registrar una salida.
- **pedimentos.** Pedimento(s) que justifican la salida de la mercancía. Este parámetro es un arreglo de objetos **PedimentoSalida** con la siguiente estructura:
 - **agente.** Clave de la patente del agente aduanal que elaboró el pedimento.
 - **numero.** Número de pedimento.
 - **anio.** Año en que fue validado el pedimento en 4 dígitos.
 - **remesa.** Número de remesa cuando se trata de un pedimento parte II.
 - **factura.** Número de factura cuando se trata de un pedimento previo de consolidado.
- **autoridad.** Clave de la autoridad que está solicitando la salida. En este método este parámetro no es significativo y debe ser nulo.
- **peso.** Peso de la mercancía que sale. Sólo en el caso de la mercancía a granel este dato puede ser distinto al registrado en el manifiesto, para la salida de mercancía (tanto suelta como contenerizada) de otro tipo este valor debe ser igual al registrado en el manifiesto.
- **piezas.** Piezas de la mercancía que salen del recinto. Sólo en el caso de la mercancía a granel este dato puede ser distinto al registrado en el manifiesto, para la salida de mercancía (tanto suelta como contenerizada) de otro tipo este valor debe ser igual al registrado en el manifiesto.
- **preliberacion.** Identificador para preliberación, el cual podrá recibir los valores de *true* o *false*.
- **contingencia.** Identificador para contingencia, el cual podrá recibir los valores de *true* o *false*.

MENSAJE SOAP DE SALIDA CON UNO O MÁS PEDIMENTOS

Este es el formato del mensaje SOAP para registrar la salida de mercancías del almacén del recinto fiscalizado con uno o más pedimentos.

```
POST /wsRefis/RecintosFis.asmx HTTP/1.1
Host: satcpnmxch02
Content-Type: text/xml; charset=utf-8
Content-Length: length
```



```
SOAPAction: "http://Recintos/ServiciosWebXml/registraSalidaPedim"

<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <EncabezadoSOAP xmlns="http://Recintos/ServiciosWebXml/">
      <idUsuario>string</idUsuario>
      <firma>base64Binary</firma>
      <ipCliente>string</ipCliente>
    </EncabezadoSOAP>
  </soap:Header>
  <soap:Body>
    <registraSalidaPedim xmlns="http://Recintos/ServiciosWebXml/">
      <paramSalida>
        <numEntrada>int</numEntrada>
        <pedimentos>
          <PedimentoSalida>
            <agente>string</agente>
            <numero>string</numero>
            <anio>string</anio>
            <factura>string</factura>
            <remesa>string</remesa>
          </PedimentoSalida>
          <PedimentoSalida>
            <agente>string</agente>
            <numero>string</numero>
            <anio>string</anio>
            <factura>string</factura>
            <remesa>string</remesa>
          </PedimentoSalida>
        </pedimentos>
        <autoridad>string</autoridad>
        <numDocumento>string</numDocumento>
        <observacion>string</observacion>
        <peso>string</peso>
      </paramSalida>
    </registraSalidaPedim>
  </soap:Body>
</soap:Envelope>
```

```
<piezas>string</piezas>
<preliberacion>boolean</preliberacion>
<contingencia>string</contingencia>
</paramSalida>
</registraSalidaPedim>
</soap:Body>
</soap:Envelope>
```

RESPUESTA DEL MÉTODO

Este método regresa un objeto del tipo **SalidaMercancia**. Una vez que el método terminó de procesar la solicitud del usuario, regresa el resultado de la operación con el siguiente formato:

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <registraSalidaPedimResponse xmlns="http://Recintos/ServiciosWebXml/">
      <registraSalidaPedimResult>
        <error>int</error>
        <descripcionError>string</descripcionError>
        <numSalida>int</numSalida>
        <acuse>string</acuse>
        <autorizacion>int</autorizacion>
        <acuseAutorizacion>string</acuseAutorizacion>
      </registraSalidaPedimResult>
    </registraSalidaPedimResponse>
  </soap:Body>
</soap:Envelope>
```

Donde:

error. Es el número de error que sucedió durante el registro de la salida de la mercancía del almacén.

descripError. Es una cadena de caracteres que tiene los errores relacionados con el proceso de salida. Cada error esta encerrado en los caracteres “[]”.

numSalida. Es el número de salida que calculó el sistema para esta operación.

acuse. Es el acuse de recibo de la operación.

autorizacion. En este caso este dato es cero.

acuseAutorizacion. En este caso este dato es nulo.

NOTA: En caso de enviar más de una vez la salida a una mercancía, la respuesta entregará el número de error correspondiente además del número de salida y acuse.

EJEMPLO EN C#

```
//Crea el objeto que tiene los parámetros de la salida
ParametroSalida paramSalida = new ParametroSalida();
paramSalida.pedimentos = new PedimentoSalida[1];

paramSalida.numEntrada = 3;
PedimentoSalida pedSalida = new PedimentoSalida();
pedSalida.agente = "0448";
pedSalida.numero = "021892";
pedSalida.anio = "2005";
pedSalida.remesa = "";
pedSalida.factura = "";
paramSalida.pedimentos[0] = pedSalida;

//Cuando la mercancía no es a granel estos datos debe ser iguales a los declarados
//en el manifiesto.
paramSalida.peso = "8192";
paramSalida.piezas = "819";
```

```
//En este caso la propiedad no es significativa y debe declararse
//como "false" o no definirla.
//paramSalida.preliberacion = false;

//Identificador para contingencia, "true" o "false".
paramSalida.contingencia = "true";

//Genera el proxy del servicio
ProxyRecintos.Recintos recinto = new Recintos();

/* Define los datos del encabezado */
EncabezadoSOAP enc = new EncabezadoSOAP();
enc.idUsuario = "RECINTO69";
enc.firma = new byte[60];

SalidaMercancia res;
recinto.EncabezadoSOAPValue = enc;

res = recinto.registraSalidaPedim(paramSal);
```

Cancela la salida de mercancías del almacén.

El recinto fiscalizado debe usar este método para cancelar una salida de mercancía de su almacén.

MÉTODO

Cancelacion cancelaOperacion(ParametroCancela paramCancela)

Parámetros:

ParametroCancela:

- **tipoOperacion.** Este dato indica el tipo de operación a cancelar, para cancelar salidas este dato debe ser **2**.
- **operacion.** Indica el número de salida a cancelar.
- **parcialidad.** Este dato siempre debe ser cero.

MENSAJE SOAP

Este es el formato del mensaje SOAP para cancelar la salida de mercancías del almacén del recinto fiscalizado.

```
POST /wsRefis/RecintosFis.asmx HTTP/1.1
Host: satcpnpxch02
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Recintos/ServiciosWebXml/cancelaOperacion"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <EncabezadoSOAP xmlns="http://Recintos/ServiciosWebXml/">
      <idUsuario>string</idUsuario>
      <firma>base64Binary</firma>
      <ipCliente>string</ipCliente>
    </EncabezadoSOAP>
  </soap:Header>
  <soap:Body>
    <cancelaOperacion xmlns="http://Recintos/ServiciosWebXml/">
      <paramCancela>
        <tipoOperacion>int</tipoOperacion>
        <operacion>string</operacion>
        <parcialidad>int</parcialidad>
      </paramCancela>
    </cancelaOperacion>
  </soap:Body>
</soap:Envelope>
```

```
</paramCancela>  
</cancelaOperacion>  
</soap:Body>  
</soap:Envelope>
```

RESPUESTA DEL MÉTODO

Este método regresa un objeto del tipo **Cancelación** con el siguiente formato.

```
HTTP/1.1 200 OK  
Content-Type: text/xml; charset=utf-8  
Content-Length: length  
  
<?xml version="1.0" encoding="utf-8"?>  
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
  <soap:Body>  
    <cancelaOperacionResponse xmlns="http://Recintos/ServiciosWebXml/">  
      <cancelaOperacionResult>  
        <error>int</error>  
        <descripError>string</descripError>  
      </cancelaOperacionResult>  
    </cancelaOperacionResponse>  
  </soap:Body>  
</soap:Envelope>
```

Donde:

error. Es el número de error que sucedió durante la cancelación de la salida de mercancía del almacén.

descripError. Es una cadena de caracteres que tiene los errores de tipo general encontrados en el proceso. Cada error esta encerrado en los caracteres “[]”.

EJEMPLO EN C#

```
//Crea los parámetros para cancelar una salida
ParametroCancela param = new ParametroCancela();
param.tipoOperacion = 2; //El tipo de operación a cancelar es 2 = SALIDA
param.operacion = "21"; //Número de salida a cancelar
param.parcialidad = 0; //En este caso siempre es cero.

//Genera el proxy del servicio
ProxyRecintos.Recintos recinto = new Recintos();

//Define los datos del encabezado
EncabezadoSOAP enc = new EncabezadoSOAP();
enc.idUsuario = "RECINTO69";
enc.firma = new byte[60];

Cancelacion res;
recinto.EncabezadoSOAPValue = enc;

res = recinto.cancelaOperacion(paramCancela);
```

3ª Etapa Transferencia y Traspaleo

Transferencia de mercancía entre recintos

Esta operación representa el movimiento físico de una mercancía que está en un recinto (llamado recinto origen) hacia otro recinto (llamado recinto destino). Los dos recintos deben pertenecer a la misma aduana.

La transferencia de mercancía entre recintos fiscalizados debe darse en 3 pasos:

1. El recinto origen registra una salida de mercancías por transferencia.
2. El recinto destino consulta las mercancías registradas en una salida de mercancías por transferencia a su recinto
3. El recinto destino debe hacer una entrada de mercancía por transferencia entre recintos.

Primer paso. Registra la salida de transferencia de mercancías

El recinto fiscalizado origen que requiere transferir mercancía a otro recinto puede usar este método para registrar una salida por transferencia de mercancía.

MÉTODO

ResultadoTransfer registraSalidaTransfer (ParametroTransfer paramTrans)

Parámetros:

ParametroTransfer:

- **recintoDestino.** Es la clave de usuario del recinto destino de la operación de transferencia de mercancías.
- **listaEntradas.** Es un arreglo de objetos del tipo **string** que indican los números de entrada de las mercancías que se van a transferir del recinto origen al recinto destino.

MENSAJE SOAP

Este es el formato del mensaje SOAP para registrar una salida de transferencia de mercancías del almacén del recinto fiscalizado origen.

```
POST /wsRefis/RecintosFis.asmx HTTP/1.1
Host: satcpnmxch02
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Recintos/ServiciosWebXml/registraSalidaTransfer"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <EncabezadoSOAP xmlns="http://Recintos/ServiciosWebXml/">
      <idUsuario>string</idUsuario>
      <firma>base64Binary</firma>
      <ipCliente>string</ipCliente>
    </EncabezadoSOAP>
  </soap:Header>
  <soap:Body>
    <registraSalidaTransfer xmlns="http://Recintos/ServiciosWebXml/">
      <paramTrans>
        <recintoOrigen>string</recintoOrigen>
        <listaEntradas>
```

```
<string>string</string>
<string>string</string>
</listaEntradas>
<agente>string</agente>
<recintoDestino>string</recintoDestino>
<tarja>string</tarja>
</paramTrans>
</registraSalidaTransfer>
</soap:Body>
</soap:Envelope>
```

RESPUESTA DEL MÉTODO

Este método regresa un objeto del tipo **ResultadoTransfer**. Una vez que el método terminó de procesar la solicitud del usuario, regresa el resultado de la operación con el siguiente formato:

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <registraSalidaTransferResponse xmlns="http://Recintos/ServiciosWebXml/">
      <registraSalidaTransferResult>
        <error>int</error>
        <descripError>string</descripError>
        <acuse>string</acuse>
        <listaEntradas>
          <anyType />
          <anyType />
        </listaEntradas>
        <listaSalidas>
          <anyType />
```

```
<anyType />
</listaSalidas>
</registraSalidaTransferResult>
</registraSalidaTransferResponse>
</soap:Body>
</soap:Envelope>
```

Donde:

error. Es el número del último error que sucedió durante el registro de la salida por transferencia.

descripError. Es una cadena de caracteres que tiene los errores de tipo general encontrados en el proceso. Cada error está encerrado en los caracteres “[]”.

listaEntradas. Este dato es nulo.

listaSalidas. Este dato contiene el resultado de las salidas generadas como una lista de elementos del tipo **ResultadoSalidaTrans** con la siguiente estructura:

- **entrada.** Es el número de entrada al que se le registró la salida por transferencia.
- **salida.** Es el número de salida calculado por el sistema.
- **acuse.** Es el acuse de recibo calculado por el sistema.
- **listaErrores.** Es una cadena de caracteres que tiene los errores encontrados en el procesamiento de la salida. Cada error está encerrado en los caracteres “[]” .

EJEMPLO CON C#

```
//Crea los parámetros de la salida
ParametroTransfer paramTrans = new ParametroTransfer();
paramTrans.recintoDestino = "69";

string[] listaEntradas = new string[4];

listaEntradas[0] = "13";
listaEntradas[1] = "14";
listaEntradas[2] = "17";
listaEntradas[3] = "22";

paramTrans.listaEntradas = listaEntradas;

//Genera el proxy del servicio
ProxyRecintos.Recintos recinto = new Recintos();

//Define los datos del encabezado
EncabezadoSOAP enc = new EncabezadoSOAP();
enc.idUsuario = "RECINTO68";
enc.firma = new byte[60];

ResultadoTransfer res;
recinto.EncabezadoSOAPValue = enc;

res = recinto.registraSalidaTransfer(paramTrans);

lblError.Text = "Error: " + res.error;
lblDesc.Text = res.descripcion;

//Obtiene los número de salida que corresponden a la entrada
ResultadoSalidaTrans resSalida;
int nEntrada, nSalida;
string acuseSalida, strListaErrores;
for(i =0; i< res.listaSalidas.Length; i++)
{
    resSalida = (ResultadoSalidaTrans)res.listaSalidas[i];
    nEntrada = resSalida.entrada;
    nSalida = resSalida.salida;
    acuseSalida = resSalida.acuse;
    strListaErrores = resSalida.listaErrores;
}
```

Cancela una salida de transferencia de mercancías

Para cancelar una salida de este tipo, el recinto fiscalizado **origen** de la transferencia debe ejecutar el siguiente método.

MÉTODO

Cancelacion cancelaOperacion(ParametroCancela paramCancela)

Parámetros:

ParametroCancela:

- **tipoOperacion**. Este dato indica el tipo de operación a cancelar, para cancelar salidas de transferencia este dato debe ser **4**.
- **operacion**. Indica el número de salida a cancelar.
- **parcialidad**. Este dato debe ser cero.

MENSAJE SOAP

Este es el formato del mensaje SOAP para cancelar una salida de transferencia de mercancías.

```
POST /wsRefis/RecintosFis.asmx HTTP/1.1
Host: satcpnpxch02
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Recintos/ServiciosWebXml/cancelaOperacion"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <EncabezadoSOAP xmlns="http://Recintos/ServiciosWebXml/">
      <idUsuario>string</idUsuario>
      <firma>base64Binary</firma>
      <ipCliente>string</ipCliente>
    </EncabezadoSOAP>
  </soap:Header>
  <soap:Body>
    <cancelaOperacion xmlns="http://Recintos/ServiciosWebXml/">
      <paramCancela>
        <tipoOperacion>int</tipoOperacion>
        <operacion>string</operacion>
        <parcialidad>int</parcialidad>
      </paramCancela>
    </cancelaOperacion>
  </soap:Body>
</soap:Envelope>
```

```
</paramCancela>  
</cancelaOperacion>  
</soap:Body>  
</soap:Envelope>
```

RESPUESTA DEL MÉTODO

Este método regresa un objeto del tipo **Cancelación** con el siguiente formato.

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <cancelaOperacionResponse xmlns="http://Recintos/ServiciosWebXml/">
      <cancelaOperacionResult>
        <error>int</error>
        <descripError>string</descripError>
      </cancelaOperacionResult>
    </cancelaOperacionResponse>
  </soap:Body>
</soap:Envelope>
```

Donde:

error. Es el número de error que sucedió durante la cancelación de la salida por transferencia de mercancía.

descripError. Es una cadena de caracteres que tiene los errores de tipo general encontrados en el proceso. Cada error esta encerrado en los caracteres “[]”.

EJEMPLO CON C#

```
//Crea los parámetros de la cancelación
ParametroCancela paramCancela = new ParametroCancela();
paramCancela.tipoOperacion = 4; //Cancela salida por transferencia.
paramCancela.operacion = "389"; //Este es el número salida a cancelar.
paramCancela.parcialidad = 0; //En este caso siempre es cero.

//Genera el proxy del servicio
ProxyRecintos.Recintos recinto = new Recintos();

//Define los datos del encabezado
```

```
EncabezadoSOAP enc = new EncabezadoSOAP();  
enc.idUsuario = "RECINTO69";  
enc.firma = new byte[60];  
  
Cancelacion res;  
recinto.EncabezadoSOAPValue = enc;  
  
res = recinto.cancelaOperacion(paramCancela);
```


Segundo paso. Consulta las salidas registradas por el recinto origen en transferencia

El recinto fiscalizado **destino** debe usar este método para consultar las mercancías que tienen registrada una salida de mercancía en transferencia hacia el almacén de su recinto.

MÉTODO

resEntradas consultaEntPorTransfer(ParametroConsEntrada paramCons) Parámetros:

ParametroConsEntrada:

- **recinto**. Clave de usuario del recinto fiscalizado origen de la transferencia.
- **Todas**. Bandera para hacer compatible el metodo con la version anterior, el valor **true** muestra todas las transferencias, **false** las filtra por número de recinto destino

MENSAJE SOAP

Este es el formato del mensaje SOAP para realizar la consulta de mercancías registradas en una salida de mercancías del almacén del recinto fiscalizado origen en una transferencia.

```
POST /wsRefis/RecintosFis.asmx HTTP/1.1
Host: satcpnpxch02
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Recintos/ServiciosWebXml/consultaEntPorTransfer"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <EncabezadoSOAP xmlns="http://Recintos/ServiciosWebXml/">
      <idUsuario>string</idUsuario>
      <firma>base64Binary</firma>
      <ipCliente>string</ipCliente>
    </EncabezadoSOAP>
  </soap:Header>
  <soap:Body>
    <consultaEntPorTransfer xmlns="http://Recintos/ServiciosWebXml/">
      <paramCons>
        <recinto>int</recinto>
        <todas>boolean</todas>
      </paramCons>
    </consultaEntPorTransfer>
  </soap:Body>
</soap:Envelope>
```

```
</paramCons>  
</consultaEntPorTransfer>  
</soap:Body>  
</soap:Envelope>
```

RESPUESTA DEL MÉTODO

Una vez que el método terminó de procesar la solicitud del usuario, regresa el resultado de la operación con el siguiente formato:

```
HTTP/1.1 200 OK  
Content-Type: text/xml; charset=utf-8  
Content-Length: length  
  
<?xml version="1.0" encoding="utf-8"?>  
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
  <soap:Body>  
    <consultaEntPorTransferResponse xmlns="http://Recintos/ServiciosWebXml/">  
      <consultaEntPorTransferResult>  
        <mercanciaEnRecinto>  
          <anyType />  
          <anyType />  
        </mercanciaEnRecinto>  
      </consultaEntPorTransferResult>  
    </consultaEntPorTransferResponse>  
  </soap:Body>  
</soap:Envelope>
```

Donde:

error. Es el número de error que sucedió durante la operación.

descripError. Es una cadena de caracteres que tiene los errores de tipo general encontrados en el proceso. Cada error esta encerrado en los caracteres “[]”.

mercanciaEnRecinto. En este arreglo se regresa una lista de objetos del tipo **EntradaEnRecinto** con la siguiente estructura:

- **recinto.** Es el número recinto fiscalizado origen que registro la salida de mercancía en transferencia.
- **entrada.** Es el número de entrada de mercancía registrado en la salida de mercancía en transferencia.
- **contenedor.** Es el número de contenedor relacionado al numero de entrada, para el caso de mercancía suelta muestra la leyenda “SUELTA”.
- **descripcion.** Es la descripción de la mercancía relacionada al número de entrada.

EJEMPLO CON C#

```
//Crea los parámetros de la consulta
ParametroConsEntrada paramCons = new ParametroConsEntrada();
paramCons.recinto = "69";
paramCons.todas = false;

//Genera el proxy del servicio
ProxyRecintos.Recintos recinto = new Recintos();

//Define los datos del encabezado
EncabezadoSOAP enc = new EncabezadoSOAP();
enc.idUsuario = "RECINTO69";
enc.firma = new byte[60];

ProxyRecintos.resEntradas res;

recinto.EncabezadoSOAPValue = enc;
```

```
res = recinto.consultaEntPorTransfer (paramCons);

Response.Output.WriteLine("Error " + res.error + " " + res.descripcion + "<br>");
Response.Output.WriteLine("Entradas encontradas: " + res.registros + "<BR>");

if(res.mercanciaEnRecinto != null)
{
    Response.Output.WriteLine("<table border=\"1\"><br><br><br>");
    for(int i=0; i < res.mercanciaEnRecinto.Length;i++)
    {
        ProxyRecintos.EntradaEnRecinto entRecinto = (ProxyRecintos.EntradaEnRecinto) res.mercanciaEnRecinto[i];

        Response.Output.WriteLine("<th bgColor=\"cyan\">ENTRADA</th>");
        Response.Output.WriteLine("<th bgColor=\"cyan\">CONTENEDOR</th>");
        Response.Output.WriteLine("<th bgColor=\"cyan\">MERCANCIA</th>");

        Response.Output.WriteLine("<tr>");
        Response.Output.WriteLine("<td>" + entRecinto.entrada + "</td>");
        Response.Output.WriteLine("<td>" + entRecinto.contenedor + "</td>");
        Response.Output.WriteLine("<td>" + entRecinto.descripcion + "</td>");
        Response.Output.WriteLine("</tr>");
    }
    Response.Output.WriteLine("</table>");
}
```

Tercer paso. Registra la entrada de mercancías por transferencia al recinto destino

El recinto fiscalizado destino debe usar este método para registrar la entrada de las mercancías por transferencia en su almacén.

MÉTODO

ResultadoTransfer registraEntradaTransfer (ParametroTransfer paramTrans)

Parámetros:

ParametroTransfer:

- **recintoOrigen.** Clave de usuario del recinto origen de la transferencia.
- **listaEntradas.** Es un arreglo de objetos del tipo **string** que contiene los números de entrada de la mercancía que desea ingresar al recinto por transferencia.

MENSAJE SOAP

Este es el formato del mensaje SOAP para registrar una entrada de mercancías al almacén del recinto fiscalizado destino en una transferencia.

```
POST /wsRefis/RecintosFis.asmx HTTP/1.1
Host: satcpnpxch02
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Recintos/ServiciosWebXml/registraEntradaTransfer"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <EncabezadoSOAP xmlns="http://Recintos/ServiciosWebXml/">
      <idUsuario>string</idUsuario>
      <firma>base64Binary</firma>
      <ipCliente>string</ipCliente>
    </EncabezadoSOAP>
  </soap:Header>
  <soap:Body>
    <registraEntradaTransfer xmlns="http://Recintos/ServiciosWebXml/">
      <paramTrans>
        <recintoOrigen>string</recintoOrigen>
        <listaEntradas>
```

```
<string>string</string>
<string>string</string>
</listaEntradas>
<agente>string</agente>
<recintoDestino>string</recintoDestino>
<tarja>string</tarja>
</paramTrans>
</registraEntradaTransfer>
</soap:Body>
</soap:Envelope>
```

RESPUESTA DEL MÉTODO

Este método regresa un objeto del tipo **ResultadoTransfer**. Una vez que el método terminó de procesar la solicitud del usuario, regresa el resultado de la operación con el siguiente formato:

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <registraEntradaTransferResponse xmlns="http://Recintos/ServiciosWebXml/">
      <registraEntradaTransferResult>
        <error>int</error>
        <descripcionError>string</descripcionError>
        <acuse>string</acuse>
        <listaEntradas>
          <anyType />
          <anyType />
        </listaEntradas>
        <listaSalidas>
          <anyType />
          <anyType />
        </listaSalidas>
      </registraEntradaTransferResult>
    </registraEntradaTransferResponse>
  </soap:Body>
</soap:Envelope>
```

```
</registraEntradaTransferResponse>  
</soap:Body>  
</soap:Envelope>
```

Donde:

error. Es el número del último error que sucedió durante la operación.

descripcionError. Es una cadena de caracteres que tiene los errores de tipo general encontrados en el proceso. Cada error está encerrado en los caracteres “[]”.

acuse. En esta operación es nulo.

listaEntradas. Contiene las entradas generadas por la transferencia como una lista de objetos del tipo **ResultadoEntrada** que tiene la siguiente estructura:

- **consecutivo.** Es un número que corresponde a cada elemento del arreglo **listaEntradas** e inicia con 0 (cero).
- **numEntrada.** Es el número de entrada por transferencia calculado por el sistema.
- **acuse.** Acuse electrónico de la entrada por transferencia.
- **listaErrores.** Es una cadena de caracteres que tiene los errores encontrados en el procesamiento de la entrada. Cada error está encerrado en los caracteres “[]”.

listaSalidas. En esta operación este campo es nulo.

EJEMPLO CON C#

```
ParametroTransfer paramTrans = new ParametroTransfer();  
paramTrans.recintoOrigen = "68";  
  
string[] listaEntradas = new string[4];  
  
listaEntradas[0] = "13";  
listaEntradas[1] = "14";  
listaEntradas[2] = "17";  
listaEntradas[3] = "22";  
  
paramTrans.listaEntradas = listaEntradas;  
  
//Genera el proxy del servicio  
ProxyRecintos.Recintos recinto = new Recintos();  
  
//Define los datos del encabezado
```

```
EncabezadoSOAP enc = new EncabezadoSOAP();
enc.idUsuario = "RECINTO69"
enc.firma      = new byte[60];

ResultadoTransfer res;
recinto.EncabezadoSOAPValue = enc;

res = recinto.registraEntradaTransfer(paramTrans);

lblError.Text = "Error: " + res.error;
lblDesc.Text  = res.descripcion;

ResultadoEntrada rEntrada = null;
int nEntrada;
string acuse, strListaErrores;
for(i=0; i<res.listaEntradas.Length;i++)
{
    rEntrada = (ResultadoEntrada)res.listaEntradas[i];
    strListaErrores = rEntrada.listaErrores;
    nEntrada = rEntrada.numEntrada;
    acuse = rEntrada.acuse;
}
```


Cancela la entrada de mercancías por transferencia al recinto destino

Una vez que el recinto destino le ha dado entrada a la mercancía en una transferencia, puede cancelarla con el siguiente método.

MÉTODO

Cancelacion cancelaOperacion(ParametroCancela paramCancela)

Parámetros:

ParametroCancela:

- **tipoOperacion.** Este dato indica el tipo de operación a cancelar, para cancelar la entrada de mercancía en el recinto destino por transferencia este dato debe ser **5**.
- **operacion.** Indica el número de entrada a cancelar en el recinto destino.
- **parcialidad.** En este dato se indica la clave del recinto origen de la transferencia.

MENSAJE SOAP

Este es el formato del mensaje SOAP para cancelar una entrada de mercancías por transferencia.

```
POST /wsRefis/RecintosFis.asmx HTTP/1.1
Host: satcpnpxch02
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Recintos/ServiciosWebXml/cancelaOperacion"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <EncabezadoSOAP xmlns="http://Recintos/ServiciosWebXml/">
      <idUsuario>string</idUsuario>
      <firma>base64Binary</firma>
      <ipCliente>string</ipCliente>
    </EncabezadoSOAP>
  </soap:Header>
  <soap:Body>
    <cancelaOperacion xmlns="http://Recintos/ServiciosWebXml/">
      <paramCancela>
        <tipoOperacion>int</tipoOperacion>
        <operacion>string</operacion>
      </paramCancela>
    </cancelaOperacion>
  </soap:Body>
</soap:Envelope>
```

```
<parcialidad>int</parcialidad>  
</paramCancela>  
</cancelaOperacion>  
</soap:Body>  
</soap:Envelope>
```

RESPUESTA DEL MÉTODO

Este método regresa un objeto del tipo **Cancelación** con el siguiente formato.

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <cancelaOperacionResponse xmlns="http://Recintos/ServiciosWebXml/">
      <cancelaOperacionResult>
        <error>int</error>
        <descripError>string</descripError>
      </cancelaOperacionResult>
    </cancelaOperacionResponse>
  </soap:Body>
</soap:Envelope>
```

Donde:

error. Contiene el número del último error ocurrido en la operación.

descripError. Es una cadena de caracteres que tiene los errores encontrados en el proceso. Cada error esta encerrado en los caracteres “[]”.

EJEMPLO CON C#

```
//Crea los parámetros de la cancelación
ParametroCancela paramCancela = new ParametroCancela();
paramCancela.tipoOperacion = 5; //Cancela entrada por transferencia.

//Este es el número de entrada a cancelar.
paramCancela.operacion = "211";

//Este es el número recinto origen de la transferencia.
paramCancela.parcialidad = 69;

//Genera el proxy del servicio
```

```
ProxyRecintos.Recintos recinto = new Recintos();

//Define los datos del encabezado */
EncabezadoSOAP enc = new EncabezadoSOAP();
enc.idUsuario = "RECINTO68";
enc.firma = new byte[60];

Cancelacion res;
recinto.EncabezadoSOAPValue = enc;

res = recinto.cancelaOperacion(paramCancela);
```

Traspaleo de mercancía de uno o más contenedores a otro u otros contenedores.

Esta operación representa el movimiento físico de una mercancía que está en uno o más contenedores (contenedores origen) hacia otro u otros contenedores (contenedores destino). Tanto los contenedores origen como los contenedores destino deben estar en el mismo recinto fiscalizado.

El traspaleo de mercancía debe darse en 2 pasos:

1. El recinto registra el aviso de traspaleo de contenedores para “apartar” a los contenedores involucrados (origen y destino) y que éstos no puedan usarse en ninguna otra operación.
2. El recinto registra qué contenedor o contenedores destino del traspaleo ya terminaron su operación.

Primer paso. Registra el aviso de traspaleo de contenedores

El recinto fiscalizado que requiere hacer un traspaleo de mercancía de uno o más contenedores debe usar este método para registrar el aviso de traspaleo de mercancía.

MÉTODO

ResultadoTraspaleo registraAvisoTraspaleo (ParametroTraspaleo paramTraspaleo)

Parámetros:

ParametroTraspaleo:

- **numAviso.** Es el número de aviso de traspaleo asignado por el recinto.
- **listaContOrig.** Es una lista de objetos del tipo **ContOrig** que indican a los contenedores origen en la operación de traspaleo. Estos objetos tienen la siguiente estructura:
 - **entrada.** Indica el número de entrada del contenedor origen del traspaleo.
 - **total.** Indica si el traspaleo va a ser total o parcial. Un traspaleo total deja al contenedor origen vacío al terminar la operación. Los valores posibles para este campo son: “si”, “verdadero”, “1”, cualquier otro valor representa al valor FALSO.
 - **observación.** Notas que se van a adjuntar al contenedor para esta operación de traspaleo.
- **fecha.** Indica la fecha programada del traspaleo.
- **listaContDest.** Es una lista de objetos del tipo **ContDest** que indican a los contenedores destino del traspaleo. Estos objetos tienen la siguiente estructura:
 - **entrada.** Identifica el número de entrada del contenedor destino del traspaleo.
 - **contenedor.** Identifica al contenedor a usar como destino del traspaleo.
- **tarja.** Este dato debe ser nulo.
- **listaContFinal.** Este dato debe ser nulo.

MENSAJE SOAP

Este es el formato del mensaje SOAP para registrar un aviso de traspaleo de mercancías del almacén del recinto fiscalizado.

```
POST /wsRefis/RecintosFis.asmx HTTP/1.1
Host: satcpnpxch02
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Recintos/ServiciosWebXml/registraAvisoTraspaleo"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <EncabezadoSOAP xmlns="http://Recintos/ServiciosWebXml/">
      <idUsuario>string</idUsuario>
      <firma>base64Binary</firma>
      <ipCliente>string</ipCliente>
    </EncabezadoSOAP>
  </soap:Header>
  <soap:Body>
    <registraAvisoTraspaleo xmlns="http://Recintos/ServiciosWebXml/">
      <paramTraspaleo>
        <numAviso>string</numAviso>
        <listaContOrig>
          <ContOrig>
            <entrada>string</entrada>
            <total>string</total>
            <observacion>string</observacion>
          </ContOrig>
          <ContOrig>
            <entrada>string</entrada>
            <total>string</total>
            <observacion>string</observacion>
          </ContOrig>
        </listaContOrig>
        <fecha>string</fecha>
        <listaContDest>
```

```
<ContDest>
  <entrada>string</entrada>
  <contenedor>string</contenedor>
</ContDest>
<ContDest>
  <entrada>string</entrada>
  <contenedor>string</contenedor>
</ContDest>
</listaContDest>
<tarja>string</tarja>
<listaContFinal>
  <ContFinal>

    <entrada>string</entrada>
    <descrip>string</descrip>
    <marcas>string</marcas>
    <peso>string</peso>
    <piezas>string</piezas>
    <unidadMed>string</unidadMed>
    <sellos xsi:nil="true" />
  </ContFinal>
  <ContFinal>
    <entrada>string</entrada>
    <descrip>string</descrip>
    <marcas>string</marcas>
    <peso>string</peso>
    <piezas>string</piezas>
    <unidadMed>string</unidadMed>
    <sellos xsi:nil="true" />
  </ContFinal>
</listaContFinal>
</paramTraspaleo>
</registraAvisoTraspaleo>
</soap:Body>
</soap:Envelope>
```

RESPUESTA DEL MÉTODO

Este método regresa un objeto del tipo **ResultadoTraspaleo**. Una vez que el método terminó de procesar la solicitud del usuario, regresa el resultado de la operación con el siguiente formato:

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <registraAvisoTraspaleoResponse xmlns="http://Recintos/ServiciosWebXml/">
      <registraAvisoTraspaleoResult>
        <error>int</error>
        <descripError>string</descripError>
        <acuse>string</acuse>
        <listaEntradas>
          <anyType />
          <anyType />
        </listaEntradas>
        <listaOrigen>
          <anyType />
          <anyType />
        </listaOrigen>
      </registraAvisoTraspaleoResult>
    </registraAvisoTraspaleoResponse>
  </soap:Body>
</soap:Envelope>
```

Donde:

error. Es el número del último error que sucedió durante la operación.

descripError. Es una cadena de caracteres que tiene los errores de tipo general encontrados en el proceso. Cada error está encerrado en los caracteres "[]".

acuse. Es el acuse de recibo del aviso de traspaleo.

listaEntradas. Contiene una lista de objetos del tipo **ResultadoEntrada** que corresponde a cada uno de los contenedores destino del traspaleo definidos en los parámetros de entrada. Este objeto tiene la siguiente estructura:

- **consecutivo.** Contiene al número de contenedor destino del traspaleo.
- **numEntrada.** Es el número de entrada asignado por el sistema al contenedor destino del traspaleo.
- **acuse.** Acuse de recibo de la operación de entrada.
- **listaErrores.** Es la lista de errores que sucedieron al procesar la entrada del contenedor destino del traspaleo.

listaOrigen. Contiene una lista de objetos del tipo **ResultadoEntrada** que corresponden a la validación de cada uno de los contenedores origen definidos en los parámetros de entrada.

EJEMPLO CON C#

```
ParametroTraspaleo paramTras = new ParametroTraspaleo();
ContOrig cOrig;
ContDest cDest;

//Identifica a los contenedores origen del traspaleo.
ContOrig[] listaOrig = new ContOrig[5];

cOrig = new ContOrig();
cOrig.entrada = "11";
cOrig.total = "si";
listaOrig[0] = cOrig;

cOrig = new ContOrig();
cOrig.entrada = "17";
cOrig.total = "si";
listaOrig[1] = cOrig;

cOrig = new ContOrig();
cOrig.entrada = "18";
cOrig.total = "si";
listaOrig[2] = cOrig;

cOrig = new ContOrig();
cOrig.entrada = "25";
cOrig.total = "si";
listaOrig[3] = cOrig;

cOrig = new ContOrig();

cOrig.entrada = "28";
cOrig.total = "si";
```

```
listaOrig[4] = cOrig;

//Identifica a los contenedores destino del traspaleo.
ContDest[] listaDest = new ContDest[2];

cDest = new ContDest();
cDest.contenedor = "SUDU049142"
listaDest[0] = cDest;

cDest = new ContDest();
cDest.contenedor = "SUDU21112";
listaDest[1] = cDest;

//Asigna valores a parámetros
paramTras.numAviso           = "TRASPALCO-001";
paramTras.fecha              = "2005-02-01 00:00:00";
paramTras.tarja               = "TARJA-00X";
paramTras.listaContOrig      = listaOrig;
paramTras.listaContDest      = listaDest;

//Genera el proxy del servicio
ProxyRecintos.Recintos recinto = new Recintos();

//Define los datos del encabezado
EncabezadoSOAP enc           = new EncabezadoSOAP();
enc.idUsuario                 = TBUusuario.Text;
enc.firma                     = new byte[60];
recinto.EncabezadoSOAPValue   = enc;

ResultadoTraspaleo res;
res = recinto.registraAvisoTraspaleo(paramTras);

//Muestra errores de los contenedores origen del traspaleo.
ResultadoEntrada resEnt;
for(i=0; i<res.listaOrigen.Length; i++)
{
    resEnt = (ResultadoEntrada)res.listaOrigen[i];
    switch(i)
    {
        case 0:
            lblOrg1.Text = resEnt.listaErrores;
            break;
        case 1:
            lblOrg2.Text = resEnt.listaErrores;
            break;
        case 2:
            lblOrg3.Text = resEnt.listaErrores;
```

```
        break;
    case 3:
        lblOrg4.Text = resEnt.listaErrores;
        break;
    case 4:
        lblOrg5.Text = resEnt.listaErrores;
        break;
    }
}
```

```
//Muestra errores de los contenedores destino del traspaleo.
for(i=0; i<res.listaEntradas.Length; i++)
{
    resEnt = (ResultadoEntrada)res.listaEntradas[i];
    switch(i)
    {
        case 0:
            lblDes1.Text = resEnt.listaErrores;
            break;
        case 1:
            lblDes2.Text = resEnt.listaErrores;
            break;
        case 2:
            lblDes3.Text = resEnt.listaErrores;
            break;
        case 3:
            lblDes4.Text = resEnt.listaErrores;
            break;
        case 4:
            lblDes5.Text = resEnt.listaErrores;
            break;
    }
}

//Obtiene los errores generales del traspaleo.
string erroresGenerales = res.descripcionError;
```



Cancela un aviso de traspaleo de contenedores

Cuando el recinto fiscalizado tenga necesidad de cancelar un aviso de traspaleo, debe usar el siguiente método.

MÉTODO

Cancelacion cancelaOperacion(ParametroCancela paramCancela)

Parámetros:

ParametroCancela:

- **tipoOperacion.** Este dato indica el tipo de operación a cancelar, para cancelar un aviso de traspaleo este dato debe ser **6**.
- **operacion.** Este es el número del aviso que se desea cancelar.
- **parcialidad.** Este dato debe ser nulo.

MENSAJE SOAP

Este es el formato del mensaje SOAP para cancelar un aviso de traspaleo.

```
POST /wsRefis/RecintosFis.asmx HTTP/1.1
Host: satcpnpxch02
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Recintos/ServiciosWebXml/cancelaOperacion"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <EncabezadoSOAP xmlns="http://Recintos/ServiciosWebXml/">
      <idUsuario>string</idUsuario>
      <firma>base64Binary</firma>
      <ipCliente>string</ipCliente>
    </EncabezadoSOAP>
  </soap:Header>
  <soap:Body>
    <cancelaOperacion xmlns="http://Recintos/ServiciosWebXml/">
      <paramCancela>
        <tipoOperacion>int</tipoOperacion>
        <operacion>string</operacion>
        <parcialidad>int</parcialidad>
      </paramCancela>
    </cancelaOperacion>
  </soap:Body>
</soap:Envelope>
```

```
</paramCancela>  
</cancelaOperacion>  
</soap:Body>  
</soap:Envelope>
```

RESPUESTA DEL MÉTODO

Este método regresa un objeto del tipo **Cancelación** con el siguiente formato.

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <cancelaOperacionResponse xmlns="http://Recintos/ServiciosWebXml/">
      <cancelaOperacionResult>
        <error>int</error>
        <descripError>string</descripError>
      </cancelaOperacionResult>
    </cancelaOperacionResponse>
  </soap:Body>
</soap:Envelope>
```

Donde:

error. Contiene el número del último error ocurrido en la operación.

descripError. Es una cadena de caracteres que tiene los errores de tipo general encontrados en el proceso. Cada error esta encerrado en los caracteres “[]”.

listaOrigen. Es un arreglo que contiene objetos del tipo **ResultadoEntrada** que contiene el resultado de la validación de los contenedores origen del traspaleo para realizar la cancelación del aviso.

listaEntradas. Es un arreglo que contiene objetos del tipo **ResultadoEntrada** con el resultado de la validación de los contenedores destino del traspaleo para realizar la cancelación del aviso.

EJEMPLO CON C#

```
//Genera los parámetros de la cancelación
ParametroCancela paramCancela = new ParametroCancela();
paramCancela.tipoOperacion = 6; //Cancela un aviso de traspaleo.
paramCancela.operacion = "TRASPALAO0001";

//Genera el proxy del servicio
```

```
ProxyRecintos.Recintos recinto = new Recintos();

/* Define los datos del encabezado */
EncabezadoSOAP enc = new EncabezadoSOAP();
enc.idUsuario = "RECINTO68";
enc.firma = new byte[60];

Cancelacion res;
recinto.EncabezadoSOAPValue = enc;

res = recinto.cancelaOperacion(paramCancela);

//Obtiene los errores de los contenedores origen.
ResultadoEntrada resEnt;
for(i=0; i<res.listaOrigen.Length; i++)
{
    resEnt = (ResultadoEntrada)res.listaOrigen[i];
    switch(i)
    {
        case 0:
            lblOrg1.Text = resEnt.listaErrores;
            break;
        case 1:
            lblOrg2.Text = resEnt.listaErrores;
            break;
        case 2:
            lblOrg3.Text = resEnt.listaErrores;
            break;
        case 3:
            lblOrg4.Text = resEnt.listaErrores;
            break;
        case 4:
            lblOrg5.Text = resEnt.listaErrores;
            break;
    }
}

//Obtiene los errores de los contenedores destino.
for(i=0; i<res.listaEntradas.Length; i++)
{
    resEnt = (ResultadoEntrada)res.listaEntradas[i];
    switch(i)
    {
        case 0:
            lblDes1.Text = resEnt.listaErrores;
            break;
        case 1:
    }
```



```
        lblDes2.Text = resEnt.listaErrores;  
        break;  
    case 2:  
        lblDes3.Text = resEnt.listaErrores;  
        break;  
    case 3:  
        lblDes4.Text = resEnt.listaErrores;  
        break;  
    case 4:  
        lblDes5.Text = resEnt.listaErrores;  
        break;  
    }  
}  
  
//Obtiene los errores generales.  
lblError.Text = "Error: " + res.error;  
lblDesc.Text = res.descripcion;
```

Segundo paso. Registra la conclusión del traspaleo de uno o más contenedores

El recinto fiscalizado origen debe usar este método para registrar el final de una operación de traspaleo en uno o más contenedores destino con el siguiente método.

MÉTODO

ResultadoTraspaleo registraFinTraspaleo (ParametroTraspaleo paramTraspaleo)

Parámetros:

ParametroTraspaleo:

- **numAviso**. Es el número de aviso de traspaleo asignado por el recinto.
- **listaContOrig**. Este dato debe ser nulo.
- **fecha**. Indica la fecha programada del traspaleo.
- **listaContDest**. Este dato debe ser nulo.
- **tarja**. Este dato debe ser nulo.
- **listaContFinal**. Es una lista de objetos del tipo **ContFinal** que indican a los contenedores que ya concluyeron la operación de traspaleo y tienen la siguiente estructura:
 - **entrada**. Es el número de entrada del contenedor destino del traspaleo.
 - **descrip**. Es la descripción de la mercancía.
 - **marcas**. Identifica a las marcas del contenedor.
 - **peso**. Es el peso de la mercancía “traspaleada” más la mercancía que ya tenía el contenedor antes de la operación de traspaleo.
 - **piezas**. Es la cantidad de piezas “traspaleadas” más las piezas que ya existían en el contenedor antes de la operación de traspaleo.
 - **unidadMed**. Es la clave de la unidad de medida del peso.
 - **sellos**. Es la lista de sellos que tiene el contenedor destino al terminar la operación de traspaleo. Esta es una lista de objetos del tipo **psello** y tiene la siguiente estructura:
 - **número**. Identifica al número de sello.

MENSAJE SOAP

Este es el formato del mensaje SOAP para registrar el fin del traspaleo de un contenedor.

```
POST /wsRefis/RecintosFis.asmx HTTP/1.1
Host: satcpnmxch02
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Recintos/ServiciosWebXml/registraFinTraspaleo"
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <EncabezadoSOAP xmlns="http://Recintos/ServiciosWebXml/">

      <idUsuario>string</idUsuario>
      <firma>base64Binary</firma>
      <ipCliente>string</ipCliente>
    </EncabezadoSOAP>
  </soap:Header>
  <soap:Body>
    <registraFinTraspaleo xmlns="http://Recintos/ServiciosWebXml/">
      <paramTraspaleo>
        <numAviso>string</numAviso>
        <listaContOrig>
        </listaContOrig>
        <fecha>string</fecha>
        <listaContDest>
        </listaContDest>
        <tarja>string</tarja>
        <listaContFinal>
          <ContFinal>
            <entrada>string</entrada>
            <descrip>string</descrip>
            <marcas>string</marcas>
            <peso>string</peso>
            <piezas>string</piezas>
            <unidadMed>string</unidadMed>
            <sellos xsi:nil="true" />
          </ContFinal>
          <ContFinal>
            <entrada>string</entrada>
            <descrip>string</descrip>
            <marcas>string</marcas>
            <peso>string</peso>
          </ContFinal>
        </listaContFinal>
      </paramTraspaleo>
    </registraFinTraspaleo>
  </soap:Body>
</soap:Envelope>
```

```
<piezas>string</piezas>
<unidadMed>string</unidadMed>
<sellos xsi:nil="true" />
</ContFinal>
</listaContFinal>
</paramTraspaleo>
</registraFinTraspaleo>
</soap:Body>
</soap:Envelope>
```

RESPUESTA DEL MÉTODO

Este método regresa un objeto del tipo **ResultadoTraspaleo**. Una vez que el método terminó de procesar la solicitud del usuario, regresa el resultado de la operación con el siguiente formato:

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <registraFinTraspaleoResponse xmlns="http://Recintos/ServiciosWebXml/">
      <registraFinTraspaleoResult>
        <error>int</error>
        <descripError>string</descripError>
        <acuse>string</acuse>
        <listaEntradas>
          <anyType />
          <anyType />
        </listaEntradas>
        <listaOrigen>
          <anyType />
          <anyType />
        </listaOrigen>
      </registraFinTraspaleoResult>
    </registraFinTraspaleoResponse>
```

```
</soap:Body>  
</soap:Envelope>
```

Donde:

error. Contiene el número del último error ocurrido en la operación.

descripError. Es una cadena de caracteres que tiene los errores de tipo general encontrados en el proceso. Cada error esta encerrado en los caracteres “[]”.

acuse. En esta operación este campo es nulo.

listaOrigen. Es un arreglo que contiene objetos del tipo **ResultadoEntrada** que contiene el resultado de la validación de los contenedores origen del traspaleo.

listaEntradas. Es un arreglo que contiene objetos del tipo **ResultadoEntrada** con el resultado de la validación de los contenedores destino del traspaleo.

EJEMPLO CON C#

```
ParametroTraspaleo paramTras = new ParametroTraspaleo();  
ContFinal cFinal; //datos del contenedor a finalizar la op. de traspaleo.  
ContFinal[] listaFinal = new ContFinal[1];  
psello sello;  
psello[] listaSellos;  
  
//Aunque pueden establecerse varios contenedores en la lista  
//de fin de operación de traspaleo, este ejemplo sólo emplea uno.  
cFinal = new ContFinal();  
cFinal.entrada = TBCont.Text;  
cFinal.descrip = TBDescrip.Text;  
cFinal.marcas = TBMarcas.Text;  
cFinal.peso = TBPeso.Text;  
cFinal.piezas = TBPiezas.Text;  
cFinal.unidadMed = TBUMedida.Text;
```

```
//Crea la lista de sello del contenedor
listaSellos = new psello[5];

sello = new psello();
sello.numero = "SELLO-1";
listaSellos[0] = sello;

sello = new psello();
sello.numero = "SELLO-2";
listaSellos[1] = sello;

sello = new psello();
sello.numero = "SELLO-3";
listaSellos[2] = sello;

sello = new psello();
sello.numero = "SELLO-4";
listaSellos[3] = sello;

sello = new psello();
sello.numero = "SELLO-5";
listaSellos[4] = sello;

cFinal.sellos = listaSellos;
listaFinal[0] = cFinal;

//Asigna valores a parámetros
paramTras.numAviso = "TRASPALCO-001";
paramTras.fecha = "2005-02-01 00:00:00";
paramTras.tarja = "TARJA-00X";

paramTras.listaContFinal = listaFinal;

//Genera el proxy del servicio
ProxyRecintos.Recintos recinto = new Recintos();

/* Define los datos del encabezado */
EncabezadoSOAP enc = new EncabezadoSOAP();
enc.idUsuario = "RECINTO69";
enc.firma = new byte[60];

ResultadoTraspaleo res;
recinto.EncabezadoSOAPValue = enc;

res = recinto.registraFinTraspaleo(paramTras);

//Obtiene los errores de los contenedores origen.
```

```
ResultadoEntrada resEnt;
for(i=0; i<res.listaOrigen.Length; i++)
{
    resEnt = (ResultadoEntrada)res.listaOrigen[i];
    switch(i)
    {
        case 0:
            lblOrg1.Text = resEnt.listaErrores;
            break;
        case 1:
            lblOrg2.Text = resEnt.listaErrores;
            break;
        case 2:
            lblOrg3.Text = resEnt.listaErrores;
            break;
        case 3:
            lblOrg4.Text = resEnt.listaErrores;
            break;
        case 4:
            lblOrg5.Text = resEnt.listaErrores;
            break;
    }
}

//Obtiene los errores de los contenedores destino.
for(i=0; i<res.listaEntradas.Length; i++)
{
    resEnt = (ResultadoEntrada)res.listaEntradas[i];
    switch(i)
    {
        case 0:
            lblDes1.Text = resEnt.listaErrores;
            break;
        case 1:
            lblDes2.Text = resEnt.listaErrores;
            break;
        case 2:
            lblDes3.Text = resEnt.listaErrores;
            break;
        case 3:
            lblDes4.Text = resEnt.listaErrores;
            break;
        case 4:
            lblDes5.Text = resEnt.listaErrores;
            break;
    }
}
```

```
//Obtiene los errores generales.  
lblError.Text = "Error: " + res.error;  
lblDesc.Text = res.descripcion;
```


4ª Etapa Desconsolidación y registro de incidencias

Desconsolidación de un contenedor

La desconsolidación de un contenedor es vaciar total o parcialmente la mercancía que está dentro de un contenedor. El resultado de esta operación son N entradas de mercancía suelta nuevas que son definidas por el recinto fiscalizado al momento de concretar la operación. La desconsolidación de un contenedor debe darse en 2 pasos:

1. El recinto registra el aviso de desconsolidación de un contenedor.
2. El recinto registra la conclusión de la operación de desconsolidación en mercancía suelta.

Primer paso. Registra el aviso de desconsolidación de un contenedor

El recinto fiscalizado que requiere realizar la desconsolidación de la mercancía de un contenedor que ya está en el recinto, debe usar este método para registrar un aviso de desconsolidación.

MÉTODO

Desconsolidacion registraAvisoDescon(ParametroAvisoDes paramDes)

Parámetros:

ParametroAvisoDes:

- **numAviso.** Es el número del aviso asignado por el recinto.
- **entrada.** Es el número de entrada del contenedor que se va a desconsolidar.
- **total.** Identifica si la desconsolidación será total o parcial. Este parámetro puede tener el valor "verdadero", "si", "true" ó "1" para indicar que la desconsolidación será total, cualquier otro valor indica que la desconsolidación será parcial.
- **fechaInicio.** Indica la fecha en que se tiene planeado iniciar la desconsolidación del contenedor. Esta fecha debe estar en el formato AAAA-MM-DD.

MENSAJE SOAP

Este es el formato del mensaje SOAP para registrar un aviso de desconsolidación de un contenedor que está en el recinto fiscalizado.

```
POST /wsRecintos/RecintosFis.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Recintos/ServiciosWebXml/registraAvisoDescon"
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <EncabezadoSOAP xmlns="http://Recintos/ServiciosWebXml/">
      <idUsuario>string</idUsuario>
      <firma>base64Binary</firma>
      <ipCliente>string</ipCliente>
    </EncabezadoSOAP>
  </soap:Header>
  <soap:Body>
    <registraAvisoDescon xmlns="http://Recintos/ServiciosWebXml/">
      <paramDes>
        <numAviso>string</numAviso>
        <entrada>string</entrada>
        <total>string</total>
        <fechaInicio>string</fechaInicio>
      </paramDes>
    </registraAvisoDescon>
  </soap:Body>
</soap:Envelope>
```

RESPUESTA DEL MÉTODO

Una vez que el método terminó de procesar la solicitud del usuario, regresa el resultado de la operación con el siguiente formato:

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <registraAvisoDesconResponse xmlns="http://Recintos/ServiciosWebXml/">
```

```
<registraAvisoDesconResult>
  <acuse>string</acuse>
  <listaEntradas>
    <ResultadoEntrada>
      <consecutivo>string</consecutivo>
      <numEntrada>int</numEntrada>
      <acuse>string</acuse>
    </ResultadoEntrada>
    <ResultadoEntrada>
      <consecutivo>string</consecutivo>
      <numEntrada>int</numEntrada>
      <acuse>string</acuse>
    </ResultadoEntrada>
  </listaEntradas>
  <listaOrigen>
    <anyType />
    <anyType />
  </listaOrigen>
</registraAvisoDesconResult>
</registraAvisoDesconResponse>
</soap:Body>
</soap:Envelope>
```

Donde:

error. Es el número del último error que sucedió durante el registro del aviso.

descripError. Es una cadena de caracteres que tiene los errores de tipo general encontrados en el proceso. Cada error está encerrado en los caracteres “[]”.

acuse. Es el acuse de recibo.

listaEntradas. En esta operación este campo es nulo.

listaOrigen. Es un arreglo que contiene un solo objeto del tipo **ResultadoEntrada**, este objeto tiene el resultado de la validación de la entrada origen de la desconsolidación.

EJEMPLO CON C#

```
ResultadoEntrada resEnt;

//Crea los parámetros del aviso
```

```
ParametroAvisoDes paramDes = new ParametroAvisoDes();
paramDes.numAviso      = "DESCONSOLIDACION-001";
paramDes.total         = "true";
paramDes.entrada       = "321";
paramDes.fechaInicio  = "2005-02-28";

//Genera el proxy del servicio
ProxyRecintos.Recintos recinto = new Recintos();

//Define los datos del encabezado
EncabezadoSOAP enc = new EncabezadoSOAP();
enc.idUsuario = "RECINTO69";
enc.firma     = new byte[60];
recinto.EncabezadoSOAPValue = enc;

Desconsolidacion res;
res = recinto.registraAvisoDescon(paramDes);

//Obtiene el resultado de la validación del contenedor origen de la desconsolidación
if(res.listaOrigen.Length > 0)
{
    resEnt = (ResultadoEntrada)res.listaOrigen[0];
    lblErrEntAviso.Text = "Entrada: " + resEnt.listaErrores;
}

//Obtiene los errores generales de la desconsolidación
string lblError1 = "Error: " + res.error.ToString();
string lblDescrip1 = res.descripcionError;
```

Cancela el aviso de desconsolidación

Para cancelar un aviso de desconsolidación que ya haya sido registrado, el recinto debe usar el siguiente método:

MÉTODO

Cancelacion cancelaOperacion(ParametroCancela paramCancela)

Parámetros:

ParametroCancela:

- **tipoOperacion.** Este dato indica el tipo de operación a cancelar, para cancelar un aviso de desconsolidación este dato debe ser **7**.
- **operacion.** Este es el número del aviso que se desea cancelar.
- **parcialidad.** Este dato debe ser nulo.

Una vez que se cancela un aviso de desconsolidación, esta operación cancela también a todas las mercancías que se hayan desconsolidado hasta este momento para el aviso.

MENSAJE SOAP

Este es el formato del mensaje SOAP para cancelar un aviso de desconsolidación.

```
POST /wsRefis/RecintosFis.asmx HTTP/1.1
Host: satcpnpxch02
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Recintos/ServiciosWebXml/cancelaOperacion"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <EncabezadoSOAP xmlns="http://Recintos/ServiciosWebXml/">
      <idUsuario>string</idUsuario>
      <firma>base64Binary</firma>
      <ipCliente>string</ipCliente>
    </EncabezadoSOAP>
  </soap:Header>
  <soap:Body>
    <cancelaOperacion xmlns="http://Recintos/ServiciosWebXml/">
```

```
<paramCancela>  
  <tipoOperacion>int</tipoOperacion>  
  <operacion>string</operacion>  
  <parcialidad>int</parcialidad>  
</paramCancela>  
</cancelaOperacion>  
</soap:Body>  
</soap:Envelope>
```

RESPUESTA DEL MÉTODO

Este método regresa un objeto del tipo **Cancelación** con el siguiente formato.

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <cancelaOperacionResponse xmlns="http://Recintos/ServiciosWebXml/">
      <cancelaOperacionResult>
        <error>int</error>
        <descripError>string</descripError>
      </cancelaOperacionResult>
    </cancelaOperacionResponse>
  </soap:Body>
</soap:Envelope>
```

Donde:

error. Contiene el número del último error ocurrido en la operación.

descripError. Tiene la descripción completa de los errores ocurridos en la operación.

listaOrigen. Es un arreglo que contiene un solo objeto del tipo **ResultadoEntrada**, este objeto tiene el resultado de la validación de la entrada origen de la desconsolidación.

listaEntradas. Es una arreglo que contiene el resultado de la validación hecha a las mercancías sueltas desconsolidadas hasta el momento del contenedor origen. Por cada desconsolidación registrada hay un elemento **ResultadoEntrada**.

EJEMPLO CON C#

```
ResultadoEntrada resEnt;

//Genera los parámetros de la cancelación
ParametroCancela paramCancela = new ParametroCancela();
paramCancela.tipoOperacion = 7;
paramCancela.operacion = "DESCONSOLIDACION-001";

//Genera el proxy del servicio
ProxyRecintos.Recintos recinto = new Recintos();

/* Define los datos del encabezado */
EncabezadoSOAP enc = new EncabezadoSOAP();
enc.idUsuario = "RECINTO69";
enc.firma = new byte[60];

Cancelacion res;
recinto.EncabezadoSOAPValue = enc;

res = recinto.cancelaOperacion(paramCancela);

//Obtiene el resultado de la validación del contenedor origen de la desconsolidación
if(res.listaOrigen.Length > 0)
{
    resEnt = (ResultadoEntrada)res.listaOrigen [0];
    lblErrEntAviso.Text = "Entrada: " + resEnt.listaErrores;
}

//Obtiene el resultado de la validación a las mercancías ya desconsolidadas del //contendor
if(res.listaEntradas.Length > 0)
{
    for(int i=0; i< res.listaEntradas.Length; i++)
    {
        resEnt = (ResultadoEntrada)res.listaEntradas[i];
        lblErrEntrada.Text = "Entrada: " + resEnt.listaErrores;
    }
}

//Obtiene los errores generales.
lblError.Text = "Error: " + res.error.ToString();
lblDesc.Text = res.descripcionError;
```


Segundo paso. Registra la conclusión de la desconsolidación del contenedor

El recinto fiscalizado origen debe usar este método para:

- Registrar las partes de la desconsolidación sin terminar la operación de desconsolidación, esta operación genera nuevas entradas de mercancía.
- Registrar la(s) última(s) parte(s) de la desconsolidación y la conclusión de la operación de desconsolidación del contenedor origen.

Aunque no siempre se registra la conclusión de la operación de desconsolidación con este método, le hemos dado ese nombre para simplificar su identificación.

MÉTODO

Desconsolidacion desconsolida (ParametroDesconsolida paramDes)

Parámetros:

ParametroDesconsolida:

- **numAviso**. Es el número de aviso de desconsolidación que se registró anteriormente.
- **listaMercancia**. Contiene los detalles de la mercancía que va a desconsolidarse del contenedor. Los detalles de la mercancía que va a desconsolidarse deben entregarse dentro de un objeto del tipo **ParametroSuelta** que tiene la siguiente estructura:
 - **esParcial**. En esta operación este dato no es significativo y por lo tanto es nulo.
 - **mercancia**. Es un número consecutivo que se usará como referencia al momento de entregar el resultado del método.
 - **peso**. Es el peso de la nueva mercancía suelta.
 - **piezas**. Es el número de piezas de la nueva mercancía suelta.
 - **detalleMerc**. Contiene más detalles particulares de la mercancía en un objeto del tipo **DetalleMercancia** que tiene la siguiente estructura:
 - **descripcion**. Es la descripción textual de la mercancía suelta.
 - **embalaje**. Es la clave del embalaje de la mercancía suelta.
 - **marcas**. Son las marcas y números de la mercancía suelta.
 - **unidMedida**. Es la clave de la unidad de medida del peso.
 - **tipoMercancia**. Es un arreglo de objetos del tipo **TipoMercancia** con la clave de los tipos de mercancía suelta que tienen la siguiente estructura:
 - **cTipoMerc**. En este método este campo es nulo.
 - **cuTipoMerc**. Contiene la clave de usuario del tipo de mercancía.
 - **nomTipoMerc**. En este método este campo es nulo.
 - **fAbandono**. En este método este campo es nulo.
- **finOperacion**. Identifica si esta es la última operación de desconsolidación del contenedor. Para indicar que sí es la última operación puede tener el valor "verdadero", "sí", "true" ó "1". Cualquier otro valor indica que no es la última operación.

MENSAJE SOAP

Este es el formato del mensaje SOAP para registrar el término de una operación de desconsolidación de un contenedor.

```
POST /wsRecintos/RecintosFis.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Recintos/ServiciosWebXml/desconsolida"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <EncabezadoSOAP xmlns="http://Recintos/ServiciosWebXml/">
      <idUsuario>string</idUsuario>
      <firma>base64Binary</firma>
      <ipCliente>string</ipCliente>
    </EncabezadoSOAP>
  </soap:Header>
  <soap:Body>
    <desconsolida xmlns="http://Recintos/ServiciosWebXml/">
      <paramDes>
        <numAviso>string</numAviso>
        <listaMercancia>
          <ParametroSuelta>
            <esParcial>boolean</esParcial>
            <mercancia>int</mercancia>
            <peso>double</peso>
            <piezas>int</piezas>
            <detalleMerc xsi:nil="true" />
          </ParametroSuelta>
          <ParametroSuelta>
            <esParcial>boolean</esParcial>
            <mercancia>int</mercancia>
            <peso>double</peso>
            <piezas>int</piezas>
            <detalleMerc xsi:nil="true" />
          </ParametroSuelta>
        </listaMercancia>
      </paramDes>
    </desconsolida>
  </soap:Body>
</soap:Envelope>
```

```
</ParametroSuelta>  
</listaMercancia>  
<finOperacion>string</finOperacion>  
</paramDes>  
</desconsolida>  
</soap:Body>  
</soap:Envelope>
```

RESPUESTA DEL MÉTODO

Una vez que el método terminó de procesar la solicitud del usuario, regresa el resultado de la operación con el siguiente formato:

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <desconsolidaResponse xmlns="http://Recintos/ServiciosWebXml/">
      <desconsolidaResult>
        <acuse>string</acuse>
        <listaEntradas>
          <ResultadoEntrada>
            <consecutivo>string</consecutivo>
            <numEntrada>int</numEntrada>
            <acuse>string</acuse>
          </ResultadoEntrada>
          <ResultadoEntrada>
            <consecutivo>string</consecutivo>
            <numEntrada>int</numEntrada>
            <acuse>string</acuse>
          </ResultadoEntrada>
        </listaEntradas>
        <listaOrigen>
          <anyType />
          <anyType />
        </listaOrigen>
      </desconsolidaResult>
    </desconsolidaResponse>
  </soap:Body>
</soap:Envelope>
```

Donde:

error. Es el número del último error que sucedió durante la desconsolidación.

descripError. Es una cadena de caracteres que tiene los errores de tipo general encontrados en el proceso. Cada error esta encerrado en los caracteres “[]”.

acuse. En este momento este campo no es significativo y su valor siempre es nulo.

listaOrigen. Es un arreglo de objetos del tipo **ResultadoEntrada** con un solo elemento que tiene la validación del contenedor origen de la desconsolidación.

listaEntradas. En este arreglo regresa los números de entrada que el sistema le asignó a la mercancía suelta declarada en los parámetros del método en objetos del tipo **ResultadoEntrada**.

EJEMPLO CON C#

```
ResultadoEntrada resEnt;
ParametroSuelta paramSuelta;
DetalleMercancia detalle;
TipoMercancia tMerc;
TipoMercancia[] listaTipoMerc;

//Crea los parámetros para registrar la desconsolidación.
ParametroDesconsolida paramDes = new ParametroDesconsolida();

//Este ejemplo va a desconsolidar 2 mercancías del contenedor.
ParametroSuelta[] listaMercancia = new ParametroSuelta[2];

//Datos de la primera parte de la desconsolidación.
paramDes.numAviso = "DESCONSOLIDACION-001";
paramDes.finOperacion = "falso";

paramSuelta = new ParametroSuelta();
paramSuelta.mercancia = 1;
paramSuelta.peso = 185.13;
paramSuelta.piezas = 80;

//Crea el detalle de la mercancía.
detalle = new DetalleMercancia();
detalle.embalaje = "26";
detalle.unidMedida = "1";
detalle.descripcion = "DESCRIPCION DE LA MERCANCIA DESCONSOLIDADA UNO";
detalle.marcas = "MARCAS DE LA MERCANCIA DESCONSOLIDADA UNO";

//Crea el arreglo de tipo de mercancía para la desconsolidación.
listaTipoMerc = new TipoMercancia[1];
tMerc = new TipoMercancia();
tMerc.cuTipoMerc = "0";
listaTipoMerc[0] = tMerc;

//Agrega el tipo de mercancía al detalle
detalle.tipoMercancia = listaTipoMerc;

//Agrega el detalle a los datos de la mercancía
paramSuelta.detalleMerc = detalle;

//Agrega los datos de la mercancía a desconsolidar
listaMercancia[0] = paramSuelta;

//Datos de la segunda parte de la desconsolidación.
paramSuelta = new ParametroSuelta();
```

```
paramSuelta.mercancia      = 2;
paramSuelta.peso           = 833.19;
paramSuelta.piezas         = 443;

//Crea el detalle de la mercancía.
detalle = new DetalleMercancia();
detalle.embalaje           = "27";
detalle.unidMedida         = "1";
detalle.descripcion        = "DESCRIPCION DE LA MERCANCIA DESCONSOLIDADA DOS";
detalle.marcas             = "MARCAS DE LA MERCANCIA DESCONSOLIDADA DOS";

//Crea el arreglo de tipo de mercancía para la desconsolidación.
listaTipoMerc = new TipoMercancia[1];
tMerc = new TipoMercancia();
tMerc.cuTipoMerc = "0";
listaTipoMerc[0] = tMerc;

//Agrega el tipo de mercancía al detalle
detalle.tipoMercancia = listaTipoMerc;

//Agrega el detalle a los datos de la mercancía
paramSuelta.detalleMerc = detalle;

//Agrega los datos de la mercancía a desconsolidar
listaMercancia[1] = paramSuelta;
paramDes.listaMercancia = listaMercancia;

//Genera el proxy del servicio
ProxyRecintos.Recintos recinto = new Recintos();

//Define los datos del encabezado
EncabezadoSOAP enc = new EncabezadoSOAP();
enc.idUsuario = TBUusuario2.Text;
enc.firma      = new byte[60];

Desconsolidacion res;
recinto.EncabezadoSOAPValue = enc;

res = recinto.desconsolida(paramDes);

//Obtiene el resultado de la validación del contenedor origen de la desconsolidación
if(res.listaOrigen.Length > 0)
{
    resEnt = (ResultadoEntrada)res.listaOrigen[0];
    lblErrEntAviso.Text = "Entrada: " + resEnt.listaErrores;
}
}
```

```
if(res.error == 0)
{
    //Muestra los números de entrada que le corresponden a las desconsolidaciones
    ResultadoEntrada resEntrada = (ResultadoEntrada)res.listaEntradas[0];
    lblEntrada1.Text = "Número de entrada " + resEntrada.numEntrada + " : " + resEntrada.acuse;

    ResultadoEntrada resEntrada = (ResultadoEntrada)res.listaEntradas[1];
    lblEntrada2.Text = "Número de entrada " + resEntrada.numEntrada + " : " + resEntrada.acuse;
}

//Obtiene los errores generales
lblError2.Text = "Error: " + res.error.ToString();
lblDescrip2.Text = res.descripcionError;
```


Registro de incidencias

El recinto fiscalizado puede registrar incidencias a la mercancía que ya está en su almacén para llevar un control más estricto de su estado. Las incidencias que puede registrar el recinto son:

1. **Mercancía robada.**
2. **Mercancía dañada.**
3. **Mercancía extraviada.**
4. **Mercancía localizada.**
5. **Mercancía violada.**

Registra un aviso de incidencia a una mercancía

El recinto fiscalizado que requiere registrar un incidente a una mercancía que ya está en el recinto, debe usar este método.

MÉTODO

`ResultadoIncidencia registraIncidencia (ParametroIncidencia paramIncid)`

Parámetros:

ParametroIncidencia:

- **string numAviso.** Es el número de aviso que el recinto asigna a la incidencia.
- **string tipo.** Identifica al tipo de aviso, hasta el momento están registrados los siguientes:

TIPO	DESCRIPCIÓN
6	ROBO
7	DAÑO
8	EXTRAVIO
9	LOCALIZACIÓN
10	VIOLACIÓN

- **string entrada.** Indica el número de entrada de la mercancía a la que se le asignará la incidencia.
- **string observacion.** Descripción textual de la incidencia.

MENSAJE SOAP

Este es el formato del mensaje SOAP para registrar un aviso de incidencia a una mercancía que está en el recinto fiscalizado.

```
POST /wsRecintos/RecintosFis.asmx HTTP/1.1
```

```
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Recintos/ServiciosWebXml/registraIncidencia"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

  <soap:Header>
    <EncabezadoSOAP xmlns="http://Recintos/ServiciosWebXml/">
      <idUsuario>string</idUsuario>
      <firma>base64Binary</firma>
      <ipCliente>string</ipCliente>
    </EncabezadoSOAP>
  </soap:Header>
  <soap:Body>
    <registraIncidencia xmlns="http://Recintos/ServiciosWebXml/">
      <paramIncid>
        <numAviso>string</numAviso>
        <tipo>string</tipo>
        <entrada>string</entrada>
        <observacion>string</observacion>
      </paramIncid>
    </registraIncidencia>
  </soap:Body>
</soap:Envelope>
```

RESPUESTA DEL MÉTODO

Una vez que el método terminó de procesar la solicitud del usuario, regresa el resultado de la operación con el siguiente formato:

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
```

```
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <registraIncidenciaResponse xmlns="http://Recintos/ServiciosWebXml/">
      <registraIncidenciaResult>
        <error>int</error>
        <descripError>string</descripError>
        <acuse>string</acuse>
      </registraIncidenciaResult>
    </registraIncidenciaResponse>
  </soap:Body>
</soap:Envelope>
```

Donde:

error. Es el número de error que sucedió durante la operación.

descripError. Es una cadena de caracteres que tiene los errores encontrados en el proceso. Cada error esta encerrado en los caracteres “[]”.

acuse. Es el acuse de recibo del aviso.

EJEMPLO CON C#

```
ParametroIncidencia paramIncid = new ParametroIncidencia();
paramIncid.numAviso = "ROBO-005";
paramIncid.tipo = "6"; //Aviso de mercancía robada.
paramIncid.entrada = "3";
paramIncid.observacion = "SE IDENTIFICO QUE LA MERCANCIA FUE ROBADA EL 11-FEB-2005.";
//Genera el proxy del servicio
ProxyRecintos.Recintos recinto = new Recintos();

//Define los datos del encabezado
EncabezadoSOAP enc = new EncabezadoSOAP();
enc.idUsuario = "RECINTO69";
enc.firma = new byte[60];

ResultadoIncidencia res;
recinto.EncabezadoSOAPValue = enc;
```

```
res = recinto.registraIncidencia(paramIncid);  
  
lblError.Text = "Error: " + res.error.ToString();  
lblDescrip.Text = res.descripcion;
```

Cancela una incidencia

El recinto fiscalizado que requiera cancelar un incidente a una mercancía que ya está en el recinto, debe usar este método.

MÉTODO

Cancelacion `cancelaOperacion` (`ParametroCancela paramCancela`)

Parámetros:

ParametroCancela:

- **tipoOperacion.** Este dato indica el tipo de operación a cancelar, en el caso de cancelar entradas este dato debe ser **8**.
- **operacion.** Indica el número de aviso a cancelar.

MENSAJE SOAP

Este es el formato del mensaje SOAP para cancelar una incidencia.

```
POST /wsRefis/RecintosFis.asmx HTTP/1.1
Host: satcpnpxch02
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Recintos/ServiciosWebXml/cancelaOperacion"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <EncabezadoSOAP xmlns="http://Recintos/ServiciosWebXml/">
      <idUsuario>string</idUsuario>
      <firma>base64Binary</firma>
      <ipCliente>string</ipCliente>
    </EncabezadoSOAP>
  </soap:Header>
  <soap:Body>
    <cancelaOperacion xmlns="http://Recintos/ServiciosWebXml/">
      <paramCancela>
        <tipoOperacion>int</tipoOperacion>
        <operacion>string</operacion>
        <parcialidad>int</parcialidad>
      </paramCancela>
    </cancelaOperacion>
  </soap:Body>
</soap:Envelope>
```

```
</cancelaOperacion>  
</soap:Body>  
</soap:Envelope>
```

RESPUESTA DEL MÉTODO

Este método regresa un objeto del tipo **Cancelación** con el siguiente formato.

```
HTTP/1.1 200 OK  
Content-Type: text/xml; charset=utf-8  
Content-Length: length  
  
<?xml version="1.0" encoding="utf-8"?>  
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
  <soap:Body>  
    <cancelaOperacionResponse xmlns="http://Recintos/ServiciosWebXml/">  
      <cancelaOperacionResult>  
        <error>int</error>  
        <descripError>string</descripError>  
      </cancelaOperacionResult>  
    </cancelaOperacionResponse>  
  </soap:Body>  
</soap:Envelope>
```

Donde:

error. Contiene el número del último error ocurrido en la operación.

descripError. Es una cadena de caracteres que tiene los errores encontrados en el proceso. Cada error esta encerrado en los caracteres “[]”.

EJEMPLO CON C#

```
ParametroCancela param = new ParametroCancela();
param.tipoOperacion = 8;           //Cancela una incidencia
param.operacion = "EXTRAVIO-012"; //Número de incidencia a cancelar

//Genera el proxy del servicio
ProxyRecintos.Recintos recinto = new Recintos();

//Define los datos del encabezado
EncabezadoSOAP enc = new EncabezadoSOAP();
enc.idUsuario = "RECINTO69";
enc.firma = new byte[60];

Cancelacion res;
recinto.EncabezadoSOAPValue = enc;

res = recinto.cancelaOperacion(paramCancela);

//Obtiene los errores generales.
lblError.Text = "Error: " + res.error.ToString();
lblDesc.Text = res.descripError;
```

5ª Etapa: registro de entradas y salidas por órdenes de una autoridad

En ocasiones es necesario darle entrada y salida a la mercancía del recinto con base en una solicitud de una autoridad, en estos casos el recinto fiscalizado debe utilizar estos métodos.

Registra entradas por órdenes de una autoridad

MÉTODO

EntradaPorAutoridad registra **EntradaXAutoridad**(ParametroEntrada paramEnt)

Parámetros:

ParametroEntrada:

- **manifiesto.** Este dato debe ser nulo.
- **nomConocimiento.** Este dato debe ser nulo.
- **numMercancia.** Este dato debe ser nulo.
- **cuPais.** Este dato debe ser nulo.
- **cuPuerto.** Este dato debe ser nulo.
- **autoridad.** Es la clave de la autoridad que está solicitando la entrada de la mercancía de acuerdo con el **Catálogo de autoridad.**
- **numOrden.** Número de orden de la autoridad.
- **tipoEntrada.** Identifica el tipo de entrada de acuerdo con el **Catálogo de tipo de entrada**, el tipo puede ser:
 - Por PAMA. Valor en el catálogo = 7.
 - Por retención. Valor en el catálogo = 8.
 - Por verificación. Valor en el catálogo = 9.
- **listaMercancia (arreglo de mercancía suelta)**

Este arreglo debe usarse siempre que la mercancía a ingresar no tenga como embalaje a un contenedor y debe contener a un objeto del tipo **ParametroSuelta** por cada mercancía suelta a ingresar al almacén.

- **ParametroSuelta**
 - **entrada.** Este dato debe ser nulo.
 - **esParcial.** Este dato debe ser nulo.
 - **mercancia.** Es un número consecutivo para cada elemento del arreglo y debe comenzar con 0 (cero) e incrementarse de uno en uno.
 - **peso.** Indica el peso de la mercancía a ingresar.
 - **piezas.** Indica la cantidad de piezas de la mercancía a ingresar.
 - **detalleMerc.** Contiene el detalle de la mercancía a ingresar.
 - **descripcion.** Contiene la descripción de la mercancía a entrar.
 - **embalaje.** Es la clave del embalaje de la mercancía de acuerdo con el **Catálogo de embalajes.**

- **tipoMercancia.** Es un arreglo que contiene las claves del tipo de mercancía de acuerdo con el **Catálogo de tipos de mercancía.**
 - **cuTipoMerc.** Es la clave del tipo de mercancía de acuerdo con el **Catálogo de tipos de mercancía.**
 - **marcas.** Contiene las marcas y números que identifican a la mercancía.
 - **unidadMedida.** Es la clave de unidad de medida de acuerdo con el **Catálogo de unidades de medida.**
- **listaContenedor (arreglo de mercancía contenerizada)**

Este arreglo debe usarse siempre que la mercancía a ingresar tenga como embalaje a un contenedor y debe contener un objeto del tipo **ParametroCont** por cada contenedor a ingresar al almacén.

 - **ParametroCont**
 - **nomContenedor.** Identificador del contenedor de la mercancía.
 - **peso.** Peso del contenedor con la mercancía.
 - **piezas.** Piezas que están en el contenedor.
 - **tipoContenedor.** Tipo de contenedor de acuerdo con el **Catálogo de tipo de contenedor.**
 - **tipoServicio.** Tipo de servicio del contenedor de acuerdo con el **Catálogo de tipo de servicio de contenedor.**
 - **listaSellos.** Lista de sellos que tiene el contenedor.
 - **numSello.** Número de sello del contenedor.
 - **detalleMerc.** Tiene la misma estructura que **detalleMerc** de **ParametroSuelta** descrito en **listaMercancia.**

MENSAJE SOAP

Este es el formato del mensaje SOAP para registrar una entrada de este tipo.

```
POST /wsRefis/RecintosFis.asmx HTTP/1.1
Host: satcpnpxch02
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Recintos/ServiciosWebXml/registraEntradaXAutoridad"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <EncabezadoSOAP xmlns="http://Recintos/ServiciosWebXml/">
      <idUsuario>string</idUsuario>
      <firma>base64Binary</firma>
    </EncabezadoSOAP>
  </soap:Header>
  <body>
    <ParametroCont>
      <nomContenedor>string</nomContenedor>
      <peso>float</peso>
      <piezas>int</piezas>
      <tipoContenedor>string</tipoContenedor>
      <tipoServicio>string</tipoServicio>
      <listaSellos>
        <numSello>int</numSello>
      </listaSellos>
      <detalleMerc>
        <tipoMercancia>string</tipoMercancia>
        <marcas>string</marcas>
        <unidadMedida>string</unidadMedida>
      </detalleMerc>
    </ParametroCont>
  </body>
</soap:Envelope>
```

```
<ipCliente>string</ipCliente>
</EncabezadoSOAP>
</soap:Header>
<soap:Body>
  <registraEntradaXAutoridad xmlns="http://Recintos/ServiciosWebXml/">
    <paramEnt>
      <manifiesto>string</manifiesto>
      <nomConocimiento>string</nomConocimiento>

      <cuPais>string</cuPais>
      <cuPuerto>string</cuPuerto>
      <numAviso>string</numAviso>
      <autoridad>string</autoridad>
      <numDocumento>string</numDocumento>
      <tipoEntrada>int</tipoEntrada>
      <observacion>string</observacion>
      <listaMercancia>
        <ParametroSuelta>
          <entrada>int</entrada>
          <esParcial>boolean</esParcial>
          <mercancia>int</mercancia>
          <peso>double</peso>
          <piezas>int</piezas>
          <detalleMerc xsi:nil="true" />
        </ParametroSuelta>
        <ParametroSuelta>
          <entrada>int</entrada>
          <esParcial>boolean</esParcial>
          <mercancia>int</mercancia>
          <peso>double</peso>
          <piezas>int</piezas>
          <detalleMerc xsi:nil="true" />
        </ParametroSuelta>
      </listaMercancia>
      <listaContenedor>
        <ParametroCont>
          <nomContenedor>string</nomContenedor>
          <peso>double</peso>
```

```
<piezas>int</piezas>
<tipoContenedor>string</tipoContenedor>
<tipoServicio>string</tipoServicio>
<listaSellos xsi:nil="true" />
<detalleMerc xsi:nil="true" />
</ParametroCont>
<ParametroCont>
  <nomContenedor>string</nomContenedor>
  <peso>double</peso>
  <piezas>int</piezas>
  <tipoContenedor>string</tipoContenedor>
  <tipoServicio>string</tipoServicio>
  <listaSellos xsi:nil="true" />
  <detalleMerc xsi:nil="true" />
</ParametroCont>
</listaContenedor>
<recintoOrigen>string</recintoOrigen>
</paramEnt>
</registraEntradaXAutoridad>
</soap:Body>
</soap:Envelope>
```

RESPUESTA DEL MÉTODO

Una vez que el método terminó de procesar la solicitud del usuario, regresa el resultado de la operación con el siguiente formato:

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <registraEntradaXAutoridadResponse xmlns="http://Recintos/ServiciosWebXml/">
      <registraEntradaXAutoridadResult>
        <error>int</error>
        <descripError>string</descripError>
        <listaEntradas>
        </listaEntradas>
      </registraEntradaXAutoridadResult>
    </registraEntradaXAutoridadResponse>
  </soap:Body>
</soap:Envelope>
```

Donde:

error. Es el número del último error que sucedió durante el registro de la entrada.

descripError. Es una cadena de caracteres que tiene los errores de tipo general encontrados en el proceso. Cada error está encerrado en los caracteres “[]”.

listaEntradas. En este arreglo regresa los números de entrada que el sistema le asignó a la mercancía declarada en los parámetros del método. Este arreglo contiene objetos del tipo **ResultadoEntrada** que tiene la siguiente estructura:

- **consecutivo.** Es el número consecutivo/nombre del contenedor declarado en los parámetros de entrada del método.
- **numEntrada.** Es el número de entrada de la mercancía asignado por el método.
- **acuse.** Es el acuse de la entrada calculado por el método.
- **listaErrores.** Es una cadena de caracteres que tiene los errores encontrados en el procesamiento de la entrada. Cada error está encerrado en los caracteres “[]”.

EJEMPLO CON C#

```
Recintos recinto;
EntradaPorAutoridad res;

//Crea los parámetros del método
ParametroEntrada paramEnt = new ParametroEntrada();
paramEnt.autoridad = TBAutoridad.Text;
paramEnt.numDocumento = "ACTAENT-001";
//Entrada por PAMA
paramEnt.tipoEntrada = 7;
//Crea un elemento de mercancía suelta
ParametroSuelta[] suelta = new ParametroSuelta[1];
suelta[0] = new ParametroSuelta();
suelta[0].mercancia = 0;
suelta[0].peso = 3221.33;
suelta[0].piezas = 5467;
//Crea un los datos del detalle de la mercancía suelta
suelta[0].detalleMerc = new DetalleMercancia();
suelta[0].detalleMerc.unidMedida = "1";
suelta[0].detalleMerc.descripcion = "DESCRIPCION MERCANCIA SUELTA 1";
suelta[0].detalleMerc.marcas = "MARCAS MERCANCIA SUELTA 1";
suelta[0].detalleMerc.embalaje = "26";
//Como la mercancía puede tener varios tipos, crea el arreglo de tipos de merc.
//En este caso tiene sólo un elemento.
TipoMercancia[] tMerca = new TipoMercancia[1];
tMerca[0] = new TipoMercancia();
tMerca[0].cuTipoMerc = "0";
suelta[0].detalleMerc.tipoMercancia = tMerca;
paramEnt.listaMercancia = suelta;

//Crea una instancia del proxy.
recinto = new Recintos();

//Declara el encabezado del mensaje.
EncabezadoSOAP encabezado = new ProxyRecintos.EncabezadoSOAP();
byte[] firma;

//En este momento no se está autenticando la firma del mensaje pero debe estar
//declarada como un arreglo de bytes.
firma = new byte[100];

//Establece los valores del encabezado.
encabezado.idUsuario = "RECINTO69";
encabezado.firma = firma;

//Asigna el encabezado a la instancia del proxy.
recinto.EncabezadoSOAPValue = encabezado;
```

```
//Ejecuta el método de entrada
res = recinto.registraEntradaXAutoridad(paramEnt);

lblResultado.Text = "Error: " + res.error.ToString();
lblResultado.Text+= " " + res.descripcion;
if(res.error == 0)
{
    ResultadoEntrada entrada = (ResultadoEntrada)res.listaEntradas[0];
    lblEntrada.Text = "Entrada: " + entrada.numEntrada +
        " Acuse: " + entrada.acuse;
}
```

Cancela entradas por órdenes de una autoridad

El recinto fiscalizado puede usar este método para cancelar una entrada de mercancía a su almacén registrada previamente.

MÉTODO

Cancelacion `cancelaOperacion (ParametroCancela paramCancela)`

Parámetros:

ParametroCancela:

- **tipoOperacion.** Este dato indica el tipo de operación a cancelar. Para las entradas de este tipo este dato debe ser **9**.
- **operacion.** Indica el número de entrada a cancelar.
- **parcialidad.** Este dato debe tener el valor 0 (cero).

MENSAJE SOAP

Este es el formato del mensaje SOAP para cancelar la entrada de mercancías en el almacén del recinto fiscalizado.

```
POST /wsRefis/RecintosFis.asmx HTTP/1.1
Host: satcpnpxch02
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Recintos/ServiciosWebXml/cancelaOperacion"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <EncabezadoSOAP xmlns="http://Recintos/ServiciosWebXml/">
      <idUsuario>string</idUsuario>
      <firma>base64Binary</firma>
      <ipCliente>string</ipCliente>
    </EncabezadoSOAP>
  </soap:Header>
  <soap:Body>
    <cancelaOperacion xmlns="http://Recintos/ServiciosWebXml/">
      <paramCancela>
        <tipoOperacion>int</tipoOperacion>
        <operacion>string</operacion>
        <parcialidad>int</parcialidad>
      </paramCancela>
    </cancelaOperacion>
  </soap:Body>
</soap:Envelope>
```

```
</paramCancela>  
</cancelaOperacion>  
</soap:Body>  
</soap:Envelope>
```


RESPUESTA DEL MÉTODO

Este método regresa un objeto del tipo **Cancelación** con el siguiente formato.

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <cancelaOperacionResponse xmlns="http://Recintos/ServiciosWebXml/">
      <cancelaOperacionResult>
        <error>int</error>
        <descripError>string</descripError>
      </cancelaOperacionResult>
    </cancelaOperacionResponse>
  </soap:Body>
</soap:Envelope>
```

Donde:

error. Es el número de error que sucedió durante la cancelación del ingreso de la mercancía al almacén.

descripError. Es una cadena de caracteres que tiene los errores encontrados en el proceso. Cada error esta encerrado en los caracteres “[]”.

EJEMPLO CON C#

```
ParametroCancela param = new ParametroCancela();
param.tipoOperacion = 9; //Cancela una entrada por órdenes de una autoridad
param.operacion = "1665"; //Número de entrada a cancelar
param.parcialidad = 0;

//Genera el proxy del servicio
ProxyRecintos.Recintos recinto = new Recintos();

//Define los datos del encabezado
EncabezadoSOAP enc = new EncabezadoSOAP();
enc.idUsuario = "RECINTO69";
enc.firma = new byte[60];
```

```
Cancelacion res;  
recinto.EncabezadoSOAPValue = enc;  
  
res = recinto.cancelaOperacion(paramCancela);  
  
//Obtiene los errores generales.  
lblError.Text = "Error: " + res.error.ToString();  
lblDesc.Text = res.descripcion;
```

Registra la salida de mercancías por órdenes de una autoridad

Cada vez que el recinto necesite registrar la salida de mercancía por órdenes de una autoridad debe usar este método.

MÉTODO

SalidaPorAutoridad registraSalidaXAutoridad(ParametroSalida paramSalida)

Parámetros:

ParametroSalida:

- **numEntrada**. Indica el número de entrada del que se desea hacer la salida.
- **autoridad**. Clave de la autoridad que está solicitando la salida de acuerdo con el **Catálogo de autoridades**.
- **numDocumento**. Es el número de documento que justifica la salida.
- **observacion**. Notas que puede añadir el recinto en la salida.
- **peso**. Peso de la mercancía que va a salir del recinto.
- **piezas**. Piezas de la mercancía que salen del recinto.
- **preliberacion**. Para marcar la solicitud de autorización como preliberada. En este método este parámetro no es significativo y debe ser *false*.

MENSAJE SOAP DE SALIDA POR ÓRDENES DE UNA AUTORIDAD

Este es el formato del mensaje SOAP para registrar la salida de mercancías del almacén del recinto fiscalizado con uno o más pedimentos.

```
POST /wsrefis/recintosfis.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Recintos/ServiciosWebXml/registraSalidaXAutoridad"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <EncabezadoSOAP xmlns="http://Recintos/ServiciosWebXml/">
      <idUsuario>string</idUsuario>
      <firma>base64Binary</firma>
      <ipCliente>string</ipCliente>
    </EncabezadoSOAP>
  </soap:Header>
```

```
<soap:Body>
  <registraSalidaXAutoridad xmlns="http://Recintos/ServiciosWebXml/">
    <paramSalida>
      <numEntrada>int</numEntrada>
      <pedimentos>
        <PedimentoSalida>
          <agente>string</agente>

          <numero>string</numero>
          <remesa>string</remesa>
          <factura>string</factura>
        </PedimentoSalida>
        <PedimentoSalida>
          <agente>string</agente>
          <numero>string</numero>
          <remesa>string</remesa>
          <factura>string</factura>
        </PedimentoSalida>
      </pedimentos>
      <autoridad>string</autoridad>
      <numDocumento>string</numDocumento>
      <observacion>string</observacion>
      <peso>string</peso>
      <piezas>string</piezas>
      <preliberacion>boolean</preliberacion>
    </paramSalida>
  </registraSalidaXAutoridad>
</soap:Body>
</soap:Envelope>
```

RESPUESTA DEL MÉTODO

Este método regresa un objeto del tipo **salidaPorAutoridad**. Una vez que el método terminó de procesar la solicitud del usuario, regresa el resultado de la operación con el siguiente formato:

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
```

Content-Length: **length**

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <registraSalidaXAutoridadResponse xmlns="http://Recintos/ServiciosWebXml/">
      <registraSalidaXAutoridadResult />
    </registraSalidaXAutoridadResponse>
  </soap:Body>
</soap:Envelope>
```

Donde:

error. Es el número de error que sucedió durante el registro de la salida de la mercancía del almacén.

descripError. Es una cadena de caracteres que tiene los errores relacionados con el proceso de salida. Cada error esta encerrado en los caracteres “[]”.

numSalida. Es el número de salida que calculó el sistema para esta operación.

acuse. Es el acuse de recibo de la operación.

Nota: En caso de enviar mas de una vez la salida a una mercancía, la respuesta entregará un error además de el número de salida y su acuse.

EJEMPLO EN C#

```
//Crea el objeto que tiene los parámetros de la salida
ParametroSalida paramSalida = new ParametroSalida();
paramSalida.autoridad = "1";
paramSalida.numDocumento = "ACTA DE SALIDA 6";
paramSalida.numEntrada = 19;
paramSalida.observacion = "OBSERVACIONES SALIDA POR AUTORIDAD ENTRADA 16";
paramSalida.peso = "4345.12";
paramSalida.piezas = "221";

//En este caso la propiedad no es significativa y debe declararse
//como "false" o no definirla.
//paramSalida.preliberacion = false;

//Genera el proxy del servicio
ProxyRecintos.Recintos recinto = new Recintos();
```

```
/* Define los datos del encabezado */
EncabezadoSOAP enc = new EncabezadoSOAP();
enc.idUsuario = "RECINTO69";
enc.firma      = new byte[60];

SalidaPorAutoridad res;
recinto.EncabezadoSOAPValue = enc;

res = recinto.registraSalidaXAutoridad(paramSal);

lblResultado.Text = "Error: " + res.error.ToString();
lblResultado.Text+= " " + res.descripcion;
if(res.error == 0)
    lblEntrada.Text = "Salida: " + res.numSalida + " Acuse: " + res.acuse;
```

Cancela la salida por órdenes de una autoridad.

El recinto fiscalizado debe usar este método para cancelar una salida de mercancía por órdenes de una autoridad.

MÉTODO

Cancelacion cancelaOperacion(ParametroCancela paramCancela)

Parámetros:

ParametroCancela:

- **tipoOperacion.** Este dato indica el tipo de operación a cancelar, para cancelar salidas de este tipo este dato debe ser **10**.
- **operacion.** Indica el número de salida a cancelar.
- **parcialidad.** Este dato debe ser 0 (cero).

MENSAJE SOAP

Este es el formato del mensaje SOAP para cancelar la salida de mercancías del almacén del recinto fiscalizado.

```
POST /wsRefis/RecintosFis.asmx HTTP/1.1
Host: satcpnpxch02
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Recintos/ServiciosWebXml/cancelaOperacion"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <EncabezadoSOAP xmlns="http://Recintos/ServiciosWebXml/">
      <idUsuario>string</idUsuario>
      <firma>base64Binary</firma>
      <ipCliente>string</ipCliente>
    </EncabezadoSOAP>
  </soap:Header>
  <soap:Body>
    <cancelaOperacion xmlns="http://Recintos/ServiciosWebXml/">
      <paramCancela>
        <tipoOperacion>int</tipoOperacion>
        <operacion>string</operacion>
        <parcialidad>int</parcialidad>
      </paramCancela>
    </cancelaOperacion>
  </soap:Body>
</soap:Envelope>
```

```
</paramCancela>  
</cancelaOperacion>  
</soap:Body>  
</soap:Envelope>
```


RESPUESTA DEL MÉTODO

Este método regresa un objeto del tipo **Cancelación** con el siguiente formato.

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <cancelaOperacionResponse xmlns="http://Recintos/ServiciosWebXml/">
      <cancelaOperacionResult>
        <error>int</error>
        <descripError>string</descripError>
      </cancelaOperacionResult>
    </cancelaOperacionResponse>
  </soap:Body>
</soap:Envelope>
```

Donde:

error. Contiene el número del último error ocurrido en la operación.

descripError. Es una cadena de caracteres que tiene los errores encontrados en el proceso. Cada error esta encerrado en los caracteres "[]".

EJEMPLO EN C#

```
//Crea los parámetros para cancelar una salida
ParametroCancela param = new ParametroCancela();
param.tipoOperacion = 10; //Cancela una salida por órdenes de una autoridad
param.operacion = "21"; //Número de salida a cancelar
param.parcialidad = 0;

//Genera el proxy del servicio
ProxyRecintos.Recintos recinto = new Recintos();

//Define los datos del encabezado
EncabezadoSOAP enc = new EncabezadoSOAP();
enc.idUsuario = "RECINTO69";
```

```
enc.firma      = new byte[60];

Cancelacion res;
recinto.EncabezadoSOAPValue = enc;

res = recinto.cancelaOperacion(paramCancela);

//Obtiene los errores generales.
lblError.Text = "Error: " + res.error.ToString();
lblDesc.Text  = res.descripcion;
```

6a etapa Separación y Subdivisión de mercancía.

Los recintos fiscalizados también tienen necesidad de “dividir” la mercancía suelta en partes más pequeñas que la mercancía original. Existen 2 procesos mediante los cuales pueden realizar esta actividad:

- **Separación.** Sucede cuando una mercancía suelta se divide en 2 ó más partes pero que el resultado debe tener el mismo embalaje que la mercancía origen de la separación.
- **Subdivisión.** Sucede cuando una mercancía suelta se divide en 2 ó más partes que deben tener un embalaje distinto al de la mercancía origen de la subdivisión.

En ambos casos la mercancía origen continúa en el recinto pero con una existencia en peso y piezas menor.

Separación de mercancía

La mecánica de operación para realizar una separación de mercancía lleva 2 pasos:

- 1) El registro del aviso de separación.
- 2) La conclusión del proceso de separación.

Primer paso. Registra el aviso de separación

El recinto fiscalizado debe usar este método cuando necesite realizar una separación de mercancía suelta.

MÉTODO

Separacion registraAvisoSepara (ParametroAvisoDes paramDes)

Parámetros:

ParametroAvisoDes:

- **numAviso.** Es el número de aviso asignado por el recinto destino.
- **entrada.** Número de entrada de mercancía suelta a separar.
- **total.** Sólo tiene significado para las operaciones de desconsolidación, por lo tanto en esta operación este dato debe tener el valor nulo.
- **fechaInicio.** Es la fecha programada de inicio de la separación.

MENSAJE SOAP

```
POST /wsrefis/recintosfis.aspx HTTP/1.1
```

```
Host: satcpanxch02
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Recintos/ServiciosWebXml/registraAvisoSepara"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>

    <EncabezadoSOAP xmlns="http://Recintos/ServiciosWebXml/">
      <idUsuario>string</idUsuario>
      <firma>base64Binary</firma>
      <ipCliente>string</ipCliente>
    </EncabezadoSOAP>
  </soap:Header>
  <soap:Body>
    <registraAvisoSepara xmlns="http://Recintos/ServiciosWebXml/">
      <ParametroAvisoDes>
        <numAviso>string</numAviso>
        <entrada>string</entrada>
        <total>string</total>
        <fechaInicio>string</fechaInicio>
      </ParametroAvisoDes>
    </registraAvisoSepara>
  </soap:Body>
</soap:Envelope>
```

RESPUESTA DEL MÉTODO

Una vez que el método terminó de procesar la solicitud del usuario, regresa el resultado de la operación con el siguiente formato:

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
```

```
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <registraAvisoSeparaResponse xmlns="http://Recintos/ServiciosWebXml/">
      <registraAvisoSeparaResult />
    </registraAvisoSeparaResponse>
  </soap:Body>
</soap:Envelope>
```

Donde:

error. Es el último número de error que sucedió durante la operación.

descripError. Contiene la lista de errores generales detectados al procesar la operación. Cada error está delimitado con los caracteres "[]".

acuse. Es el acuse de recibo del aviso.

listaEntradas. En esta operación este campo es nulo.

listaOrigen. Contiene un elemento del tipo **ResultadoEntrada** donde se especifican los errores correspondientes a la entrada origen de la operación.

EJEMPLO CON C#

```
//En este objeto se va a examinar el resultado de la validación de la entrada origen
ResultadoEntrada resEnt;

//Crea los parámetros del aviso
ParametroAvisoDes paramDes = new ParametroAvisoDes();
paramDes.numAviso      = "SEPARACION-001";
paramDes.entrada       = "546";
paramDes.fechaInicio= "2005-06-02 00:00:00";

//Genera el proxy del servicio
Recintos recinto = new Recintos();

//Define los datos del encabezado
EncabezadoSOAP enc = new EncabezadoSOAP();
```

```
enc.idUsuario = "RECINTO69";
enc.firma      = new byte[60];
recinto.EncabezadoSOAPValue = enc;

//Define al objeto que va a recibir el resultado de la separación
Separacion res;
res = recinto.registraAvisoSepara(paramDes);

if(res.listaOrigen.Length > 0)
{
    //Obtiene lo detalles de la validación de la entrada origen
    resEnt = (ResultadoEntrada)res.listaOrigen[0];
    lblErrEntAviso.Text = "Entrada: " + resEnt.listaErrores;
}

//Obtiene lo detalles generales de la validación del regsitro del aviso
lblError1.Text = "Error: " + res.error.ToString();
lblDescrip1.Text = res.descripcionError;
```

Cancela un aviso de separación

El recinto fiscalizado debe usar este método para cancelar un aviso de separación registrado anteriormente.

MÉTODO

Cancelacion cancelaOperacion(ParametroCancela paramCancela)

Parámetros:

ParametroCancela:

- **tipoOperacion.** Este dato indica el tipo de operación a cancelar, para cancelar un aviso de este tipo este dato debe ser 11.
- **operacion.** Indica el número del aviso a cancelar.
- **parcialidad.** Este dato debe ser 0 (cero).

MENSAJE SOAP

Este es el formato del mensaje SOAP para cancelar un aviso de separación.

```
POST /wsRefis/RecintosFis.asmx HTTP/1.1
Host: satcpnpxch02
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Recintos/ServiciosWebXml/cancelaOperacion"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <EncabezadoSOAP xmlns="http://Recintos/ServiciosWebXml/">
      <idUsuario>string</idUsuario>
      <firma>base64Binary</firma>
      <ipCliente>string</ipCliente>
    </EncabezadoSOAP>
  </soap:Header>
  <soap:Body>
    <cancelaOperacion xmlns="http://Recintos/ServiciosWebXml/">
      <paramCancela>
        <tipoOperacion>int</tipoOperacion>
        <operacion>string</operacion>
        <parcialidad>int</parcialidad>
      </paramCancela>
    </cancelaOperacion>
  </soap:Body>
</soap:Envelope>
```

```
</paramCancela>  
</cancelaOperacion>  
</soap:Body>  
</soap:Envelope>
```


RESPUESTA DEL MÉTODO

Este método regresa un objeto del tipo **Cancelación** con el siguiente formato.

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <cancelaOperacionResponse xmlns="http://Recintos/ServiciosWebXml/">
      <cancelaOperacionResult>
        <error>int</error>
        <descripError>string</descripError>
      </cancelaOperacionResult>
    </cancelaOperacionResponse>
  </soap:Body>
</soap:Envelope>
```

Donde:

error. Es el número de error que sucedió durante la cancelación.

descripError. Es una cadena de caracteres que tiene los errores encontrados en el proceso. Cada error esta encerrado en los caracteres “[]”.

EJEMPLO EN C#

```
//Crea los parámetros para cancelar una salida
ParametroCancela param = new ParametroCancela();
param.tipoOperacion = 11; //Cancela aviso de separación.
param.operacion = "SEPARA-001";

//Genera el proxy del servicio
ProxyRecintos.Recintos recinto = new Recintos();

//Define los datos del encabezado
EncabezadoSOAP enc = new EncabezadoSOAP();
enc.idUsuario = "RECINTO69";
enc.firma = new byte[60];
```

```
Cancelacion res;  
recinto.EncabezadoSOAPValue = enc;  
  
res = recinto.cancelaOperacion(paramCancela);  
  
//Obtiene los errores generales.  
lblError.Text = "Error: " + res.error.ToString();  
lblDesc.Text = res.descripcion;
```

Segundo paso. Registra la conclusión de la separación

El recinto fiscalizado origen debe usar este método para registrar el término de la operación de separación de mercancía suelta origen.

MÉTODO

Separacion separaMercancia (ParametroDesconsolida paramDes)

Parámetros:

ParametroDesconsolida:

- **numAviso**. Es el número de aviso de separación que se registró anteriormente.
- **listaMercancia**. Contiene los detalles de la mercancía que va a separarse de la mercancía origen. Los detalles de la mercancía que va a separarse deben entregarse dentro de un objeto del tipo **ParametroSuelta** que tiene la siguiente estructura:
 - **esParcial**. En esta operación este dato no es significativo y por lo tanto es nulo.
 - **mercancia**. Es un número consecutivo que se usará como referencia al momento de entregar el resultado del método.
 - **peso**. Es el peso de la nueva mercancía suelta.
 - **piezas**. Es el número de piezas de la nueva mercancía suelta.
 - **detalleMerc**. Contiene más detalles particulares de la mercancía en un objeto del tipo **DetalleMercancia** que tiene la siguiente estructura:
 - **descripcion**. Es la descripción textual de la mercancía suelta.
 - **embalaje**. Es la clave del embalaje de la mercancía suelta.
 - **marcas**. Son las marcas y números de la mercancía suelta.
 - **unidMedida**. Es la clave de la unidad de medida del peso.
 - **tipoMercancia**. Es un arreglo de objetos del tipo **TipoMercancia** con la clave de los tipos de mercancía suelta que tienen la siguiente estructura:
 - **cTipoMerc**. En este método este campo es nulo.
 - **cuTipoMerc**. Contiene la clave de usuario del tipo de mercancía.
 - **nomTipoMerc**. En este método este campo es nulo.
 - **fAbandono**. En este método este campo es nulo.
- **finOperacion**. Identifica si esta es la última operación de separación del contenedor. Para indicar que sí es la última operación puede tener el valor "verdadero", "si", "true" ó "1". Cualquier otro valor indica que no es la última operación.

MENSAJE SOAP

Este es el formato del mensaje SOAP para registrar el término de una operación de separación de mercancía.



```
POST /wsrefis/recintosfis.asmx HTTP/1.1
Host: satcpanxch02
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Recintos/ServiciosWebXml/separaMercancia"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <EncabezadoSOAP xmlns="http://Recintos/ServiciosWebXml/">
      <idUsuario>string</idUsuario>
      <firma>base64Binary</firma>
      <ipCliente>string</ipCliente>
    </EncabezadoSOAP>
  </soap:Header>
  <soap:Body>
    <separaMercancia xmlns="http://Recintos/ServiciosWebXml/">
      <ParametroDesconsolida>
        <numAviso>string</numAviso>
        <listaMercancia>
          <ParametroSuelta>
            <entrada>int</entrada>
            <esParcial>boolean</esParcial>
            <mercancia>int</mercancia>
            <peso>double</peso>
            <piezas>int</piezas>
            <DetalleMercancia>
              <descripcion>string</descripcion>
              <embalaje>string</embalaje>
              <tipoMercancia>
                <TipoMercancia>
                  <cTipoMerc>string</cTipoMerc>
                  <cuTipoMerc>string</cuTipoMerc>
                  <nomTipoMerc>string</nomTipoMerc>
                  <fAbandono>string</fAbandono>
                </TipoMercancia>
              </DetalleMercancia>
            </ParametroSuelta>
          </listaMercancia>
        </ParametroDesconsolida>
      </separaMercancia>
    </soap:Body>
  </soap:Envelope>
```

```
<cTipoMerc>string</cTipoMerc>
<cuTipoMerc>string</cuTipoMerc>
<nomTipoMerc>string</nomTipoMerc>
<fAbandono>string</fAbandono>
  </TipoMercancia>
</tipoMercancia>
<marcas>string</marcas>
<unidMedida>string</unidMedida>
</DetalleMercancia>
</ParametroSuelta>
</listaMercancia>
<finOperacion>string</finOperacion>
</ParametroDesconsolida>
</separaMercancia>
</soap:Body>
</soap:Envelope>
```

RESPUESTA DEL MÉTODO

Una vez que el método terminó de procesar la solicitud del usuario, regresa el resultado de la operación con el siguiente formato:

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <separaMercanciaResponse xmlns="http://Recintos/ServiciosWebXml/">
      <separaMercanciaResult/>
    </separaMercanciaResponse>
  </soap:Body>
</soap:Envelope>
```

Donde:

error. Es el último número de error que sucedió durante la separación.

descripError. Contiene la lista de errores generales detectados al procesar la operación. Cada error está delimitado con los caracteres "["].

listaEntradas. En este arreglo regresa los números de entrada que el sistema le asignó a la mercancía suelta declarada en

listaMercancia. Este arreglo contiene objetos del tipo **ResultadoEntrada** que tiene la siguiente estructura:

- **consecutivo.** Es el número consecutivo declarado en los parámetros de entrada del método.
- **numEntrada.** Es el número de entrada asignado por el método.
- **acuse.** Es el acuse calculado por el método.
- **listaErrores.** Son los errores identificados al procesar los datos de la entrada. Cada error está delimitado con los caracteres "["].

listaOrigen. Contiene a un elemento del tipo **ResultadoEntrada** que contiene los datos de la validación de la entrada origen de la operación.

EJEMPLO CON C#

```
ParametroSuelta paramSuelta;
DetalleMercancia detalle;
```

```
TipoMercancia tMerc;
TipoMercancia[] listaTipoMerc;

//Crea los parámetros para registrar la separación.
ParametroDesconsolida paramDes = new ParametroDesconsolida();

ParametroSuelta[] listaMercancia = new ParametroSuelta[2];
paramDes.numAviso = "SEPARACION-001";
paramDes.finOperacion = "falso";

//Primer mercancía de la separación.
paramSuelta = new ParametroSuelta();
paramSuelta.mercancia = 0;
paramSuelta.peso = 867.12;
paramSuelta.piezas = 221;

detalle = new DetalleMercancia();
detalle.embalaje = "33";
detalle.unidMedida = "1";
detalle.descripcion = "DESCRIPCION DE LA MERC. SEPARADA";
detalle.marcas = "MARCAS DE LA MERC. SEPARADA";

listaTipoMerc = new TipoMercancia[1];
tMerc = new TipoMercancia();
tMerc.cuTipoMerc = "0";
listaTipoMerc[0] = tMerc;

detalle.tipoMercancia = listaTipoMerc;
paramSuelta.detalleMerc = detalle;
listaMercancia[0] = paramSuelta;

//Segunda mercancía de la separación.
paramSuelta = new ParametroSuelta();
paramSuelta.mercancia = 1;
paramSuelta.peso = 425.88;
paramSuelta.piezas = 332;

detalle = new DetalleMercancia();
detalle.embalaje = "33";
detalle.unidMedida = "1";
detalle.descripcion = "DESCRIPCION DE LA MERC. SEPARADA";
detalle.marcas = "MARCAS DE LA MERC. SEPARADA";

listaTipoMerc = new TipoMercancia[1];
tMerc = new TipoMercancia();
tMerc.cuTipoMerc = "0";
listaTipoMerc[0] = tMerc;
```

```
detalle.tipoMercancia = listaTipoMerc;
paramSuelta.detalleMerc = detalle;
listaMercancia[1] = paramSuelta;

//Establece la lista de mercancías a separar.
paramDes.listaMercancia = listaMercancia;

//Genera el proxy del servicio
Recintos recinto = new Recintos();

//Define los datos del encabezado
EncabezadoSOAP enc = new EncabezadoSOAP();
enc.idUsuario = "USUARIO-01";
enc.firma = new byte[60];

recinto.EncabezadoSOAPValue = enc;

Separacion res;
res = recinto.separaMercancia(paramDes);

//Muestra los números de entrada que le corresponden a las desconsolidaciones
ResultadoEntrada resEntrada = (ResultadoEntrada)res.listaEntradas[0];
lblEntrada1.Text = "Error: " + resEntrada.listaErrores + " Ent:" + resEntrada.numEntrada + " : " + resEntrada.acuse;

ResultadoEntrada resEntrada = (ResultadoEntrada)res.listaEntradas[1];
lblEntrada2.Text = "Error: " + resEntrada.listaErrores + " Ent:" + resEntrada.numEntrada + " : " + resEntrada.acuse;

//Muestra errores generales
lblError2.Text = "Error: " + res.error.ToString();
lblDescrip2.Text = res.descripcionError;
```


Subdivisión de mercancía

La mecánica de operación para realizar una subdivisión de mercancía lleva 2 pasos:

- 1) El registro del aviso de subdivisión.
- 2) La conclusión del proceso de subdivisión.

Primer paso. Registra el aviso de subdivisión

El recinto fiscalizado debe usar este método cuando necesite realizar una subdivisión de mercancía suelta.

MÉTODO

Subdivision registraAvisoSubdiv(ParametroAvisoDes paramDes)

Parámetros:

ParametroAvisoDes:

- **numAviso.** Es el número de aviso asignado por el recinto destino.
- **entrada.** Número de entrada de mercancía suelta a subdividir.
- **total.** Sólo tiene significado para las operaciones de desconsolidación, por lo tanto en esta operación este dato debe tener el valor nulo.
- **fechaInicio.** Es la fecha programada de inicio de la subdivisión.

MENSAJE SOAP

Este es el formato del mensaje SOAP para registrar la mercancía origen de la que se van a hacer subdivisiones.

```
POST /wsrefis/recintosfis.asmx HTTP/1.1
Host: satcpanxch02
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Recintos/ServiciosWebXml/registraAvisoSubdiv"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <EncabezadoSOAP xmlns="http://Recintos/ServiciosWebXml/">
```

```
<idUsuario>string</idUsuario>
<firma>base64Binary</firma>
<ipCliente>string</ipCliente>
</EncabezadoSOAP>
</soap:Header>
<soap:Body>
  <registraAvisoSubdiv xmlns="http://Recintos/ServiciosWebXml/">
    <ParametroAvisoDes>
      <numAviso>string</numAviso>
      <entrada>string</entrada>
      <total>string</total>

      <fechaInicio>string</fechaInicio>
    </ParametroAvisoDes>
  </registraAvisoSubdiv>
</soap:Body>
</soap:Envelope>
```

RESPUESTA DEL MÉTODO

Una vez que el método terminó de procesar la solicitud del usuario, regresa el resultado de la operación con el siguiente formato:

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <registraAvisoSubdivResponse xmlns="http://Recintos/ServiciosWebXml/">
      <registraAvisoSubdivResult />
    </registraAvisoSubdivResponse>
  </soap:Body>
</soap:Envelope>
```

Donde:

error. Es el último número de error que sucedió durante la operación.

descripError. Contiene la lista de errores generales detectados al procesar la operación. Cada error está delimitado con los caracteres "[]".

acuse. Es el acuse de recibo del aviso.

listaEntradas. En esta operación este campo es nulo.

listaOrigen. Contiene un elemento del tipo **ResultadoEntrada** donde se especifican los errores correspondientes a la entrada origen de la operación.

EJEMPLO CON C#

```
ResultadoEntrada resEnt;  
  
//Crea los parámetros del aviso  
ParametroAvisoDes paramDes = new ParametroAvisoDes();  
paramDes.numAviso      = TBavisol.Text;  
paramDes.entrada       = TBEntrada1.Text;  
paramDes.fechaInicio= TBFecha.Text;  
  
//Genera el proxy del servicio  
Recintos recinto = new Recintos();  
  
//Define los datos del encabezado
```

```
EncabezadoSOAP enc = new EncabezadoSOAP();

enc.idUsuario = TBUsuario1.Text;
enc.firma      = new byte[60];
recinto.EncabezadoSOAPValue = enc;

Subdivision res;
res = recinto.registraAvisoSubdiv(paramDes);

if(res.listaOrigen.Length > 0)
{
    resEnt = (ResultadoEntrada)res.listaOrigen[0];
    lblErrEntAviso.Text = "Entrada: " + resEnt.listaErrores;
}

lblError1.Text = "Error: " + res.error.ToString();
lblDescrip1.Text = res.descripcionError;
```

Cancela un aviso de subdivisión

El recinto fiscalizado debe usar este método para cancelar un aviso de subdivisión registrado anteriormente.

MÉTODO

Cancelacion cancelaOperacion(ParametroCancela paramCancela)

Parámetros:

ParametroCancela:

- **tipoOperacion.** Este dato indica el tipo de operación a cancelar, para cancelar un aviso de este tipo este dato debe ser **12**.
- **operacion.** Indica el número del aviso a cancelar.
- **parcialidad.** Este dato debe ser 0 (cero).

MENSAJE SOAP

Este es el formato del mensaje SOAP para cancelar un aviso de subdivisión.

```
POST /wsRefis/RecintosFis.asmx HTTP/1.1
Host: satcpnpxch02
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Recintos/ServiciosWebXml/cancelaOperacion"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <EncabezadoSOAP xmlns="http://Recintos/ServiciosWebXml/">
      <idUsuario>string</idUsuario>
      <firma>base64Binary</firma>
      <ipCliente>string</ipCliente>
    </EncabezadoSOAP>
  </soap:Header>
  <soap:Body>
    <cancelaOperacion xmlns="http://Recintos/ServiciosWebXml/">
      <paramCancela>
        <tipoOperacion>int</tipoOperacion>
        <operacion>string</operacion>
        <parcialidad>int</parcialidad>
      </paramCancela>
    </cancelaOperacion>
  </soap:Body>
</soap:Envelope>
```

```
</paramCancela>  
</cancelaOperacion>  
</soap:Body>  
</soap:Envelope>
```

RESPUESTA DEL MÉTODO

Este método regresa un objeto del tipo **Cancelación** con el siguiente formato.

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <cancelaOperacionResponse xmlns="http://Recintos/ServiciosWebXml/">
      <cancelaOperacionResult>
        <error>int</error>
        <descripError>string</descripError>
      </cancelaOperacionResult>
    </cancelaOperacionResponse>
  </soap:Body>
</soap:Envelope>
```

Donde:

error. Es el número de error que sucedió durante la cancelación..

descripError. Es una cadena de caracteres que tiene los errores encontrados en el proceso. Cada error esta encerrado en los caracteres “[]”.

listaOrigen. Es un arreglo que contiene objetos del tipo **ResultadoEntrada** que contiene el resultado de la validación de la mercancía origen de la subdivisión.

listaEntradas. Es un arreglo que contiene objetos del tipo **ResultadoEntrada** con el resultado de la validación de las mercancías resultado de la subdivisión.

EJEMPLO EN C#

```
//Crea los parámetros para cancelar una salida
ParametroCancela param = new ParametroCancela();
param.tipoOperacion = 12; //Cancela aviso de subdivisión.
param.operacion = "SUBDIVISION-001";

//Genera el proxy del servicio
ProxyRecintos.Recintos recinto = new Recintos();

//Define los datos del encabezado
```

```
EncabezadoSOAP enc = new EncabezadoSOAP();
enc.idUsuario = "RECINTO69";
enc.firma      = new byte[60];

Cancelacion res;
recinto.EncabezadoSOAPValue = enc;

res = recinto.cancelaOperacion(paramCancela);

//Obtiene los errores generales.
lblError.Text = "Error: " + res.error.ToString();
lblDesc.Text  = res.descripcion;
```


Segundo paso. Registra la conclusión de la subdivisión

El recinto fiscalizado origen debe usar este método para registrar el término de la operación de subdivisión de mercancía suelta origen.

MÉTODO

Subdivision subdivideMercancia (ParametroDesconsolida paramDes)

Parámetros:

ParametroDesconsolida:

- **numAviso.** Es el número de aviso asignado por el recinto destino.
- **listaMercancia.** Lista de mercancía suelta en que se subdividió la mercancía origen. Esta lista está hecha con objetos del tipo **ParametroSuelta** que tiene la siguiente estructura:
 - **entrada.** En esta operación este dato no es significativo.
 - **esParcial.** En esta operación este dato no es significativo.
 - **mercancia.** Es un número consecutivo que se usará como referencia en el resultado del método.
 - **peso.** Es el peso de la nueva mercancía suelta.
 - **piezas.** Es el número de piezas de la nueva mercancía suelta.
 - **detalleMerc.** Es un objeto del tipo **DetalleMercancia** y contiene más detalles particulares de la mercancía:
 - **descripcion.** Es la descripción textual de la mercancía suelta.
 - **embalaje.** Es la clave del embalaje de la mercancía suelta.
 - **marcas.** Son las marcas y números de la mercancía suelta.
 - **unidMedida.** Es la clave de la unidad de medida del peso.
 - **tipoMercancia.** Es un arreglo de objetos del tipo **TipoMercancia** que contiene la clave de los tipos de mercancía suelta.
 - **cTipoMerc.** Este dato no es significativo.
 - **cuTipoMerc.** Contiene la clave del tipo de mercancía suelta de acuerdo con el **Catálogo de tipos de mercancía**.
 - **nomTipoMerc.** Este dato no es significativo.
 - **fAbandono.** Este dato no es significativo.
- **finOperacion.** Identifica si se terminó por completo la subdivisión. Una vez que se ha indicado este parámetro en verdadero, ya no podrán realizarse más subdivisiones en la mercancía origen de este aviso.

MENSAJE SOAP

Este es el formato del mensaje SOAP para registrar el término de una operación de subdivisión de mercancía.

```
POST /wsrefis/recintosfis.asmx HTTP/1.1
Host: satcpanxch02
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Recintos/ServiciosWebXml/subdivideMercancia"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <EncabezadoSOAP xmlns="http://Recintos/ServiciosWebXml/">
      <idUsuario>string</idUsuario>
      <firma>base64Binary</firma>
      <ipCliente>string</ipCliente>
    </EncabezadoSOAP>
  </soap:Header>
  <soap:Body>
    <subdivideMercancia xmlns="http://Recintos/ServiciosWebXml/">
      <paramDes>
        <numAviso>string</numAviso>
        <listaMercancia>
          <ParametroSuelta>
            <entrada>int</entrada>
            <esParcial>boolean</esParcial>
            <mercancia>int</mercancia>
            <peso>double</peso>
            <piezas>int</piezas>
            <DetalleMercancia>
              <descripcion>string</descripcion>
              <embalaje>string</embalaje>
              <tipoMercancia>
                <TipoMercancia>
                  <cTipoMerc>string</cTipoMerc>
                  <cuTipoMerc>string</cuTipoMerc>
                </TipoMercancia>
              </DetalleMercancia>
            </piezas>
          </ParametroSuelta>
        </listaMercancia>
      </paramDes>
    </subdivideMercancia>
  </soap:Body>
</soap:Envelope>
```

```
<nomTipoMerc>string</nomTipoMerc>
  <fAbandono>string</fAbandono>
</TipoMercancia>
<TipoMercancia>
  <cTipoMerc>string</cTipoMerc>
  <cuTipoMerc>string</cuTipoMerc>
  <nomTipoMerc>string</nomTipoMerc>
  <fAbandono>string</fAbandono>
</TipoMercancia>
</tipoMercancia>
<marcas>string</marcas>
<unidMedida>string</unidMedida>
</DetalleMercancia>
</ParametroSuelta>
<ParametroSuelta>
  <entrada>int</entrada>
  <esParcial>boolean</esParcial>
  <mercancia>int</mercancia>
  <peso>double</peso>
  <piezas>int</piezas>
  <detalleMerc xsi:nil="true" />
</ParametroSuelta>
</listaMercancia>
<finOperacion>string</finOperacion>
</paramDes>
</subdivideMercancia>
</soap:Body>
</soap:Envelope>
```

RESPUESTA DEL MÉTODO

Una vez que el método terminó de procesar la solicitud del usuario, regresa el resultado de la operación con el siguiente formato:

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
```

```
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <subdivideMercanciaResponse xmlns="http://Recintos/ServiciosWebXml/">
      <subdivideMercanciaResult />
    </subdivideMercanciaResponse>
  </soap:Body>
</soap:Envelope>
```

Donde:

error. Es el último número de error que sucedió durante la subdivisión.

descripError. Contiene la lista de errores generales detectados al procesar la operación. Cada error está delimitado con los caracteres "[]".

listaEntradas. En este arreglo regresa los números de entrada que el sistema le asignó a la mercancía suelta declarada en **listaMercancia**.

listaOrigen. Contiene a un elemento del tipo **ResultadoEntrada** que contiene los datos de la validación de la entrada origen de la subdivisión.

EJEMPLO CON C#

```
ParametroSuelta paramSuelta;  
DetalleMercancia detalle;  
TipoMercancia tMerc;  
TipoMercancia[] listaTipoMerc;  
  
//Crea los parámetros para registrar la subdivisión.  
ParametroDesconsolida paramDes = new ParametroDesconsolida();  
  
ParametroSuelta[] listaMercancia = new ParametroSuelta[2];  
paramDes.numAviso = "SUBDIVISION-001";  
paramDes.finOperacion = "falso";  
  
//Primer mercancía de la subdivisión.  
paramSuelta = new ParametroSuelta();  
paramSuelta.mercancia = 0;  
paramSuelta.peso = 867.12;  
paramSuelta.piezas = 221;  
  
detalle = new DetalleMercancia();  
detalle.embalaje = "33";  
detalle.unidMedida = "1";  
detalle.descripcion = "DESCRIPCION DE LA MERC. SUBDIVIDIDA ";  
detalle.marcas = "MARCAS DE LA MERC. SUBDIVIDIDA ";  
  
listaTipoMerc = new TipoMercancia[1];  
tMerc = new TipoMercancia();  
tMerc.cuTipoMerc = "0";  
listaTipoMerc[0] = tMerc;  
  
detalle.tipoMercancia = listaTipoMerc;  
paramSuelta.detalleMerc = detalle;  
listaMercancia[0] = paramSuelta;  
  
//Segunda mercancía de la subdivisión.  
paramSuelta = new ParametroSuelta();  
paramSuelta.mercancia = 1;  
paramSuelta.peso = 425.88;  
paramSuelta.piezas = 332;  
  
detalle = new DetalleMercancia();  
detalle.embalaje = "33";  
detalle.unidMedida = "1";  
detalle.descripcion = "DESCRIPCION DE LA MERC. SUBDIVIDIDA";  
detalle.marcas = "MARCAS DE LA MERC. SUBDIVIDIDA";  
  
listaTipoMerc = new TipoMercancia[1];
```

```
tMerc = new TipoMercancia();
tMerc.cuTipoMerc = "0";
listaTipoMerc[0] = tMerc;

detalle.tipoMercancia = listaTipoMerc;
paramSuelta.detalleMerc = detalle;
listaMercancia[1] = paramSuelta;

//Establece la lista de mercancías a subdividir.
paramDes.listaMercancia = listaMercancia;

//Genera el proxy del servicio
Recintos recinto = new Recintos();

//Define los datos del encabezado
EncabezadoSOAP enc = new EncabezadoSOAP();
enc.idUsuario = "USUARIO-01";
enc.firma = new byte[60];
recinto.EncabezadoSOAPValue = enc;

Subdivision res;
res = recinto.subdivideMercancia(paramDes);

//Muestra los números de entrada que le corresponden a las desconsolidaciones
ResultadoEntrada resEntrada = (ResultadoEntrada)res.listaEntradas[0];
lblEntrada1.Text = "Error: " + resEntrada.listaErrores + " Ent:" + resEntrada.numEntrada + " : " + resEntrada.acuse;

ResultadoEntrada resEntrada = (ResultadoEntrada)res.listaEntradas[1];
lblEntrada2.Text = "Error: " + resEntrada.listaErrores + " Ent:" + resEntrada.numEntrada + " : " + resEntrada.acuse;

//Muestra errores generales
lblError2.Text = "Error: " + res.error.ToString();
lblDescrip2.Text = res.descripcionError;
```

Eventos e incidencias

Eventos de una mercancía en el recinto

Este método puede usarlo el recinto fiscalizado para obtener información de las operaciones que le ha registrado a una mercancía.

MÉTODO

resEvento obtieneEventos(**string** aduana, **int** recinto, **int** entrada)

Parámetros:

- **aduana**. En este caso este dato debe estar vacío.
- **recinto**. Clave del recinto fiscalizado de acuerdo con el **Catálogo de recintos**.
- **entrada**. Es el número de entrada en el recinto de la mercancía que se van a obtener los eventos.

MENSAJE SOAP PARA OBTENER LOS EVENTOS DE UNA MERCANCÍA EN EL RECINTO

```
POST /wsrefis/recintosfis.asmx HTTP/1.1
Host: satcpaxch02
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Recintos/ServiciosWebXml/obtieneEventosEntrada"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <EncabezadoSOAP xmlns="http://Recintos/ServiciosWebXml/">
      <idUsuario>string</idUsuario>
      <firma>base64Binary</firma>
      <ipCliente>string</ipCliente>
    </EncabezadoSOAP>
  </soap:Header>
  <soap:Body>
    <obtieneEventosEntrada xmlns="http://Recintos/ServiciosWebXml/">
      <aduana>string</aduana>
      <recinto>int</recinto>
      <entrada>int</entrada>
    </obtieneEventosEntrada>
  </soap:Body>
</soap:Envelope>
```

```
</soap:Body>  
</soap:Envelope>
```


RESPUESTA DEL MÉTODO

Este método regresa un objeto del tipo **resEvento**. Una vez que el método terminó de procesar la solicitud del usuario, regresa el resultado de la operación con el siguiente formato:

```
HTTP/1.1 200 OK

Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <obtieneEventosEntradaResponse xmlns="http://Recintos/ServiciosWebXml/">
      <obtieneEventosEntradaResult>
        <listaEvento>
          <anyType />
          <anyType />
        </listaEvento>
      </obtieneEventosEntradaResult>
    </obtieneEventosEntradaResponse>
  </soap:Body>
</soap:Envelope>
```

Donde:

error. Es el número de error que sucedió durante la operación.

descripcionError. Es una cadena de caracteres que tiene los errores relacionados con el proceso de salida. Cada error esta encerrado en los caracteres “[]”.

listaEvento. Es un arreglo que contiene a objetos del tipo **EventoEnRecinto** con la siguiente estructura:

- **cEntrada**. Es el número de entrada de la mercancía.
- **cEvento**. Es la clave del evento.
- **nEvento**. Es el número de evento registrado a la mercancía.
- **fecha**. Es la fecha en que se registró el evento.
- **descripcion**. Es la descripción del evento.
- **nCveDRecinto**. Clave del recinto destino en una salida por transferencia.

- **salida.** Número de salida en una operación de salida.
- **AcuseEntrada.** Acuse de la entrada en una operación de entrada.
- **AcuseSalida.** Acuse de la salida en una operación de salida.

EJEMPLO EN C#

```
//Genera el proxy del servicio
ProxyRecintos.Recintos recinto = new Recintos();

//Define los datos del encabezado
EncabezadoSOAP enc = new EncabezadoSOAP();
enc.idUsuario = "RECINTO69";
enc.firma = new byte[60];
recinto.EncabezadoSOAPValue = enc;

resEvento res;
res = recinto.obtieneEventos("", 69, 7);
```

Incidencias de una mercancía en el recinto

Este método puede usarlo el recinto fiscalizado para obtener información de las incidencias que le ha registrado a una mercancía.

MÉTODO

resIncidencia obtieneIncidencia(**string** aduana, **int** recinto, **int** entrada)

Parámetros:

- **aduana**. En este caso este dato debe estar vacío.
- **recinto**. Clave del recinto fiscalizado de acuerdo con el **Catálogo de recintos**.
- **entrada**. Es el número de entrada de la mercancía que se van a obtener las incidencias en el recinto.

MENSAJE SOAP PARA OBTENER LAS INCIDENCIAS DE UNA MERCANCÍA EN EL RECINTO

```
POST /wsrefis/recintosfis.asmx HTTP/1.1
Host: satcpanxch02
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Recintos/ServiciosWebXml/obtieneIncidenciasEntrada"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <EncabezadoSOAP xmlns="http://Recintos/ServiciosWebXml/">
      <idUserario>string</idUserario>
      <firma>base64Binary</firma>
      <ipCliente>string</ipCliente>
    </EncabezadoSOAP>
  </soap:Header>
  <soap:Body>
    <obtieneIncidenciasEntrada xmlns="http://Recintos/ServiciosWebXml/">
      <aduana>string</aduana>
      <recinto>int</recinto>
      <entrada>int</entrada>
    </obtieneIncidenciasEntrada>
  </soap:Body>
</soap:Envelope>
```

RESPUESTA DEL MÉTODO

Este método regresa un objeto del tipo **resIncidencia**. Una vez que el método terminó de procesar la solicitud del usuario, regresa el resultado de la operación con el siguiente formato:

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <obtieneIncidenciasEntradaResponse xmlns="http://Recintos/ServiciosWebXml/">
      <obtieneIncidenciasEntradaResult>
        <listaIncidencia>
          <anyType />
          <anyType />
        </listaIncidencia>
      </obtieneIncidenciasEntradaResult>
    </obtieneIncidenciasEntradaResponse>
  </soap:Body>
</soap:Envelope>
```

Donde:

error. Es el número de error que sucedió durante la operación.

descripcionError. Es una cadena de caracteres que tiene los errores relacionados con el proceso de salida. Cada error esta encerrado en los caracteres “[]”.

listaIncidencia. Es un arreglo que contiene a objetos del tipo **IncidenciaenRecinto** con la siguiente estructura:

- **centrada**. Es el número de entrada de la mercancía.
- **cIncidencia**. Es la clave de la incidencia.
- **fecha**. Es la fecha en que se registró la incidencia.
- **descripcion**. Es la descripción de la incidencia.
- **descripcionUsr**. Es la descripción complementaria asignada a la incidencia.

EJEMPLO EN C#

```
//Genera el proxy del servicio
ProxyRecintos.Recintos recinto = new Recintos();

//Define los datos del encabezado
EncabezadoSOAP enc = new EncabezadoSOAP();
enc.idUsuario = "RECINTO69";
enc.firma = new byte[60];
recinto.EncabezadoSOAPValue = enc;

resIncidencia res;
res = recinto.obtieneIncidencia("", 69, 7);
```

Consulta de Información de un Manifiesto

A través de esta consulta el Web Service hará una petición indicando el manifiesto el cual se solicita. Este servicio responderá un estado de ejecución 0=OK, 1 = Error.

MÉTODO

resInfomanifiesto InformacionManifiestos(**string** manifiesto)

Parámetros:

- **manifiesto**. Es el número único de manifiesto.

Este método lo usa el recinto fiscalizado para obtener los datos generales de un manifiesto específico registrado por el **Validador de manifiestos**.

MENSAJE SOAP

Este es el formato del mensaje SOAP para obtener un manifiesto específico registrado por el **Validador de manifiestos**.

```
POST /wsRefis/RecintosFis.asmx HTTP/1.1
Host: satcpnpxch02
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Recintos/ServiciosWebXml/InformacionManifiestos"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <EncabezadoSOAP xmlns="http://Recintos/ServiciosWebXml/">
      <idUsuario>string</idUsuario>
      <firma>base64Binary</firma>
      <ipCliente>string</ipCliente>
    </EncabezadoSOAP>
  </soap:Header>
  <soap:Body>
    <InformacionManifiestos xmlns="http://Recintos/ServiciosWebXml/">
```

```
<manifiesto>string</manifiesto>
</InformacionManifiestos>
</soap:Body>
</soap:Envelope>
```

RESPUESTA DEL MÉTODO

Este método regresa un objeto del tipo **resInfomanifiesto**. Una vez que el método terminó de procesar la solicitud del usuario, regresa el resultado de la operación con el siguiente formato:

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <InformacionManifiestosResponse xmlns="http://Recintos/ServiciosWebXml/">
      <InformacionManifiestosResult>
        <error>int</error>
        <descripError>string</descripError>
        <resultado>int</resultado>
      </InformacionManifiestosResult>
    </InformacionManifiestosResponse>
  </soap:Body>
</soap:Envelope>
```

Donde:

error. Es el número de error que sucedió durante la operación.

descripError. Es una cadena de caracteres que tiene los errores relacionados con el proceso. Cada error esta encerrado en los caracteres "[]".

resultado. Se refleja el resultado de una consulta, en la cual si el resultado de la consulta es 0 (cero), se interpretará como "La consulta se procesó con éxito", si el resultado es diferente a 0 (cero), se interpretará como "Existe un error al ejecutar la consulta."

EJEMPLO CON C#

```
//Crea los parámetros del aviso
resInfomanifiesto res = new resInfomanifiesto();

//Genera el proxy del servicio
Recintos recinto = new Recintos();

//Define los datos del encabezado
EncabezadoSOAP enc = new EncabezadoSOAP();
enc.idUsuario = Usuario2.Text;
enc.firma = new byte[60];
recinto.EncabezadoSOAPValue = enc;

res = recinto.InformacionManifiestos(numManif);
TBNError.Text = res.error.ToString();
TBDescError.Text = res.descripcionError;
```

Consulta de Información de un Pedimento

A través de esta consulta el Recinto Fiscalizado podrá obtener la información de un pedimento especificado. Este servicio responderá con el detalle del pedimento.

MÉTODO

```
public DataSet ConsultaPedimento(int agente, int numerodeDocumento, int aduana)
```

Parámetros:

- **agente.** Es el número de patente del agente aduanal que valido el pedimento
- **numerodeDocumento.** Es el número de documento asignado a ese pedimento
- **aduana.** Es la aduana en la que se despachara el pedimento

MENSAJE SOAP

Este es el formato del mensaje SOAP para obtener un pedimento específico.

```
POST /crpws/crp.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/ConsultaPedimento"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ConsultaPedimento xmlns="http://tempuri.org/">
      <agente>int</agente>
      <numerodeDocumento>int</numerodeDocumento>
      <aduana>int</aduana>
    </ConsultaPedimento>
  </soap:Body>
</soap:Envelope>
```

RESPUESTA DEL MÉTODO

Este método regresa un objeto del tipo **dataset**. Una vez que el método terminó de procesar la solicitud del usuario, regresa el resultado de la operación con el siguiente formato:

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ConsultaPedimentoResponse xmlns="http://tempuri.org/">
      <ConsultaPedimentoResult>
        <xsd:schema>schema</xsd:schema>xml</ConsultaPedimentoResult>
      </ConsultaPedimentoResponse>
    </soap:Body>
  </soap:Envelope>
```

schema. Es el esquema que define el contenido del dataset, en caso de no encontrar el pedimento en base de datos este estará vacío.

xml. Es el contenido del dataset, este esta definido por el campo echema, no necesariamente todos los campos del eschema se verán reflejados en el xml, si no encuentra información para el tipo no se generara el campo correspondiente, en caso de no encontrar el pedimento en base de datos este estará vacío.

A continuación encontraras una respuesta sin errores de la petición que el usuario realizo.

EJEMPLO:

```
<?xml version="1.0" encoding="utf-8" ?>
- <DataSet xmlns="http://tempuri.org/">
-   <xs:schema id="Pedimento_Normal_M3" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
-     xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
- <xs:element name="Pedimento_Normal_M3" msdata:IsDataSet="true" msdata:Locale="es-MX">
- <xs:complexType>
- <xs:choice maxOccurs="unbounded">
+ <xs:element name="DatosGenerales">
```

```
- <xs:element name="Fechas">
- <xs:complexType>
- <xs:sequence>
  <xs:element name="fechaentrada" type="xs:dateTime" minOccurs="0" />
  <xs:element name="fechapago" type="xs:dateTime" minOccurs="0" />
  <xs:element name="fechavalidacion" type="xs:dateTime" minOccurs="0" />
  <xs:element name="fechaseleccion" type="xs:dateTime" minOccurs="0" />
  <xs:element name="fechadesaduanamiento" type="xs:dateTime" minOccurs="0" />
  </xs:sequence>
</xs:complexType>

</xs:element>
+ <xs:element name="Pago">
- <xs:element name="impuestos">
- <xs:complexType>
- <xs:sequence>
  <xs:element name="concepto" type="xs:string" minOccurs="0" />
  <xs:element name="importe" type="xs:decimal" minOccurs="0" />
  </xs:sequence>
</xs:complexType>
</xs:element>
+ <xs:element name="formasdepago">
+ <xs:element name="relacionesimpuestosformasdepago">
+ <xs:element name="contenedores">
+ <xs:element name="casosgral">
+ <xs:element name="desistidos">
+ <xs:element name="pedimentosusa">
+ <xs:element name="saldos1">
+ <xs:element name="saldos2">
</xs:choice>
</xs:complexType>
</xs:element>
```

```
</xs:schema>
- <diffgr:diffgram xmlns:msdata="urn:schemas-microsoft-com:xml-msdata" xmlns:diffgr="urn:schemas-microsoft-com:xml-diffgram-v1">
- <Pedimento_Normal_M3 xmlns="">
- <DatosGenerales diffgr:id="DatosGenerales1" msdata:rowOrder="0">
  <clavetipooperacion>2</clavetipooperacion>
  <tipooperacion>Exportacion</tipooperacion>
  <clavepedimento>J2</clavepedimento>
  <descripcionclavepedimento>RETORNO AL EXTRANJERO DE MERCANCIAS ELABORADAS, TRANSFORMADAS O REPARADAS POR PARTE DE EMPRESAS CON PITEX (EXPORTACION).</descripcionclavepedimento>
  <firma>KKTFOJRL</firma>
  <tipocambio>10.94360</tipocambio>
  <destinomercancia>INTERIOR DEL PAIS</destinomercancia>
  <pesobruto>122588.000</pesobruto>
  <rfcimportadorexportador>ATE890522F28</rfcimportadorexportador>
  <razonsocialimportadorexportador>ALUMINIO TEXCOCO, S.A. DE C.V.</razonsocialimportadorexportador>
  <curpagenteaduanal>SIDF401204HVZLZR04</curpagenteaduanal>
  </DatosGenerales>
- <Fechas diffgr:id="Fechas1" msdata:rowOrder="0">
  <fechapago>2009-05-12T00:00:00.0000000-05:00</fechapago>
  <fechavalidacion>2009-05-12T09:37:49.0000000-05:00</fechavalidacion>
  <fechaseleccion>2009-05-12T00:00:00.0000000-05:00</fechaseleccion>
  <fechadesaduanamiento>2009-05-12T00:00:00.0000000-05:00</fechadesaduanamiento>
  </Fechas>
- <Pago diffgr:id="Pago1" msdata:rowOrder="0">
  <valorcomercial>3977976</valorcomercial>
  <valoraduana>0</valoraduana>
  <seguros>0</seguros>
  <fletes>0</fletes>
  <fechapago>2009-05-12T09:30:55.0000000-05:00</fechapago>
  <banco>CAJAS DE ADUANAS</banco>
  </Pago>
+ <impuestos diffgr:id="impuestos1" msdata:rowOrder="0">
```

```
- <impuestos diffgr:id="impuestos2" msdata:rowOrder="1">
  <concepto>PREVALIDAAA</concepto>
  <importe>161</importe>
</impuestos>
+ <formasdepago diffgr:id="formasdepago1" msdata:rowOrder="0">
- <formasdepago diffgr:id="formasdepago2" msdata:rowOrder="1">
  <formapago>EFECTIVO</formapago>
  <importe>161</importe>
</formasdepago>
+ <relacionesimpuestosformasdepago diffgr:id="relacionesimpuestosformasdepago1" msdata:rowOrder="0">
+ <relacionesimpuestosformasdepago diffgr:id="relacionesimpuestosformasdepago2" msdata:rowOrder="1">
- <contenedores diffgr:id="contenedores1" msdata:rowOrder="0">
  <secuencia>1</secuencia>
  <identificador>AMFU3062268</identificador>
  <clave>1</clave>
  <descripcion>CONTENEDOR ESTANDAR 20</descripcion>
</contenedores>
+ <contenedores diffgr:id="contenedores2" msdata:rowOrder="1">
+ <contenedores diffgr:id="contenedores3" msdata:rowOrder="2">
+ <contenedores diffgr:id="contenedores4" msdata:rowOrder="3">
+ <contenedores diffgr:id="contenedores5" msdata:rowOrder="4">
+ <contenedores diffgr:id="contenedores6" msdata:rowOrder="5">
+ <casosgral diffgr:id="casosgral1" msdata:rowOrder="0">
- <casosgral diffgr:id="casosgral2" msdata:rowOrder="1">
  <consecutivo>2</consecutivo>
  <identificador>11</identificador>
  <complemento1>ST</complemento1>
  <descripcion>OPERACION SUJETAS AL ART. 303 DEL TLCAN.</descripcion>
</casosgral>
</Pedimento_Normal_M3>
</diffgr:diffgram>
</DataSet>
```

schema. En la parte superior se puede observar la definición del schema para un pedimento normal en m3.

xml. En la parte baja se puede observar el mensaje con los datos obtenidos del pedimento consultado.

Tanto como el schema y el xml pueden variar su contenido de una llamada a otra independientemente de que el tipo del pedimento sea el mismo, esto es debido a que solo se va a serializar la información encontrada.

A continuación se mostrara un ejemplo de C#.

EJEMPLO CON C#

```
ProxyRecintos.Recintos wsConsultas;  
DataSet res;  
ProxyRecintos.EncabezadoSOAP encabezado = new ProxyRecintos.EncabezadoSOAP();  
byte[] firma;  
  
/* Genera una firma ficticia */  
firma = new byte[64];  
for(int i = 0; i < 64; i++)  
    firma[i] = (byte)i;  
  
wsConsultas = new ProxyRecintos.Recintos();  
  
encabezado.idUsuario = Usuario;  
encabezado.firma = firma;  
wsConsultas.EncabezadoSOAPValue = encabezado;  
res = (DataSet) wsConsultas.ConsultaPedimento (agente, documento, aduana);
```

Consulta los datos de un contenedor

Este método puede usarlo el recinto fiscalizado para obtener la información relacionada con un contenedor: los manifiestos y conocimientos a los que ha estado asociado y los recintos y números de entrada que le han sido asignados al haber sido ingresado en los recintos de la aduana.

MÉTODO

ResContenedor consultaContenedor (ParamConsContenedor paramCons)

Parámetros:

ParamConsContenedor:

- **aduana**. Clave de la aduana donde se encuentra el recinto que está haciendo la consulta.
- **nomContenedor**. Número de contenedor.
- **buque**. Filtro opcional, si se le establece un valor entonces el web service va regresar los manifiestos que estén relacionados con el buque indicado.
- **fechaIni y fechaFin**. Filtro opcional, si se establece un rango de fechas el web service va a regresar los manifiestos que se hayan recibido en el rango de fechas indicado y que tengan asociado al contenedor especificado en los parámetros de entrada.

MENSAJE SOAP

```
POST /wsRefis/RecintosFis.asmx HTTP/1.1
Host: satcpnpxch02
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Recintos/ServiciosWebXml/consultaContenedor"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <EncabezadoSOAP xmlns="http://Recintos/ServiciosWebXml/">
      <idUsuario>string</idUsuario>
      <firma>base64Binary</firma>
      <ipCliente>string</ipCliente>
    </EncabezadoSOAP>
  </soap:Header>
  <soap:Body>
    <consultaContenedor xmlns="http://Recintos/ServiciosWebXml/">
      <paramCons>
```

```
<aduana>string</aduana>
<nomContenedor>string</nomContenedor>
<buque>string</buque>
<fechaIni>string</fechaIni>
<fechaFin>string</fechaFin>
</paramCons>
</consultaContenedor>
</soap:Body>
</soap:Envelope>
```

RESPUESTA DEL MÉTODO

Este método regresa un objeto del tipo **ResContenedor**. Una vez que el método terminó de procesar la solicitud del usuario, regresa el resultado de la operación con el siguiente formato:

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <consultaContenedorResponse xmlns="http://Recintos/ServiciosWebXml/">
      <consultaContenedorResult>
        <error>int</error>
        <descripError>string</descripError>
        <ds>
          <xsd:schema>schema</xsd:schema>xml</ds>
        </consultaContenedorResult>
      </consultaContenedorResponse>
    </soap:Body>
  </soap:Envelope>
```

Donde:

error. Es el número de error que sucedió durante la operación.

descripError. Es una cadena de caracteres que tiene los errores relacionados con el proceso. Cada error esta encerrado en los caracteres “[]”.

ds. Colección de datos con el resultado de la consulta.

EJEMPLO EN C#

```
ProxyRecintos.Recintos recinto = new Recintos();

//Define los datos del encabezado
EncabezadoSOAP enc = new EncabezadoSOAP();
enc.idUsuario = "RECINTO69";
enc.firma = new byte[60];
recinto.EncabezadoSOAPValue = enc;

ProxyRecintos.ParamConsContenedor paramCons = new ProxyRecintos.ParamConsContenedor();
ParamConsContenedor paramCons = new ParamConsContenedor();
paramCons.aduana = "430";
paramCons.nomContenedor = "HLCU4052192";
paramCons.buque = "";
paramCons.fechaIni = "";
paramCons.fechaFin = "";

ProxyRecintos.ResContenedor res;
res = recinto.consultaContenedor(paramCons);
```

Consulta de ingresos por transferencia

Este método puede usarlo el recinto fiscalizado para obtener a los contenedores que haya puesto en transferencia y que ya cuenten con ingreso por parte del recinto destino.

MÉTODO

ResContenedor consultaIngresoPorTransfer (ParamConsTransfer paramCons)

Parámetros:

ParamConsTransfer:

- **recintoDestino.** Clave de usuario del recinto destino de la operación de transferencia, en el que se desean verificar los ingresos. Parámetro opcional.
- **numContenedor.** Número de contenedor del que se desean verificar los ingresos. Parámetro obligatorio cuando no se especifique el rango de fechas (*fechaIni* y *fechaFin*); de lo contrario se vuelve opcional.
- **fechaIni y fechaFin.** Rango de fechas en que se realizó la salida por transferencia en el recinto origen. Parámetro obligatorio cuando no se especifique el número de contenedor (*numContenedor*); de lo contrario se vuelve opcional.

MENSAJE SOAP

```
POST /wsRefis/RecintosFis.asmx HTTP/1.1
Host: satcpnpxch02
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Recintos/ServiciosWebXml/consultaIngresoPorTransfer"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <EncabezadoSOAP xmlns="http://Recintos/ServiciosWebXml/">
      <idUsuario>string</idUsuario>
      <firma>base64Binary</firma>
      <ipCliente>string</ipCliente>
    </EncabezadoSOAP>
  </soap:Header>
  <soap:Body>
    <consultaIngresoPorTransfer xmlns="http://Recintos/ServiciosWebXml/">
      <paramCons>
        <recintoDestino>string</recintoDestino>
      </paramCons>
    </consultaIngresoPorTransfer>
  </soap:Body>
</soap:Envelope>
```

```
<numContenedor>string</numContenedor>  
<fechaIni>string</fechaIni>  
<fechaFin>string</fechaFin>  
</paramCons>  
</consultaIngresoPorTransfer>  
</soap:Body>  
</soap:Envelope>
```

RESPUESTA DEL MÉTODO

Este método regresa un objeto del tipo **ResContenedor**. Una vez que el método terminó de procesar la solicitud del usuario, regresa el resultado de la operación con el siguiente formato:

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <consultaIngresoPorTransferResponse xmlns="http://Recintos/ServiciosWebXml/">
      <consultaIngresoPorTransferResult>
        <error>int</error>
        <descripError>string</descripError>
        <registros>int</registros>
        <ds>
          <xsd:schema>schema</xsd:schema>xml</ds>
        </consultaIngresoPorTransferResult>
      </consultaIngresoPorTransferResponse>
    </soap:Body>
  </soap:Envelope>
```

Donde:

error. Es el número de error que sucedió durante la operación.

descripError. Es una cadena de caracteres que tiene los errores relacionados con el proceso. Cada error esta encerrado en los caracteres "[]".

registros. Es el número de registros obtenidos en la consulta.

ds. Colección de datos con el resultado de la consulta.

Estructura de la colección de datos **ds**:

```
<xs:schema id="ResConsulta" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  <xs:element name="ResConsulta" msdata:IsDataSet="true" msdata:Locale="es-MX">
```

```
<xs:complexType>
  <xs:choice maxOccurs="unbounded">
    <xs:element name="Contenedor">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="cuAduana" type="xs:int" minOccurs="0" />
          <xs:element name="cuRecintoOrigen" type="xs:int" minOccurs="0" />
          <xs:element name="cuRecintoDestino" type="xs:int" minOccurs="0" />
          <xs:element name="entrada" type="xs:int" minOccurs="0" />
          <xs:element name="numContenedor" type="xs:string" minOccurs="0" />
          <xs:element name="fSalida" type="xs:string" minOccurs="0" />
          <xs:element name="fIngreso" type="xs:string" minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:choice>
</xs:complexType>
</xs:element>
</xs:schema>
```

Donde:

- cuAduana.** Es la clave de la aduana.
- cuRecintoOrigen.** Es la clave del recinto origen.
- cuRecintoDestino.** Es la clave del recinto destino.
- entrada.** Es el número de entrada.
- numContenedor.** Es el número de contenedor.
- fSalida.** Es la fecha en que se registró la salida por transferencia.
- fIngreso.** Es la fecha en que se registró el ingreso por transferencia.

Por ejemplo, una respuesta sin errores sería como ésta:

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
```



```
<consultaIngresoPorTransferResponse xmlns="http://Recintos/ServiciosWebXml/">
  <consultaIngresoPorTransferResult>
    <error>0</error>
    <descripError />
    <registros>7</registros>
    <ds>
      <xs:schema id="ResConsulta" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
        <xs:element name="ResConsulta" msdata:IsDataSet="true" msdata:Locale="es-MX">
          <xs:complexType>
            <xs:choice maxOccurs="unbounded">
              <xs:element name="Contenedor">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="cuAduana" type="xs:int" minOccurs="0" />
                    <xs:element name="cuRecintoOrigen" type="xs:int" minOccurs="0" />

                    <xs:element name="cuRecintoDestino" type="xs:int" minOccurs="0" />
                    <xs:element name="entrada" type="xs:int" minOccurs="0" />
                    <xs:element name="numContenedor" type="xs:string" minOccurs="0" />
                    <xs:element name="fSalida" type="xs:string" minOccurs="0" />
                    <xs:element name="fIngreso" type="xs:string" minOccurs="0" />
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:choice>
          </xs:complexType>
        </xs:element>
      </xs:schema>
      <diffgr:diffgram xmlns:msdata="urn:schemas-microsoft-com:xml-msdata" xmlns:diffgr="urn:schemas-
microsoft-com:xml-diffgram-v1">
        <ResConsulta xmlns="">
          <Contenedor diffgr:id="Contenedor1" msdata:rowOrder="0" diffgr:hasChanges="inserted">
            <cuAduana>43</cuAduana>
            <cuRecintoOrigen>69</cuRecintoOrigen>
            <cuRecintoDestino>68</cuRecintoDestino>
            <entrada>1</entrada>
            <numContenedor>CSVU2505801</numContenedor>
          </Contenedor>
        </ResConsulta>
      </diffgr:diffgram>
    </ds>
  </consultaIngresoPorTransferResult>
</consultaIngresoPorTransferResponse>
```

```
<fSalida>01/01/1993 12:00:00 a.m.</fSalida>
<fIngreso>01/01/1993 12:00:00 a.m.</fIngreso>
</Contenedor>
<Contenedor diffgr:id="Contenedor2" msdata:rowOrder="1" diffgr:hasChanges="inserted">
  <cuAduana>43</cuAduana>
  <cuRecintoOrigen>69</cuRecintoOrigen>
  <cuRecintoDestino>68</cuRecintoDestino>
  <entrada>2</entrada>
  <numContenedor>LBIU2002630</numContenedor>
  <fSalida>01/01/1993 12:00:00 a.m.</fSalida>
  <fIngreso>01/01/1993 12:00:00 a.m.</fIngreso>
</Contenedor>
<Contenedor diffgr:id="Contenedor3" msdata:rowOrder="2" diffgr:hasChanges="inserted">
  <cuAduana>43</cuAduana>
  <cuRecintoOrigen>69</cuRecintoOrigen>
  <cuRecintoDestino>68</cuRecintoDestino>
  <entrada>3</entrada>
  <numContenedor>MLCU32552"5</numContenedor>
  <fSalida>01/01/1993 12:00:00 a.m.</fSalida>
  <fIngreso>01/01/1993 12:00:00 a.m.</fIngreso>
</Contenedor>
<Contenedor diffgr:id="Contenedor4" msdata:rowOrder="3" diffgr:hasChanges="inserted">
  <cuAduana>43</cuAduana>
  <cuRecintoOrigen>69</cuRecintoOrigen>
  <cuRecintoDestino>68</cuRecintoDestino>

  <entrada>4</entrada>
  <numContenedor>SCZU7608381</numContenedor>
  <fSalida>01/01/1993 12:00:00 a.m.</fSalida>
  <fIngreso>01/01/1993 12:00:00 a.m.</fIngreso>
</Contenedor>
<Contenedor diffgr:id="Contenedor5" msdata:rowOrder="4" diffgr:hasChanges="inserted">
  <cuAduana>43</cuAduana>
  <cuRecintoOrigen>69</cuRecintoOrigen>
  <cuRecintoDestino>68</cuRecintoDestino>
  <entrada>5</entrada>
  <numContenedor>TEXU3575918</numContenedor>
  <fSalida>01/01/1993 12:00:00 a.m.</fSalida>
```

```
<fIngreso>01/01/1993 12:00:00 a.m.</fIngreso>
</Contenedor>
<Contenedor diffgr:id="Contenedor6" msdata:rowOrder="5" diffgr:hasChanges="inserted">
  <cuAduana>43</cuAduana>
  <cuRecintoOrigen>69</cuRecintoOrigen>
  <cuRecintoDestino>68</cuRecintoDestino>
  <entrada>6</entrada>
  <numContenedor>TTNU2038575</numContenedor>
  <fSalida>01/01/1993 12:00:00 a.m.</fSalida>
  <fIngreso>01/01/1993 12:00:00 a.m.</fIngreso>
</Contenedor>
<Contenedor diffgr:id="Contenedor7" msdata:rowOrder="6" diffgr:hasChanges="inserted">
  <cuAduana>43</cuAduana>
  <cuRecintoOrigen>69</cuRecintoOrigen>
  <cuRecintoDestino>68</cuRecintoDestino>
  <entrada>7</entrada>
  <numContenedor>TTNU2246590</numContenedor>
  <fSalida>01/01/1993 12:00:00 a.m.</fSalida>
  <fIngreso>01/01/1993 12:00:00 a.m.</fIngreso>
</Contenedor>
</ResConsulta>
</diffgr:diffgram>
</ds>
</consultaIngresoPorTransferResult>
</consultaIngresoPorTransferResponse>
</soap:Body>
</soap:Envelope>
```

EJEMPLO EN C#

```
//Crea los parámetros de la consulta
ParamConsTransfer paramCons = new ParamConsTransfer();
paramCons.recintoDestino = this.txtRecintoDestino.Text.Trim();
paramCons.numContenedor = this.txtContenedor.Text.Trim();
paramCons.fechaIni = this.txtFecIni.Text.Trim();
paramCons.fechaFin = this.txtFecFin.Text.Trim();

/* Define los datos del encabezado */
EncabezadoSOAP enc = new EncabezadoSOAP();
enc.idUsuario = this.txtUsuario.Text.Trim();
enc.firma = new byte[60];

//Genera el proxy del servicio
Recintos recinto = new Recintos();
ResContenedor res;
recinto.EncabezadoSOAPValue = enc;

//Realiza la búsqueda de ingresos por transferencia
res = recinto.consultaIngresoPorTransferencia(paramCons);

lblError.Text = "Error: " + res.error;
lblDesc.Text = res.descripcion;

this.lblResultado.Text = "<strong>Resultado de la Consulta:</strong><br>";

// Muestra los registros obtenidos
if(res.ds != null && res.ds.Tables.Count > 0)
{
    string strResultado = "";

    strResultado += "<table border='1'>";

    foreach(DataColumn dtcColumna in res.ds.Tables["Contenedor"].Columns)
    {
        strResultado += "<th bgColor='cyan'>" + dtcColumna.Caption + "</th>";
    }

    foreach(DataRow dtrRegistro in res.ds.Tables["Contenedor"].Rows)
    {
        strResultado += "<tr>";
        strResultado += "<td>" + Convert.ToString(dtrRegistro["cuAduana"]) + "</td>";
        strResultado += "<td>" + Convert.ToString(dtrRegistro["cuRecintoOrigen"]) + "</td>";
        strResultado += "<td>" + Convert.ToString(dtrRegistro["cuRecintoDestino"]) + "</td>";
        strResultado += "<td>" + Convert.ToString(dtrRegistro["entrada"]) + "</td>";
        strResultado += "<td>" + Convert.ToString(dtrRegistro["numContenedor"]) + "</td>";
        strResultado += "<td>" + Convert.ToString(dtrRegistro["fSalida"]) + "</td>";
    }
}
```

```
        strResultado += "<td>" + Convert.ToString(dtrRegistro["fIngreso"]) + "</td>";  
        strResultado += "</tr>";  
    }  
  
    strResultado += "</table>";  
    this.lblResultado.Text += strResultado;  
}
```

Catálogo de errores

Los errores están clasificados en 6 niveles a saber:

1. **SEGURIDAD.** Aplica para todos los errores que representen un riesgo de seguridad.
2. **APLICACIÓN.** Identifica a todos los errores generados por la aplicación en el servidor web. Estos errores no se encuentran en el catálogo porque su descripción la da el controlador de excepciones del servidor web.
3. **BASE_DATOS.** Son todos los errores reportados por el manejador de base de datos.
4. **PARAMETROS.** Hay alguna inconsistencia en los parámetros entregados a un método.
5. **VALIDACION.** Al ejecutar un método algunos datos no pasaron las validaciones para considerar a la operación correcta.
6. **ADVERTENCIA.** En estos casos la operación si se realizó pero con las reservas indicadas en la advertencia.

Número	Descripción	Tipo
-2	ERROR AL REGISTRAR EL DESEMPEÑO DE LA OPERACION	5
-1	ERROR DE BASE DE DATOS	2
2	EL USUARIO NO EXISTE	1
3	EL USUARIO ESTA DADO DE BAJA	1
4	EL CERTIFICADO DEL USUARIO NO EXISTE	1
5	EL CERTIFICADO ES NULO EN LA BASE DE DATOS	1
6	EL CERTIFICADO YA NO ES VIGENTE	1
7	ERROR AL CONVERTIR EL CERTIFICADO DEL USUARIO	1
8	CERTIFICADO NO VALIDO	1
9	NO ES POSIBLE DECODIFICAR LA LLAVE PZBLICA DEL CERTIFICADO	1
10	NO ES POSIBLE OBTENER EL MODULO DE LA LLAVE PUBLICA DEL CERTIFICADO	1
11	NO COINCIDE LA FIRMA DEL USUARIO	1
12	LA FECHA DE ENTRADA DEBE SER MENOR O IGUAL A LA FECHA ACTUAL	5
13	LA IDENTIFICACION DE LA MERCANCMA DEL sam NO ESTA COMPLETA	5
14	DEBE DEFINIR AL MENOS UNA DE LAS 2 LISTAS (MERCANCIAS O CONTENEDORES)	5
15	LAS OPERACIONES CON MERCANCIA SUELTA DEBEN DEFINIR OBLIGATORIAMENTE: FECHA DE ENTRADA, PARCIALIDAD, NUMERO DE MERCANCIA, PESO Y PIEZAS	5
16	EL CONOCIMIENTO NO EXISTE, NO PUEDE REALIZAR LA OPERACION	5
17	EL ESTADO DEL CONOCIMIENTO NO ESTA PERMITIDO PARA INGRESAR MERCANCIAS, NO PUEDE REALIZAR LA OPERACION	5
18	LA MERCANCIA NO EXISTE EN LA BASE DE DATOS	5
19	NO PUEDE INGRESAR MERCANCIA CONTENERIZADA COMO MERCANCIA SUELTA	5
20	YA EXISTE UNA ENTRADA AL RECINTO PARA LA MERCANCIA	5

21	SOLO LA MERCANCIA SUELTA A GRANEL PUEDE INGRESARSE POR PARTES	5
22	LA SUMA DEL PESO Y LAS PIEZAS DE ENTRADAS PARCIALES NO PUEDE SER MAYOR AL DECLARADO EN EL MANIFIESTO	5
23	EL ESTADO DE LA MERCANCIA EN EL RECINTO YA NO PERMITE HACER MAS ENTRADAS PARCIALES	5
24	NINGUN ELEMENTO DE LA LISTA DE MERCANCIAS A INGRESAR PUEDE SER NULL	5
25	NO PUEDE INGRESAR NI MAS PESO NI MAS PIEZAS AL ALMACEN DE LA MERCANCIA DEL QUE FUE DECLARADO EN EL MANIFIESTO	5
26	EL CONTENEDOR NO TIENE MERCANCIAS RELACIONADAS EN EL CONOCIMIENTO	5
27	YA EXISTE UNA ENTRADA DEL CONTENEDOR CON ESE CONOCIMIENTO	5
28	EL PESO A INGRESAR ES DISTINTO AL DECLARADO EN EL CONOCIMIENTO	5
29	EL NUMERO DE PIEZAS A INGRESAR ES DISTINTO AL DECLARADO EN EL CONOCIMIENTO	5
30	TODAS LAS ENTRADAS DEBEN DEFINIR UNA FECHA DE INGRESO AL ALMACEN	5
31	TODAS LAS ENTRADAS DEBEN DEFINIR UNA FECHA DE INGRESO Y UN CONTENEDOR	5
32	EL ESTADO DE LA MERCANCIA NO PERMITE CANCELAR LA ENTRADA	5
33	LA MERCANCIA YA TIENE SALIDAS, NO PUEDE CANCELAR SU ENTRADA	5
34	LA ENTRADA NO EXISTE	5
35	NO PUEDE CANCELARSE LA ENTRADA, SI ES UNA ENTRADA CON PARCIALIDADES DEBE CANCELARLAS PRIMERO	5
36	LA ENTRADA PARCIAL NO EXISTE	5
37	LA ENTRADA YA FUE CANCELADA ANTES	5
38	EL ESTADO DE LA MERCANCIA NO PERMITE HACER SALIDAS	5
39	LA FECHA DE SALIDA NO PUEDE SER MENOR QUE LA FECHA DE ENTRADA	5
40	UNA SALIDA PUEDE HACERSE SOLO A TRAVES DE UN PEDIMENTO O A PETICION DE UNA AUTORIDAD	5
41	PARA REALIZAR UNA SALIDA DEBE INDICAR LA PATENTE DEL AGENTE Y EL NUMERO DE PEDIMENTO	5
42	EL SP NO REGRESO NINGUN VALOR	5
43	EL PEDIMENTO NO EXISTE EN LA ADUANA	5
44	NO EXISTE LA FACTURA DEL PREVIO DE CONSOLIDADO	5
45	EL TIPO DE PEDIMENTO NO ESTA PERMITIDO PARA HACER SALIDAS	5
46	EL PESO O LAS PIEZAS EN LA SALIDA NO PUEDEN SER MAYORES A LA EXISTENCIA	5

47	LAS SALIDAS PARCIALES SOLO APLICAN PARA MERCANCIA A GRANEL	5
48	EN EL INGRESO DE MERCANCIAS DE ESTE TIPO DEBEN DEFINIRSE TODOS LOS DATOS	5
49	EN EL INGRESO DE MERCANCIAS DE ESTE TIPO DEBE DEFINIR EL TIPO DE MERCANCIA	5
50	EN EL INGRESO DE MERCANCIAS DE ESTE TIPO DEBE DEFINIR LA LISTA DE SELLOS DEL CONTENEDOR	5
51	LA SALIDA NO EXISTE	5
52	EL TIPO DE SALIDA NO PUEDE CANCELARSE	5
53	EL CONTENEDOR NO EXISTE EN EL PEDIMENTO	5
54	EL CONTENEDOR YA SALIO DEL RECINTO	5
55	EL PEDIMENTO NO ES VALIDO PARA REALIZAR SALIDAS	5
56	EL PEDIMENTO NO ES VALIDO PARA REALIZAR SALIDAS	5
57	EL AVISO YA ESTA REGISTRADO	5
58	EL TIPO DE ENTRADA NO ESTA PERMITIDO EN UN AVISO	5
59	EL ESTADO DE LA ENTRADA NO ESTA PERMITIDO EN UN AVISO	5
60	LA ENTRADA DE UN AVISO NO PUEDE TENER SALIDAS	5
61	LA PATENTE NO ES VALIDA	5
62	PARA UN AVISO DE TRANSFERENCIA DEBE DECLARAR: NUMERO DE AVISO, LA PATENTE DEL AGENTE, LA FECHA DEL AVISO, EL RECINTO ORIGEN, LA TARJA DE LA OPERACION Y LA LISTA DE ENTRADAS.	5
63	NINGUNA DE LAS ENTRADAS DE LA LISTA PUEDE SER NULO	5
64	EL AVISO NO EXISTE	5
65	EL ESTADO DE LAS ENTRADAS NO PERMITE CANCELAR EL AVISO	5
66	DEBE INDICAR EL NUMERO DE AVISO	5
67	EL TIPO DE ENTRADA NO ES VALIDO	5
68	EL TIPO DE SALIDA NO ES POR TRANSFERENCIA	5
69	LA ENTRADA NO ESTA RELACIONADA CON ESTE AVISO	5
70	EL ESTADO DE LA ENTRADA NO ES VALIDO	5
71	LA ENTRADA DEL RECINTO ORIGEN NO EXISTE.	5
72	PARA UN AVISO DE TRASPALEO DEBE DECLARAR: NUMERO DE AVISO, LA FECHA DEL AVISO, Y LA LISTA DE CONTENEDORES ORIGEN Y LA LISTA DE CONTENEDORES DESTINO.	5
73	CADA CONTENEDOR DE LA LISTA DE CONTENEDORES ORIGEN DEBE DEFINIR SU ENTRADA/CONTENEDOR Y SI ES UN TRASPALEO TOTAL O NO (VERDADERO/FALSO).	5
74	EN UN TRASPALEO LA LISTA DE CONTENDORES DESTINO DEBE DEFINIR LA	5

	ENTRADA DEL CONTENEDOR O EL NUMERO DE CONTENEDOR.	
75	LA ENTRADA INDICADA DEBE ESTAR IDENTIFICADA COMO CONTENERIZADA EN LA BASE DE DATOS.	5
76	NO FUE POSIBLE ASIGNAR AL CONTENEDOR DESTINO DEL TRASPALEO.	5
77	EN ESTA OPERACION HAY ALGUN(OS) PARAMETRO(S) NO VALIDO(S).	5
78	LA CLAVE NO EXISTE	5
79	EL EMBALAJE NO PUEDE SER CONTENEDOR	5
80	EL PESO O LAS PIEZAS A DESCONSOLIDAR NO PUEDEN SER MAYORES A LA EXISTENCIA.	5
81	DEBE INDICAR EL RECINTO DESTINO.	5
82	DEBE INDICAR EL RECINTO ORIGEN.	5
83	LA ADUANA DE AMBOS RECINTO (ORIGEN Y DESTINO) DEBE SER LA MISMA.	5
84	LA ENTRADA RELACIONADA CON LA SALIDA NO EXISTE.	5
85	DEBE INDICAR LA CLAVE DE AUTORIDAD	5
86	DEBE INDICAR EL DOCUMENTO QUE JUSTIFICA LA OPERACION	5
87	DEBE INDICAR EL TIPO DE ENTRADA	5
88	DEBE INDICAR LA LISTA DE MERCANCIA SUELTA O LA LISTA DE CONTENEDORES	5
89	DEBE INDICAR EL CONSECUTIVO DE LA MERCANCIA	5
90	DEBE INDICAR EL PESO	5
91	DEBE INDICAR LAS PIEZAS	5
92	DEBE INDICAR LA DESCRIPCION DE LA MERCANCIA	5
93	DEBE INDICAR LA CLAVE DEL EMBALAJE	5
94	DEBE INDICAR EL TIPO DE MERCANCIA	5
95	DEBE INDICAR LA UNIDAD DE MERDIDA PARA EL PESO	5
96	DEBE INDICAR LA UNIDAD DE MERDIDA PARA EL PESO	5
97	DEBE INDICAR LA LISTA DE SELLOS	5
98	LA ORDEN DE E/S YA EXISTE	5
99	LA CLAVE DE AUTORIDAD NO EXISTE	5
100	YA HAY UNA ENTRADA REGISTRADA PARA EL CONTENEDOR	5
101	LA CLAVE DE EMBALAJE NO EXISTE	5
102	LA CLAVE DE UNIDAD DE MEDIDA NO EXISTE	5
103	LA CLAVE DEL TIPO DE CONTENEDOR NO EXISTE	5
104	LA CLAVE DEL TIPO DE MERCANCIA NO EXISTE	5
105	DEBE INDICAR EL NUMERO DE ENTRADA	5
106	EL PESO Y LAS PIEZAS DEBEN SER IGUALES A LOS DE LA ENTRADA	5
107	EL TIPO DE LA SALIDA NO PERMITE CANCELARLA	5

108	LA LISTA DE CONTENEDORES NO DEBE ESTAR VACIA	5
109	NO HAY MERCANCIA RELACIONADA CON EL SAM	5
110	EL AVISO NO TIENE ENTRADAS RELACIONADAS	5
111	LA MERCANCIA (PESO Y PIEZAS) DEBE SER MENOR A LA MERCANCIA ORIGEN	5
112	LA CLAVE DE EMBALAJE DE LA MERCANCIA NUEVA DEBE SER IGUAL AL DE LA MERCANCIA ORIGEN	5
113	LA CLAVE DE EMBALAJE DE LA MERCANCIA NUEVA DEBE SER DIFERENTE AL DE LA MERCANCIA ORIGEN	5
114	UN CONTENEDOR ORIGEN DE UN TRASPALEO NO PUEDE SER DESTINO AL MISMO TIEMPO	5
115	UN CONTENEDOR ESTA DEFINIDO DOS VECES EN LA LISTA	5
116	LA MERCANCIA ORIGEN DE UNA DESCONSOLIDACION DEBE SER UN CONTENEDOR	5
117	EL TIPO DE MERCANCIA DECLARADO EN LA ENTRADA NO COINCIDE CON LOS DECLARADOS EN EL MANIFIESTO	5
118	NO PUEDE DECLARA EL MISMO TIPO DE MERCANCIA DOS VECES EN UNA MERCANCIA	5
119	ESTE NO ES UN MENSAJE SOAP VALIDO	5
120	NO ESTAN TODAS LAS PARTES REQUERIDAS DEL MENSAJE SOAP	5
121	EL IDENTIFICADOR DEL CERTIFICADO NO COINCIDE	5
122	EL PEDIMENTO HA SIDO RECTIFICADO	5
123	EL CONTENEDOR HA SIDO RECTIFICADO	5
124	LA MERCANCIA YA ESTA AUTORIZADA PARA SALIR	5
125	LA SALIDA POR TRANSFERENCIA ESTA DESTINADA A OTRO RECINTO	5
126	OTRA OPERACION CON LAS MISMAS CARACTERISTICAS SE ENCUENTRA EN PROCESO	5

Descripción detallada de errores

Nota: Los errores del 4 al 11 están reservados para uso futuro ya que en este momento no se está autenticando con certificados digitales.

-1 (MENOS 1)

ERROR DE BASE DE DATOS

Descripción:

Sucedió un error en el manejador de base de datos del sistema.

Solución:

Reintente la operación. Si el problema persiste levante un reporte de mesa de ayuda para su atención.

-2 (MENOS 2)

ADVERTENCIA: ERROR AL REGISTRAR DESEMPEÑO DE LA OPERACIÓN

Descripción:

Sucedió un error en el manejador de base de datos del sistema al registrar el desempeño de la operación.

Solución:

Si este es el único código de error recibido la operación se registró correctamente, reintente la operación y observe los códigos de error devueltos, éstos deben indicar que la operación anterior se proceso correctamente.

2

EL USUARIO NO EXISTE

Descripción:

El identificador de usuario proporcionado en el encabezado del mensaje no es reconocido por el sistema.

Solución:

Corrija el identificador de usuario en el encabezado del mensaje y reintente la operación. Si el problema persiste levante un reporte de mesa de ayuda para su atención.

3

EL USUARIO ESTA DADO DE BAJA

Descripción:

El identificador de usuario proporcionado en el encabezado del mensaje ya está dado de baja y no es válido para el sistema.

Solución:

Corrija el identificador de usuario en el encabezado del mensaje y reintente la operación. Si el problema persiste levante un reporte de mesa de ayuda para su atención.

4

EL CERTIFICADO DEL USUARIO NO EXISTE

Descripción:

No fue posible obtener el certificado digital del usuario que está solicitando la operación.

Solución:

Levante un reporte de mesa de ayuda para su atención.

5

EL CERTIFICADO ES NULO EN LA BASE DE DATOS

Descripción:

No fue posible obtener el certificado digital del usuario que está solicitando la operación.

Solución:

Levante un reporte de mesa de ayuda para su atención.

6

EL CERTIFICADO YA NO ES VIGENTE

Descripción:

El certificado digital del usuario que está solicitando la operación ya caducó y no es posible realizar operaciones con él.

Solución:

Levante un reporte de mesa de ayuda para su atención.

7

ERROR AL CONVERTIR EL CERTIFICADO DEL USUARIO

Descripción:

No fue posible obtener el certificado digital del usuario que está solicitando la operación.

Solución:

Si el problema persiste levante un reporte de mesa de ayuda para su atención.

8

CERTIFICADO NO VALIDO

Descripción:

No fue posible obtener el certificado digital del usuario que está solicitando la operación.

Solución:

Si el problema persiste levante un reporte de mesa de ayuda para su atención.

9

NO ES POSIBLE DECODIFICAR LA LLAVE PUBLICA DEL CERTIFICADO

Descripción:

No fue posible obtener el certificado digital del usuario que está solicitando la operación.

Solución:

Si el problema persiste levante un reporte de mesa de ayuda para su atención.

10

NO ES POSIBLE OBTENER EL MODULO DE LA LLAVE PUBLICA DEL CERTIFICADO

Descripción:

No fue posible obtener el certificado digital del usuario que está solicitando la operación.

Solución:

Si el problema persiste levante un reporte de mesa de ayuda para su atención.

11

NO COINCIDE LA FIRMA DEL USUARIO

Descripción:

La firma incluida en el mensaje no pudo ser validada con el certificado del usuario.

Solución:

Use la llave privada que corresponde al certificado y reintente la operación. Si el problema persiste levante un reporte de mesa de ayuda para su atención.

12

LA FECHA DE ENTRADA DEBE SER MENOR O IGUAL A LA FECHA ACTUAL

Descripción:

En esta versión este error ya no se presenta.

Solución:

No hay.

13

LA IDENTIFICACION DE LA MERCANCIA DEL sam NO ESTA COMPLETA

Descripción:

Al intentar registrar una entrada de mercancía faltó definir todos los datos que la identifican en un manifiesto.

Solución:

Defina todos los parámetros que identifican a la mercancía en el manifiesto y reintente la operación. Para mayor información vea [Registra la entrada de mercancías al almacén](#).

14

DEBE DEFINIR AL MENOS UNA DE LAS 2 LISTAS (MERCANCIAS O CONTENEDORES)

Descripción:

Al intentar registrar una entrada de mercancía faltó definir la lista de mercancía suelta o la lista de contenedores en los parámetros de entrada.

Solución:

Defina la lista de que se trate la mercancía y reintente la operación. Para mayor información vea [Registra la entrada de mercancías al almacén](#).

15

LAS OPERACIONES CON MERCANCIA SUELTA DEBEN DEFINIR OBLIGATORIAMENTE: PARCIALIDAD, NUMERO DE MERCANCIA, PESO Y PIEZAS

Descripción:

Al intentar registrar una entrada de mercancía suelta faltó definir todos los datos obligatorios.

Solución:

Defina todos los datos de la mercancía a ingresar y reintente la operación. Para mayor información vea [Entrada de mercancía suelta](#).

16

EL CONOCIMIENTO NO EXISTE, NO PUEDE REALIZAR LA OPERACIÓN

Descripción:

Al intentar registrar una entrada de mercancía el conocimiento indicado en los parámetros no existe en la base de datos.

Solución:

Defina todos los datos que identifican al contenedor y reintente la operación. Para mayor información vea [Para mercancía que está en el SAM](#).

17

EL ESTADO DEL CONOCIMIENTO NO ESTA PERMITIDO PARA INGRESAR MERCANCIAS, NO PUEDE REALIZAR LA OPERACIÓN

Descripción:

Al intentar registrar una entrada de mercancía el conocimiento indicado en los parámetros tiene un estado que no permite usarlo para ingresar mercancía en el recinto.

Solución:

Asegúrese que sea el conocimiento correcto y reintente la operación. Si el problema persiste levante un reporte de mesa de ayuda para su atención.

18

LA MERCANCIA NO EXISTE EN LA BASE DE DATOS

Descripción:

Al intentar registrar una entrada de mercancía suelta el número de mercancía indicado no existe en el manifiesto.

Solución:

Asegúrese que el número de mercancía indicado sea correcto y reintente la operación. Para mayor información vea [Entrada de mercancía suelta](#). Si el problema persiste levante un reporte de mesa de ayuda para su atención.

19

NO PUEDE INGRESAR MERCANCIA CONTENERIZADA COMO MERCANCIA SUELTA

Descripción:

Al intentar registrar una entrada de contenedor el sistema la indentificó como mercancía suelta. Asegúrese que está intentando registrar el ingreso de la mercancía correcta y reintente la operación.

Solución:

Asegúrese que el número de mercancía indicado sea correcto y reintente la operación. Para mayor vea [Entrada de mercancía contenerizada](#). Si el problema persiste levante un reporte de mesa de ayuda para su atención.

20

YA EXISTE UNA ENTRADA AL RECINTO PARA LA MERCANCIA

Descripción:

Este es un código de advertencia porque una operación anterior registró la entrada de la mercancía en el recinto.

Solución:

Puede obtener más información en la sección [Registra la entrada de mercancías al almacén](#).

21

SOLO LA MERCANCIA SUELTA A GRANEL PUEDE INGRESARSE POR PARTES

Descripción:

Al realizar un registro de entrada de mercancía el peso y/o las piezas declaradas en los parámetros de entrada son menores a los declarados en el manifiesto.

Solución:

Indique las cantidades de peso y piezas igual a como están declaradas en el manifiesto en los parámetros de entrada del método y reintente la operación. Puede obtener más información en la sección [Registra la entrada de mercancías al almacén](#).

22

LA SUMA DEL PESO Y LAS PIEZAS DE ENTRADAS PARCIALES NO PUEDE SER MAYOR AL DECLARADO EN EL MANIFIESTO

Descripción:

Al realizar un registro de entrada de mercancía por partes el peso y/o las piezas declaradas en los parámetros de entrada se han ido acumulando y ahora son mayor a los declarados en el manifiesto.

Solución:

La suma en las entradas de mercancía por partes nunca debe ser mayor a las declaradas en el manifiesto. Puede obtener más información en la sección [Registra la entrada de mercancías al almacén](#).

23

EL ESTADO DE LA MERCANCIA EN EL RECINTO YA NO PERMITE HACER MAS ENTRADAS PARCIALES

Descripción:

La mercancía que se ha identificado para ingresarse por partes tiene un estado que no permite registrar más entradas parciales.

Solución:

Para registrar la entrada de varias partes de una mercancía, el registro de la entrada inicial debe conservar el estado: INGRESADA AL ALMACÉN o el estado: AUTORIZADA PARA SALIR.

24

NINGUN ELEMENTO DE LA LISTA DE MERCANCIAS A INGRESAR PUEDE SER NULO

Descripción:

Al intentar ingresar mercancía al recinto se está entregando un elemento con valor null en el arreglo de lista de mercancía suelta o en el arreglo de lista de contenedores de los parámetros de entrada.

Solución:

Asegúrese que ninguno de los elementos de la lista entregada en los parámetros de entrada del método sea null y reintente la operación. Para más detalles vea [Registra la entrada de mercancías al almacén](#).

25

NO PUEDE INGRESAR NI MAS PESO NI MAS PIEZAS AL ALMACEN DE LA MERCANCIA DEL QUE FUE DECLARADO EN EL MANIFIESTO

Descripción:

Al intentar ingresar mercancía al recinto se está entregando un elemento con valor null en el arreglo de lista de mercancía suelta o en el arreglo de lista de contenedores de los parámetros de entrada.

Solución:

Asegúrese que ninguno de los elementos de la lista entregada en los parámetros de entrada del método sea null y reintente la operación.

26

EL CONTENEDOR NO TIENE MERCANCIAS RELACIONADAS EN EL CONOCIMIENTO

Descripción:

Al intentar ingresar una contenedor el sistema no encontró ninguna mercancía relacionada con el en el manifiesto dado en los parámetros de entrada.

Solución:

Verifique que los datos proporcionados en los parámetros de entrada sean correctos. Para obtener los datos correctos del manifiesto vea .

27	YA EXISTE UNA ENTRADA DEL CONTENEDOR CON ESE CONOCIMIENTO	5
28	EL PESO A INGRESAR ES DISTINTO AL DECLARADO EN EL CONOCIMIENTO	5
29	EL NUMERO DE PIEZAS A INGRESAR ES DISTINTO AL DECLARADO EN EL CONOCIMIENTO	5
30	TODAS LAS ENTRADAS DEBEN DEFINIR UNA FECHA DE INGRESO AL ALMACEN	5
31	TODAS LAS ENTRADAS DEBEN DEFINIR UNA FECHA DE INGRESO Y UN CONTENEDOR	5
32	EL ESTADO DE LA MERCANCIA NO PERMITE CANCELAR LA ENTRADA	5
33	LA MERCANCIA YA TIENE SALIDAS, NO PUEDE CANCELAR SU ENTRADA	5
34	LA ENTRADA NO EXISTE	5
35	NO PUEDE CANCELARSE LA ENTRADA, SI ES UNA ENTRADA CON PARCIALIDADES DEBE CANCELARLAS PRIMERO	5
36	LA ENTRADA PARCIAL NO EXISTE	5
37	LA ENTRADA YA FUE CANCELADA ANTES	5
38	EL ESTADO DE LA MERCANCIA NO PERMITE HACER SALIDAS	5
39	LA FECHA DE SALIDA NO PUEDE SER MENOR QUE LA FECHA DE ENTRADA	5
40	UNA SALIDA PUEDE HACERSE SOLO A TRAVES DE UN PEDIMENTO O A PETICION DE UNA AUTORIDAD	5
41	PARA REALIZAR UNA SALIDA DEBE INDICAR LA PATENTE DEL AGENTE Y EL NUMERO DE PEDIMENTO	5
42	EL SP NO REGRESO NINGUN VALOR	5
43	EL PEDIMENTO NO EXISTE EN LA ADUANA	5
44	NO EXISTE LA FACTURA DEL PREVIO DE CONSOLIDADO	5
45	EL TIPO DE PEDIMENTO NO ESTA PERMITIDO PARA HACER SALIDAS	5
46	EL PESO O LAS PIEZAS EN LA SALIDA NO PUEDEN SER MAYORES A LA EXISTENCIA	5
47	LAS SALIDAS PARCIALES SOLO APLICAN PARA MERCANCIA A GRANEL	5
48	EN EL INGRESO DE MERCANCIAS DE ESTE TIPO DEBEN DEFINIRSE TODOS LOS DATOS	5
49	EN EL INGRESO DE MERCANCIAS DE ESTE TIPO DEBE DEFINIR EL TIPO DE MERCANCIA	5
50	EN EL INGRESO DE MERCANCIAS DE ESTE TIPO DEBE DEFINIR LA LISTA DE SELLOS DEL CONTENEDOR	5
51	LA SALIDA NO EXISTE	5
52	EL TIPO DE SALIDA NO PUEDE CANCELARSE	5
53	EL CONTENEDOR NO EXISTE EN EL PEDIMENTO	5
54	EL CONTENEDOR YA SALIO DEL RECINTO	5
55	EL PEDIMENTO NO ES VALIDO PARA REALIZAR SALIDAS	5
56	EL PEDIMENTO NO ES VALIDO PARA REALIZAR SALIDAS	5
57	EL AVISO YA ESTA REGISTRADO	5
58	EL TIPO DE ENTRADA NO ESTA PERMITIDO EN UN AVISO	5
59	EL ESTADO DE LA ENTRADA NO ESTA PERMITIDO EN UN AVISO	5
60	LA ENTRADA DE UN AVISO NO PUEDE TENER SALIDAS	5
61	LA PATENTE NO ES VALIDA	5
62	PARA UN AVISO DE TRANSFERENCIA DEBE DECLARAR: NUMERO DE AVISO, LA PATENTE DEL AGENTE, LA FECHA DEL AVISO, EL RECINTO ORIGEN, LA TARJA DE LA OPERACION Y LA LISTA DE ENTRADAS.	5
63	NINGUNA DE LAS ENTRADAS DE LA LISTA PUEDE SER NULO	5
64	EL AVISO NO EXISTE	5

65	EL ESTADO DE LAS ENTRADAS NO PERMITE CANCELAR EL AVISO	5
66	DEBE INDICAR EL NUMERO DE AVISO	5
67	EL TIPO DE ENTRADA NO ES VALIDO	5
68	EL TIPO DE SALIDA NO ES POR TRANSFERENCIA	5
69	LA ENTRADA NO ESTA RELACIONADA CON ESTE AVISO	5
70	EL ESTADO DE LA ENTRADA NO ES VALIDO	5
71	LA ENTRADA DEL RECINTO ORIGEN NO EXISTE.	5
72	PARA UN AVISO DE TRASPALO DEBE DECLARAR: NUMERO DE AVISO, LA FECHA DEL AVISO, Y LA LISTA DE CONTENEDORES ORIGEN Y LA LISTA DE CONTENEDORES DESTINO.	5
73	CADA CONTENEDOR DE LA LISTA DE CONTENEDORES ORIGEN DEBE DEFINIR SU ENTRADA/CONTENEDOR Y SI ES UN TRASPALO TOTAL O NO (VERDADERO/FALSO).	5
74	EN UN TRASPALO LA LISTA DE CONTENEDORES DESTINO DEBE DEFINIR LA ENTRADA DEL CONTENEDOR O EL NUMERO DE CONTENEDOR.	5
75	LA ENTRADA INDICADA DEBE ESTAR IDENTIFICADA COMO CONTENERIZADA EN LA BASE DE DATOS.	5
76	NO FUE POSIBLE ASIGNAR AL CONTENEDOR DESTINO DEL TRASPALO.	5
77	EN ESTA OPERACION HAY ALGUN(OS) PARAMETRO(S) NO VALIDO(S).	5
78	LA CLAVE NO EXISTE	5
79	EL EMBALAJE NO PUEDE SER CONTENEDOR	5
80	EL PESO O LAS PIEZAS A DESCONSOLIDAR NO PUEDEN SER MAYORES A LA EXISTENCIA.	5
81	DEBE INDICAR EL RECINTO DESTINO.	5
82	DEBE INDICAR EL RECINTO ORIGEN.	5
83	LA ADUANA DE AMBOS RECINTO (ORIGEN Y DESTINO) DEBE SER LA MISMA.	5
84	LA ENTRADA RELACIONADA CON LA SALIDA NO EXISTE.	5
85	DEBE INDICAR LA CLAVE DE AUTORIDAD	5
86	DEBE INDICAR EL DOCUMENTO QUE JUSTIFICA LA OPERACION	5
87	DEBE INDICAR EL TIPO DE ENTRADA	5
88	DEBE INDICAR LA LISTA DE MERCANCIA SUELTA O LA LISTA DE CONTENEDORES	5
89	DEBE INDICAR EL CONSECUTIVO DE LA MERCANCIA	5
90	DEBE INDICAR EL PESO	5
91	DEBE INDICAR LAS PIEZAS	5
92	DEBE INDICAR LA DESCRIPCION DE LA MERCANCIA	5
93	DEBE INDICAR LA CLAVE DEL EMBALAJE	5
94	DEBE INDICAR EL TIPO DE MERCANCIA	5
95	DEBE INDICAR LA UNIDAD DE MEDIDA PARA EL PESO	5
96	DEBE INDICAR LA UNIDAD DE MEDIDA PARA EL PESO	5
97	DEBE INDICAR LA LISTA DE SELLOS	5
98	LA ORDEN DE E/S YA EXISTE	5

99	LA CLAVE DE AUTORIDAD NO EXISTE	5
100	YA HAY UNA ENTRADA REGISTRADA PARA EL CONTENEDOR	5
101	LA CLAVE DE EMBALAJE NO EXISTE	5
102	LA CLAVE DE UNIDAD DE MEDIDA NO EXISTE	5
103	LA CLAVE DEL TIPO DE CONTENEDOR NO EXISTE	5
104	LA CLAVE DEL TIPO DE MERCANCIA NO EXISTE	5
105	DEBE INDICAR EL NUMERO DE ENTRADA	5
106	EL PESO Y LAS PIEZAS DEBEN SER IGUALES A LOS DE LA ENTRADA	5
107	EL TIPO DE LA SALIDA NO PERMITE CANCELARLA	5
108	LA LISTA DE CONTENEDORES NO DEBE ESTAR VACIA	5
109	NO HAY MERCANCIA RELACIONADA CON EL SAM	5
110	EL AVISO NO TIENE ENTRADAS RELACIONADAS	5
111	LA MERCANCIA (PESO Y PIEZAS) DEBE SER MENOR A LA MERCANCIA ORIGEN	5
112	LA CLAVE DE EMBALAJE DE LA MERCANCIA NUEVA DEBE SER IGUAL AL DE LA MERCANCIA ORIGEN	5
113	LA CLAVE DE EMBALAJE DE LA MERCANCIA NUEVA DEBE SER DIFERENTE AL DE LA MERCANCIA ORIGEN	5
114	UN CONTENEDOR ORIGEN DE UN TRASPALCO NO PUEDE SER DESTINO AL MISMO TIEMPO	5
115	UN CONTENEDOR ESTA DEFINIDO DOS VECES EN LA LISTA	5
116	LA MERCANCIA ORIGEN DE UNA DESCONSOLIDACION DEBE SER UN CONTENEDOR	5
117	EL TIPO DE MERCANCIA DECLARADO EN LA ENTRADA NO COINCIDE CON LOS DECLARADOS EN EL MANIFIESTO	5
118	NO PUEDE DECLARAR EL MISMO TIPO DE MERCANCIA DOS VECES EN UNA MERCANCIA	5
119	ESTE NO ES UN MENSAJE SOAP VALIDO	5
120	NO ESTAN TODAS LAS PARTES REQUERIDAS DEL MENSAJE SOAP	5
121	EL IDENTIFICADOR DEL CERTIFICADO NO COINCIDE	5
122	EL PEDIMENTO HA SIDO RECTIFICADO	5
123	EL CONTENEDOR HA SIDO RECTIFICADO	5
124	LA MERCANCIA YA ESTA AUTORIZADA PARA SALIR	5
125	LA SALIDA POR TRANSFERENCIA ESTA DESTINADA A OTRO RECINTO	5
126	OTRA OPERACION CON LAS MISMAS CARACTERISTICAS SE ENCUENTRA EN PROCESO	5
127	LOS PEDIMENTOS PROPORCIONADOS NO COINCIDEN CON LOS REGISTRADOS EN LA AUTORIZACION DE SALIDA	5
128	EL PEDIMENTO NO HA SIDO PAGADO	5
129	EL PEDIMENTO HA SIDO PAGADO	5
130	EL PEDIMENTO HA SIDO MODULADO	5
131	EL PEDIMENTO NO HA SIDO MODULADO	5
132	UNA AUTORIZACION DE SALIDA YA HA SIDO REGISTRADA CON EL MISMO PARAMETRO DE REMESA/FACTURA	5
133	DEBE INDICAR EL NUMERO DE FACTURA	5
134	DEBE INDICAR EL NUMERO DE REMESA	5
135	SE HA EXEDIDO EL NUMERO DE REMESAS PARA EL PEDIMENTO	5
136	EL CONOCIMIENTO NO ESTA LIBERADO	5
137	EL MANIFIESTO NO EXISTE	5

138	EL CONTENEDOR ESTA BLOQUEADO	5
139	EL CONTENEDOR ESTA EN REVISION POR LA ADUANA	5
140	EL ARRIBO SE REGISTRO EN SAAPA PERO NO EN SICREFIS	5
141	EL CAAT NO COINCIDE	5
142	EL ARRIBO NO SE PUDO REGISTRAR EN SAAPA	5
143	LA CLAVE DEL TIPO DE SERVICIO DEL CONTENEDOR ES ERRONEA	5
144	LA CLAVE DEL TIPO DE CONTENEDOR ES ERRONEA	5