



ESPECIFICACIÓN DE SERVICIOS WEB
(Revisión 15.0)

Tabla de contenido

Historial cambios.....	4
Uso indebido de los servicios	4
Servicios XML	5
Uso de los servicios en PHP	5
Uso de servicios desde ASP	6
Ejemplo 1.....	6
Ejemplo 2.....	6
Servicios Sidetours	8
Código 100: Lista de destinos	9
Código 101: Lista de zonas por destino	10
Código 102: Lista de categorías	11
Código 103: Lista de países.....	12
Código 104: Idiomas	13
Código 105: Regímenes	14
Código 106: Servicios de establecimientos	15
Código 107: Servicios por establecimiento	16
Código 108: Provincias	17
Código 110: Destinos y zonas	18
Código 111: Tipos de turismo	19
Código 112: Tipos de acomodación	20
Código 200: Detalles de un establecimiento	21
Código 201: Lista de establecimientos.....	25
Código 202: Catálogo de establecimientos	28
Código 500: Reserva básica	30
Código 501: Catálogo de reservas	34
Código 502: Detalles reserva.....	37
Código 503: Cancelación de reserva	39
Código 504: Confirmación de reserva.....	40
Código 505: Simulación de reserva.....	41
Código 600: Tarifas.....	43
Código 601: Disponibilidad diaria	44
Código 602: Paros de venta.....	46
Código 620: Cabecera de ofertas.....	48
Código 1202: Catálogo de edificios extendido.....	50

Código 1400: Disponibilidad avanzada.....	52
Código 1503: Cancelación de reserva con gastos	61
Código 1504: Confirmación de reservas extendida.....	62

Historial cambios

Versión	Cambios
14.3	
	Nuevo servicio 112: Devuelve un listado con todos los tipos de acomodación contemplados en el sistema
14.4	
	Se incluyen observaciones de la oferta en el servicio de disponibilidad 1400, en la nueva etiqueta COMMENT dentro de OFFER.
14.4.1	
	Indicamos que la codificación de nuestros webservices es ISO-8859-1 Corregimos bug en el DTD de respuesta del servicio 501 <ul style="list-style-type: none"> • Añadimos criterio “file” a los parámetros y la respuesta del servicio. • Incluimos atributos “source” y “state” en la respuesta.
14.4.2	
	Añadimos información al servicio 200 de detalles de un edificio.
15.0	
	Añadimos la etiqueta <i>babies</i> así como las etiquetas para tercer y cuarto niño donde faltaba en los servicios de disponibilidad y reserva. Actualizamos descripción de los servicios de caché (600, 601 ...)

Uso indebido de los servicios

El servicio de disponibilidad (1400) está preparado para ser usado para saber la disponibilidad con determinados filtros, Zona, Destino Turístico, Provincias incluso edificios sueltos, aunque este último no debe de ser utilizado para saber la disponibilidad en una zona, destino turístico o provincias sino utilizar el filtro adecuado, ya que eso puede provocar una sobrecarga del sistema perjudicando a los demás clientes.

Las fotografías que ofrece Sidetours son para que nuestros clientes directos las descarguen en su sistema y así poderlas ofrecer directamente a sus clientes, en ningún caso deberá ofrecer los links directos a Sidetours, ya que solamente serán accesibles desde las IPs permitidas del cliente.

Tenemos un servicio de cancelación 503 para cancelar reservas. Llamar a este servicio no implica que la reserva se haya cancelado, porque el servicio puede haber dado respuesta negativa.

Servicios XML

En este documento facilitado por VIAJES SIDETOURLS S.A. (en adelante SIDETOURLS) se explican los detalles de los servicios definidos.

La definición de la estructura XML para cada servicio se ha realizado mediante ficheros DTD. Se definen los elementos admisibles dentro del documento, los atributos posibles y (opcionalmente) los valores de atributo permitidos para cada elemento. Es más, describirá los anidamientos y ocurrencias de elementos.

Uso de los servicios en PHP

En lugar de explicar la teoría de servicios web, utilizaremos algunos ejemplos de invocación vía PHP para ilustrar cómo manejarlos.

- Cómo indicar a PHP que se quieren emplear los servicios de Sidetours:

La librería *nusoap.php* contiene todas las clases y métodos necesarios para utilizar *web-services* a través del protocolo de intercambio de datos SOAP. Esta librería habría de estar en el mismo directorio que la página que quiere hacer uso del servicio. En caso de estar en algún otro lugar, se deberá indicar en la instrucción *require_once*.

```
require_once("nusoap.php");
```

En un entorno de desarrollo, para hacer pruebas, utilizaremos la dirección siguiente:

```
$wsdl="http://www.viajessidetours.com/webservices/side_services.php?wsdl";
```

- Declaración de un cliente para atacar a los servicios (utilizando *SoapToolKit 3.0*)

```
$client = new soapclient($wsdl, "wsdl");
```

- Llamada a un servicio sin parámetros:

```
$Resultado=$client->call("nombre_servicio", array());
```

- Llamada a un servicio con parámetros:

```
$Resultado1=$client->call("servicio1", array($param1));
```

```
$Resultado2=$client->call("servicio2", array($param1, $param2));
```

- Uso del resultado

El tratamiento del valor devuelto por el servicio dependerá fuertemente del contexto. A modo de ejemplo aquí imprimimos en pantalla el resultado recibido.

```
printf("Datos devueltos por el servicios XML: $Resultado\n");
```

Uso de servicios desde ASP

Ponemos dos ejemplos sobre cómo invocar los servicios desde ASP, uno empleando SOAP ToolKit y otro pasando las cabeceras directamente.

Ejemplo 1

En este ejemplo se accede al servicio 110 (Destinos y zonas) por el usuario MI_USUARIO con contraseña MI_PASSWORD.

```
<%
    parametros = "<PARAMETERS zone="" destination=""5""
building_name="" ... "

Los parámetros no pueden contener los símbolos < ó >, de modo que hay
que sustituirlos por &lt; y &gt; respectivamente.

    parametros = replace(replace(parametros, "<", "&lt;"), ">", "&gt;")

    strEnvelope= "<SOAP-ENV:Envelope xmlns:SOAP-
ENV='http://schemas.xmlsoap.org/soap/envelope/' " & _
        "xmlns:xsi='http://www.w3.org/1999/XMLSchema-instance' " & _
        "xmlns:xsd='http://www.w3.org/1999/XMLSchema'> " & _
        "<SOAP-ENV:Body> " & _
        "<ns1:service xmlns:ns1='urn:xmethods-service' " & _
        " SOAP-
ENV:encodingStyle='http://schemas.xmlsoap.org/soap/encoding/'> " & _
        "<codigo xsi:type='xsd:string'>110</codigo> " & _
        "<usuario xsi:type='xsd:string'>MI_USUARIO</usuario> " & _
        "<clave xsi:type='xsd:string'>MI_PASSWORD</clave> " & _
        "<parametros xsi:type='xsd:string'>" + parametros +
"</parametros> " & _
        "</ns1:service> " & _
        "</SOAP-ENV:Body> " & _
        "</SOAP-ENV:Envelope> "

    Set xmlhttp = Server.Createobject("Msxml2.ServerXMLHTTP.4.0")
    xmlhttp.Open "POST",
"http://mdserver/webservices/side_services.php", False
    xmlhttp.setRequestHeader "Accept", "text/xml"
    xmlhttp.setRequestHeader "Cache-Control", "no-cache"
    xmlhttp.setRequestHeader "SOAPAction",
"http://mdserver/webservices/side_services.php?wsdl/service"
    xmlhttp.setRequestHeader "Content-Type", "text/xml; charset=ISO-
8859-1"
    xmlhttp.setRequestHeader "Host", "mdserver"
    On Error Resume Next
    xmlhttp.send strEnvelope

    response.write xmlhttp.ResponseText

    xmlhttp = nothing
%>
```

Ejemplo 2

Este ejemplo accede a los servicios 110 y 400 empleando una librería para utilizar servicios XML mediante SOAP.

```

<%@LANGUAGE="VBSCRIPT" CODEPAGE="1252"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-
1" />
<title>Untitled Document</title>
</head>

<body>
<%
    Dim oSOAP
    Set oSOAP = Server.CreateObject("MSSOAP.SoapClient30")
    oSOAP.ClientProperty("ServerHTTPRequest") = True
    oSOAP.mssoapinit("http://mdserver/webservices/side_services.php?wsd
1")
%>
<form>

<h1>Service 110</h1>
<textarea name="service1" cols="120" rows="3">
<%
    Response.write(oSOAP.service("110", "MI_USUARIO", "MI_PASSWORD", "
") & "<BR>")
%>
</textarea>
-----
<%
    ' Lista de destinos.
    dim parametros
    parametros = "<PARAMETERS zone="" destination=""5""
building_name="" "
    parametros = parametros & "check_in=""2006-06-25"" "
    parametros = parametros & "check_out=""2006-07-03"" "
    parametros = parametros & "start_id=""0"" "
    parametros = parametros & "pag_size=""130"" "
    parametros = parametros & "language=""2"">"
    parametros = parametros & "<ROOM adults=""2"" "
    parametros = parametros & "children=""0"" "
    parametros = parametros & "age0=""0"" "
    parametros = parametros & "age1=""0""/>"
    parametros = parametros & "</PARAMETERS>"

%>
<h1>Service 400</h1>
<textarea name="service" cols="120" rows="50">
<%
    Response.write(oSOAP.service("400", "MI_USUARIO", "MI_PASSWORD",
parametros))
%>
</textarea>
</form>
</body>
</html>

```

Servicios Sidetours

Todos los servicios tienen un código para identificarlo. Se utilizan mediante un “super-servicio” llamado SERVICE que nos permite acceder al servicio concreto a través de éste.

Para hacer uso de dicho “super-servicio”, es imprescindible estar registrado en la base de datos de SIDETOOURS. El registro ofrecerá un **nombre de usuario** y una **contraseña** que acompañarán a todas las peticiones de servicio que se realicen.

Para utilizar un servicio concreto de SIDETOOURS, hay que emplear la operación SERVICE con los siguientes parámetros:

```
$Result = $client->call("SERVICE", array($cod, $user, $pass, $param));
```

Donde:

- \$cod: Código de servicio que se solicita.
- \$user: Identificación del usuario para tener acceso a nuestros servicios.
- \$pass: Clave de acceso.
- \$param: Parámetros necesarios para el servicio. Tomará el valor "" en el caso de que no tenga parámetros.

La codificación de caracteres que se emplea tanto en los parámetros como en la respuesta es ISO-8859-1

En caso de que un usuario no se valide correctamente, el error se devolvería en formato XML siguiendo el siguiente esquema:

```
<!ELEMENT ERROR EMPTY>
<!ATTLIST ERROR
    cod CDATA #REQUIRED
    msg CDATA #REQUIRED
    extra CDATA #IMPLIED >

<!ELEMENT RESPONSE (ERROR)>
<!ATTLIST RESPONSE cod CDATA #REQUIRED>
```


Código 100: Lista de destinos

Devuelve en formato XML los destinos donde trabaja SIDETOURLS que disponen de hoteles para la agencia. Como parámetro se le pasa el país que queremos consultar (ver servicio 103: *Lista de países*). Si no se le pasa nada, devuelve los destinos de España.

Resultado del servicio

La información devuelta para cada destino es su identificador (único en la base de datos) y un nombre descriptivo del destino. Este identificador es el que hay que emplear en cualquier otro servicio que pida un destino como parámetro.

La estructura DTD del resultado se detalla a continuación:

<!ELEMENT RESPONSE (DESTINATION)+>	
<!ATTLIST RESPONSE cod CDATA #REQUIRED>	
<!ELEMENT DESTINATION>	
<!ATTLIST DESTINATION	
id CDATA #REQUIRED	Código de destino.
name CDATA #REQUIRED	Nombre del destino.

Ejemplo

La llamada quedaría tal que así.

```
$Result = $client->call("SERVICE", array("100", $user, $pass, "1"));
```

Donde *100* recordamos que es el código del servicio *Lista de destinos*.

La respuesta guardada en *\$Result* sería muy similar a la que se expone a continuación.

```
<RESPONSE cod="100">
  <DESTINATION id="2" name="Mallorca"/>
  <DESTINATION id="3" name="Menorca"/>
  <DESTINATION id="4" name="Ibiza"/>
  <DESTINATION id="5" name="Costa del Sol y de la Luz"/>
</RESPONSE>
```

Entonces por ejemplo en un servicio donde el destino fuera uno de los parámetros (como en el servicio 400 “disponibilidad básica”), si quisiéramos referenciar a “Mallorca” utilizaríamos el código 2.

Código 101: Lista de zonas por destino

Este servicio devuelve las zonas contenidas en un destino pasado por parámetro al servicio. Dada la sencillez del parámetro no se emplea el encapsulamiento XML para pasarlo, sino que se indica directamente en la llamada el destino que queremos interrogar.

Resultado del servicio

El resultado sí que viene encapsulado en formato XML debido a que no se conoce a priori la cantidad de zonas que puede contener cada destino concreto.

Los valores devueltos para el destino solicitado es una relación de zonas, facilitándose para cada una de ellas su identificador (único en la base de datos) y un nombre descriptivo de esta zona.

Si el código de destino pasado por parámetro no existiera, se devolvería un conjunto de zonas vacío.

El resultado se devuelve atendiendo al siguiente esquema DTD:

```
<!ELEMENT RESPONSE (PARAMETERS,(ZONE)*)>
<!ATTLIST RESPONSE cod CDATA #REQUIRED>

<!ELEMENT PARAMETERS EMPTY>
<!ATTLIST PARAMETERS destination CDATA #REQUIRED>
                                destination = Destino para el que se piden las zonas.

<!ELEMENT ZONE EMPTY>
<!ATTLIST ZONE
    id CDATA #REQUIRED          Código de la zona.
    name CDATA #REQUIRED        Nombre de la zona.
```

Ejemplo

La llamada para conocer las zonas del destino 2 (Mallorca), sería algo así:

```
$xZonas = $client->call("SERVICE", array("101", $user, $pass, "2"));
```

Resultado

Al invocar a este servicio con el código de destino 2 (Mallorca), obtendríamos dentro de la variable `$xZonas` el siguiente XML con las zonas (sólo se han escrito las primeras a título ilustrativo).

```
<RESPONSE cod="101">
  <PARAMETERS destination="2" />
  <ZONE id="1" name="Puig d'en Ros"/>
  <ZONE id="2" name="Son Verí"/>
  <ZONE id="3" name="El Arenal"/>
  <ZONE id="4" name="Playa de Palma"/>
  <ZONE id="5" name="Ca'n Pastilla"/>
  <ZONE id="6" name="Ciudad Jardín"/>
  <ZONE id="7" name="Palma Ciudad"/>
  <ZONE id="8" name="Son Vida"/>
  <ZONE id="9" name="Calamayor"/>
  ...
</RESPONSE>
```

Código 102: Lista de categorías

Muestra todas las categorías aplicables a los establecimientos. El resultado viene desglosado por tipos de establecimiento. Como parámetro se pasa el idioma en el que se quieren los textos.

Los códigos de idioma los relaciono a continuación, aunque para asegurarse debería de acudir al servicio 104.

- language = 2 – Castellano
- language = 3 – Inglés
- language = 4 – Alemán
- language = 5 – Francés

El resultado viene en formato XML con la estructura detallada en el siguiente DTD:

<!ELEMENT RESPONSE (PARAMETERS,(BUILDING_TYPE)*)>	
<!ATTLIST RESPONSE cod CDATA #REQUIRED>	
<!ELEMENT PARAMETERS EMPTY>	
<!ATTLIST PARAMETERS language CDATA #REQUIRED>	
<!ELEMENT BUILDING_TYPE (CATEGORY)*>	
<!ATTLIST BUILDING_TYPE	
id CDATA #REQUIRED	Código del tipo de establecimiento
name CDATA #REQUIRED	Nombre del tipo de establecimiento.
img CDATA #REQUIRED>	URL con una imagen ilustrativa.
<!ELEMENT CATEGORY EMPTY>	
<!ATTLIST CATEGORY	
id CDATA #REQUIRED	Código de la categoría del hotel.
name CDATA #REQUIRED>	Nombre de esta categoría.

Ejemplo

Para conocer las categorías aplicables según el tipo de establecimiento en castellano, haríamos la siguiente llamada:

```
$xxx = $client->call("service", array("102", $user, $pass, "2"));
```

Donde el “2” es el código de idioma para el castellano.

La respuesta sería del siguiente modo:

```
<RESPONSE cod="102">
  <PARAMETERS language="2"/>
  <BUILDING_TYPE id="4" name="Hotel">
    <CATEGORY id="7" name="3 estrellas" img="http://www...com/i3star.gif" />
    <CATEGORY id="8" name="4 estrellas" img="http://www...com/i4star.gif" />
    ...
  </BUILDING_TYPE >
  <BUILDING_TYPE id="5" name="Apartamentos">
    <CATEGORY id="9" name="1 llave" img="http://www...com/s/ilkey.gif" />
    <CATEGORY id="13" name="2 llaves" img="http://www...com/s/ilkey.gif" />
    ...
  </BUILDING_TYPE>
  ...
</RESPONSE>
```

Código 103: Lista de países

Devuelve la relación de todos los países con los que trabaja SIDETOURLS.

Este servicio no tiene parámetros, pues se devuelven siempre todos los países. El resultado está encapsulado en formato XML, ajustándose al siguiente modelo DTD:

<!ELEMENT COUNTRY EMPTY>	
<!ATTLIST COUNTRY	
id CDATA #REQUIRED	Código del país.
name CDATA #REQUIRED	Nombre del país.
short_name CDATA #REQUIRED>	Código de dos letras asociado.
<!ELEMENT RESPONSE (COUNTRY)+>	
<!ATTLIST RESPONSE	
cod CDATA #REQUIRED>	Código del servicio (103)

Ejemplo

La llamada quedaría tal que así.

```
$Result = $client->call("SERVICE", array("103", $user, $pass, ""));
```

Al invocar a este servicio, obtendríamos el siguiente XML. En *id* tenemos los códigos de país que tendríamos que utilizar en los servicios que lo requieran.

```
<RESPONSE cod="103">
  <COUNTRY id="1" name="ESPAÑA" short_name="SP"/>
  <COUNTRY id="4" name="REINO UNIDO" short_name="UK"/>
</RESPONSE>
```

Código 104: Idiomas

Devuelve todos los idiomas con los que trabaja SIDETOURLS.

Este servicio no tiene parámetros. El resultado viene en formato XML siguiendo el siguiente esquema DTD:

<code><!ELEMENT LANGUAGE EMPTY></code>	
<code><!ATTLIST LANGUAGE</code>	
<code>id CDATA #REQUIRED</code>	Código del idioma
<code>name CDATA #REQUIRED</code>	Nombre
<code>short_name CDATA #REQUIRED></code>	Código de 3 letras asociado.
<code><!ELEMENT RESPONSE (LANGUAGE)+></code>	
<code><!ATTLIST RESPONSE</code>	
<code>cod CDATA #REQUIRED></code>	Código del servicio (104)

Ejemplo

La llamada quedaría tal que así.

```
$Result = $client->call("SERVICE", array("104", $user, $pass, ""));
```

Al invocar a este servicio, obtendríamos el siguiente XML. En *id* tenemos los códigos del idioma que tendríamos que utilizar en los servicios que lo requieran.

```
<RESPONSE cod="104">
  <LANGUAGE id="1" name="CASTELLANO" short_name="ESP"/>
  <LANGUAGE id="4" name="INGLÉS" short_name="ENG"/>
  ...
</RESPONSE>
```

Código 105: Regímenes

Un servicio sencillo que devuelve todos los regímenes a los que se podría optar en una habitación. Como parámetro se le pasa el idioma en el que queremos el resultado (ver servicio 104).

La llamada al servicio es:

```
$xxx = $client->call("service", array("105", $user, $pass, $idioma);
```

Y el resultado sería (si se pidieran en castellano):

```
<RESPONSE cod="105">
  <BOARDS>
    <BOARD id="0"
      name="Cualquiera"/>
    <BOARD id="T"
      name="Todo incluido"/>
    <BOARD id="C"
      name="Pensión completa"/>
    <BOARD id="M"
      name="Media pensión"/>
    <BOARD id="D"
      name="Alojamiento y desayuno"/>
    <BOARD id="A"
      name="Solo alojamiento"/>
  </BOARDS>
</RESPONSE>
```

Creo conveniente remarcar que el régimen “cualquiera” se representa con el número cero y no con la letra “O”.

Código 106: Servicios de establecimientos

Devuelve todos los servicios que podría tener un hotel. Como parámetro se pasa el idioma en el que se quiere recibir el resultado.

La estructura del resultado es:

<code><!ELEMENT RESPONSE (SERVICES)></code>	
<code><!ATTLIST RESPONSE cod CDATA #REQUIRED></code>	
<code><!ELEMENT SERVICES (SERVICE)+></code>	
<code><!ATTLIST SERVICES</code>	
<code>count CDATA #REQUIRED</code>	Número de servicios devueltos.
<code>language CDATA #REQUIRED></code>	Idioma en el que se devuelve el resultado.
<code><!ELEMENT SERVICE EMPTY></code>	
<code><!ATTLIST SERVICE</code>	
<code>id CDATA #REQUIRED</code>	Código del servicio
<code>name CDATA #REQUIRED</code>	Nombre del servicio.
<code>type CDATA #REQUIRED></code>	Tipo de Servicio: de habitación, general o de entretenimiento.

Ejemplo

Para obtener en castellano todos los servicios a los que se puede optar en los hoteles en general, haríamos una llamada parecida a esta:

```
$xxx = $client->call("service", array("106", $user, $pass, "2");
```

El resultado sería también parecido al siguiente:

```
<RESPONSE cod="106">
  <SERVICES language="2" count="66">
    <SERVICE id="26" name="Comedor"/>
    <SERVICE id="43" name="Piscina cubierta"/>
    <SERVICE id="47" name="Televisión"/>

    <!--MAS SERVICIOS -->

  </SERVICES>
</RESPONSE>
```

Código 107: Servicios por establecimiento

Devuelve una relación de todos los servicios que ofrece un establecimiento pasado por parámetro.

Los parámetros que necesita el servicio son el código de establecimiento y el lenguaje en el que se quiere el resultado. La estructura DTD de la petición es la siguiente:

```
<!ELEMENT PARAMETERS EMPTY>
<!ATTLIST PARAMETERS
    building_id CDATA #REQUIRED
    language CDATA #REQUIRED>
```

El DTD de la respuesta es el siguiente:

```
<!ELEMENT RESPONSE (SERVICES)>
<!ATTLIST RESPONSE cod CDATA #REQUIRED>

<!ELEMENT SERVICES (SERVICE)+>
<!ATTLIST SERVICES
    building_id CDATA #REQUIRED      Código del establecimiento consultado.
    count CDATA #REQUIRED           Nº de servicios encontrados en el hotel.
    language CDATA #REQUIRED        Código del idioma del resultado.

<!ELEMENT SERVICE EMPTY>
<!ATTLIST SERVICE
    id CDATA #REQUIRED              Código del servicio.
    name CDATA #REQUIRED            Nombre del servicio.
    type (HAB | GEN | ENT) "ENT">  Tipo de Servicio: de habitación , general o
                                    de entretenimiento.
```

Ejemplo

Supongamos que queremos saber los servicios del establecimiento con código 2069. Un código PHP que haría esto posible sería el siguiente:

```
$xParams = "<PARAMETERS building_id=\"2069\" language=\"2\" />";
$xxx = $client->call("service", array("107", $xUser, $xPass, $xParams));
```

Y el XML resultado sería algo parecido al siguiente:

```
<RESPONSE cod="107">
    <SERVICES building_id="2069" language="2" count="31">
        <SERVICE id="26" name="Comedor" type="GEN"/>
        <SERVICE id="47" name="Televisión" type="HAB "/>
        <SERVICE id="64" name="Buffet" type="GEN"/>
        <SERVICE id="50" name="Air. Acondicionado" type="HAB "/>
        <SERVICE id="12" name="Alquiler de coches" type="ENT "/>

        <!-- Más servicios -->

    </SERVICES>
</RESPONSE>
```


Código 108: Provincias

Devuelve la relación de todas las provincias donde SIDETOURLS tiene hoteles.

Este servicio no tiene parámetros. El resultado está encapsulado en formato XML, ajustándose al siguiente modelo DTD:

<!ELEMENT PROVINCE EMPTY>	
<!ATTLIST PROVINCE	
id CDATA #REQUIRED	Código de provincia.
name CDATA #REQUIRED>	Nombre de la provincia .
<!ELEMENT RESPONSE (PROVINCE)+>	
<!ATTLIST RESPONSE	
cod CDATA #REQUIRED>	Código del servicio (108)

Ejemplo

La llamada quedaría tal que así.

```
$Result = $client->call("SERVICE", array("108", $user, $pass, ""));
```

Al invocar a este servicio, obtendríamos el siguiente XML. En *id* tenemos los códigos de provincia que tendríamos que utilizar en los servicios que lo requieran.

```
<RESPONSE cod="108">
  <PROVINCE id="55" name="Algarve - Portugal"/>
  <PROVINCE id="3" name="Alicante"/>
  <PROVINCE id="9" name="Barcelona"/>
  <PROVINCE id="12" name="Cádiz"/>
  <PROVINCE id="21" name="Granada"/>
  <PROVINCE id="24" name="Huelva"/>
  <PROVINCE id="8" name="Islas Baleares"/>
  <PROVINCE id="31" name="Málaga"/>
  <PROVINCE id="33" name="Murcia"/>
  <PROVINCE id="48" name="Valencia"/>
</RESPONSE>
```

Código 110: Destinos y zonas

Devuelve una relación de todos los destinos con sus zonas. Se podría conseguir el mismo resultado mediante los servicios 100 (destinos) y 101 (zonas por destino), pero con este servicio se consigue todo de un solo acceso a la base de datos, consiguiendo en general más velocidad.

El servicio no tiene parámetros.

La estructura del retorno se representa mediante el siguiente DTD:

<!ELEMENT RESPONSE (DESTINATION)+>	
<!-- ATTLIST RESPONSE cod CDATA #REQUIRED -->	
<!ELEMENT DESTINATION (ZONE)+>	
<!-- ATTLIST DESTINATION -->	
id CDATA #REQUIRED	Código del destino turístico.
name CDATA #REQUIRED	Nombre del destino.
id_country CDATA #REQUIRED	Código de país al que pertenece el destino.
nzones CDATA #REQUIRED	Zonas encontradas en este destino.
<!-- ELEMENT ZONE EMPTY -->	
<!-- ATTLIST ZONE -->	
id CDATA #REQUIRED	Código de la zona turística.
name CDATA #REQUIRED	Nombre de la zona turística.

Ejemplo

```
$xxx = $client->call("service", array("110", $user, $pass, ""));
```

El resultado sería algo tal que así:

```
<RESPONSE cod="110">
  <DESTINATION id="9" name="Costa de Almeria" id_country="1" nzones="9">
    <ZONE id="240" name="Adra"/>
    <ZONE id="243" name="Aguadulce"/>
    <ZONE id="244" name="Almeria"/>
    <ZONE id="241" name="Almerimar"/>
    <ZONE id="245" name="El Toyo"/>
    <ZONE id="255" name="La Envia"/>
    <ZONE id="249" name="Mojacar"/>
    <!-- MAS ZONAS -->
  </DESTINATION>
  <DESTINATION id="6" name="Costa de la Luz" id_country="1" nzones="8">
    <ZONE id="205" name="Algeciras"/>
    <ZONE id="211" name="Cadiz"/>
    <ZONE id="209" name="Chiclana"/>
    <ZONE id="215" name="Chipiona"/>
    <ZONE id="208" name="Conil"/>
  </DESTINATION>
  <!-- MORE DESTINATIONS -->
</RESPONSE>
```

Código 111: Tipos de turismo

Devuelve los tipos de turismo que puede albergar un hotel.

El único parámetro de entrada es el idioma del resultado (ver servicio 104).

Los valores devueltos siguen el siguiente DTD:

```
<!ELEMENT RESPONSE (TOURISM_TYPE)+>
<!ATTLIST RESPONSE cod CDATA #REQUIRED>

<!ELEMENT TOURISM_TYPE EMPTY>
<!ATTLIST TOURISM_TYPE
  id CDATA #REQUIRED
  name CDATA #REQUIRED>
```

Ejemplo

```
$Result = $client->call("SERVICE", array("111", $user, $pass, "2"));
```

Resultado:

```
<RESPONSE cod="111">
  <TOURISM_TYPE id="C" name="Ciudad"/>
  <TOURISM_TYPE id="I" name="Interior"/>
  <TOURISM_TYPE id="M" name="Montaña"/>
  <TOURISM_TYPE id="B" name="Playa"/>
  <TOURISM_TYPE id="R" name="Rural"/>
</RESPONSE>
```

Código 112: Tipos de acomodación

Devuelve todos los tipos de acomodación que contempla Sidetours. Todas las acomodaciones de los hoteles tienen un tipo de acomodación.

Los parámetros de entrada se pasan en formato XML (actualmente el único atributo es el idioma del resultado). La estructura de la petición es la que representa el siguiente DTD.

<pre><!ELEMENT PARAMETERS EMPTY> <!ATTLIST PARAMETERS language CDATA #REQUIRED></pre>	Código del idioma del resultado.
---	----------------------------------

Los valores devueltos siguen el siguiente DTD:

<pre><!ELEMENT RESPONSE (ACCOMODATION_TYPE)+> <!ATTLIST RESPONSE cod CDATA #REQUIRED> <!ELEMENT ACCOMODATION_TYPE EMPTY> <!ATTLIST ACCOMODATION_TYPE id CDATA #REQUIRED name CDATA #REQUIRED></pre>
--

Ejemplo de petición

```
$Result = $client->call("SERVICE", array("112", $user, $pass,
    "<PARAMETERS language=\"2\" />")));
```

Resultado

```
<RESPONSE cod="112">
  <ACCOMODATION_TYPE id="475" name="Abuhardillada"/>
  <ACCOMODATION_TYPE id="476" name="Apartamento"/>
  <ACCOMODATION_TYPE id="477" name="Apartamento 4/5 pax"/>
  <ACCOMODATION_TYPE id="478" name="Apartamento 1 Dormitorio"/>
  <ACCOMODATION_TYPE id="479" name="Apartamento 1 Dormitorio 2/4
pax"/>
  <ACCOMODATION_TYPE id="480" name="Apartamento 1 Dormitorio 3/4
pax"/>
  <ACCOMODATION_TYPE id="481" name="Apartamento 1 Dormitorio
exterior"/>
  <ACCOMODATION_TYPE id="482" name="Apartamento 1 Dormitorio
interior"/>
  <ACCOMODATION_TYPE id="483" name="Apartamento 1 Dormitorio vista
mar"/>
  . . .
</RESPONSE>
```

Código 200: Detalles de un establecimiento

Muestra los detalles al completo de un edificio concreto pasado por parámetro en un idioma también pasado por parámetro.

La condición para poder acceder a este edificio es que trabaje con un mercado igual con el que trabaja la agencia.

Los parámetros de entrada se pasan en formato XML. La estructura de la petición es la que representa el siguiente DTD.

<!ELEMENT PARAMETERS EMPTY>	
<!ATTLIST PARAMETERS	
language CDATA #REQUIRED	Código del idioma del resultado.
building_id CDATA #REQUIRED>	Código del establecimiento del que queremos detalles.

Resultado del servicio

Este resultado también está encapsulado en un XML con la siguiente estructura:

<!ELEMENT RESPONSE (PARAMETERS, BUILDING)>	
<!ATTLIST RESPONSE cod CDATA #REQUIRED>	
<!ELEMENT PARAMETERS EMPTY>	
<!ATTLIST PARAMETERS	
language CDATA #REQUIRED	Código del idioma (ver servicio 104)
building_id CDATA #REQUIRED>	Código del establecimiento.
<!ELEMENT BUILDING (DESCRIPTION, SERVICES, CHARACTERISTICS, ACOMODATIONS, PHOTOS)>	
<!ATTLIST BUILDING	
id CDATA #REQUIRED	Código del establecimiento.
building_type CDATA #REQUIRED	Código del tipo de establecimiento.
building_type_name CDATA #REQUIRED	Nombre del tipo.
destination CDATA #REQUIRED	Código de destino turístico.
destination_name CDATA #REQUIRED	Nombre del destino turístico.
country CDATA #REQUIRED	Código de país.
country_name CDATA #REQUIRED	Nombre del país.
zone CDATA #REQUIRED	Código de la zona.
zone_name CDATA #REQUIRED	Nombre de la zona.
category CDATA #REQUIRED	Código de categoría del establecimiento.
category_name CDATA #REQUIRED	Nombre de la categoría.
category_img CDATA #REQUIRED	URL de una imagen ilustrativa de la categ.
name CDATA #REQUIRED	Nombre del establecimiento.
address CDATA #REQUIRED	Dirección.
postcode CDATA #REQUIRED	Código postal.
town CDATA #REQUIRED	Población.
city CDATA #REQUIRED	Ciudad.
telephone1 CDATA #REQUIRED	Primer teléfono de contacto.
telephone2 CDATA #REQUIRED	Segundo teléfono de contacto.
fax CDATA #REQUIRED	Fax de contacto.
email CDATA #REQUIRED	Correo electrónico de contacto.
latitude CDATA #REQUIRED	Latitud geográfica para google maps.
longitude CDATA #REQUIRED	Longitud geográfica para google maps.
url CDATA #REQUIRED>	Página web del establecimiento.

<!ELEMENT DESCRIPTION (#PCDATA)>	Descripción larga del hotel.
<!ELEMENT SERVICES (SERVICE)*>	Lista de servicios que ofrece.
<!ATTLIST SERVICES count CDATA #REQUIRED>	Número de servicios encontrados.
<!ELEMENT SERVICE EMPTY>	
<!ATTLIST SERVICE id CDATA #REQUIRED type (HAB ENT GEN) "HAB" name CDATA #REQUIRED>	Código de servicio dado por el hotel. Habitación, Entretenimiento o General. Nombre del servicio.
<!ELEMENT CHARACTERISTICS (CHARACTERISTIC)*>	Características del hotel.
<!ELEMENT CHARACTERISTIC EMPTY>	
<!ATTLIST CHARACTERISTIC id CDATA #REQUIRED type (CAP DIS EDI) "CAP" name CDATA #REQUIRED value CDATA #REQUIRED>	Código de la característica. Capacidad, Distancia o Edificio. Nombre. Valor (ej: si es distancia, pues nº de Km)
<!ELEMENT ACOMODATIONS (ACOMODATION)*>	Lista de acomodaciones.
<!ATTLIST ACOMODATIONS count CDATA #REQUIRED>	Nº de acomodaciones.
<!ELEMENT ACOMODATION count>	
<!ATTLIST ACOMODATION id CDATA #REQUIRED name CDATA #REQUIRED >	Código de acomodación. Nombre de la acomodación.
<!ELEMENT PHOTOS (PHOTO)*>	Conjunto de fotografías.
<!ATTLIST PHOTOS count CDATA #REQUIRED>	
<!ELEMENT PHOTO EMPTY>	
<!ATTLIST PHOTO url CDATA #REQUIRED title CDATA #REQUIRED description CDATA #REQUIRED main (1 0) "0" >	Localización de la fotografía. Título de la foto. Descripción de la foto. 1: Principal, 0: normal

Errores

En este servicio se contemplan los siguientes errores:

- Código 201: No se ha pasado el parámetro del idioma.
- Código 202: No se ha pasado el parámetro del código de edificio.
- Código 203: El idioma no está definido en la base de datos.
- Código 204: El hotel no está definido o no tiene contrato.

Ejemplo

Si quisiéramos los detalles del edificio 1965 en castellano, haríamos la siguiente petición:

```
$xxx = $client->call("service", array("200", $user, $pass,
    "<PARAMETERS language=\"2\" building_id=\"1965\"/>"));
```

De donde obtendríamos el siguiente XML resultado:

```
<RESPONSE cod="200">
  <PARAMETERS language="2" building_id="2345"/>
  <BUILDING id="2345" building_type="5"
    building_type_name="Apartamentos" category="13"
    category_name="2 llaves"
    category_img="20041125133654_2key.gif"
    name="Golf Center"
    address="Paseo Transversal"
    postcode="04741"
    town="0"
    city="Croquetas de Mar"
    telephone1=""
    telephone2=""
    fax=""
    email=""
    latitude=""
    longitude=""
    url="">
    <DESCRIPTION>Situados en un enclave privilegiado de la
Urbanización de Playa Serena de Roquetas de Mar, en el centro de un
espléndido Campo de Golf de 18 hoyos, a 500 metros de la playa, zona
comercial y nocturna.
```

Este complejo al estilo de un pueblo andaluz dispone de dos grandes piscinas-lago con zona infantil y solarium. Un área social con bar-cafetería, restaurante y salón TV-SAT, recepción, teléfono público, aparcamiento privado y vigilancia nocturna.

Los apartamentos son de uno y dos dormitorios y disponen de cuarto de baño, cocina totalmente equipada, salón-comedor, con sofá-cama nido para dos personas, terraza y TV-SAT.

Todos los apartamentos tienen vista al campo de golf o piscina y jardines.

Otros complementos: caja fuerte en el apartamento, calefacción. Tumbonas y parasoles en la piscina.</DESCRIPTION>

```
<SERVICES count="29">
  <SERVICE id="29" name="Sala TV-Sat"/>
  <SERVICE id="67" name="Televisión vía satélite"/>
  <SERVICE id="7" name="Cambio de divisias"/>
  <SERVICE id="42" name="Piscina aire libre"/>
  ...
</SERVICES>
<CHARACTERISTICS count="12">
<CHARACTERISTIC id="3" name="Aeropuerto" value="30 km"/>
<CHARACTERISTIC id="13" name="Apartamentos" value="150"/>
  ...
</CHARACTERISTICS>
<ACOMODATIONS count="1">
  <ACOMODATION id="5597" name="Apartamento"/>
</ACOMODATIONS>
<PHOTOS count="5">
  <PHOTO url="http://... /20060120230704_golf5.jpg"
    title="Alrededores" description=""/>
  <PHOTO url="http://... /20060120230619_golf4.jpg"
    title="Apartamento" description=""/>
```

```
    . . .  
    </PHOTOS>  
  </BUILDING>  
</RESPONSE>
```


Código 201: Lista de establecimientos

Ofrece una relación de todos los establecimientos a los que puede acceder la agencia. Tiene la opción de filtrar los resultados, mostrando por ejemplo sólo los de un destino concreto, una categoría concreta, etc.

Como parámetro se le pasa la condición del filtro en formato XML. Se pasará la cadena vacía en el caso de que queramos todos los edificios.

```
$R = $client->call("SERVICE", array("201", $user, $pass, $xml_filtro));
```

Nos fijamos en la definición recursiva de la estructura del filtro. Esto permite tener condiciones anidadas:

<!ELEMENT OR ((OR)*, (AND)*, (FILTER)*)>	
<!ELEMENT AND ((OR)*, (AND)*, (FILTER)*)>	
<!ELEMENT FILTER EMPTY>	
<!ATTLIST FILTER	
field CDATA #REQUIRED	Columna que queremos filtrar.
op (EQ DF LT GT) "EQ"	Operación que queremos realizar.
value CDATA #REQUIRED>	Valor con el que comparamos la columna.
<!ELEMENT PARAMETERS ((OR)*, (AND)*, (FILTER)*)>	
<!ATTLIST PARAMETERS	
language CDATA #REQUIRED>	Código del idioma del resultado.

Con más detalle tenemos que:

- *field* es la columna que queremos filtrar y puede tomar los valores
 - category: Categoría del edificio.
 - zone: Zona donde se encuentra el edificio.
 - destination: Destino donde se encuentra el edificio.
 - building_type: Tipo de edificio.
 - last_updated: Fecha de última modificación en formato yyyy-mm-dd. Si se especifica se devolverán los hoteles que se hayan modificado a partir de este valor.
 - building_id: Para cuando queramos buscar información de un edificio concreto.
- *op*: Indica cómo vamos a comparar los campos. Puede tomar los valores:
 - EQ: Para comparaciones tipo “igual”
 - DF: Para comparaciones tipo “diferente”
 - LT: Para comparaciones tipo “menor que”
 - GT: Para comparaciones tipo “mayor que”

El esquema de la respuesta sigue el siguiente esquema DTD:

<!ELEMENT BUILDING EMPTY>	
<!ATTLIST BUILDING	
id CDATA #REQUIRED	Código del establecimiento.
name CDATA #REQUIRED	Nombre.

building_type CDATA #REQUIRED	Tipo de establecimiento.
destination CDATA #REQUIRED	Código del destino turístico.
zone CDATA #REQUIRED	Código de la zona turística.
text CDATA #REQUIRED	Texto corto descriptivo del hotel.
photo CDATA #REQUIRED	Foto principal del edificio.
category CDATA #REQUIRED	Código de la categoría del edificio.
category_name CDATA #REQUIRED	Nombre de la categoría.
category_img CDATA #REQUIRED >	Imagen ilustrativa de la categoría.
<!ELEMENT BUILDING_FILTER (BUILDING)+>	
<!ATTLIST BUILDING_FILTER	
language CDATA #REQUIRED	Código del idioma del resultado.
count CDATA #REQUIRED >	Número de resultados encontrados.
<!ELEMENT RESPONSE (BUILDING_FILTER)>	
<!ATTLIST RESPONSE cod CDATA #REQUIRED>	

Ejemplos

Ejemplos de expresiones válidas son:

Para representar:

```
(tipo_edificio = 1) o (tipo_edificio = 2)
      y
(categoría = 3) o (categoría = 4)
```

escribiremos:

```
<PARAMETERS language="2">
  <OR>
    <FILTER field="building_type" op="EQ" value="1" />
    <FILTER field="building_type" op="EQ" value="2" />
  </OR>
  <OR>
    <FILTER field="category " op="EQ" value="3" />
    <FILTER field="category " op="EQ" value="4" />
  </OR>
</PARAMETERS>
```

Para representar:

```
(tipo_edificio = 1)
      o
(zona = 2) y (destino = 2)
```

escribiremos:

```
<PARAMETERS language="2">
  <OR>
    <FILTER field="tipo_edificio" op="EQ" value="1" />
    <AND>
      <FILTER field="zona" op="EQ" value="2" />
      <FILTER field="destino" op="EQ" value="2" />
    </AND>
  </OR>
</PARAMETERS>
```

Un ejemplo completo de llamada a este servicio es:

```
$condicion = "<PARAMETERS language=\"2\">".
"<OR>".
"<FILTER field=\"building_type\" op=\"EQ\" value=\"3\" />".
"<FILTER field=\"building_type\" op=\"EQ\" value=\"4\" />".
"</OR>".
```

```
"<OR>".
"<FILTER field=\"destination\" op=\"EQ\" value=\"5\" />".
"<FILTER field=\"destination\" op=\"EQ\" value=\"6\" />".
"</OR>".
"<FILTER field=\"category\" op=\"EQ\" value=\"8\" />".
"</PARAMETERS>";

$xxx=$client->call("service", array("201", $user, $pass, $condicion);
```

Y un posible resultado sería:

```
<RESPONSE cod="201">
  <BUILDING_FILTER>
    <BUILDING id="1125" name="ESTABLECIMIENTO 1"
      building_type="3" destination="5" zone="189" category="8"
      category_name="3 estrellas"
      category_img="http://... /3_estrellas.gif" />

    <BUILDING id="2529" name=" ESTABLECIMIENTO 2"
      building_type="4" destination="5" zone="189" category="8"
      category_name="3 estrellas"
      category_img="http://... /3_estrellas.gif" />

    <BUILDING id="2731" name=" ESTABLECIMIENTO 3"
      building_type="4" destination="6" zone="189" category="8"
      category_name="3 estrellas"
      category_img="http://... /3_estrellas.gif" />

    ...

  </BUILDING_FILTER>
</RESPONSE>
```

Código 202: Catálogo de establecimientos

Devuelve información básica de todos establecimientos que encajan en un patrón determinado. Es una versión simplificada del servicio 201 *Lista de establecimientos*.

El DTD de la petición es:

<!ELEMENT CATALOGUE (SERVICE)*>	Establecimientos con servicios concretos.
<!ATTLIST CATALOGUE	
building_type CDATA ""	Código de tipo de edificio deseado.
category CDATA ""	Código de categoría de edificio deseada.
destination CDATA ""	Código del destino turístico donde queremos establecimientos
zone CDATA "">	Código de zona turística.
province CDATA "">	Código de la provincia.
<!ELEMENT SERVICE EMPTY>	
<!ATTLIST SERVICE	Código de servicio que deseamos que tenga el establecimiento (ver servicio 106).
id CDATA #REQUIRED>	

El resultado está estructurado según el siguiente DTD:

<!ELEMENT RESPONSE (CATALOGUE)>	
<!ATTLIST RESPONSE cod CDATA #REQUIRED>	
<!ELEMENT CATALOGUE (BUILDING)+>	
<!ATTLIST CATALOGUE	
count CDATA #REQUIRED	Número de establecimientos encontrados.
language CDATA #REQUIRED>	Código del idioma del resultado.
<!ELEMENT BUILDING EMPTY>	
<!ATTLIST BUILDING	
building_type CDATA #REQUIRED	Código del tipo de establecimiento.
category CDATA #REQUIRED	Código de su categoría.
category_img CDATA #REQUIRED	URL de la imagen de la categoría.
category_name CDATA #REQUIRED	Nombre de la categoría.
destination CDATA #REQUIRED	Código de destino turístico.
destination_name CDATA #REQUIRED	Nombre del destino turístico.
id CDATA #REQUIRED	Código del establecimiento.
name CDATA #REQUIRED	Nombre del establecimiento.
photo CDATA #REQUIRED	URL de una foto de portada del hotel.
text CDATA #REQUIRED	Breve descripción del hotel.
zone CDATA #REQUIRED	Código de zona turística.
zone_name CDATA #REQUIRED>	Nombre de la zona turística.
province CDATA #REQUIRED>	Código de la provincia
provincename CDATA #REQUIRED>	Nombre de la provincia

Ejemplo

Vamos a averiguar los establecimientos que se encuentran en Costa del Sol y disponen de los servicios 12 (Alquiler de coches) y 7 (cambio de divisas).

```
$xParams      =      "<PARAMETERS      language=\"2\"      destination=\"5\"
province=\"8\">
    <SERVICE id=\"12\" />
    <SERVICE id=\"7\" />
</PARAMETERS>";
```

```
$xxx = $client->call("service", array("202", $xUser, $xPass, $xParams));
```

El resultado sería algo parecido a esto:

```
<RESPONSE cod="202">
  <CATALOGUE language="2" count="108">
    <BUILDING id="2566" name="ACUARIUM" building_type="4"
destination="5" destination_name="Costa del Sol"
zone="16" zone_name="Fuengirola"
text="El hotel está situado en la turística localidad de Fuengirola, a
unos minutos de la playa y del centro de la localidad. En los
alrededores podrá encontrar varias zonas con tiendas, boutiques,
bares, r..."
photo="http://www.sidetours.com/.../2000517_photo1.jpg"
category="7" category_name="3 estrellas"
category_img="http://www.sidetours.com/.../3star.gif"/>

    <BUILDING id="2338" name="AGH ESTEPONA" building_type="4"
destination="5" destination_name="Costa del Sol"
zone="Costa del Sol" zone_name="Estepona"
text="el hotel agh estepona, se encuentra situado en la urbanización
paraiso beach, en la carretera n-340, en el km 166,5; a tan solo 150
mts. De la playa, junto a marbella y a 10 minutos de puert banus.
E..."
photo="http://www.sidetours.com/.../53104-photo1.jpg"
category="18" category_name="4 estrellas SUP"
category_img="http://www.sidetours.com/... /20045_4starplus.gif"/
province="8" provincename="Islas Baleares">

    <!-- MAS EDIFICIOS -->
  </CATALOGUE>
</RESPONSE>
```

Código 500: Reserva básica

Permite reservar en un establecimiento un conjunto de habitaciones tanto en cupo como en precio.

El formato de la petición es:

<!ELEMENT ROOM ANY>	
<!ATTLIST ROOM	
adults CDATA #REQUIRED	Número de adultos en la habitación.
children CDATA #REQUIRED	Número de niños.
age0 CDATA "	Edad del primer niño si lo hay.
age1 CDATA "	Edad del segundo niño si lo hay.
age2 CDATA "	Edad del tercer niño si lo hay.
age4 CDATA "	Edad del cuarto niño si lo hay.
contact CDATA #REQUIRED	Persona de contacto (responsable)
aco CDATA #REQUIRED>	Código de acomodación (dado por dispo).
<!ELEMENT PARAMETERS (#PCDATA ROOM) *>	
<!ATTLIST PARAMETERS	
language CDATA #REQUIRED	Código del idioma del resultado.
check_in CDATA #REQUIRED	Fecha entrada en format yyyy-mm-dd
check_out CDATA #REQUIRED	Fecha de salida en formato yyyy-mm-dd
building_id CDATA #REQUIRED	Código de edificio.
file CDATA #REQUIRED	Número de expediente.
board_id CDATA #REQUIRED	Código de régimen (servicio 105)
no_booking CDATA "">	Opcional. Si se especifica no se reserva.

Explicamos con más detalle alguno de los atributos:

- no_booking: Es opcional. Si no pone, se hace la reserva tal cual, se envía al establecimiento y se mete en la cola de reservas para facturar. Si se especifica, se hace la reserva igualmente y se devuelve el mismo resultado, solo que el resto de procesos se quedan en pausa hasta que se confirma que realmente se desea esta reserva. El tiempo máximo que se guarda es de 20 minutos. Los servicios para confirmar la reserva es el 504 o el 1504.
- file: La longitud máxima del expediente es de 20 caracteres.

La respuesta depende de si ha habido éxito en la reserva. En caso de error se devuelve un XML con la descripción de éste. En caso de que vaya bien, se devuelve un XML con la información resumen de la reserva.

Respuesta en caso de error

Los errores contemplados son:

- Código 500: XML mal formado. Se da este error cuando el XML enviado por la agencia para realizar la petición de búsqueda de disponibilidad no está bien formado según las normas XML.
- Código 500-1: Fecha no válida – O no se ha pasado o tiene el formato incorrecto. El formato adecuado de una fecha es YYYY-MM-DD. Ejemplos de fechas correctas son: 2007-12-20, 2006-03-24, etc.

- Código 500-2: Error en parámetros. No se ha introducido el edificio en el atributo *building_id* de la etiqueta *PARAMETERS*.
- Código 500-3: Error en parámetros. No se ha introducido la acomodación en el atributo *aco* de la etiqueta *ROOM* en alguna de las habitaciones.
- Código 500-4: Error en parámetros. No se ha introducido la edad del niño 1 en el atributo *age0* de la etiqueta *ROOM* en alguna de las habitaciones.
- Código 500-5: Error en parámetros. No se ha introducido la edad del niño 2 en el atributo *age1* de la etiqueta *ROOM* en alguna de las habitaciones.
- Código 500-6: Error en parámetros. No se ha introducido el responsable de la habitación en el atributo *contact* de la etiqueta *ROOM* en alguna de las habitaciones.
- Código 500-7: Error en reserva. Si la agencia no admite reserva directa, ha de utilizar el servicio de pre-reserva.
- Código del 500-10 al 500-12: No hay disponibilidad.
- Código 500-13 en adelante: Errores de bases de datos y otros conceptos, propios de Sidetours, cuyos códigos y mensajes son útiles para la localización rápida de dichos errores.

El formato de la respuesta será el siguiente:

```
<ERROR cod="500" msg="Error en reserva" extra="mensaje adicional" />
```

Respuesta en caso de éxito

```
<!ELEMENT RESERVATION (NOTES, (ROOM)+)>
<!ATTLIST RESERVATION
    board_id CDATA #REQUIRED          Código de regimen.
    building_id CDATA #REQUIRED        Código del edificio.
    check_in CDATA #REQUIRED           Fecha de entrada al establecimiento.
    check_out CDATA #REQUIRED          Fecha de salida.
    data CDATA #REQUIRED               Fecha de la reserva
    file CDATA #REQUIRED               Expediente de la agencia.
    locata CDATA #REQUIRED             Localizador de SIDETOURLS
    nrooms CDATA #REQUIRED             Número de habitaciones.
    price CDATA #REQUIRED              Precio neto para la agencia.
    email (OK | NOK) "OK">           OK: Si se envió el e-mail a la agencia.
                                     NOK: en caso contrario.

<!ELEMENT ROOM ((PERIOD)+, (OFFER)*, (SUPPLEMENT)*)>
<!ATTLIST ROOM
    contact CDATA #REQUIRED            Persona de contacto de la habitación.
    adults CDATA #REQUIRED             Número de adultos.
    children CDATA #REQUIRED           Número de niños.
    age0 CDATA " "                    Edad del primer niño (si lo hay).
    age1 CDATA " "                    Edad del segundo niño (si lo hay).
    age2 CDATA " "                    Edad del tercer niño (si lo hay).
    age3 CDATA " "                    Edad del cuarto niño (si lo hay).
    price CDATA #REQUIRED>           Precio total de la habitación.
```

<!ELEMENT PERIOD EMPTY>	
<!ATTLIST PERIOD	
id CDATA #REQUIRED	Código de periodo.
price CDATA #REQUIRED	Precio total del periodo.
days CDATA #REQUIRED	Días que coge este periodo.
adult1 CDATA "	Precio del primer adulto.
adult2 CDATA "	Precio del segundo adulto.
adult3 CDATA "	Precio del tercer adulto.
adult4 CDATA "	Precio del cuarto adulto en adelante.
child1 CDATA "	Precio del primer niño.
child2 CDATA "	Precio del segundo niño.
child3 CDATA "	Precio del tercer niño.
child4 CDATA ">	Precio del cuarto niño.
<!ELEMENT OFFER EMPTY>	
<!ATTLIST OFFER	
id CDATA #REQUIRED	Código de la oferta.
type CDATA #REQUIRED	Tipo de oferta.
name CDATA #REQUIRED	Nombre de la oferta.
title CDATA #REQUIRED	Título de la oferta.
price CDATA #REQUIRED>	Precio de la oferta.
<!ELEMENT SUPLEMENT EMPTY>	
<!ATTLIST SUPLEMENT	
id CDATA #REQUIRED	Código del suplemento.
name CDATA #REQUIRED	Nombre del suplemento.
price CDATA #REQUIRED>	Precio del suplemento.

Ejemplo

Supongamos que en la variable *\$params* se almacena el siguiente XML para reservar del 26 de Junio al 11 de Julio de 2006 dos habitaciones en pensión completa.

```
$params = "<PARAMETERS language=\"2\" building_id=\"2069\"
check_in=\"2006-06-26\" check_out=\"2006-07-11\"
file=\"exp_023\" board_id=\"C\">
<ROOM adults=\"2\" children=\"0\" age0=\"0\" age1=\"0\"
contact=\"Gato con botas\" aco=\"5051\"/>
<ROOM adults=\"1\" children=\"0\" age0=\"0\" age1=\"0\"
contact=\"Gato con botas II\" aco=\"5051\"/>
Comentario. Es importante que no hayan líneas en blanco ni
tabuladores, pues esto se graba todo todo todo...
</PARAMETERS>";
```

Entonces la llamada a este servicio sería algo así:

```
$Result = $client->call("SERVICE", array("500", $user, $pass, $params));
```

Y la respuesta del servicio quedaría de esta forma:

```
<RESERVATION locata="011414" file="expediente" check_in="2006-06-26"
check_out="2006-07-11" nrooms="2" data="2006-06-05"
price="1456.75">
<ROOM adults="2" children="0" age0="0" age1="0" price="734.50">
<PERIOD id="29680" price="196.50" days="5"
adult1="98.25" adult2="98.25"/>
<PERIOD id="29681" price="538.00" days="10"
adult1="269.00" adult2="269.00"/>
</ROOM>
<ROOM adults="1" children="0" age0="0" age1="0" price="722.25">
```



```
<PERIOD id="29680" price="98.25"
      days="5" adult1="98.25"/>
<PERIOD id="29681" price="269.00"
      days="10" adult1="269.00"/>
<SUPPLEMENT id="3056" name="SUP INDIVIDUAL" price="95.00"/>
<SUPPLEMENT id="3057" name="SUP INDIVIDUAL" price="260.00"/>
</ROOM>
</RESERVATION>
```

Código 501: Catálogo de reservas

Permite averiguar un conjunto de reservas de una agencia.

Los parámetros de entrada se facilitan en XML mediante la siguiente estructura DTD:

<!ELEMENT PARAMETERS EMPTY>	
<!ATTLIST PARAMETERS	
building_name CDATA ""	Nombre del establecimiento.
building_id CDATA ""	Id del establecimiento.
check_in CDATA ""	Fecha de entrada al hotel: yyyy-mm-dd
check_out CDATA ""	Fecha de salida al hotel: yyyy-mm-dd
date_from CDATA ""	Reserva hecha a partir de "date_from"
date_to CDATA ""	Reserva hecha no más tarde de "date_to"
destination CDATA ""	Código de destino geográfico.
zone CDATA ""	Código de zona geográfica.
language CDATA ""	Idioma en el que queremos el resultado.
locata CDATA ""	Localizador de reserva.
file CDATA ""	Expediente de la agencia.
npax CDATA ""	Número de personas.
state (0 2 3) "0"	0: Pendiente no cancelada. 2: Cancelada, 3: No cancelada ya pasada.
source (WEB BCK XML) "WEB">	Origen de la reserva.

Donde:

- Todas las fechas tienen el formato yyyy-mm-dd.
- building_name = Nombre del establecimiento. Pueden utilizarse los caracteres comodines "%" para la expresión de este nombre.
- check_in: Fecha de entrada de referencia en formato yyyy-mm-dd. Se cogerán las reservas con fecha de entrada posterior a la indicada en este parámetro. Si no se indica, se devuelven por defecto las reservas hechas a partir del día actual.
- check_out: Fecha de salida de referencia en formato yyyy-mm-dd. Se cogerán las reservas con fecha de salida anterior a la indicada en este parámetro.
- locata: Localizador de la reserva. Hay que tener en cuenta que el localizador de una reserva es único, de modo que dando un valor a este parámetro la longitud del resultado será como mucho de un registro.
- npax: Número de personas que hubieron en la reserva. Se cogerán las reservas con al menos el número de personas especificadas en este parámetro.
- source: XML son las hechas a través de los webservices. WEB son las hechas a través de nuestra web <http://www.sidetours.com>. BCK son las hechas por Sidetours bajo petición del cliente (reservas especiales por ejemplo).

La respuesta será la información de un conjunto de reservas. Esta información viene según el siguiente DTD:

<!ELEMENT RESPONSE (RESERVATIONS)>	
<!ATTLIST RESPONSE cod CDATA #REQUIRED>	
<!ELEMENT RESERVATIONS (RESERVATION)+>	
<!ATTLIST RESERVATIONS	

count CDATA #REQUIRED>	Número de reservas encontradas.
<!ELEMENT RESERVATION EMPTY>	
<!ATTLIST RESERVATION	
building_category CDATA #REQUIRED	Categoría del establecimiento.
building_category_img CDATA #REQUIRED	Imagen asociada a la categoría.
building_category_name CDATA #REQUIRED	Nombre de la categoría.
building_id CDATA #REQUIRED	Código del establecimiento.
building_name CDATA #REQUIRED	Nombre del establecimiento
check_in CDATA #REQUIRED	Fecha en la que se entró al hotel.
check_out CDATA #REQUIRED	Fecha en la que se salió del hotel.
date CDATA #REQUIRED	Fecha en la que se hizo la reserva.
locata CDATA #REQUIRED	Localizador de la reserva.
file CDATA "	Expediente de la agencia.
npax CDATA #REQUIRED	Paxes que protagonizaron la reserva.
price CDATA #REQUIRED	Precio neto de la reserva.
state (0 2 3) "0"	0: Pendiente no cancelada. 2: Cancelada, 3: No cancelada ya pasada.
source (WEB BCK XML) "WEB">	Origen de la reserva.

Ejemplo:

Para ver todas las reservas hechas en el destino turístico Costa del Sol, haremos lo siguiente:

```
$xParams = '<PARAMETERS destination="5" language="2" />';
$xxx = $client->call("service", array("501", $xUser, $xPass, $xParams));
```

Y obtendremos un resultado parecido al siguiente:

```
<RESPONSE cod="501">
  <RESERVATIONS count="6">
    <RESERVATION locata="011079" file="xxx" building_id="12070"
building_name="SOL PRINCIPE" building_category="7"
building_category_name="3 estrellas"
building_category_img="http://www.sidetours.com/.../2405_3star.gif"
date="2006-05-29 10:31:39"
check_in="2006-06-26"
check_out="2006-07-02"
price="478.20"
npax="2"
state="2"
source="XML"/>

    <RESERVATION locata="011114" file="yyy" building_id="20691"
building_name="LAS CHAPAS-PALACIO DEL SOL" building_category="7"
building_category_name="3 estrellas"
building_category_img="http://www.sidetours.com/... /205_3star.gif"
date="2006-05-30 12:34:00"
check_in="2006-06-26"
check_out="2006-07-11"
price="1599.50"
npax="4"
state="2"
source="XML"/>
  />

  <!-- MAS RESERVAS -->
```

</RESERVATIONS>
</RESPONSE>

Código 502: Detalles reserva

Recupera información de una reserva en concreto. Como parámetro se pasa el localizador de la reserva y el idioma en el que se quiere el resultado.

El DTD de estos parámetros es:

```
<!ELEMENT PARAMETERS EMPTY>
<!ATTLIST PARAMETERS
  locata CDATA #REQUIRED
  language CDATA #REQUIRED>
```

La respuesta es el detalle de la reserva. Viene representada por el siguiente DTD:

<pre><!ELEMENT RESPONSE (RESERVATION)> <!ATTLIST RESPONSE cod CDATA #REQUIRED></pre>	
<pre><!ELEMENT RESERVATION (ROOM)+> <!ATTLIST RESERVATION board_id CDATA #REQUIRED board_name CDATA #REQUIRED building_id CDATA #REQUIRED building_name CDATA #REQUIRED check_in CDATA #REQUIRED check_out CDATA #REQUIRED date CDATA #REQUIRED destination_id CDATA #REQUIRED destination_name CDATA #REQUIRED file CDATA #REQUIRED locata CDATA #REQUIRED price CDATA #REQUIRED zone_id CDATA #REQUIRED zone_name CDATA #REQUIRED> date_sent CDATA #REQUIRED> state CDATA #REQUIRED></pre>	<p>Código del régimen elegido.</p> <p>Nombre descriptivo del régimen.</p> <p>Código del establecimiento.</p> <p>Nombre del establecimiento.</p> <p>Fecha de entrada al hotel (yyyy-mm-dd).</p> <p>Fecha de salida (yyyy-mm-dd)</p> <p>Fecha de la reserva (yyyy-mm-dd)</p> <p>Código del destino turístico del hotel.</p> <p>Nombre del destino turístico.</p> <p>Expediente dado x la agencia (texto libre)</p> <p>Código único de la reserva.</p> <p>Precio neto que se pagó por la reserva.</p> <p>Código de zona turística del hotel.</p> <p>Nombre de la zona turística.</p> <p>Fecha en la que la reserva se ha enviado al hotel (yyyy-mm-dd).</p> <p>Estado de la reserva:</p> <p>1: Pendiente de confirmar.</p> <p>2: Cancelada</p> <p>3: Fecha de entrada anterior día de hoy</p> <p>0: cualquier otro caso.</p>
<pre><!ELEMENT ROOM (PAX)+> <!ATTLIST ROOM aco_id CDATA #REQUIRED aco_name CDATA #REQUIRED contact CDATA #REQUIRED></pre>	<p>Código de la acomodación.</p> <p>Nombre de la acomodación</p> <p>Nombre de la persona de contacto.</p>
<pre><!ELEMENT PAX EMPTY> <!ATTLIST PAX age CDATA #REQUIRED type (a na nb) "a"></pre>	<p>Edad del niño (no aplicable en adultos).</p> <p>Tipo de inquilino: Adulto, Niño tipo A o B</p>

Ejemplo

Supongamos que queremos los detalles de la reserva con localizador “011104” en castellano. Escribiríamos un código parecido al siguiente:

```
$xParams = "<PARAMETERS locata=\"011104\" language=\"2\" />";  
$xxx = $client->call("service", array("502", $xUser, $xPass, $xParams));
```

El resultado muestra una reserva de 2 habitaciones con 2 adultos cada una. Puede verse en el código XML siguiente:

```
<RESPONSE cod="502">  
  <RESERVATION locata="011104" file="" building_id="2072"  
building_name="EH EL PINAR"  
destination_id="5" destination_name="Costa del Sol"  
zone_id="189" zone_name="Torremolinos"  
date="2006-05-29 10:34:57"  
check_in="2006-06-26"  
check_out="2006-07-05"  
board_id="M"  
board_name="Media pensión"  
price="588.8">  
    <ROOM aco_id="5057" aco_name="Doble Standard" contact="">  
      <PAX type="a" age="0"/>  
      <PAX type="a" age="0"/>  
    </ROOM>  
    <ROOM aco_id="5057" aco_name="Doble Standard" contact="">  
      <PAX type="a" age="0"/>  
      <PAX type="a" age="0"/>  
    </ROOM>  
  </RESERVATION>  
</RESPONSE>
```

Código 503: Cancelación de reserva

Permite cancelar una reserva concreta.

Sólo tiene un parámetro de entrada, por lo que no se ha encapsulado en formato XML y se pasa directamente. El parámetro es el localizador de la reserva que se desea cancelar.

El resultado de la cancelación puede ser positivo o negativo. Si es negativo puede ser debido a que ese localizador no existe, o no pertenece a la agencia que invoca al servicio de reserva. Si es positivo se devuelve también el coste que ha supuesto la cancelación de esta reserva.

Ejemplo

Un ejemplo de la cancelación de la reserva con localizador 320325 en PHP sería:

```
$xxx = $client->call("service", array("503",$xUsuario,$xClave, "320325"));
```

Y los resultados posibles serían:

Caso de éxito:

```
<CANCELLATION locata="320325 " result="OK" expenses="0" />
```

Caso de error:

```
<CANCELLATION locata="320325 " result="NOK" expenses="0" />
```

Código 504: Confirmación de reserva

Se utiliza para confirmar las reservas que se han hecho a través del servicio 500 con la cláusula *no_booking*.

Sólo tiene un parámetro, el localizador.

Un ejemplo de llamada en PHP sería del siguiente modo:

```
$client = new soapclient(WSDL);  
$result = $client->service("504", user_name, password, "106551");
```

Si va todo bien recibiríamos la siguiente respuesta:

```
<CONFIRMATION locata="106551" date_time="2009-02-25" email="1" />
```

Pero si ocurre algún error, recibiremos lo siguiente:

```
<ERROR cod="504-001" msg="Localizador no válido o se han superado  
los 20 minutos de tiempo para confirmar" extra="106551" />
```


Código 505: Simulación de reserva

Es un simulador del servicio 500, es decir que simula una reserva sin llegarla a realizar.

En la respuesta no se encuentra el localizador pero sí una nueva etiqueta llamada CANCEL_POLICIES que cuelga de la etiqueta ya utilizada RESERVATION.

Las ventajas de este servicio es la velocidad de respuesta (ya que no genera registros en el sistema de Sidetours) y el poder obtener los gastos de cancelación antes de hacer la reserva.

Parámetros de entrada.

Son iguales, calcados, a los del servicio 500 de reserva estándar.

Respuesta:

El DTD se muestra a continuación. Se explican los cambios añadidos:

```
<!ELEMENT RESPONSE (RESERVATION)>
<!ATTLIST RESPONSE cod CDATA #REQUIRED>

<!ELEMENT RESERVATION (ROOM)>
<!ATTLIST RESERVATION
    building_id CDATA #REQUIRED
    locata CDATA #REQUIRED
    signature CDATA #REQUIRED
    pay_mode CDATA #REQUIRED
    file CDATA #REQUIRED
    check_in CDATA #REQUIRED
    check_out CDATA #REQUIRED
    nrooms CDATA #REQUIRED
    data CDATA #REQUIRED
    price CDATA #REQUIRED
    board_id CDATA #REQUIRED
    email CDATA #REQUIRED>

<!ELEMENT ROOM (PERIOD,EXTRA,CANCEL_POLICIES)>
<!ATTLIST ROOM
    id CDATA #REQUIRED
    id_acomod CDATA #REQUIRED
    contact CDATA #REQUIRED
    adults CDATA #REQUIRED
    children CDATA #REQUIRED
    price CDATA #REQUIRED>

<!ELEMENT PERIOD EMPTY>
<!ATTLIST PERIOD
    id CDATA #REQUIRED
    price CDATA #REQUIRED
    days CDATA #REQUIRED
    adult1 CDATA ""
    adult2 CDATA ""
    adult3 CDATA ""
    adult4 CDATA ""
    child1 CDATA ""
    child2 CDATA ""
    child3 CDATA ""
    child4 CDATA "">

<!ELEMENT EXTRA EMPTY>
<!ATTLIST EXTRA
```

Modificadores de precio (sptos, descuentos...)

id CDATA #REQUIRED	Código
name CDATA #REQUIRED	Nombre
price CDATA #REQUIRED>	Precio (positivo o negativo)
<!ELEMENT CANCEL_POLICIES (POLICY)>	Políticas de cancelación.
<!ELEMENT POLICY EMPTY>	
<!ATTLIST POLICY	
from CDATA #REQUIRED	Inicio de fecha de penalización.
to CDATA #REQUIRED	Fin de fecha de penalización.
price CDATA #REQUIRED>	Precio de la penalización.

Ejemplo:

Podríamos hacer una petición con los siguientes parámetros XML:

```
<PARAMETERS language="2" building_id="5190" check_in="2012-08-09" check_out="2012-08-11"
file="" board_id="A" pay_mode="A" >
  <ROOM adults="1" children="0" contact="Fernando Sidetours" aco="17923" />
</PARAMETERS>
```

En PHP, si hacemos que la variable \$xParametros tuviera el anterior XML de petición, la llamada podría ser algo así:

```
$xxx = service("505", "un_usuario", "una_clave", $xParametros);
```

La respuesta es la misma que la del servicio 500, solo que no hay localizador ni firma de autenticidad porque realmente no hace la reserva.

Una posible respuesta sería:

- En verde los campos que cambian de valor: ni locata ni signature tienen sentido en la simulación de reserva.
- En rojo la nueva información añadida, que en este caso concreto significa:
 - Cancelar la reserva del 6 al 7 de Agosto cuesta 79.46 euros.
 - Cancelar la reserva el día 8 o un “no-show” cuesta 120.00 euros.

```
<RESPONSE cod="505">
  <RESERVATION building_id="5190" locata="" signature="" pay_mode="A" file=""
check_in="2012-08-09" check_out="2012-08-11" nrooms="1" data="2012-07-04 10:25:37"
price="158.91" board_id="A" email="0">
    <ROOM id="0" id_acomod="17923" contact="Fernando Sidetours" adults="1"
children="0" price="158.91">
      <PERIOD id="0" price="65.73" days="1" adult1="65.73"/>
      <PERIOD id="0" price="93.18" days="1" adult1="93.18"/>
      <CANCEL_POLICIES>
        <POLICY from="2012-08-08" to="2012-08-09" price="120.00"/>
        <POLICY from="2012-08-06" to="2012-08-07" price="79.46"/>
      </CANCEL_POLICIES>
    </ROOM>
  </RESERVATION>
</RESPONSE>
```

Código 600: Tarifas

Recupera información de las tarifas establecidas por Sidetours para sus establecimientos:

- La información se encuentra empaquetada en un archivo ZIP que se descarga desde una dirección HTTP.
- El archivo se actualiza cada 4 horas, aunque esta frecuencia podría cambiar por diversas razones, por lo que si interesa, hay que preguntar a Sidetours la frecuencia actual de refresco.

Este servicio actualmente no tiene parámetros y lo que devuelve es precisamente la URL el archivo ZIP. Un ejemplo de respuesta sería la siguientes:

```
<RESPONSE cod="600">
  <SIDETOURLS
    url="http://fotosbck.sidetours.com/tarifas/s600\_pack\_814\_0aXX3e2.zip"
    file="s600_pack_814_0aXX3e2.zip" />
  </RESPONSE>
```

La documentación sobre este archive ZIP se trata en un documento aparte que hay que pedir a Sidetours.

Código 601: Disponibilidad diaria

Devuelve la ocupación diaria de los hoteles por cada una de sus acomodaciones.

Se le pasa un parámetro opcional: el idioma (ver servicio 104). De no pasarse, se ofrecerán los resultados en castellano.

La respuesta. Viene representada por el siguiente DTD:

<!ELEMENT RESPONSE (DAILY_AVAILABILITY)>	
<!ATTLIST RESPONSE cod CDATA #REQUIRED>	
<!ELEMENT DAILY_AVAILABILITY (BUILDING)+>	
<!ELEMENT BUILDING (BOARD)+>	
<!ATTLIST BUILDING	
id CDATA #REQUIRED	Código del establecimiento.
name CDATA #REQUIRED	Nombre del establecimiento.
building_type CDATA #REQUIRED	Código del tipo de establecimiento.
building_type_name CDATA #REQUIRED	Nombre del tipo (hotel, apto...)
category CDATA #REQUIRED	Código categoría.
category_name CDATA #REQUIRED	Nombre de la categoría.
category_img CDATA #REQUIRED	Imagen representativa de la categoría.
destination CDATA #REQUIRED	Código de destino turístico del hotel.
zone CDATA #REQUIRED	Código de la zona.
destination_name CDATA #REQUIRED	Nombre del destino turístico.
zone_name CDATA #REQUIRED>	Nombre de la zona dentro del destino.
<!ELEMENT BOARD (DAY)+>	
<!ATTLIST BOARD id CDATA #REQUIRED>	
<!ELEMENT DAY EMPTY>	
<!ATTLIST DAY	
id CDATA #REQUIRED	Fecha.
booked CDATA #REQUIRED	Plazas ocupadas.
quote CDATA #REQUIRED>	Plazas contratadas.

Ejemplo

Para saber el nivel de ocupación diaria de una agencia concreta, utilizaremos la siguiente sintaxis en PHP:

```
$xxx = $client->call("service", array("601", $xUsuario, $xClave, "2"));
```

El resultado se aprecia en el XML siguiente:

```
<RESPONSE cod="601">
  <DAILY_AVAILABILITY>
    <BUILDING id="1910" name="LEO" building_type="4"
      building_type_name="Hotel" category="7" category_name="3 estrellas"
      category_img="http://www.sidetours.com/.../20405_3star.gif"
      destination="2" zone="5" destination_name="Mallorca"
      zone_name="Ca'n Pastilla">
      <BOARD id="4582">
        <DAY id="06-09-2006" quote="20" booked="0"/>
        <DAY id="07-09-2006" quote="20" booked="0"/>
        ... MORE DAYS ...
      </BOARD>
      ... MORE BOARDS ...
    </BUILDING>
  </DAILY_AVAILABILITY>
</RESPONSE>
```

```
... MORE BUILDINGS ...  
</DAILY_AVAILABILITY>  
</RESPONSE>
```


ACOMOD.base_day: es el día en el que empieza a definirse el paro de ventas.

ACOMOD.stop_info: Es la información de paros. Hay una posición por día, siendo la primera posición de la cadena el estado de paros del día *ACOMOD.base_day*, la segunda posición el estado del día siguiente, y así sucesivamente.

Un 1 en la posición k (con k un valor mayor o igual a 0) indica que el día *ACOMOD.base_day* + k tiene un paro de ventas. Del mismo modo, si el valor fuera 0 indicaría que ese día no hay paro de ventas explícito.

Como máximo *ACOMOD.stop_info* tendrá una longitud de 365 posiciones, es decir que como mucho se informa de los cierres de un año.

Código 620: Cabecera de ofertas

Devuelve las cabeceras de las ofertas que existen en el sistema Sidetours.

La petición se hace conforme al siguiente DTD:

<!ELEMENT PARAMETERS EMPTY>	
<!ATTLIST PARAMETERS	
language CDATA ""	Idioma en el que se quieren los resultados.
building_id CDATA ""	Ofertas de un edificio concreto.
building_name CDATA ""	Ofertas de los edificios con un nombre.
destination_id CDATA ""	Ofertas por destino turístico.
booking_date CDATA "">	Fecha de la supuesta reserva.

El idioma es opcional, pero de no pasarlo se asumirá el castellano (ver servicio 104: lista de idiomas).

La aplicación de una oferta depende en gran medida del día en el que se haga la reserva, porque todas ellas tienen un *release* (esto se ve claramente en las ofertas de *early-booking*). De este modo, lanzar en Febrero este servicio para conocer las ofertas que tenemos en Agosto puede dar un resultado muy distinto a lanzar este servicio en Julio, precisamente por el tema de *releases*.

La respuesta se sigue según el siguiente DTD

<!ELEMENT OFFER (#PCDATA)>	
<!ATTLIST OFFER	
id CDATA #REQUIRED	Código de la oferta.
title CDATA #REQUIRED	Título de la oferta.
type CDATA #REQUIRED	Tipo: Precio, días gratis, porcentaje
type_name CDATA #REQUIRED	Nombre del tipo.
ini_valid_date CDATA #REQUIRED	Fecha de validez inicial
end_valid_date CDATA #REQUIRED	Fecha de validez final.
ini_application_date CDATA #REQUIRED	Fecha de aplicación inicial.
end_application_date CDATA #REQUIRED>	Fecha de aplicación final.
<!ELEMENT BUILDING (OFFER)+>	
<!ATTLIST BUILDING	
id CDATA #REQUIRED	
name CDATA #REQUIRED	
destination CDATA #REQUIRED	
destination_name CDATA #REQUIRED	
zone CDATA #REQUIRED	
zone_name CDATA #REQUIRED	
category CDATA #REQUIRED	
category_name CDATA #REQUIRED	
category_img CDATA #REQUIRED	
count CDATA #REQUIRED>	
<!ELEMENT BUILDING_SET (BUILDING)+>	
<!ATTLIST BUILDING_SET count CDATA #REQUIRED>	
<!ELEMENT RESPONSE (BUILDING_SET)>	
<!ATTLIST RESPONSE cod CDATA #REQUIRED>	

Ejemplo

Para saber las ofertas que se aplican en Mallorca (destino id = 2), haría una petición tal que así:

```
$xParams = '<PARAMETERS language="2" destination_id="2" />';  
$xxx = $client->call("service", array("620", $xUsuario, $xClave, $xParams));
```

El resultado se aprecia en el XML siguiente:

```
<RESPONSE cod="620">  
  <BUILDING_SET count="78">  
    <BUILDING id="2112" name=" HSM LINDA PLAYA" destination="2"  
destination_name="Mallorca" zone="21" zone_name="Paguera" category="7"  
category_name="3 estrellas"  
category_img="http://www.sidetours.com/imgwebs/icono/20041125133405_3s  
tar.gif" count="1">  
      <OFFER id="7033" title="WINTER OFFER" type="1"  
type_name="precio" ini_valid_date="2008-01-20" end_valid_date="2008-  
04-27" ini_application_date="2008-03-09" end_application_date="2008-  
04-27"></OFFER>  
    </BUILDING>  
  
    <BUILDING id="2113" name=" HSM REGANA" destination="2"  
destination_name="Mallorca" zone="46" zone_name="Cala Ratjada"  
category="7" category_name="3 estrellas"  
category_img="http://www.sidetours.com/imgwebs/icono/20041125133405_3s  
tar.gif" count="1">  
      <OFFER id="7666" title="OFFER IN MAY" type="1"  
type_name="precio" ini_valid_date="2008-03-15" end_valid_date="2008-  
05-29" ini_application_date="2008-05-01" end_application_date="2008-  
05-29"></OFFER>  
    </BUILDING>  
  
    ... MÁS EDIFICIOS ...  
  
  </BUILDING_SET>  
</RESPONSE>
```

Código 1202: Catálogo de edificios extendido

Al igual que el servicio 202, devuelve una lista de establecimientos, salvo que ahora también para cada uno de ellos incluye los tipos de turismo que acepta (ver el servicio 111: Tipos de turismo).

El DTD de la petición es:

<code><!ELEMENT CATALOGUE (SERVICE)*></code>	Establecimientos con servicios concretos.
<code><!ATTLIST CATALOGUE</code>	
<code>language CDATA ""</code>	Idioma en el que se quieren los resultados.
<code>building_type CDATA ""</code>	Código de tipo de edificio deseado.
<code>category CDATA ""</code>	Código de categoría de edificio deseada.
<code>destination CDATA ""</code>	Código del destino turístico donde queremos establecimientos
<code>zone CDATA ""</code>	Código de zona turística.
<code>province CDATA ""</code>	Código de la provincia.
<code><!--ELEMENT SERVICE EMPTY--></code>	
<code><!ATTLIST SERVICE</code>	Código de servicio que deseamos que
<code>id CDATA #REQUIRED</code>	tenga el establecimiento (ver servicio 106).

El idioma es opcional, pero de no pasarlo se asumirá el castellano.

El resultado está estructurado según el siguiente DTD:

<code><!ELEMENT RESPONSE (CATALOGUE)></code>	
<code><!ATTLIST RESPONSE cod CDATA #REQUIRED</code>	
<code><!--ELEMENT CATALOGUE (BUILDING)+--></code>	
<code><!ATTLIST CATALOGUE</code>	
<code>count CDATA #REQUIRED</code>	Número de establecimientos encontrados.
<code>language CDATA #REQUIRED</code>	Código del idioma del resultado.
<code><!--ELEMENT BUILDING (TOURISM_TYPES) --></code>	
<code><!ATTLIST BUILDING</code>	
<code>building_type CDATA #REQUIRED</code>	Código del tipo de establecimiento.
<code>building_type_name CDATA #REQUIRED</code>	Nombre del tipo de establecimiento.
<code>category CDATA #REQUIRED</code>	Código de su categoría.
<code>category_img CDATA #REQUIRED</code>	URL de la imagen de la categoría.
<code>category_name CDATA #REQUIRED</code>	Nombre de la categoría.
<code>destination CDATA #REQUIRED</code>	Código de destino turístico.
<code>destination_name CDATA #REQUIRED</code>	Nombre del destino turístico.
<code>id CDATA #REQUIRED</code>	Código del establecimiento.
<code>name CDATA #REQUIRED</code>	Nombre del establecimiento.
<code>address CDATA #REQUIRED</code>	Dirección del establecimiento.
<code>telephone1 CDATA #REQUIRED</code>	Primer número de teléfono.
<code>telephone2 CDATA #REQUIRED</code>	Segundo número de teléfono.
<code>fax CDATA #REQUIRED</code>	Fax del establecimiento.
<code>email CDATA #REQUIRED</code>	e-mail del establecimiento.
<code>photo CDATA #REQUIRED</code>	URL de una foto de portada del hotel.
<code>text CDATA #REQUIRED</code>	Breve descripción del hotel.
<code>zone CDATA #REQUIRED</code>	Código de zona turística.
<code>zone_name CDATA #REQUIRED</code>	Nombre de la zona turística.
<code>province CDATA #REQUIRED</code>	Código de la provincia
<code>province_name CDATA #REQUIRED</code>	Nombre de la provincia

<code><!ELEMENT TOURISM_TYPES (TOURISM_TYPE)* ></code>	
<code><!ELEMENT TOURISM_TYPE EMPTY ></code>	
<code><!ATTLIST TOURISM_TYPE</code>	
<code>id CDATA #REQUIRED</code>	Código del tipo de turismo.
<code>name CDATA #REQUIRED></code>	Nombre del tipo de turismo.

Ejemplo

Vamos a averiguar los establecimientos que se encuentran en Costa del Sol y disponen de los servicios 12 (Alquiler de coches) y 7 (cambio de divisas).

```
$xParams = '<PARAMETERS language="2" destination="5" province="8" >
  <SERVICE id="12" />
  <SERVICE id="7" />
</PARAMETERS>';
```

```
$xxx = $client->call("service", array("202", $xUser, $xPass, $xParams));
```

El resultado sería algo parecido a esto:

```
<RESPONSE cod="202">
  <CATALOGUE language="2" count="108">
    <BUILDING id="2566" name="ACUARIUM" building_type="4"
destination="5" destination_name="Costa del Sol"
zone="Costa del Sol" zone_name="Fuengirola"
text="El hotel está situado en la turística localidad de Fuengirola, a
unos minutos de la playa y del centro de la localidad. En los
alrededores podrá encontrar varias zonas con tiendas, boutiques,
bares, r..."
photo="http://www.sidetours.com/.../2000517_photo1.jpg"
category="7" category_name="3 estrellas"
category_img="http://www.sidetours.com/.../3star.gif">
      <TOURISM_TYPES>
        <TOURISM_TYPE id="1" name="ciudad" />
        <!-- MAS TIPOS DE TURISMO -->
      </TOURISM_TYPES>
    </BUILDING>

    <!-- MAS EDIFICIOS -->
  </CATALOGUE>
</RESPONSE>
```

Código 1400: Disponibilidad avanzada

Este servicio ofrece dos modos de pago: Venta anticipada (la habitual donde el cliente paga a la agencia) y Pago directo (donde el cliente paga al hotel).

Los parámetros son pasados en XML siguiendo el siguiente esquema:

<code><!ELEMENT ROOM EMPTY></code>	Especificación de una habitación.
<code><!ATTLIST ROOM</code>	
<code>adults CDATA #REQUIRED</code>	Número de adultos en la habitación.
<code>children CDATA #REQUIRED</code>	Número de niños (0, 1 ó 2).
<code>age0 CDATA #REQUIRED</code>	Edad del primero niño (si lo hubiera).
<code>age1 CDATA #REQUIRED</code>	Edad del segundo niño (si lo hubiera).
<code>idmv CDATA #REQUIRED></code>	Código de agrupación.
<code><!ELEMENT PARAMETERS (ROOM)+></code>	Se requiere al menos una habitación.
<code><!ATTLIST PARAMETERS</code>	
<code>building_name CDATA ""</code>	Filtro por nombre de edificio.
<code>building_id CDATA ""</code>	Filtro por código de edificio.
<code>destination CDATA ""</code>	Filtro por destino turístico.
<code>zone CDATA ""</code>	Filtro por zona turística.
<code>province CDATA ""</code>	Filtro por provincia.
<code>country CDATA ""</code>	Filtro por país.
<code>check_in CDATA #REQUIRED</code>	Fecha de entrada en el hotel: yyyy-mm-dd
<code>check_out CDATA #REQUIRED</code>	Fecha de salida del hotel: yyyy-mm-dd
<code>category CDATA ""</code>	Categoría mínima solicitada
<code>board CDATA ""</code>	Régimen mínimo solicitado.
<code>tourism_type CDATA ""</code>	Tipo de turismo.
<code>pay_mode (A D B) "A"</code>	Modo de pago.
<code>language CDATA #REQUIRED</code>	Idioma del resultado.
<code>options CDATA ""</code>	Opciones de respuesta
<code>filter CDATA "" ></code>	Filtro de datos

Explicamos más en detalle alguno de los parámetros:

- **building_name:** Si se especifica, los resultados mostrados corresponderán al patrón de este nombre. Se permite el carácter comodín '%' para las búsquedas, que significa 'cualquier cosa' (Ej. '%CASA' dará todos los establecimientos que comiencen por cualquier cosa y sigan con CASA).
- **building_id:** Si se especifica, busca disponibilidad únicamente en el establecimiento con código el indicado en este parámetro.
- **destination:** Si se especifica, se muestran sólo los establecimientos con disponibilidad pertenecientes al destino turístico codificado con el valor de este parámetro.
- **zone:** Si se especifica, se muestran sólo los establecimientos con disponibilidad pertenecientes a la zona geográfica codificada con el valor de este parámetro.
- **province:** Si se especifica, se busca sólo establecimientos con disponibilidad dentro de la provincia indicada.
- **country:** Si se especifica, se busca sólo establecimientos con disponibilidad dentro del país indicado.

- **Category:** Mínima categoría solicitada. Los valores posibles para este atributo está en el servicio 102. Si se eligen “hoteles de 3 estrellas”, no se devolverá ninguno de categoría inferior (1 ó 2 estrellas).
- **Board:** Mínimo régimen solicitado. Los valores los cogemos del servicio 105. Los órdenes de las categorías son: Sólo alojamiento, alojamiento y desayuno, media pensión, pensión completa y todo incluido.
- **tourism_type:** Tipo de turismo. Los valores posibles son: C – Ciudad, B – Playa, I- Interior, R: Rural, M: Montaña.
- **pay_mode:** Modo de pago de la reserva. Si no se especifica, se supone el tipo A.
 - A: Todos los hoteles que se ofrezcan permiten venta anticipada. No se muestra ningún precio de pago directo.
 - D: Todos los hoteles que se ofrezcan permitirán pago directo. Se muestran ambos precios, el de pago directo y el de venta anticipada.
 - B: Se muestra cualquier hotel. Se muestran ambos precios si el hotel admite pago directo.
- **idmv:** Código de agrupación. Sólo tiene sentido cuando hay más de una habitación. Todas las habitaciones que tengan el mismo valor para este atributo mostrarán el tipo de acomodaciones. Dos habitaciones con valores distintos para este atributo no tiene porqué ofrecer las mismas acomodaciones.
- **options:** Opciones (filtros) en la devolución de la respuesta. Permite personalizar la respuesta de modo que no contenga información redundante o innecesaria para el cliente. Esta opción es interesante de cara a bajar el tráfico, pudiendo ahorrar hasta un 40% aproximadamente en función de los filtros elegidos.

Los modificadores para este parámetro son

N	No incluye el campo BUILDING.name en la respuesta.
D	No campo BUILDING.destination_name
Z	No campo BUILDING.zone_name
V	No campo BUILDING.province_name
I	No campo BUILDING.category_image
U	No campo BUILDING.photo_url
B	No campo BOARD.name (nombre de régimen)
A	No campo ACOMODATION.name (nombre acomodación)
R	No campo ACOMODATION.short_name (nombre corto acomodación)
O	No campo ACOMODATION.offer_title (titulo de la oferta)
P	No hace el desglose por periodos (etiqueta PERIODS) en la respuesta.
S	Sin desglose de suplementos (etiqueta SUPPLEMENT) en la respuesta
T	No campo <NOTES>.

- **Filter:** Filtros sobre los resultados. Es un string que puede tener una combinación de los siguientes valores:

P: Los resultados incluirán hoteles con precios en PVP

R: Los resultados tendrán acomodaciones con la cualidad de “no reembolsables”

El atributo *filter* es opcional. En caso de no incluirse se entiende que no se desea información de edificios con PVP ni de habitaciones no reembolsables.

La respuesta también se ofrece en formato XML. Puede ser exitosa o dar un error.

Respuesta en caso de error

Los errores contemplados son:

- **Código 400:** XML mal formado. Se da este error cuando el XML enviado por la agencia para realizar la petición de búsqueda de disponibilidad no está bien formado según las normas XML.
- **Código 401:** Fecha de entrada fuera del margen de prepago. Si la agencia es de prepago, ha de reservar al menos con una anterioridad igual a los días de prepago. Este control solamente se realiza en venta anticipada.
- **Código 402:** Sin disponibilidad. Se da este error cuando no hay disponibilidad con los criterios elegidos. Bien porque sea una fecha pasada al día actual, bien porque en ningún hotel se cumplan las condiciones de *release* o estancia mínima, o por cualquier otra cosa.
- **Código 403:** Fecha no válida – O no se ha pasado o tiene el formato incorrecto. El formato adecuado de una fecha es YYYY-MM-DD. Ejemplos de fechas correctas son: 2007-12-20, 2006-03-24, etc.
- **Código 404:** Error en la base de datos. Se da por motivos ajenos a la petición realizada por la agencia.
- **Código 405:** Error de régimen. El valor indicado en el atributo *board* de la etiqueta *PARAMETERS* es un régimen no válido.
- **Código 410:** Se da en el caso de que se marque en el atributo *children* de una habitación por ejemplo 2 pero luego se pase sólo la edad de un niño o de ninguno. En general se da este error cuando no concuerda el valor de *children* con el número de edades facilitadas en una habitación.
- **Código 411:** Error en la cantidad de niños. No se admiten más de dos niños en las peticiones de disponibilidad y en el atributo *children* de la etiqueta *ROOM* aparece un valor superior a dos.

Respuesta en caso de éxito

Los resultados se facilitan ahora ordenados por precio, mostrándose primero el de precio más barato de todos los regímenes y acomodaciones encontradas.

Dentro de un hotel, el resultado se muestra ordenado por régimen, de modo que saldrá antes Sólo alojamiento que Media pensión en el caso de que Sólo alojamiento disponga de la acomodación más barata dentro del hotel.

Dentro de un régimen, los resultados se muestran ordenados por precio según la acomodación.

<!ELEMENT RESPONSE (BUILDING_SET)>	Engloba el conjunto de resultados.
<!ATTLIST RESPONSE cod CDATA #REQUIRED>	Código de servicio al que respondemos.
<!ELEMENT BUILDING_SET (BUILDING)+>	Conjunto de edificios.
<!ATTLIST BUILDING_SET	Atributos communes a los edificios.
check_in CDATA #REQUIRED	Fecha de entrada.
check_out CDATA #REQUIRED	Fecha de salida.
count CDATA #REQUIRED>	Número de edificios recopilados.
<!ELEMENT BUILDING ((NOTES)?,BOARD+)>	
<!ATTLIST BUILDING	Atributos de un edificio.
id CDATA #REQUIRED	Código de edificio.
name CDATA #REQUIRED	Nombre
p100_agency CDATA #REQUIRED	Divisor del neto para dar el PVP
type CDATA #REQUIRED	Tipo de edificio (servicio 102)
category CDATA #REQUIRED	Código de categoría.
category_img CDATA #REQUIRED	Imagen representativa de la categoría.
category_name CDATA #REQUIRED	Nombre de la categoría.
province CDATA #REQUIRED	Código de provincia.
province_name CDATA #REQUIRED	Nombre de la provincia.
destination CDATA #REQUIRED	Código de destino turístico.
zone CDATA #REQUIRED	Código de zona turística.
destination_name CDATA #REQUIRED	Nombre del destino turístico.
zone_name CDATA #REQUIRED	Nombre de la zona turística.
photo_url CDATA #REQUIRED	Dirección con una foto del edificio.
pay_mode (A D) "A"	Modo de pago ofrecido.
min_price_pd CDATA ""	Mín precio general de pago directo.
min_price_pvp CDATA ""	Mín precio general en PVP si procede.
min_price CDATA #REQUIRED>	Mínimo precio entre los regímenes.
<!ELEMENT NOTES (#PCDATA) >	Notas del establecimiento al cliente.
<!ELEMENT BOARD (ROOM)+>	Componente de un régimen: habitaciones.
<!ATTLIST BOARD	Atributos del régimen
id CDATA #REQUIRED	Código del régimen.
min_price_pd CDATA ""	Mín precio entre habs. de pago directo.
min_price_pvp CDATA ""	Mín precio entre regs. en PVP si procede.
min_price CDATA #REQUIRED	Mínimo precio de las combinaciones.
pvp_type CDATA (R V) "R"	R: Recomendado, V: Vinculante.
name CDATA #REQUIRED>	Nombre del régimen.
<!ELEMENT ROOM (ACOMODATION)+>	Componentes habitación: Acomodaciones
<!ATTLIST ROOM	Atributos de la habitación.
adults CDATA #REQUIRED	Número de adultos.
age0 CDATA ""	Edad del primer niño (si lo hay)
age1 CDATA ""	Edad del segundo niño (si lo hay)
age2 CDATA ""	Edad del segundo niño (si lo hay)
age3 CDATA ""	Edad del segundo niño (si lo hay)
children CDATA #REQUIRED	Número de niños.
babies CDATA #REQUIRED	Número de bebés.
min_price_pd CDATA ""	Mín precio entre habs. de pago directo.
min_price_pvp CDATA ""	Mín precio entre habs. en PVP si procede.
min_price CDATA #REQUIRED>	Mínimo precio entre las acomodaciones.

<pre> <!ELEMENT ACOMODATION (PERIOD+,SUPLEMENT*,(OFFERS)?,(DISCOUNTS)?)> <!ATTLIST ACOMODATION pay_mode (A D) "A" id CDATA #REQUIRED short_name CDATA #REQUIRED name CDATA #REQUIRED count CDATA #REQUIRED price CDATA #REQUIRED price_pvp CDATA "" price_pd CDATA "" price_type CDATA #REQUIRED offer_type CDATA #REQUIRED offer_title CDATA #REQUIRED adults_in_hotel CDATA #REQUIRED children_in_hotel CDATA #REQUIRED children_a CDATA #REQUIRED children_b CDATA #REQUIRED babies CDATA #REQUIRED non_refundable (0 1) 0> </pre>	<p>Atributos de una acomodación.</p> <p>Modo de pago ofrecido.</p> <p>Código (necesario para la reserva).</p> <p>Nombre corto de la acomodación.</p> <p>Nombre.</p> <p>Número de periodos que la componen.</p> <p>Precio de la habitación en la acomodación</p> <p>Precio de venta al público.</p> <p>Precio en caso de admitirse pago directo.</p> <p>PERSON/DAY o ROOM/DAY según si el precio se calculó por persona/día o por habitación/día.</p> <p>N: No hay, O: oferta, S: Super oferta.</p> <p>Título de la oferta.</p> <p>Nº de adultos según el hotel.</p> <p>Nº de niños según el hotel.</p> <p>Nº de niños tipo A según el hotel.</p> <p>Nº de niños tipo B según el hotel.</p> <p>Nº de bebés en la petición.</p> <p>1 si es “no reembolsable”.</p>
<pre> <!ELEMENT PERIOD EMPTY> <!ATTLIST PERIOD adult1 CDATA "" adult2 CDATA "" adult3 CDATA "" adult4 CDATA "" child1 CDATA "" child2 CDATA "" adult1_pd CDATA "" adult2_pd CDATA "" adult3_pd CDATA "" adult4_pd CDATA "" child1_pd CDATA "" child2_pd CDATA "" adult1_pvp CDATA "" adult2_pvp CDATA "" adult3_pvp CDATA "" adult4_pvp CDATA "" child1_pvp CDATA "" child2_pvp CDATA "" days CDATA #REQUIRED id CDATA #REQUIRED price CDATA #REQUIRED price_pvp CDATA "" price_pd CDATA ""> </pre>	<p>Periodo de tiempo en una acomodación.</p> <p>Atributos.</p> <p>Precio adultos, según tarifa sin descuentos ni ofertas ni suplementos.</p> <p>Precio primer niño si lo hay</p> <p>Del segundo niño si lo hay.</p> <p>Precio adultos, según tarifa en p. Directo.</p> <p>Precio primer niño si lo hay</p> <p>Del segundo niño si lo hay.</p> <p>Precio adultos en PVP si procede.</p> <p>Idem para los niños</p> <p>Días que dura este periodo.</p> <p>Código del periodo.</p> <p>Precio del periodo.</p> <p>Precio del periodo en PVP</p> <p>Precio del periodo en pago directo.</p>
<pre> <!ELEMENT OFFERS (OFFER)+> <!ELEMENT OFFER (COMMENT)> <!ATTLIST OFFER id CDATA #REQUIRED name CDATA #REQUIRED title CDATA #REQUIRED type CDATA #REQUIRED </pre>	<p>Conjunto de ofertas.</p> <p>Una oferta concreta.</p> <p>Las ofertas se descuentan del precio base.</p> <p>Código de la oferta.</p> <p>Nombre de la oferta.</p> <p>Título de la oferta.</p> <p>Tipo de la oferta: Precio, días gratis...</p>

price CDATA #REQUIRED	Valor en euros de la oferta.
price_pvp CDATA ""	Valor en PVP si procede
price_pd CDATA "">	Valor en euros de la oferta en p. directo.
<!ELEMENT COMMENT (#PCDATA)>	Comentario de la oferta
<!ELEMENT SUPLEMENT EMPTY>	Info suplemento.
<!ATTLIST SUPLEMENT	
id CDATA #REQUIRED	Código del suplemento.
name CDATA #REQUIRED	Nombre
price CDATA #REQUIRED	Valor del suplemento.
price_pvp CDATA ""	Valor en PVP si procede.
price_pd CDATA "">	Valor del suplemento para pago directo.
<!ELEMENT DISCOUNTS (DISCOUNT)*>	Conjunto de descuentos.
<!ELEMENT DISCOUNT EMPTY>	Info de descuento especial.
<!ATTLIST DISCOUNT	
id CDATA #REQUIRED	Código del descuento.
name CDATA #REQUIRED	Nombre
price CDATA #REQUIRED	Valor del descuento.
price_pvp CDATA ""	Precio en PVP si procede.
price_pd CDATA "">	Valor del descuento para pago directo.

Dentro de una etiqueta ACOMODATION nos encontramos etiquetas PERIOD, SUPLEMENT y OFFER. Si todo va bien, si sumamos los atributos de precio de todos los periodos y suplementos y restamos los precios de las ofertas, ha de dar lo que diga la acomodación. Es decir, lo que hay dentro de la etiqueta ACOMODATION es el detalle de cómo se ha calculado el precio.

El atributo *pvp* nos lo sugiere el hotel. En estos casos *pvp_type* dirá si este PVP es vinculante (que no se puede modificar) o recomendado. En el caso de que el hotel no nos indique un PVP entonces *pvp_type* valdrá X y se ha de ignorar el valor de la etiqueta *pvp* (no procederá).

El atributo *non_refundable* indica si la habitación permite ser cancelada o no una vez formalizada la reserva. Notamos que en una reserva basta que haya una única acomodación no reembolsable para que no se pueda cancelar mediante webservices, teniendo que recurrir entonces al personal de Sidetours.

Ejemplo

Llamada al servicio

Se piden dos habitaciones en cualquier zona del destino 5 entrando el 26 de Junio del 2007 y saliendo el 11 de Julio de 2007. Hay 2 adultos en la primera habitación y uno sólo en la segunda. No hay ninguna restricción en cuanto al régimen.

Pediremos todos los hoteles con independencia del modo de pago que se acepten, ya sea pago directo o venta anticipada (*pay_mode*="B")

La petición sería tal que así

```
$params = '<PARAMETERS destination="5" check_in="2007-06-26"
check_out="2007-07-01" language="2" pay_mode="B">
<ROOM adults="2" children="1" age0="4" age1="0" idmv="1"/>
<ROOM adults="1" children="0" age0="0" age1="0" idmv="1"/>
</PARAMETERS>'
```

Si se hubiera deseado información de habitaciones en PVP (cosa que algunos hoteles casi están a punto de exigir), entonces la petición sería del tipo:

```
$params = '<PARAMETERS destination="5" check_in="2007-06-26"
        check_out="2007-07-01" language="2" pay_mode="B" filter="P" >
        <ROOM adults="2" children="1" age0="4" age1="0" idmv="1"/>
        <ROOM adults="1" children="0" age0="0" age1="0" idmv="1"/>
</PARAMETERS>'
```

Y si además se quisieran también las habitaciones no reembolsables, la petición hubiera cambiado a:

```
$params = '<PARAMETERS destination="5" check_in="2007-06-26"
        check_out="2007-07-01" language="2" pay_mode="B" filter="PR" >
        <ROOM adults="2" children="1" age0="4" age1="0" idmv="1"/>
        <ROOM adults="1" children="0" age0="0" age1="0" idmv="1"/>
</PARAMETERS>'
```

Con independencia de los parámetros, la llamada en PHP utilizando la librería nusoap.php queda del siguiente modo:

```
$xxx = $client->call("service", array("1400", $user, $pass, $params));
```

La respuesta se muestra a continuación.

Los atributos con texto *pvp* (como *pvp_type*, *min_price_pvp*, etc) saldrían sólo si se hubiera especificado el atributo *filter* en los parámetros y se hubiera incluido la opción *P*. Del mismo modo las etiquetas *non_refundable* saldrían sólo en el caso de haber incluido la etiqueta *filter* pero ahora con el valor *R*.

```
<RESPONSE cod="1400">
  <BUILDING_SET check_in="2007-06-26" check_out="2007-07-01" count="2">
```

- Se han encontrado 2 edificios entre las fechas pedidas.

```
  <BUILDING pl00_agency="0.94" id="3493" name="HM JAIME III"
    destination="5" zone="7" destination_name="Mallorca" zone_name="Palma Ciudad"
    pay_mode="D" category="8" category_name="4 estrellas"
    min_price="371.00" min_price_pd="446.98" min_price_pvp="450.00" pvp_type="V"
    category_img="http://www...com/imgwebs/icono/20041125_4star.gif"
    photo_url="http://www...com/imgwebs/artgaleria/edi3493/Imagen_158.jpg">
```

```
  <NOTES>Media Pensión será desayuno y cena.
  Half board with dinner.</NOTES>
```

- Sobre el régimen: 371.00 es el precio más bajo de todos los precios que pueda dar el hotel HM JAUME III en Sólo alojamiento. Los 371 será la suma de los precios de las dos habitaciones.

```
  <BOARD id="A" name="Solo alojamiento" min_price="371.00"
    min_price_pd="446.98" min_price_pvp="380.00" >
```

- Primera habitación. La doble. Es una acomodación "reembolsable", es decir que podría cancelarse según la política de cancelación vigente.

```
<ROOM adults="2" children="0" min_price="197.50"
min_price_pd="237.95" min_price_pvp="200.00">
```

```
<ACOMODATION pay_mode="D" id="7847" short_name="DB" name="Doble
Standard" count="1" price="197.50" price_pd="237.95" price_type="ROOM/DAY"
offer_type="N" offer_title="" adults_in_hotel="2" children_in_hotel="0"
children_a="0" children_b="0" babies="0" price_pvp="200.00" non_refundable
="0">
```

```
<PERIOD id="277591" days="5" price="197.50" price_pd="237.95"
adult1="98.75" adult1_pd="118.97" adult2="98.75" adult2_pd="118.97"
adult1_pvp="100.00" adult2_pvp="100.00" />
```

```
</ACOMODATION>
```

```
</ROOM>
```

- Segunda habitación. La sencilla que tiene un suplemento. Además es "no reembolsable" lo que hace que si se hiciera la reserva con esta acomodación, no pudiera cancelarse.

```
<ROOM adults="1" children="0" min_price="173.50"
min_price_pd="209.03" min_price_pvp="180.00" >
```

```
<ACOMODATION pay_mode="D" id="7847" short_name="DB" name="Doble
Standard" count="1" price="173.50" price_pd="209.03" price_type="ROOM/DAY"
offer_type="N" offer_title="" adults_in_hotel="1" children_in_hotel="0"
children_a="0" children_b="0" babies="0" price_pvp="180.00" non_refundable
="1">
```

```
<PERIOD id="277591" days="5" price="98.78" price_pd="118.97"
adult1="98.75" adult1_pd="118.97" price_pvp="100.00" adult1_pvp="100.00" />
```

```
<SUPPLEMENT id="142935" name="Spto. Individual" price="74.75"
price_pd="90.06" price_pvp="80.00"/>
```

```
</ACOMODATION>
```

```
</ROOM>
```

```
</BOARD>
```

```
</BUILDING>
```

- Ahora más edificios, son iguales que el resto. El siguiente concretamente lleva una oferta.

Aquí pvp_type = "X", lo que significa que no hay que tener en cuenta el valor de la etiqueta pvp.

```
<BUILDING p100_agency="0.94" id="2405" name="TRH PARAISO" destination="5"
zone="199" destination_name="Costa del Sol" zone_name="Estepona" pay_mode="D"
category="8" category_name="4 estrellas"
min_price="450.00" min_price_pd="542.16" min_price_pvp="0" pvp_type="X"
category_img="http://www...com/imgwebs/icono/20041125_4star.gif"
photo_url="http://www...com/imgwebs/artgaleria/edi2405/fachada.jpg" >
```

```
<BOARD id="D" name="Alojamiento y desayuno" min_price="450.00"
min_price_pd="542.16" min_price_pvp="0" >
```

```
<ROOM adults="2" children="0" min_price="225.00"
min_price_pd="271.08" min_price_pvp="0" >
```

```
<ACOMODATION pay_mode="D" id="7415" short_name="DB" name="Doble
Standard" count="1" price="197.50" price_pd="237.95" price_type="ROOM/DAY"
offer_type="N" offer_title="" adults_in_hotel="2" children_in_hotel="0"
children_a="0" children_b="0" babies="0" price_pvp="0" non_refundable="0">
```

```
<PERIOD id="277591" days="5" price="250.00" price_pd="301.2"
adult1="125.00" adult1_pd="150.6" adult2="125.00" adult2_pd="150.6"
price_pvp="0" adult1_pvp="0" adult2_pvp="0" />
```

```
<OFFERS>
  <OFFER type="2" name="PORCENTAJE" title="Oferta Dto. venta
anticipada" value="25" price="25" price_pd="30.12" price_pvp="0" />
</OFFERS>
  </ACOMODATION>
</ROOM>

  <ROOM adults="1" children="0" min_price="225.00"
min_price_pd="271.08" min_price_pvp="0" >

    <ACOMODATION pay_mode="D" id="7415" short_name="DB" name="Doble
Standard" count="1" price="197.50" price_pd="237.95" price_type="ROOM/DAY"
offer_type="N" offer_title="" adults_in_hotel="2" children_in_hotel="0"
children_a="0" children_b="0" babies="0" price_pvp="0" non_refundable="0" >
      <PERIOD id="277591" days="5" price="250.00" price_pd="301.2"
adult1="125.00" adult1_pd="150.6" adult2="125.00" adult2_pd="150.6"
price_pvp="0" adult1_pvp="0" adult2_pvp="0" />
    <OFFERS>
      <OFFER type="2" name="PORCENTAJE" title="Oferta Dto. venta
anticipada" value="25" price="25" price_pd="30.12" price_pvp="0" >
        <COMMENT> This is a comment... </COMMENT>
      </OFFER>
    </OFFERS>
  </ACOMODATION>
</ROOM>
</BOARD>
</BUILDING>

</BUILDING_SET>
</RESPONSE>
```

Nota:

Recordamos que los precios dados por el servicio 400 (o 1400) no son válidos para confirmar reservas ya que estos son en tiempo real, por lo que pueden cambiar de un momento para otro, haciendo que el precio que el cliente ha visto a la hora de decidir haya variado a la hora de pagar (muy improbable pero posible, sobre todo si ha tardado mucho). Para eso se ha creado el servicio 500 y 504.

Código 1503: Cancelación de reserva con gastos

Permite cancelar o simular la cancelación de una reserva.

A diferencia del servicio 503, este servicio devuelve los gastos de cancelación a día de la llamada al servicio. Digo “a diferencia” porque el servicio 503 sólo permitía cancelar si en el momento de la cancelación la reserva no tenía gastos (en cualquier otro caso se tenía que poner en contacto con el departamento de reservas de Sidetours).

Los parámetros de la llamada también son distintos y se rigen según el siguiente DTD:

<!ELEMENT PARAMETERS EMPTY>	
<!ATTLIST PARAMETERS	
locata CDATA #REQUIRED	Localizador a cancelar.
action CDATA #REQUIRED>	Acción:
	cancel: Se cancela
	¬ cancel: Se simula la cancelación.

Como se ve, una reserva puede cancelarse o sólo simularse para cancelarse. La simulación nos servirá, por ejemplo, para obtener los gastos de cancelación sin cancelarla.

Existe un tipo de habitación cuyos gastos de cancelación son conocidos por ser invariables, que son los asociados a las habitaciones “no reembolsables”. Este tipo de habitaciones siempre tiene un gasto del 100% del importe sea cual sea el momento en el que se cancele.

El gasto de cancelación total será la suma de los gastos de cancelación de cada habitación.

El DTD de la respuesta es el siguiente:

<!ELEMENT CANCELLATION (PENALTY)+>	
<!ATTLIST CANCELLATION	
locata CDATA #REQUIRED	Localizador pedido.
result (OK DEMO) "DEMO"	Resultado: DEMO – No cancelada
expenses CDATA #REQUIRED>	Gastos totales.
<!ELEMENT PENALTY EMPTY>	
<!ATTLIST PENALTY	
room_id CDATA #REQUIRED	Id de habitación.
price CDATA #REQUIRED>	Gasto de la habitación.

Ejemplo:

Una petición de cancelación de ejemplo podría ser:

```
<PARAMETERS locata="1839618" action=""></PARAMETERS>
```

Y la respuesta:

```
<RESPONSE cod="1503">
  <CANCELLATION locata="1839618" result="DEMO" expenses="195.04">
    <PENALTY room_id="1" price="97.52"/>
    <PENALTY room_id="2" price="97.52"/>
  </CANCELLATION>
</RESPONSE>
```

La cancelación no se ha hecho porque el parámetro *action* no tenía el valor “cancel”. Si se hubiera así establecido, el resultado hubiera sido el mismo salvo que en *result* tendríamos el valor *OK*.

Código 1504: Confirmación de reservas extendida

La confirmación 504 tiene el problema de que se necesita saber toda la información en el momento de hacer la pre-reserva. Según la experiencia con nuestros clientes parece ser que esto no es siempre fácil de conseguir, de modo que se creó este servicio que permite confirmar la reserva al tiempo de completarla, dando la información de expediente y nombre de la persona responsable de la habitación.

Parámetros:

DTD

<![ELEMENT PARAMETERS (CUSTOMER_NOTES)>	
<![ATTLIST PARAMETERS	
locata CDATA #REQUIRED	Localizador que quiere confirmarse.
file CDATA " "	Expediente a completar.
contact CDATA " " >	Responsable de la habitaciones.
<![ELEMENT CUSTOMER_NOTES (#PCDATA)>	
Observaciones del cliente.	

Respuesta

Pueden haber dos tipos de respuesta. En caso de éxito se devolverá la información de la reserva confirmada. En caso de error se devolverá un XML de error. Los errores posibles son:

1. Que la reserva no exista en el repositorio de la agencia.
2. Que se haya confirmado fuera del periodo de confirmación.

DTD caso de éxito

<![ELEMENT CONFIRMATION EMPTY>	
<![ATTLIST CONFIRMATION	
date_time CDATA #REQUIRED	Fecha de confirmación Y-m-d h:i:s
email (0 1) "1"	1: Enviada confirmación email al hotel
locata CDATA #REQUIRED>	Localizador confirmado

DTD caso de error.

<![ELEMENT ERROR EMPTY>	
<![ATTLIST ERROR	
cod CDATA #REQUIRED	Valor 1504-001
msg CDATA #REQUIRED	Mensaje de error.
extra CDATA #REQUIRED>	Localizador que ha fallado

Ejemplo

A continuación se expone una petición de confirmación con comentarios del cliente:

```
<PARAMETERS locata="015890" file="test" contact="Bon Jovi" >
  <CUSTOMER_NOTES>
    Observaciones del cliente para el hotel.
  </CUSTOMER_NOTES>
</PARAMETERS>
```

Respuesta de confirmación con éxito:

```
<CONFIRMATION locata="015890" date_time="2007-04-16 18:23:50 email="1" />
```