

API didimo Marketing - Aplicateca

Manual de integración

Versión 2.0

API didimo Marketing - Manual de Integración

Contenido

1. Servicios SMS	2
2. Credenciales.....	2
3. Destinos internacionales.....	2
4. Remitente.....	3
5. Recepción de mensajes entrantes (MO)	3
6. Funcionalidades	4
7. Métodos.....	5
8. Ejemplo de integración.....	11
9. Códigos de Error	14
10. Estados de mensaje	14
11. Caracteres admitidos	15
12. Terminología.....	16
13. Anexo	16

1.Servicios SMS

Para utilizar los servicios SMS del API, debe integrar un cliente con el siguiente servicio web:

<https://aplicateca.didimo.es/custws/service.asmx>

2.Credenciales

Para hacer uso del servicio debe disponer de unas credenciales (usuario y contraseña) facilitadas por soporte.

Las credenciales son específicas del API y distintas de las utilizadas para acceder a Aplicateca.

Todos los métodos del servicio deben ser informados con las credenciales.

Puede solicitarlas contactando por email a soporteaplicateca@didimo.es indicando el nombre de la empresa y persona de contacto.

3.Destinos internacionales

El servicio permite realizar envíos a móviles nacionales e internacionales. El coste del SMS a móviles internacionales es el mismo que un SMS a un destino nacional.

Un móvil internacional válido debe comenzar por 00 o por +.
Ejemplo: +44xxxxxxxxx o 0033xxxxxxxxx

Estos son los países y prefijos permitidos:

Países	Prefijo
Andorra	376
Armenia	374
Austria	43
Bielorusia	375
Bélgica	32
Bosnia y Herzegovina	387
Bulgaria	359
Croacia	385
Chipre	357
Chequia	420
Dinamarca	45
Estonia	372
Finlandia	358

API didimo Marketing - Manual de Integración

Francia	33
Georgia	995
Alemania	49
Gibraltar	350
Hungria	36
Islandia	354
Irlanda	353
Italia	39
Latvia	371
Liechtenstein	423
Lituania	370
Luxemburgo	352
Macedonia	389
Malta	356
Moldavia	373
Montenegro	382
Holanda	31
Noruega	47
Polonia	48
Portugal	351
Reino Unido	44
Rumania	40
Serbia	381
Eslovaquia	421
Eslovenia	386
Suecia	46
Suiza	41
Turquía	90
Ucrania	380

4. Remitente

Para hacer uso del servicio es recomendable disponer de un remitente pre-definido en el servicio didimo Marketing. Para ello, debe haber entrado en el interfaz Web y haber definido un remitente en el menú “Configuración”.

Los remitentes pueden informarse en el uso del API, pero en caso de no hacerlo se utiliza el pre-definido.

5. Recepción de mensajes entrantes (MO)

El cliente puede recibir en sus sistemas los mensajes entrantes(MO) que envían los usuarios de terminales hacia la plataforma. Para ello, es necesario informar de la URL del cliente donde

API didimo Marketing - Manual de Integración

quiere recibir dichos mensajes. El cliente debe informar por email de dicha URL a sopORTEaplicateca@didimo.es

Los usuarios deberán enviar SMS al 215365 (funciona con Movistar, Vodafone y Orange), el cual es un nº corto compartido entre todos los clientes. Para recibir solamente sus SMS, los usuarios deben enviar un texto único llamado “alias” previamente definido. El alias se define en el interfaz web, menú “Crear Campaña”.

Al recibir los mensajes entrantes en la URL, el cliente puede decidir si quiere utilizar la respuesta automática configurada en la campaña del servicio o si quiere enviar un mensaje de respuesta utilizando el API.

Este es el formato de la URL del cliente donde se reciben las peticiones HTTP/POST con los datos correspondientes a los SMS recibidos en la plataforma correspondientes a su usuario:

<http://urlusuario.es/getSMS.aspx?param1=value1¶m2=value2...paramN=valueN>

Los datos correspondientes a los SMS entrantes:

- Identificador único del mensaje
- Numero corto donde el cliente ha enviado el SMS, en este caso Aplicateca tiene asignado el 215365
- Operador móvil correspondiente al terminal del cliente.
- Número móvil correspondiente al terminal del cliente.
- Texto del mensaje entrante.
- Alias o código de respuesta asociado al mensaje entrante.
- Indicador de si el mensaje entrante es una baja
- Fecha/hora correspondiente al mensaje entrante.
- Texto de respuesta que se le ha enviado al terminal del mensaje entrante.

La información que quiere recibir con los parámetros relacionados, por ej: si quiere recibir el número de móvil en el parámetro “movil” y el texto en el parámetro “text”, la petición a la url quedaría así: [http://urlusuario.es/getSMS.aspx?movil="+346666666&text=texto del mensaje entrante](http://urlusuario.es/getSMS.aspx?movil=).

El cliente debe enviar por email a sopORTEaplicateca@didimo.es una lista de parámetros con el valor que quiere recibir según el listado que se ha descrito arriba.

6. Funcionalidades

El API ofrece las siguientes funcionalidades:

- Consulta de disponibilidad del servicio (método [Ping](#))
- Obtención de identificador único (GUID) de mensaje (método [GetMessageId](#))
- Envío de un mensaje (método [CreateMessage](#))
- Envío de N mensajes (método [CreateSend](#))
- Obtención del estado de un mensaje (método [GetMessageStatus](#))
- Consulta de Créditos disponible para enviar SMS (método [GetCredits](#))

API didimo Marketing - Manual de Integración

- Consulta de los mensajes recibidos (MO) (método [GetReceptionsMessages](#))

7.Métodos

El conjunto de métodos es el siguiente:

Service

Las operaciones siguientes son compatibles. Para una definición formal, revise la [descripción de servicios](#).

- [CreateMessage](#)
- [CreateSend](#)
- [GetCredits](#)
- [GetMessageId](#)
- [GetMessageStatus](#)
- [GetReceptionsMessages](#)
- [Ping](#)

Todos los métodos deben informarse con las credenciales de uso de API facilitadas por soporte y cualquier otro dato necesario para su funcionalidad.

Parámetros de credenciales:

- `login` : nombre de usuario (ver credenciales), obligatorio, tipo `String`, de longitud no definida
- `password` : contraseña de usuario (ver credenciales), obligatorio, tipo `String`, de longitud no definida

Todos los métodos devuelven una estructura informando del éxito o fallo en la invocación del método. Dicha estructura presenta los siguientes miembros:

- `ErrorCode` : Integer, con valores 0 para éxito y distinto de 0 para error (ver tabla de códigos de error)
- `ErrorDescription` : en caso de que `ErrorCode` sea distinto de 0 informa una descripción del error.
- `Result` : informa el retorno específico al método invocado (ver cada método en detalle)

Ping

Haga uso de este método para consultar la disponibilidad del servicio.

Entrada (parámetros):

- `login`
- `password`

Salida: estructura `PingResult` con los siguientes miembros:

- `ErrorCode`
- `ErrorDescription`

API didimo Marketing - Manual de Integración

- `Result` : cadena formada por el literal `OK` y la fecha/hora de la invocación en formato `yyyyMMdd HHmmss`, ejemplo `OK - 20120411 213140`.

GetMessageId

Haga uso de este método para obtener un `identificador único (GUID)` de mensaje (ver métodos de envío de mensajes).

Entrada (parámetros):

- `login`
- `password`

Salida: estructura `GetMessageIdResult` con los siguientes miembros:

- `ErrorCode`
- `ErrorDescription`
- `Result` : GUID como `String`, ejemplo `6293a9bb-566b-4f43-b087-33afcfecl1f58`.

CreateMessage

Haga uso de este método para enviar un único mensaje.

Entrada (parámetros):

- `login`
- `password`
- `id` : `String`, identificador GUID del mensaje, opcional, en caso de informarse debe ser un GUID válido, en caso de no hacerlo el API generará uno. Puede hacer uso del método `GetMessageId` como utilidad para obtener identificadores GUID válidos.
- `name` : `String`, nombre del envío, opcional, en caso de informarse debe ser inferior a 250 caracteres, en caso de no informarse el API generará un nombre con formato `API_SEND_YYYYMMDDHHMM`, ejemplo `API_SEND_201204111810`.
- `sender` : `String`, remitente del mensaje, opcional, en caso de informarse debe tener longitud máxima de 11 caracteres. En caso de no informarse, el API utilizará el remitente pre-definido del usuario (ver Remitente)
- `text` : `String`, texto del mensaje, obligatorio, puede tener hasta 306 caracteres. En caso de tener hasta 160 caracteres resultará en un único SMS, para más de 160 caracteres y hasta 306 resultará en 2 SMS (ver Mensajes).
- `mobile` : `String`, destinatario del mensaje, obligatorio, puede informarse con formato internacional (p.e. 003469999999 ó +34699999999), o en formato nacional (p.e. 699999999).
- `scheduleDate` : `String`, fecha/hora de envío del mensaje, opcional, en caso de informarse debe ser una fecha/hora válida en formato `YYYY-MM-DDThh:mm:ss`, en caso de no informarse el API utilizará la fecha/hora actual y el envío será inmediato

Salida: estructura `CreateMessageResult` con los siguientes miembros:

- `ErrorCode`
- `ErrorDescription`
- `Result` : estructura `CreateMessageResultDetail` con los siguientes miembros
 - o `Id` : `String`, identificador GUID del mensaje informado por el cliente o generado por el sistema.
 - o `TotalMessages` : `Integer`, número de SMS resultantes a partir del texto.

API didimo Marketing - Manual de Integración

Ejemplo 1: Se informan datos correctos, sin identificador de mensaje y el resultado es satisfactorio:

```
Llamada: servicio.CreateMessage("usuario", "clave", null, "APISend 1 mensaje", "mysender1", "test de 1 mensaje", "607903114", null);
```

Retorno (serializado como XML):

```
<result>
  <ErrorCode>0</ErrorCode>
  <ErrorDescription></ErrorDescription>
  <Result>
    <Id>9fcbe8d9-3ffe-44ac-b9e2-9d67648041ac</Id>
    <TotalMessages>1</TotalMessages>
  </Result>
</result>
```

Ejemplo 2: Se informan datos correctos, con identificador de mensaje y el resultado es satisfactorio:

```
Llamada: servicio.CreateMessage("usuario", "clave", "7F360C11-B655-4EB0-AD54-11BEFA99800A", "APISend 1 mensaje", "mysender2", "test de 1 mensaje", "607903114", null);
```

Retorno (serializado como XML):

```
<result>
  <ErrorCode>0</ErrorCode>
  <ErrorDescription></ErrorDescription>
  <Result>
    <Id>7F360C11-B655-4EB0-AD54-11BEFA99800A</Id>
    <TotalMessages>1</TotalMessages>
  </Result>
</result>
```

Ejemplo 3: Se informan datos con móvil incorrecto, sin identificador de mensaje y el resultado es de error:

```
Llamada: servicio.CreateMessage("usuario", "clave", null, "APISend 1 mensaje", "mysender3", "test de 1 mensaje", "807903114", null);
```

Retorno (serializado como XML):

```
<result>
  <ErrorCode>3</ErrorCode>
  <ErrorDescription>Error de validación: El móvil informado ('807903114') no es válido </ErrorDescription>
  <Result>
    <Id></Id>
    <TotalMessages></TotalMessages>
  </Result>
</result>
```

CreateSend

Haga uso de este método para enviar uno o más mensajes.

API didimo Marketing - Manual de Integración

Entrada (parámetros):

- `login`
- `password`
- `send` : estructura tipo `Send`, que contiene los siguientes miembros:
 - o `Name` : `String`, nombre del envío, opcional, en caso de informarse debe ser inferior a 250 caracteres, en caso de no informarse el API generará un nombre con formato `API_SEND_YYYYMMDDHHMM`, ejemplo `API_SEND_201204111810`.
 - o `Sender` : `String`, remitente del mensaje, opcional, en caso de informarse debe tener longitud máxima de 11 caracteres. En caso de no informarse el API utilizará el remitente pre-definido del usuario (ver Remitente)
 - o `ScheduleDate` : `String`, fecha/hora de envío del mensaje, opcional. En caso de informarse debe ser una fecha/hora válida en formato `YYYY-MM-DDThh:mm:ss`, en caso de no informarse el API utilizará la fecha/hora actual y el envío será inmediato
 - o `Messages` : lista de estructuras `Message`, conteniendo los datos específicos de cada mensaje, donde cada elemento `Message` contiene los siguientes miembros:
 - `id` : `String`, identificador GUID del mensaje, opcional. En caso de informarse debe ser un GUID válido. En caso de no hacerlo el API generará uno. Puede hacer uso del método `GetMessageId` como utilidad para obtener identificadores GUID válidos.
 - `text` : `String`, texto del mensaje, obligatorio, puede tener hasta 306 caracteres. En caso de tener hasta 160 caracteres resultará en un único SMS, para más de 160 caracteres y hasta 306 resultará en 2 SMS.
 - `mobile` : `String`, destinatario del mensaje, obligatorio, puede informarse con formato internacional (p.e. 0034699999999 ó +34699999999), o en formato nacional (p.e. 699999999).

Salida: estructura `CreateSendResult` con los siguientes miembros:

- `ErrorCode`
- `ErrorDescription`
- `Result` : lista de estructuras `CreateSendResultDetail` informando del detalle de resultado para cada mensaje, donde cada `CreateSendResultDetail` contiene los siguientes miembros
 - o `ErrorCode`
 - o `ErrorDescription`
 - o `Id` : `String`, identificador GUID del mensaje informado por el cliente o generado por el sistema.
 - o `TotalMessages` : `Integer`, número de SMS resultantes a partir del texto.

En este método hay 2 niveles de error:

- Global: afecta a todo el envío y es informado por `ErrorCode` y `ErrorDescription` a nivel de `CreateSendResult`. En caso de haber error global no se llega a procesar ningún mensaje.
- Detallado: cuando a nivel global no ha habido error y se procesan mensajes, este nivel detalla el resultado para cada uno de los mismos, y es informado por `ErrorCode` y `ErrorDescription` a nivel de `CreateSendResultDetail`.

Ejemplos de errores globales y detallados:

- Error Global

API didimo Marketing - Manual de Integración

- se informa un envío con credenciales no válidas y el método devuelve un resultado con `ErrorCode 2` (Error de autenticación) en `CreateSendResult`.
- se informa un envío con una fecha/hora no válidas 2020-01-01X09:00:00 y el método devuelve un resultado con `ErrorCode 3` (Error de validación: La programación informada no es una fecha/hora válida) en `CreateSendResult`.
- Error Detallado:
 - se informa un envío con un mensaje cuyo texto no es válido y el método devuelve un resultado con `ErrorCode 0` (no hay error) en `CreateSendResult` y con `ErrorCode 3` (Error de validación) en `CreateSendResultDetail` de dicho mensaje.
 - se informa un envío con un mensaje cuyo móvil no es válido y el método devuelve un resultado con `ErrorCode 0` (no hay error) en `CreateSendResult` y con `ErrorCode 3` (Error de validación) en `CreateSendResultDetail` de dicho mensaje.

GetMessageStatus

Haga uso de este método para obtener el estado de un mensaje (ver métodos de envío de mensajes).

Entrada (parámetros):

- `login`
- `password`
- `id : String`, identificador GUID del mensaje a consultar su estado

Salida: estructura `GetMessageIdResult` con los siguientes miembros:

- `ErrorCode`
- `ErrorDescription`
- `Result` : estructura `GetMessageIdResultDetail` con los siguientes miembros
 - `StatusCode` : `Integer`, informa un código de estado (ver estados de mensaje).
 - `StatusDescription` : `String`, informa una descripción de estado (ver estados de mensaje).

Ejemplos de estados (retornos serializados como XML):

```
<result>
  <ErrorCode>0</ErrorCode>
  <ErrorDescription></ErrorDescription>
  <Result>
    <StatusCode>10</StatusCode>
    <StatusDescription>Pendiente Entrega a Operador</StatusDescription>
  </Result>
</result>

<result>
  <ErrorCode>0</ErrorCode>
  <ErrorDescription></ErrorDescription>
  <Result>
    <StatusCode>14</StatusCode>
    <StatusDescription>Pendiente Entrega a Destinatario</StatusDescription>
  </Result>
</result>

<result>
  <ErrorCode>0</ErrorCode>
```

API didimo Marketing - Manual de Integración

```
<ErrorDescription></ErrorDescription>
<Result>
  <StatusCode>19</StatusCode>
  <StatusDescription>Finalizado en Destinatario</StatusDescription>
</Result>
</result>
```

GetCredits

Este método sirve para consultar el número de créditos para enviar SMS.

Entrada (parámetros):

- login
- password

Salida: estructura `GetCreditsResult` con los siguientes miembros:

- `ErrorCode`
- `ErrorDescription`
- `Credits: integer`, Total de créditos disponibles.

GetReceptionsMessages

Este método permite obtener un listado de los mensajes recibidos.

Entrada (parámetros):

- login
- password
- `mobile: String`, número de móvil del terminal que ha originado el mensaje SMS MO
- `campaign: String`, nombre de la campaña a la que corresponde el Alias del mensajes recibido
- `alias: String`, alias o código de respuesta asociado al mensaje recibido
- `MOText: String`, texto del mensaje recibido
- `MTText: String`, texto del mensaje de respuesta
- `isResign: Boolean`, indica si el número del móvil del terminal está de baja. Posibles valores: "True", "False", null.
- `dateFrom: String`, realizará la búsqueda "a partir" de una fecha/hora de recepción del mensaje. En caso de informarse debe ser una fecha/hora válida en formato `YYYY-MM-DDThh:mm:ss`.
- `dateTo: String`, realizará la búsqueda "hasta" una fecha/hora de recepción del mensaje. En caso de informarse debe ser una fecha/hora válida en formato `YYYY-MM-DDThh:mm:ss`.

Salida: estructura `GetReceptionsMessagesResult` con los siguientes miembros:

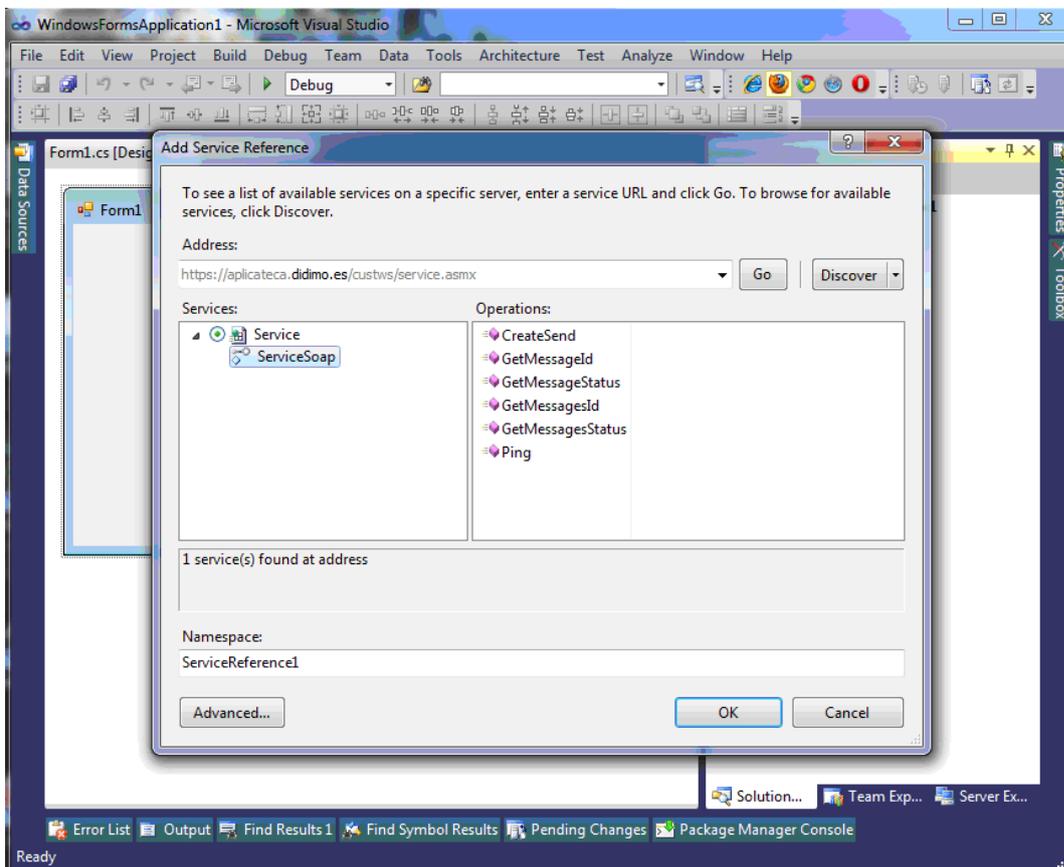
- `ErrorCode`
- `ErrorDescription`
- `Result` : lista de estructuras `GetReceptionsMessagesItem` informando del detalle de resultado para cada mensaje recibido, donde cada `GetReceptionsMessagesItem` contiene los siguientes miembros
 - o `ErrorCode`
 - o `ErrorDescription`
 - o `isResign: String`, indica si es una baja o no.

API didimo Marketing - Manual de Integración

- **Mobile**: *String*, número de móvil del terminal que ha originado el mensaje SMS MO.
- **Campaign**: *String*, nombre de la campaña a la que corresponde el Alias del mensajes recibido
- **Alias**: *String*, alias o código de respuesta asociado al mensaje recibido
- **MOText**: *String*, texto del mensaje recibido
- **MTText**: *String*, texto del mensaje de respuesta
- **Date**: *String*, fecha/hora correspondiente a la recepción del mensaje.

8. Ejemplo de integración

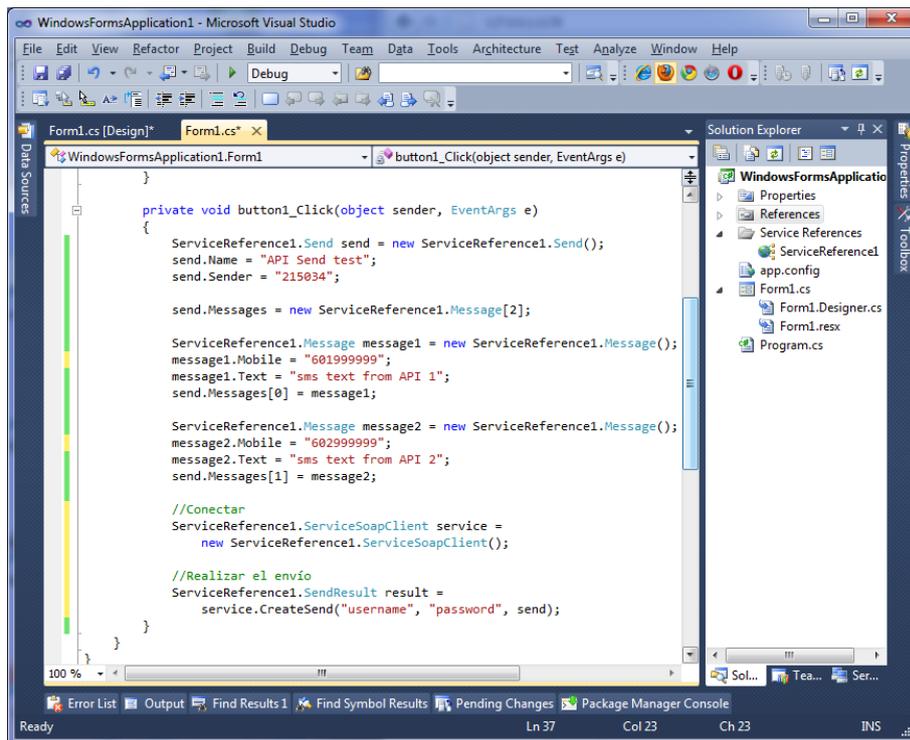
Como ejemplo se muestran capturas de pantalla de una integración usando Microsoft VisualStudio 2010



Importación de métodos 1

Tras la importación de métodos por WSDL se autogenera código para poder utilizar el servicio. Un ejemplo de la realización de un envío sería:

API didimo Marketing - Manual de Integración



Ejemplo de envío con CreateMessage:

```
string id = new Guid().ToString();
string name = "API Send test";
string sender = "miremitente";
string text = "sms text from API";
string mobile = "601999999";
string scheduleDate = "";

//Conectar
ServiceReference1.ServiceSoapClient service =
    new ServiceReference1.ServiceSoapClient();

//Realizar el envío
ServiceReference1.CreateMessageResult result =
    service.CreateMessage(
        "username", "password"
        , id, name, sender, text, mobile, scheduleDate
    );

//Resultado
if(result.ErrorCode != 0)
{
    // acciones ante error
    // ...mostramos error
    Console.WriteLine(String.Format("error: {0} ({1})"
        , result.ErrorCode
        , result.ErrorDescription
    ));
    // ...otras acciones
}
else
{
    // acciones ante éxito
    // ...mostramos id del mensaje y total de SMS
}
```

API didimo Marketing - Manual de Integración

```
Console.WriteLine(String.Format("id: {0}"
    , result.Result.Id
    ));
Console.WriteLine(String.Format("sms: {0}"
    , result.Result.TotalMessages
    ));
}
```

Ejemplo de envío con CreateSend:

```
ServiceReference1.Send send = new ServiceReference1.Send();
send.Name = "API Send test";
send.Sender = "miremitente";

send.Messages = new ServiceReference1.Message[2];

ServiceReference1.Message message1 = new ServiceReference1.Message();
message1.Mobile = "601999999";
message1.Text = "sms text from API 1";
message1.Id = new Guid().ToString();
send.Messages[0] = message1;

ServiceReference1.Message message2 = new ServiceReference1.Message();
message2.Mobile = "602999999";
message2.Text = "sms text from API 2";
message2.Id = new Guid().ToString();
send.Messages[1] = message2;

//Conectar
ServiceReference1.ServiceSoapClient service =
    new ServiceReference1.ServiceSoapClient();

//Realizar el envío
ServiceReference1.SendResult result =
    service.CreateSend("username", "password", send);

//Resultado
if(result.ErrorCode != 0)
{
    // acciones ante error global
    // ...mostramos error
    Console.WriteLine(String.Format("error: {0} ({1})"
        , result.ErrorCode
        , result.ErrorDescription
        ));
    // ...otras acciones
}
else
{
    // acciones ante éxito global
    // ...mostramos los mensajes
    for (int i = 0; i < result.Result.Length; i++)
    {
        if(result.Result[i].ErrorCode != 0)
        {
            // acciones ante error en el mensaje
        }
        else
        {
            // acciones ante éxito en el mensaje
        }
    }
}
```

API didimo Marketing - Manual de Integración

}

Nota: Para la prueba de los ejemplos puede utilizarse, entre otros entornos, Microsoft Visual C# Express disponible de forma gratuita en <http://www.microsoft.com/express>

9. Códigos de Error

Errores del API (ErrorCode, ErrorDescription):

- 0 éxito, no tiene descripción
 - Tiene lugar en cualquier método
- 1 "Error desconocido"
 - Puede tener lugar en cualquier método
- 2 "Error de autenticación: ..."
 - Puede tener lugar en cualquier método
- 3 "Error de validación: ..."
 - Puede tener lugar en cualquier método donde haya otros parámetros además de las credenciales
- 4 "Error de crédito: ..."
 - Puede tener lugar en los métodos CreateMessage y CreateSend
- 5 "Error al preparar los mensajes: ..."
 - Puede tener lugar en los métodos CreateMessage y CreateSend
- 6 "Error al crear el envío: ..."
 - Puede tener lugar en los métodos CreateMessage y CreateSend
- 7 "Mensaje no encontrado: ..."
 - Puede tener lugar en el método GetMessageStatus

10. Estados de mensaje

CODIGO NUMERICO - DESCRIPCION DE ESTADO

- 1 Pendiente Entrega a Operador
- 3 Entregando a Plataforma
- 4 Finalizado. Entrega a Plataforma Fallida *** estado final ***
- 5 Pendiente Entrega a Plataforma
- 7 Supera Logitud Máxima *** estado final ***
- 8 Pendiente Entrega a Operador
- 9 Pendiente Entrega a Operador
- 10 Pendiente Entrega a Operador
- 11 Pendiente Entrega a Operador
- 12 Pendiente Entrega a Destinatario
- 13 Pendiente Entrega a Destinatario. Apagado / Sin Cobertura
- 14 Pendiente Entrega a Destinatario
- 15 Pendiente Entrega a Destinatario
- 16 Pendiente Finalización. Entregado a Destinatario
- 17 Finalizado en Operador *** estado final ***
- 19 Finalizado en Destinatario *** estado final ***
- 20 Finalizado. Rechazado en Operador *** estado final ***
- 21 Finalizado. Expirado en Operador *** estado final ***

API didimo Marketing - Manual de Integración

- 22 Finalizado. Rechazado en Operador *** estado final ***
- 23 Finalizado. Destinatario Incorrecto en Operador *** estado final ***
- 25 Finalizado. Expirado en Plataforma *** estado final ***
- 26 Finalizado. Rechazado en Plataforma *** estado final ***
- 27 Finalizado. Rechazado en Plataforma. Crédito Insuficiente *** estado final ***
- 30 Finalizado. Servicio No Disponible *** estado final ***
- 31 Finalizado. Rechazado en Plataforma *** estado final ***
- 32 Finalizado. Rechazado en Plataforma. Mensaje No Válido *** estado final ***
- 33 Cancelado por duplicidad *** estado final ***
- 34 Cancelado por contener caracteres no válidos *** estado final ***
- 35 Cancelado por Robinson *** estado final ***
- 36 Cancelado por no válido *** estado final ***

11. Caracteres admitidos

El texto solamente admite caracteres GSM7, aunque algunos caracteres no GSM7 (por ejemplo, vocales acentuadas), son reemplazadas por caracteres equivalentes para que sean admitidos (por ejemplo, á es sustituido por a).

Puede encontrar información sobre GSM7 en los siguientes enlaces:

http://en.wikipedia.org/wiki/GSM_03.38

<http://www.developershome.com/sms/gsmAlphabet.asp>

<http://www.devindia.net/software/misc/validgsmchar.pdf>

La longitud máxima del texto es de 306 caracteres, siempre teniendo en cuenta que los caracteres GSM7 Extendidos (^ {} \ ~ [] | €) cuentan como 2 caracteres. Por ejemplo:

- el texto **El carácter (es un paréntesis** ocupa 30 caracteres y se computa por 30
- el texto **El carácter [es un corchete** ocupa 28 caracteres pero se computa por 29

Como se ha mencionado, el texto del mensaje se trata para realizar sustituciones de caracteres no GSM7 de forma que el mensaje sea admisible. Así, determinadas vocales acentuadas (por ejemplo á) son sustituidas por sus versiones sin acento. Y aquellos caracteres no GSM7 que no tienen un equivalente son reemplazados por ? (por ejemplo °).

Reemplazos:

CARACTERES NO GSM7	SUSTITUCIONES
á,â,ã (mayusc. y minusc.)	a
ê,ë,é (mayusc. y minusc.)	e
í,î,ï (mayusc. y minusc.)	i
ô,ó,ô (mayusc. y minusc.)	o
ú,û (mayusc. y minusc.)	u
ý,ÿ (mayusc. y minusc.)	y
ç	Ç
,	,
o a	Estos caracteres se eliminan
otros	?

12. Terminología

MT (Mobile Terminated): Esta terminología es usada para referirse al envío de un mensaje a hacia el Terminal

MO (Mobile Originated): Esta terminología es usada para referirse al envío de un mensaje desde el Terminal.

Push: Se denomina con esta terminología a la acción de enviar un mensaje a un terminal móvil.

Pull: Se denomina con esta terminología a la acción de envío de un mensaje por parte de un Terminal móvil a la plataforma.

Pull - Push: Se denomina con esta terminología a la acción conjunta de envío por parte de un Terminal móvil de un mensaje a la plataforma, y la respuesta de la misma.

13. Anexo

1. La aplicación cliente del servicio web debe tener en cuenta si la salida a Internet requiere o no de un cortafuegos para implementar en la misma la correspondiente funcionalidad de autenticación.
2. Para realizar pruebas de desarrollo póngase en contacto con sopORTEaplicateca@didimo.es