

Manual de uso del API de Web SMS

Cómo enviar SMS con tus propios desarrollos

Índice

1. Cómo usar SMSSend
2. Ejemplos de utilización de SMSSend
3. Webservice SMS

Desde arsys se ha desarrollado un componente que ofrece a los programadores de aplicaciones una herramienta que les **permite realizar el envío de mensajes SMS a móviles desde sus propios desarrollos.**

No solo podrás hacer uso de este componente en tus programas web, sino que podrás utilizarlo como valor añadido en el desarrollo de tus aplicaciones cliente con Visual Basic, Visual C++, Delphi, etc.

Para facilitar el uso de este componente mediante diferentes tecnologías de programación, se han desarrollado **tres versiones diferentes del mismo:**

- Componente ActiveX (SMSCOM.dll).
- Módulo de Perl (SMSSend.pm).
- Librería PHP (SMSSend.inc).
- Webservices.

El servicio Web SMS de arsys te permite usarlo en cualquiera de sus dos modalidades:

- Mediante la **aplicación** Web SMS.
- A través de tus propias aplicaciones, mediante el componente SMSSend.

1. Cómo usar SMSSend

Este componente permite **enviar un mensaje SMS a una lista de destinatarios o programar un nuevo mensaje para su envío en una fecha/hora determinada.**

Su **utilización** es muy sencilla:

- Si dispones de un **Servidor Dedicado**, necesitas permitir tráfico saliente al puerto 8080 TCP del Servidor SMS.
- El primer paso consiste en la **creación de una instancia del objeto** en su script.
- Utilizar uno de los siguientes **métodos** que te proponemos.

Método para realizar un envío.

`setAccount (account as String)`

Establece el usuario (`account`) de la cuenta desde la que se va a realizar el envío. Este dato será la dirección de correo con que activó tu cuenta SMS.

setPwd (pwd as String)

Establece la contraseña de la cuenta desde la que se va a realizar el envío.

setText (text as String)

Establece el texto del mensaje SMS (máximo: 160 caracteres).

setFrom (text as String)

Establece el remite para el SMS (máximo: 11 caracteres). No es obligatorio.

setTo (para as String)

Establece los destinatarios del mensaje (separados por ;).

Método para programar un envío.

setDescriptionEP (text as String)

Establece la descripción del Envío programado.

setDateEP (text as String)

Establece la fecha del Envío programado ("dd/mm/aaaa").

setTimeEP (text as String)

Establece la hora del Envío programado ("hh/mm").

setPeriodEP (text as String)

Establece la periodicidad del Envío programado. Los posibles valores son:

- periodUnica
- periodDiaria
- periodSemanal
- periodMensual
- periodAnual

Método para obtener el estado de envío de un mensaje.

setIdEnvio (text as String)

Establece el identificador del envío del mensaje cuyo estado se quiere consultar. El identificador del envío es devuelto por el método "Send" (ver respuesta del método "Send" más adelante).

Método para obtener los identificadores y estado de los SMS enviados en un intervalo de tiempo.

setFechaIni (text as String)

Establece la fecha y hora de inicio para la consulta. El formato debe ser "DD/MM/AAAA HH:MM:SS".

setFechaFin (text as String)

Establece la fecha y hora de inicio para la consulta. El formato debe ser "DD/MM/AAAA HH:MM:SS".

Métodos para el Envío - Programación - Consultas.

(respuesta as String) = Send

Envía el mensaje SMS. El método Send devuelve como respuesta lo siguiente:

```
<SendSMS>
  <result>(OK o KO)</result>
  <description>Descripción del resultado</description>
  <credit>Credito restante</credit>
  <idenvio>ID del envío</idenvio>
</SendSMS>
```

O, si ha habido un error en la conexión y no se ha podido obtener respuesta, el valor:

-1

(respuesta as String) = Program

Programa el mensaje SMS. El método Program devuelve como respuesta lo siguiente:

```
<ProgramSMS>
  <result>(OK o KO)</result>
  <description>Descripción del resultado</description>
  <credit>Credito restante</credit>
</Program>
```

O, si ha habido un error en la conexión y no se ha podido obtener respuesta, el valor:

-1

(respuesta as String) = Estado

Consulta el estado de un envío. El método Estado devuelve como respuesta el estado de entrega al operador y al terminal de cada uno de los mensajes incluidos en el envío del modo siguiente:

```
<SendSMS>
  <result>OK</result>
  <description>Estado del envío obtenido</description>
  <envio>
    <item>
      <phone>34666666666</phone>
      <estado_operador>entregado</estado_operador>
      <estado_terminal>entregado</estado_terminal>
    </item>
    <item>
      <phone>34666666667</phone>
      <estado_operador>entregado</estado_operador>
      <estado_terminal>entregado</estado_terminal>
    </item>
  </envio>
</SendSMS>
```

O, si ha habido un error en la conexión y no se ha podido obtener respuesta, el valor:

-1

(respuesta as String) = Listado

Consulta el listado de SMS enviados entre una fecha y hora de inicio y otra de fin. La duración del intervalo no debe ser mayor a 24 horas. El método Listado devuelve como respuesta el ID del envío, teléfono, fecha y hora y estado de entrega al operador y al terminal de cada uno de los SMS enviados en dicho intervalo del modo siguiente:

```
<SendSMS>
  <result>OK</result>
  <description>Estado del envío obtenido</description>
  <listado>
    <item>
      <idenvio>9999991</idenvio>
      <phone>3466666666</phone>
      <timestamp>20/10/2009
      18:46:16</timestamp>
      <estado_operador>entregado</estado_operador>
      <estado_terminal>entregado</estado_terminal>
    </item>
    <item>
      <idenvio>9999991</idenvio>
      <phone>34666666667</phone>
      <timestamp>20/10/2009
      18:46:16</timestamp>
      <estado_operador>entregado</estado_operador>
      <estado_terminal>entregado</estado_terminal>
    </item>
    <item>
      <idenvio>9999992</idenvio>
      <phone>3466666666</phone>
      <timestamp>20/10/2009
      20:30:00</timestamp>
      <estado_operador>entregado</estado_operador>
      <estado_terminal>entregado</estado_terminal>
    </item>
  </listado>
</SendSMS>
```

O, si ha habido un error en la conexión y no se ha podido obtener respuesta, el valor:

-1

(respuesta as String) = ListadoEP

Consulta el listado de SMS programados. Permite de manera opcional especificar una fecha y hora de inicio y otra de fin. El método ListadoEP devuelve como respuesta el ID del envío, el destinatario, el estado del envío programado, la fecha en la que está programado el primer envío, la fecha en la que está programado el próximo envío, la fecha en la que se creó el envío programado, el nº de reintentos, la descripción del envío y el error si es que se ha producido del modo siguiente:

```
<listadoep>
  <result>ok</result>
  <description>estado de envíos programados
  obtenidos</description>
  <listado>
    <item>
      <idenvio>9999991</idenvio>
      <destinatarios>3466666666</destinatarios>
      <estado>finalizado</estado>
      <fecha_primer_envio>20/10/2010
      18:46:16</fecha_primer_envio>
      <fecha_proximo_envio>20/10/2010
      18:46:16</fecha_proximo_envio>
      <fecha_creacion_envio>20/10/2010
      14:46:16</fecha_creacion_envio>
      <reintentos>1</reintentos>
      <descripcion>descripción del
      envío</descripcion>
      <error></error>
    </item>
    <item>
      <idenvio>9999992</idenvio>
      <destinatarios>34666666667</destinatarios>
      <estado>finalizado</estado>
      <fecha_primer_envio>20/10/2010
      18:46:16</fecha_primer_envio>
      <fecha_proximo_envio>20/10/2010
      18:46:16</fecha_proximo_envio>
      <fecha_creacion_envio>20/10/2010
      14:46:16</fecha_creacion_envio>
      <reintentos>1</reintentos>
      <descripcion>descripción del
      envío</descripcion>
      <error></error>
    </item>
  </listado>
</listadoep>
```

```

    </listado>
  </listadoep>

```

O, si ha habido un error en la conexión y no se ha podido obtener respuesta, el valor:

```
-1
```

(respuesta as String) = ListadoEnvios

Consulta el listado de SMS enviados. Permite, de manera opcional, filtrar la búsqueda por una fecha y hora de inicio y otra de fin, por un destinatario y por parte del contenido de un mensaje. El método ListadoEnvios devuelve como respuesta el ID del envío, el destinatario, la fecha del envío, el número de destinatarios del envío y el contenido del mensaje del modo siguiente:

```

<ListadoSMS>
  <result>ok</result>
  <description>estado de envíos programados
  obtenidos</description>
  <listado>
    <item>
      <idenvio>9999991</idenvio>
      <phone>3466666666</phone>
      <timestamp>20/10/2010 18:46:16</timestamp>
      <num_envios>1</num_envios>
      <text>Texto de mensaje</text>
    </item>
    <item>
      <idenvio>9999992</idenvio>
      <phone>34666666667</phone>
      <timestamp>23/10/2010 18:46:16</timestamp>
      <num_envios>1</num_envios>
      <text>Texto de mensaje 2</text>
    </item>
  </listado>
</ListadoSMS>

```

O, si ha habido un error en la conexión y no se ha podido obtener respuesta, el valor:

```
-1
```


(descripcion as String) = getDescription

Recupera lo que hay entre <description></description>.

(credit as String) = getCredit

Recupera lo que hay entre <credit></credit> tras enviar un SMS con "Send" o programarlo con "Program".

(envio as String) = getIdEnvio

Recupera lo que hay entre <idenvio></idenvio> al realizar el envío de un sms con "Send".

(envio as String) = getIdEnvio

Recupera lo que hay entre <idenvio></idenvio> al realizar el envío de un SMS con "Send".

Métodos opcionales - No es necesaria su utilización para realizar el envío SMS.

(account as String) = getAccount

Recupera el usuario de la cuenta desde la que se va a realizar el envío.

(pwd as String) = getPwd

Recupera la contraseña la cuenta desde la que se va a realizar el envío.

(text as String) = getText

Recupera el texto del mensaje SMS. Máximo: 160 caracteres.

(para as String) = getTo

Recupera los destinatarios del mensaje (separados por ;).

(Remite as String) = getFrom

Recupera el remite configurado para el mensaje.

(DescriptionEP as String) = getDescriptionEP

Recupera la descripción configurada para el envío programado.

(DateEP as String) = getDateEP

Recupera la fecha configurada para el envío programado.

(TimeEP as String) = getTimeEP

Recupera la hora configurada para el envío programado.

(PeriodEP as String) = getPeriodEP

Recupera el ID de la periodicidad configurada para el envío programado.

2. Ejemplos de utilización de SMSSend

A continuación incluimos ejemplos del uso del componente para sus diferentes versiones:

Uso de SMSCOM.dll desde un archivo ASP en VBScript

Envío de mensaje

```
<%@ LANGUAGE=VBScript %>
<%
Set objSendSMS= Server.CreateObject("SMSCOM.SMSSend")
'Defino las propiedades
objSendSMS.setAccount ("micuenta@micuenta.com")
objSendSMS.setPwd ("mipwd")
objSendSMS.setText ("Este es mi texto")
objSendSMS.setTo ("6666666666")
objSendSMS.setFrom ("mi remite") ' Parámetro opcional
```

```

'Recupero los valores establecidos
Account = objSendSMS.getAccount
Pwd = objSendSMS.getPwd
Texto = objSendSMS.getText
Para = objSendSMS.getTo
Remite = objSendSMS.getFrom
'Envío del mensaje
resultado = objSendSMS.Send
'Resultado de la operación
getResult = objSendSMS.getResult
getDescription = objSendSMS.getDescription
getCredit = objSendSMS.getCredit
getIdEnvio = objSendSMS.getIdEnvio
Response.write "El resultado es "& resultado & "<br>"
Response.write "getResult es " & getResult & "<br>"
Response.write "getDescription es "& getDescription & "<br>"
Response.write "getCredit es " & getCredit & "<br>"
Response.write "getIdEnvio es " & getIdEnvio & "<br>"
Set objSendSMS=Nothing
%>

```

Consulta del estado del envío

```

<%@ LANGUAGE=VBScript %>
<%
Set objSendSMS= Server.CreateObject("SMSCOM.SMSSend")
'Defino las propiedades
objSendSMS.setAccount ("micuenta@micuenta.com")
objSendSMS.setPwd ("mipwd")
objSendSMS.setIdEnvio ("9999999")
'Recupero los valores establecidos
Account = objSendSMS.getAccount
Pwd = objSendSMS.getPwd
IdEnvio = objSendSMS.getIdEnvio
'Consulto el estado
resultado = objSendSMS.Estado
'Resultado de la operación
getResult = objSendSMS.getResult
getDescription = objSendSMS.getDescription
getEstado = objSendSMS.getEstado
Response.write "El resultado es "& resultado & "<br>"
Response.write "getResult es " & getResult & "<br>"
Response.write "getDescription es "& getDescription & "<br>"

```

```
Response.write "getEstado es " & getEstado & "<br>"
Set objSendSMS=Nothing
%>
```

Programación de mensaje

```
<%@ LANGUAGE=VBScript %>
<%
Set objSendSMS= Server.CreateObject("SMSCOM.SMSSend")
'Defino las propiedades
objSendSMS.setAccount ("micuenta@micuenta.com")
objSendSMS.setPwd ("mipwd")
objSendSMS.setText ("Este es mi texto")
objSendSMS.setTo ("666666666")
objSendSMS.setFrom ("mi remite") ' Parámetro opcional
objSendSMS.setDescriptionEP ("Descripción del envío")
objSendSMS.setDateEP ("12/10/2005")
objSendSMS.setTimeEP ("10:25")
objSendSMS.setPeriodEP ("periodSemanal")
'Recupero los valores establecidos
Account = objSendSMS.getAccount
Pwd = objSendSMS.getPwd
Texto = objSendSMS.getText
Para = objSendSMS.getTo
Remite = objSendSMS.getFrom
DescriptionEP = objSendSMS.getDescriptionEP
DateEP = objSendSMS.getDateEP
TimeEP = objSendSMS.getTimeEP
PeriodEP = objSendSMS.getPeriodEP
'Envío del mensaje
resultado = objSendSMS.Program
'Resultado de la operación
getResult = objSendSMS.getResult
getDescription = objSendSMS.getDescription
getCredit = objSendSMS.getCredit
Response.write "El resultado es "& resultado & "<br>"
Response.write "getResult es " & getResult & "<br>"
Response.write "getDescription es "& getDescription & "<br>"
Response.write "getCredit es " & getCredit & "<br>"
Set objSendSMS=Nothing
%>
```

Consulta de los mensajes por intervalo de tiempo

```

<%@ LANGUAGE=VBScript %>
<%
Set objSendSMS= Server.CreateObject("SMSCOM.SMSSend")
'Defino las propiedades
objSendSMS.setAccount ("micuenta@micuenta.com")
objSendSMS.setPwd ("mipwd")
objSendSMS.setFechaIn ("14/04/2009 00:00:00")
objSendSMS.setFechaFin ("14/04/2009 23:59:59")
'Recupero los valores establecidos
Account = objSendSMS.getAccount
Pwd = objSendSMS.getPwd
'Obtengo listado
resultado = objSendSMS.Listado
'Resultado de la operación
getResult = objSendSMS.getResult
getDescription = objSendSMS.getDescription
getListado = objSendSMS.getListado
Response.write "El resultado es "& resultado & "<br>"
Response.write "getResult es " & getResult & "<br>"
Response.write "getDescription es "& getDescription & "<br>"
Response.write "getListado es " & getListado & "<br>"
Set objSendSMS=Nothing
%>

```

Uso de SMSSend.pm desde un CGI en Perl

Envío de mensaje

```

use Arsys::SMSSend;
my $SMS = new Arsys::SMSSend;
#Defino las propiedades
$SMS->setAccount("micuenta\@micuenta.com ");
$SMS->setPwd("mipwd");
$SMS->setTo("666666666");
$SMS->setText("Este es mi texto");
$SMS->setFrom("mi remite"); # Parámetro opcional
#Recupero los valores establecidos
$Account = $SMS->getAccount;
$Pwd = $SMS->getPwd;
$To = $SMS->getTo;
$Text = $SMS->getText;
$Remite = $SMS->getFrom;

```

```

#Envío del mensaje
$resultado = $SMS->Send;
#Resultado de la operación
$resultado = $SMS->getResult;
$resultado = $SMS->getDescription;
$resultado = $SMS->getCredit;
$resultado = $SMS->getIdEnvio;
print "El resultado es $resultado\n";
print "getResult es $getResult \n";
print "getDescription es $getDescription \n";
print "getCredit es $getCredit \n";
print "getIdEnvio es $getIdEnvio \n";

```

Consulta del estado del envío

```

use Arsys::SMSSend;
my $SMS = new Arsys::SMSSend;
#Defino las propiedades
$resultado = $SMS->setAccount("micuenta\@micuenta.com ");
$resultado = $SMS->setPwd("mipwd");
$resultado = $SMS->setIdEnvio("9999999");
#Recupero los valores establecidos
$resultado = $SMS->getAccount;
$resultado = $SMS->getPwd;
$resultado = $SMS->getIdEnvio;
#Consulta el estado del envío
$resultado = $SMS->Estado;
#Resultado de la operación
$resultado = $SMS->getResult;
$resultado = $SMS->getDescription;
$resultado = $SMS->getEstado;
print "El resultado es $resultado\n";
print "getResult es $getResult \n";
print "getDescription es $getDescription \n";
print "getEstado es $getEstado \n";

```

Programación de mensaje

```

use Arsys::SMSSend;
my $SMS = new Arsys::SMSSend;
#Defino las propiedades
$resultado = $SMS->setAccount("micuenta\@micuenta.com ");

```

```

$SMS->setPwd("mipwd");
$SMS->setTo("666666666");
$SMS->setText("Este es mi texto");
$SMS->setFrom("mi remite"); # Parámetro opcional
$SMS->setDescriptionEP ("Descripción del envío");
$SMS->setDateEP ("12/10/2005");
$SMS->setTimeEP ("10:25");
$SMS->setPeriodEP (periodSemanal);
#Recupero los valores establecidos
$Account = $SMS->getAccount;
$Pwd = $SMS->getPwd;
$To = $SMS->getTo;
$Text = $SMS->getText;
$Remite = $SMS->getFrom;
$DescriptionEP = $SMS->getDescriptionEP;
$DateEP = $SMS->getDateEP;
$TimeEP = $SMS->getTimeEP;
$PeriodEP = $SMS->getPeriodEP;
#Envío del mensaje
$resultado = $SMS->Program;
#Resultado de la operación
$resultado = $SMS->getResult;
$resultado = $SMS->getDescription;
$resultado = $SMS->getCredit;
print "El resultado es $resultado\n";
print "getResult es $getResult \n";
print "getDescription es $getDescription \n";
print "getCredit es $getCredit \n";

```

Consulta de los mensajes por intervalo de tiempo

```

use Arsys::SMSSend;
my $SMS = new Arsys::SMSSend;
#Defino las propiedades
$SMS->setAccount("micuenta\@micuenta.com ");
$SMS->setPwd("mipwd");
$SMS->setFechaIni("14/04/2009 00:00:00");
$SMS->setFechaFin("14/04/2009 23:59:59");
#Recupero los valores establecidos
$Account = $SMS->getAccount;
$Pwd = $SMS->getPwd;
#Envío del mensaje
$resultado = $SMS->Listado;

```

```
#Resultado de la operación
$resultado = $SMS->getResult;
$resultado = $SMS->getDescription;
$resultado = $SMS->getListado;
print "El resultado es $resultado\n";
print "getResult es $getResult \n";
print "getDescription es $getDescription \n";
print "getListado es $getListado \n";
```

Consulta de los mensajes programados

```
use Arsys::SMSSend;
my $SMS = new Arsys::SMSSend;
#Defino las propiedades
$SMS->setAccount("micuenta\@micuenta.com ");
$SMS->setPwd("mipwd");
$SMS->setFechaIni("14/04/2009 00:00:00");
$SMS->setFechaFin("14/04/2009 23:59:59");
#Recupero los valores establecidos
$Account = $SMS->getAccount;
$Pwd = $SMS->getPwd;
#Consulta de los envíos programados
$resultado = $SMS->ListadoEP;
#Resultado de la operación
$resultado = $SMS->getResult;
$resultado = $SMS->getDescription;
$resultado = $SMS->getListado;
print "El resultado es $resultado\n";
print "getResult es $getResult \n";
print "getDescription es $getDescription \n";
print "getListado es $getListado \n";
```

Consulta de los mensajes enviados

```
use Arsys::SMSSend;
my $SMS = new Arsys::SMSSend;
#Defino las propiedades
$SMS->setAccount("micuenta\@micuenta.com ");
$SMS->setPwd("mipwd");
$SMS->setTo("666666666");
$SMS->setText("texto de ejemplo");
$SMS->setFechaIni("14/04/2009 00:00:00");
```



```

$SMS->setFechaFin("14/04/2009 23:59:59");
#Recupero los valores establecidos
$Account = $SMS->getAccount;
$Pwd = $SMS->getPwd;
$To = $SMS->getTo;
$Text = $SMS->getText;
#Consulta de los envíos realizados
$resultado = $SMS->ListadoEnvios;
#Resultado de la operación
$resultado = $SMS->getResult;
$resultado = $SMS->getDescription;
$resultado = $SMS->getListado;
print "El resultado es $resultado\n";
print "getResult es $getResult \n";
print "getDescription es $getDescription \n";
print "getListado es $getListado \n";

```

Uso de SMSSend.inc desde un script en PHP

Envío de mensaje

```

<?php
require "SMSSend.inc";
//Defino las propiedades
$testsms=new smsItem;
$testsms->setAccount("micuenta@micuenta.com");
$testsms->setPwd("mipwd");
$testsms->setTo("666666666");
$testsms->setText("Este es mi texto ");
$testsms->setFrom("mi remite"); // Parámetro opcional
//Recupero los valores establecidos
$Account = $testsms->getAccount();
$Pwd = $testsms->getPwd();
$To = $testsms->getTo();
$Text = $testsms->getText();
$Remite = $testsms->getFrom();
//Envío del mensaje
$resultado = $testsms->Send();
//Resultado de la operación
$resultado = $testsms->getResult();
$resultado = $testsms->getDescription();
$resultado = $testsms->getCredit();
$resultado = $testsms->getIdEnvio();

```

```

print_r($resultado);
print_r("<BR>");
print_r($getResult);
print_r("<BR>");
print_r($getDescription);
print_r("<BR>");
print_r($getCredit);
print_r("<BR>");
print_r($getIdEnvio);
print_r("<BR>");
?>

```

Consulta del estado del envío

```

<?php
require "SMSSend.inc";
//Defino las propiedades
$testsms=new smsItem;
$testsms->setAccount("micuenta@micuenta.com");
$testsms->setPwd("mipwd");
$testsms->setIdEnvio("9999999");
//Recupero los valores establecidos
$Account = $testsms->getAccount();
$Pwd = $testsms->getPwd();
$IdEnvio = $testsms->getIdEnvio();
//Consulto el estado
$resultado = $testsms->Estado();
//Resultado de la operación
$result = $testsms->getResult();
$resultDescription = $testsms->getDescription();
$resultEstado = $testsms->getEstado();
print_r($resultado);
print_r("<BR>");
print_r($getResult);
print_r("<BR>");
print_r($getDescription);
print_r("<BR>");
print_r($getEstado);
print_r("<BR>");
?>

```

Programación de mensaje

```

<?php
require "SMSSend.inc";
//Defino las propiedades
$testsms=new smsItem;
$testsms->setAccount("micuenta@micuenta.com");
$testsms->setPwd("mipwd");
$testsms->setTo("666666666");
$testsms->setText("Este es mi texto ");
$testsms->setFrom("mi remite"); // Parámetro opcional
$testsms->setDescriptionEP ("Descripción del envío");
$testsms->setDateEP ("12/10/2005");
$testsms->setTimeEP ("10:25");
$testsms->setPeriodEP (periodSemanal);
//Recupero los valores establecidos
$Account = $testsms->getAccount();
$Pwd = $testsms->getPwd();
$To = $testsms->getTo();
$Text = $testsms->getText();
$Remite = $testsms->getFrom();
$DescriptionEP = $testsms->getDescriptionEP;
$DateEP = $testsms->getDateEP;
$TimeEP = $testsms->getTimeEP;
$PeriodEP = $testsms->getPeriodEP;
//Envío del mensaje
$resultado = $testsms->Program();
//Resultado de la operación
$getResult = $testsms->getResult();
$getDescription = $testsms->getDescription();
$getCredit = $testsms->getCredit();
print_r($resultado);
print_r("<BR>");
print_r($getResult);
print_r("<BR>");
print_r($getDescription);
print_r("<BR>");
print_r($getCredit);
print_r("<BR>");
?>

```

Consulta de los mensajes por intervalo de tiempo

```

<?php
require "SMSSend.inc";
//Defino las propiedades
$testsms=new smsItem;
$testsms->setAccount("micuenta@micuenta.com");
$testsms->setPwd("mipwd");
$testsms->setFechaIni("14/04/2009 00:00:00");
$testsms->setFechaFin("14/04/2009 23:59:59");
//Recupero los valores establecidos
$Account = $testsms->getAccount();
$Pwd = $testsms->getPwd();
//Envío del mensaje
$resultado = $testsms->Listado();
//Resultado de la operación
$getResult = $testsms->getResult();
$getDescription = $testsms->getDescription();
$getListado = $testsms->getListado();
print_r($resultado);
print_r("<BR>");
print_r($getResult);
print_r("<BR>");
print_r($getDescription);
print_r("<BR>");
print_r($getListado);
print_r("<BR>");
?>

```

Consulta de los mensajes programados

```

<?php
require "SMSSend.inc";
//Defino las propiedades
$testsms=new smsItem;
$testsms->setAccount("micuenta@micuenta.com");
$testsms->setPwd("mipwd");
$testsms->setFechaIni("14/04/2009 00:00:00");
$testsms->setFechaFin("14/04/2009 23:59:59");
//Recupero los valores establecidos
$Account = $testsms->getAccount();
$Pwd = $testsms->getPwd();
//Consulta de los envíos programados
$resultado = $testsms->ListadoEP();

```

```
//Resultado de la operación
$resultado = $testsms->getResult();
$descripcion = $testsms->getDescription();
$listado = $testsms->getListado();
print_r($resultado);
print_r("
");
print_r($descripcion);
print_r("
");
print_r($listado);
print_r("
");
?>
```

Consulta de los mensajes enviados

```
<?php
require "SMSSend.inc";
//Defino las propiedades
$testsms=new smsItem;
$testsms->setAccount("micuenta@micuenta.com");
$testsms->setPwd("mipwd");
$testsms->setTo("666666666");
/$testsms->setText("texto de prueba ");
$testsms->setFechaIni("14/04/2009 00:00:00");
$testsms->setFechaFin("14/04/2009 23:59:59");
//Recupero los valores establecidos
$Account = $testsms->getAccount();
$Pwd = $testsms->getPwd();
//Consulta de los envíos realizados
$resultado = $testsms->ListadoEnvios();
//Resultado de la operación
$resultado = $testsms->getResult();
$descripcion = $testsms->getDescription();
$listado = $testsms->getListado();
print_r($resultado);
print_r("<BR>");
print_r($descripcion);
print_r("<BR>");
```

```
print_r($getDescription);  
print_r("<BR>");  
print_r($getListado);  
print_r("<BR>");  
?>
```

3. Webservice SMS

Todos los métodos tienen como datos obligatorios:

- Login.
- Contraseña.
- Método de envío de mensajes.
- Programar mensajes.

El resto de parámetros (fechas, destinatarios, textos) son opcionales para el filtrado del resultado.

Envío de SMS

```
https://sms.arsys.es/smsarsys/accion/enviarSms2.jsp?id=micuenta@micuenta.com&phoneNumber=666666666&psw=mipwd&textSms=texto del mensaje&remite=usuario
```

Programación de SMS

```
https://sms.arsys.es/smsarsys/accion/ProgramarSmsEP.jsp?id=micuenta@micuenta.com&phoneNumber=666666666&psw=mipwd&textSms=texto del mensaje&descripcion=descripcion&fecha=12/11/2010&hora=12:40&periodicidadEnvio=1&remite=usuario1
```

Consultar estado de un envío

```
https://sms.arsys.es/smsarsys/accion/estadoSms.jsp?id=micuenta@micuenta.com&psw=mipwd&idenvio=123456
```

Listado de envíos por intervalo de tiempo

[https://sms.arsys.es/smsarsys/accion/listadoSms.jsp?id=micuenta@micuenta.com&psw=mipwd&fechaini=12/11/2010 12:00:00&fechafin=12/11/2010 13:00:00](https://sms.arsys.es/smsarsys/accion/listadoSms.jsp?id=micuenta@micuenta.com&psw=mipwd&fechaini=12/11/2010%2012:00:00&fechafin=12/11/2010%2013:00:00)

Listado de envíos programados

[https://sms.arsys.es/smsarsys/accion/listadoEP.jsp?id=micuenta@micuenta.com&psw=mipwd&fechaini=12/11/2010 10:00:00&fechafin=12/11/2010 14:00:00](https://sms.arsys.es/smsarsys/accion/listadoEP.jsp?id=micuenta@micuenta.com&psw=mipwd&fechaini=12/11/2010%2010:00:00&fechafin=12/11/2010%2014:00:00)

Listado de envíos realizados

[https://sms.arsys.es/smsarsys/accion/listadoEnvioSms.jsp?id=micuenta@micuenta.com&psw=mipwd&fechaini=12/11/2010 10:00:00&fechafin=15/11/2010 10:00:00&phone=666666666&textSms=texto de ejemplo](https://sms.arsys.es/smsarsys/accion/listadoEnvioSms.jsp?id=micuenta@micuenta.com&psw=mipwd&fechaini=12/11/2010%2010:00:00&fechafin=15/11/2010%2010:00:00&phone=666666666&textSms=texto%20de%20ejemplo)