

[Introducción](#)[Requisitos](#)[Funciones](#)[Params](#)[To](#)[CC](#)[BCC](#)[Attachments](#)[Envío en html](#)

## Introducción

Esta Velneo Open App, componente vSendMail, permite el envío de emails desde la parte cliente (vClient). Muy útil para el envío de notificaciones, alertas, etc.

Para ello, hace uso de la aplicación de licencia GNU [mailsend](#).

Permite el envío de emails usando SSL, TLS, adjuntos, correos en html, etc. Sencillo y rápido y sin necesidad de instalar o registrar dlls.

Por favor, no utilizar esta aplicación para el envío de spam o emails masivos.

Recomendamos revisar el tutor vSendMail que se adjunta junto con el componente, donde se pone en práctica este componente y que puede ayudar a entender el uso del mismo.

## Requisitos

Este componente está disponible tanto para sistemas operativos Windows, Linux y Mac. Ha sido probado en Windows XP, Windows 7, Ubuntu 11 y OSX Mavericks. Si necesitas el binario para una distribución diferente, en la web del autor están los fuentes junto con las instrucciones para su compilación.

El envío del correo debe hacerse desde un proceso en el cliente, ya sea en primer o en segundo plano.

El binario para linux incluido se compiló usando la librería libssl 0.9.8. Es necesario que esa librería esté en el sistema. Para instalarla ejecutar desde una ventana de terminal

```
sudo apt-get install libssl0.9.8
```

Para la gestión de los múltiples parámetros que usa la Open App se hace uso de los diccionarios incluidos en el componente vTools. Es por ello que la solución hereda de vUtiles 1.1.

## Funciones

Sólo es necesario una función para realizar el envío, aunque para la definición de los parámetros usaremos algunas otras.

La función para el envío es:

```
fun:SENDMAIL@vSendMail.dat(PARAMS, TO, CC, BCC, ATTACHMENTS)
```

A continuación explicamos cada uno de los parámetros y cómo darles valor

## Params

Para la definición de los parámetros haremos uso de la función siguiente:

```
set PARAMS, fun:SET_PARAM@vSendMail.dat(PARAMS, <param>, [value])
```

donde <param> es uno de los siguientes y value puede ser opcional (dependiendo de si el parámetro lleva o no valor).

Los parámetros marcados con \* son obligatorios:

-smtp hostname/IP*	del servidor SMTP
-port SMTP port	Puerto SMTP
-d domain	Nombre del dominio para el comando HELO/EHLO
+D	No añadir la fecha a la cabecera
-f email*	email del remitente
-sub subject	Asunto

-cs character set	De los adjuntos en texto plano (por defecto es us-ascii)
-M "una linea"	Una linea del cuerpo del mensaje en texto plano
-name "Full name"	Añade el nombre completo para el remitente en la cabecera del correo
-w	Espera por un CR después de enviar el correo
-rt email	Añade Reply-to a la cabecera (responder a)
-rrr email	Se solicita que los acuses de recibo se envíen a esta dirección (no todos los servidores SMTP soportan esta opción)
-ssl	SMTP sobre SSL
-starttls	Comprueba si el servidor soporta STARTTLS. Si es así, lo usa.
-auth	Intenta la autenticación en este orden: CRAM-MD5, LOGIN, PLAIN. Usar esta como primera opción.
-auth-cram-md5	Usa autenticación CRAM-MD5
-auth-plain	Usa autenticación AUTH-PLAIN
-auth-login	Usa autenticación AUTH-LOGIN
-user username	Nombre de usuario para la autenticación ESMTP
-pass username	Password para la autenticación ESMTP

## To

Dependiendo de cómo realizamos la aplicación, es posible que tengamos los destinatarios en una sola variable separados por comas o en una tabla.

En el primero de los casos es tan sencillo como utilizar esa variable directamente en la llamada a la función SENDMAIL.

En el segundo de los casos, contamos con la siguiente función:

```
set TO, fun:ADD_TO@vSendMail.dat(TO, "email@test.com")
```

Si esa función la metemos dentro de un recorrer lista, pasando como segundo parámetro el campo donde está el email, al final tendremos en la variable TO la lista de destinatarios separados por comas.

Recordemos que este campo es **obligatorio**.

## CC

Se aplica exactamente la misma filosofía que para el valor de TO.

En este caso la función a utilizar es:

```
set CC, fun:ADD_CC@vSendMail.dat(CC, "email@test.com")
```

## BCC

Se aplica exactamente la misma filosofía que para el valor de TO.

En este caso la función a utilizar es:

```
set BCC, fun:ADD_BCC@vSendMail.dat(BCC, "email@test.com")
```

## Attachments

*Lo primero es asegurarnos de que llamamos en el inicio de nuestra aplicación al proceso CHECK\_MIME\_TYPES. Este proceso se encarga de cargar por primera vez la tabla de tipos mime.*

Para incluir adjuntos dentro de nuestro correo utilizaremos la función:

```
set ATTACHMENTS, fun:ADD_ATTACH@vSendMail.dat(ATTACHMENTS, file, 0)
```

El parámetro file indica el fichero a adjuntar. Debe utilizar la nomenclatura estándar de Velneo V7, utilizando la barra / para los directorios.

El tercer parámetro indica si el adjunto se incluye dentro del propio correo (inline) o se adjunta al mismo (ver envío en html).

El componente incluye una tabla con los [mime.types](#) y la función Add\_Attach se encarga de utilizar el mime.type adecuado en función a la extensión del archivo.

## Envío en html

Para realizar un envío de correo en html, el cuerpo html debe existir como fichero en disco y adjuntarlo al correo en forma inline. La llamada a la función sería:

```
set ATTACHMENTS, fun:ADD_ATTACH@vSendMail.dat(ATTACHMENTS, FILE_HTML, 1)
```