



Bisoños Usuarios de GNU/Linux de Mallorca y Alrededores | Bergantells Usuaris de GNU/Linux de Mallorca i Afegitons

Pasarela a MSN, conferencias y SSL con Jabber (33899 lectures)

Per **Miquel Cantallops Mora**, [mcm](http://madteleco.blogspot.com) (<http://madteleco.blogspot.com>)

Creado el 09/09/2003 03:54 modificado el 07/10/2003 00:28

NOTICIA (2003/09/09): Se estrena en el servidor Jabber de **bulmalug.net** una nueva pasarela a la MSN de Hasecorp (la hemos bautizado Hasenet). Y también soporte de SSL entre clientes y servidor como ya se dijo en la bulmailing.

[ACTUALIZACIÓN \(2003/10/27\): Paquetes de las pasarelas y "plugins" para debian.](#)

Indice

- Para empezar.
- [SSL](#).⁽¹⁾
- [Antes de reiniciar el servidor: mensaje broadcast](#).⁽²⁾
- [Pasarelas y MSN](#).⁽³⁾ [Actualizado 2003/10/27](#).
- [Darse de alta a la pasarela de MSN](#).⁽⁴⁾ **Apúntate y emigra al jabber!**
- [Chat y conferencias](#).⁽⁵⁾
- [Referencias y agradecimientos](#).⁽⁶⁾

Jabber es un protocolo de Mensajería Instantánea. Gracias a que **el protocolo es abierto** no estamos sujetos a la ruleta de la fortuna que supone depender de otros sistemas de mensajería con el protocolo de comunicaciones propietario y que puede dejar de funcionar en cualquier momento.

Además, Jabber es **versátil**. Permite chatear entre varios usuarios a través del establecimiento de salas de conferencia, envío de mensajes con el receptor offline, interconexión con otros protocolos de mensajería, directorios (JUD, algo así como unas páginas amarillas de usuarios), etc...

En realidad, la implementación del servidor suele incluir lo básico. Al menos en el caso de [Jabberd](#)⁽⁷⁾, el servidor jabber de código abierto que utilizamos en Bulma. Las funcionalidades extra suelen ofrecerse a través de *Componentes* externas pero situadas también en el lado del servidor. Igualmente hay clientes que ofrecen servicio de encriptación *punto a punto* (mediante GnuPG) a nivel de aplicación, como el Kopete y el Psi.

Para empezar.

Sabemos instalar un servidor jabber "de serie". Algo así como:

```
$ apt-get install jabber
```

A partir de aquí el servidor ya está funcionando. Aunque si queremos que sea público nos conviene configurar el uso de un dominio válido en `/etc/jabber/jabber.xml` (o localización similar).

Dentro de la sección `<service id="sessions"/>` señalamos el dominio válido. Si tenemos una cuenta en un servidor DNS dinámico, el dominio puede ser del estilo `proyecto.dyndns.org` sin problemas, pero tened en consideración que para la resolución de los dominios de cuarto nivel tenéis que activar los "wildcards" en la configuración de la cuenta del dominio dinámico. En general, para cada servicio que se ofrezca en el servidor debe



disponerse de un subdominio válido. En el jabber de Bulma se usa el dominio `bulmalug.net` :

```
<host><jabberd:cmdline flag="h">bulmalug.net</jabberd:cmdline></host>
```

SSL.

Jabber nos permite el uso de *Secure Sockets Layer* en las comunicaciones entre los clientes y el servidor, mediante el uso de una clave pública certificada y una clave privada. Por motivos económicos, nosotros mismos nos vamos a certificar nuestra clave pública en el siguiente ejemplo :-P

Ante todo, necesitamos el paquete `openssl` para la generación del par de claves.

```
$ apt-get install openssl
```

Primero generamos el par de claves. Si no nos gusta el algoritmo RSA, puede usarse el que queramos (siempre y cuando siga un esquema de clave pública-privada). Conviene no abusar de la longitud de la clave bajo pena de cargar en exceso el procesador del servidor (!y del cliente!) Además, la longitud de la clave *seguro* que no es el eslabón más débil en el servidor que vaya a montarse }:-)

```
$ openssl req -new -x509 -newkey rsa:1024 -keyout priv.pem -out key.pem
```

Una vez hemos contestado todas las preguntas acerca de la generación de la clave, eliminamos la contraseña para acceder a la clave privada, y agrupamos la clave pública "certificada" con la clave privada:

```
$ openssl rsa -in priv.pem -out priv.pem
$ cat priv.pem >> key.pem
$ rm priv.pem
```

El archivo `key.pem` puede situarse en, por ejemplo, `/etc/jabber/key.pem`. Ya que la clave privada no está protegida con un password, conviene establecer medidas extra mediante el uso de permisos (400) y establecer como dueño del archivo al mismo usuario bajo el cual "jabberd" estará funcionando (usuario *daemon* en nuestro caso).

A continuación debemos realizar una serie de modificaciones en la configuración para activar el uso de SSL.

Dentro de la etiqueta `<load main="jsm">` debemos señalar correctamente el orden de los tipos de *autenticación* que el servidor intentará utilizar con el cliente. Si lo encontráis necesario, podéis comentar la línea que permite la autenticación mediante la transmisión en plano del password de la cuenta:

```
...
<mod_auth_0k>./jsm/jsm.so</mod_auth_0k>
<mod_auth_digest>./jsm/jsm.so</mod_auth_digest>
<mod_auth_plain>./jsm/jsm.so</mod_auth_plain>
...
```

A continuación, y situado dentro del *service* `id="c2s"`, activamos el puerto a usar para las comunicaciones mediante SSL. Debe especificarse una IP, y si queremos que el servicio sea público la IP también debe serlo:

```
<ssl port="5223">130.206.130.95</ssl>
```

El siguiente paso es configurar el *path* al archivo con las claves. Ello se realiza dentro de la etiqueta `<i>`:

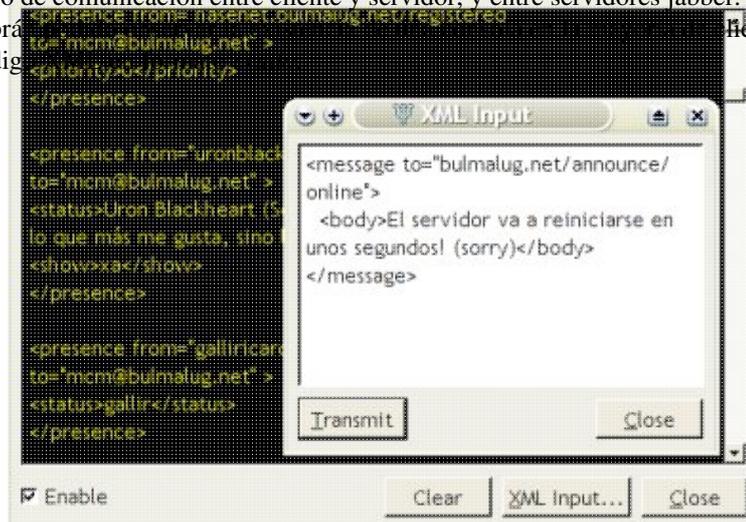
```
<ssl>
  <key ip='130.206.130.95'>/etc/jabber/key.pem</key>
</ssl>
```

A continuación sólo falta reiniciar el servidor y activar el soporte de SSL en los clientes.



Antes de reiniciar el servidor avisemos a los usuarios (mensaje broadcast)

Como ha podido observarse hasta ahora, la configuración sigue una estructura de XML. Pero no sólo la configuración, sino también el protocolo de comunicación entre cliente y servidor, y entre servidores jabber. Seguro que los habituados al jabber habrá notado que el protocolo de comunicación permite la introducción de este código:



Guillem haciendo de BOFH ;-)

Si vamos a reiniciar el demonio, lo mínimo que podemos hacer por nuestros usuarios es avisarlos del suceso mediante un mensaje broadcast. Para ello debemos ser usuario administrador con permisos de escritura globales. El permiso se establece en la configuración mediante:

```
<admin>
  <write>micuenta@dominio.org</write>
</admin>
```

Ahora ya podemos enviar el mensaje en XML al servidor (que viajará del servidor a todos los clientes conectados):

```
<message to="dominio.org/announce/online">
  <body>El servidor va a reiniciarse en unos segundos! (sorry)</body>
</message>
```

Y por último, ya podemos reiniciar el servicio con tranquilidad en el alma:

```
$ /etc/init.d/jabber restart
```

Nota: No hagais esto con vuestra **cuenta usual**. Es un mensaje a cada cliente conectado. ¿Os imagináis la cantidad de respuestas por parte de los usuarios preguntando "¿Què pasa aquí?" Conviene hacerlo desde una cuenta administrativa que impida la recepción directa de mensajes instantáneos si no queremos padecer un ataque DoS.

Pasarelas (t.c.c. Transportes): la MSNetwork (t.c.c. Hasenet)

A continuación nos disponemos a configurar una pasarela a la red de mensajería del Hasefroch. De esta manera podremos controlar nuestros contactos, conversaciones y chats con los amigos del MSN de forma centralizada desde la cuenta de jabber. Además de esta manera no será necesario un cliente "multiprotocolo".

La pasarela a la que se hace referencia a continuación es [MSN-Transport](#)⁽⁸⁾. Según los autores parece ser que dicha pasarela no gestiona correctamente el uso de la memoria. Por ello, nos dispondremos a configurar un demonio aparte que se conectará al demonio principal. La idea es bella, y de esta manera en caso de problemas podremos reiniciar el



demonio encargado de ofrecer servicio hacia MSN sin interferir en el funcionamiento del resto de cuentas jabber.

NOTA: En debian ya hay disponibles paquetes precompilados para añadir las distintas pasarelas existentes mediante **apt-get**.

A continuación siguen una serie de pasos a seguir para la instalación de la componente *msn-transport*:

1. `apt-get install curl libcurl2 libcurl2-dev`. *Curl* permite usar multitud de protocolos para la transmisión de datos entre un cliente y un servidor. Es una especie de *wget* pero más potente y orientado hacia la integración con otros programas.

2. Bajarse el código fuente del jabber. Por ejemplo, mediante `apt-get source jabber`, y en este caso el código queda disponible en `$SRC/jabber-1.4.x/`.

NOTA: Ya te puedes bajar el paquete **jabber-dev** para debian, de manera que no es necesario el paquete de fuentes y puedes abreviar los siguientes pasos ciñiéndote a la instalación de la pasarela (si no lo hay hecho ya con los paquetes precompilados).

3. Ejecutamos `./configure` para generar información necesaria para el "configure" de la pasarela msn. Si vamos a instalar el jabberd a partir del código fuente en lugar de usar el paquete precompilado mediante `apt-get`, `rpm` o lo que queramos, entonces dentro del fichero `./configure` debe modificarse la variable `WANT_SSL` al valor 1 (o no tendremos soporte de SSL).

4. Descomprimir *msn-transport-1.2.8pre10.gz* (versión usada en este artículo) dentro del directorio que contiene el código de jabber. En este ejemplo el código del módulo quedará dentro de `$SRC/jabber-1.4.x/msn-transport-1.2.8pre10/`

5. Situándose dentro de la carpeta anterior, ejecutamos `./configure --with-jabberd=$SRC/jabber-1.4.x/jabberd/`.

6. Ahora ya estamos listos para ejecutar `make` dentro del directorio del *msn-transport*.

7. Si todo ha ido bien ya tenemos compilado el módulo `msntrans.so`, el cual se encuentra en el subdirectorio `$SRC/jabber-1.4.x/msn-transport-1.2.8pre10/src/`. A continuación lo guardamos junto con los demás módulos de jabber. Por ejemplo, en `/usr/lib/jabber/msntrans/msntrans.so`

La parte de compilación ya está lista. Ahora debemos crear la configuración.

1. Seguramente en `/etc/jabber/` disponemos de la configuración del jabber instalada por el `apt-get` del principio. Todo esto no puede ser muy distinto en otras distribuciones. En este directorio creamos el fichero `msn.xml` con el siguiente contenido:

```
<jabber>
  <!-- msnlinker enlaza con los demás procesos de jabber -->
  <service id="msnlinker">
    <uplink/>
    <connect>
      <ip>127.0.0.1</ip>
      <port>1234</port>
      <secret>test</secret>
    </connect>
  </service>

  <service id="hasenet.bulmalug.net">
    <!-- Dominio válido, igual que el resto -->

    <host>chat.hasenet.bulmalug.net</host>
    <!-- Sala de chat para usuarios de MSN-->

    <msntrans xmlns="jabber:config:msntrans">
      <instructions>Enter your MSN Messenger account and password.
        Example: user@hotmail.com. Nickname's optional.
    </msntrans>
  </service>
</jabber>
```



```

        Data will be stored in the server to run transparently.
        The server will import your friends list.</instructions>
<vCard>
    <FN>Transport MSN</FN>
    <DESC>This is MSN Transport</DESC>
    <URL>http://bulma.net/</URL>
</vCard>
<conference id="chat.hasenet.bulmalug.net">
    <invite>More than one user has joined this chat session.
        Join this room to switch into groupchat mode.</invite>
    <notice>
        <join> has become available</join>
        <leave> has left</leave>
    </notice>
</conference>

<!-- Activar si no funciona el servidor por defecto
<servers>
    <ip>64.4.13.56</ip>
    <ip>64.4.13.57</ip>
</servers> -->

<headlines/>
</msntrans>

<!-- Path al módulo a partir de /usr/lib/jabber/~ -->
<load>
    <msntrans>./msntrans/msntrans.so</msntrans>
</load>

</service>

<!-- xdb y logs gestionados por el demonio principal
        mientras esté comentado

<log id="msnlog">
    <file>./msnlog</file>
</log>
<xdb id="xdb">
    <host>hasenet.bulmalug.net</host>
    <load><xdb_file>./xdb_file/xdb_file.so</xdb_file></load>
<xdb_file xmlns="jabber:config:xdb_file">
    <spool>/home/sheath/jabber/jabber2/spool</spool>
</xdb_file>
</xdb>
-->

</jabber>

```

Si queremos que se puedan establecer **conferencias entre cuentas de MSN**, no queda más remedio que configurar las resoluciones de dominio y servicios para *chat.hasenet.xxx.yy*, como subdominio de *hasenet.xxx.yy*.

2. A continuación debe modificarse el `jabber.xml`. En primer lugar añadimos el siguiente código dentro de la sección `<browse>` para que la pasarela a MSN aparezca en la lista de servicios ofrecidos (aquí la llamamos "Transport a MSN"):

```

<service type="msn" jid="hasenet.bulmalug.net" name="Transport a MSN">
    <ns>jabber:iq:gateway</ns>
    <ns>jabber:iq:register</ns>
</service>

```

3. También en `jabber.xml` debemos indicar la aceptación por el demonio principal del proceso que se ocupará de la pasarela a MSN. Para ello, dentro de la etiqueta `<jabber>` incluimos el código expuesto a continuación. Este sistema mediante el cual la componente está funcionando en un proceso independiente es aplicable también a la componente de Conferencias y en general a todas. De este modo añadimos estabilidad (el fallo de uno de los demonios secundarios no provoca la caída de todo el servicio) y la posibilidad de escalar el servicio entre varios hosts.

```

<service id="msnlinker">

```



```
<host>hasenet.bulmalug.net</host>
<host>chat.hasenet.bulmalug.net</host>
<accept>
  <!-- Datos en concordancia con los existentes en msn.xml -->
  <ip>127.0.0.1</ip>
  <port>1234</port>
  <secret>test</secret>
</accept>
</service>
```

4. En principio ya está casi listo. Reiniciamos el servicio: `/etc/init.d/jabber restart`. Y posteriormente cargamos el demonio para el transporte de MSN: `/usr/sbin/jabberd -c /etc/jabber/msn.xml &`

5. Seguramente nos interese automatizar la carga del demonio secundario. Con toda seguridad existen muchas maneras de hacerlo, pero por ejemplo podemos crear el fichero `/etc/init.d/msntrans` (con permisos 755), el cual puede contener un script tan sencillo como el siguiente:

```
#!/bin/sh
su daemon -c "/usr/sbin/jabberd -h bulmalug.net -s /var/lib/jabber -c /etc/jabber/msn.xml" &
```

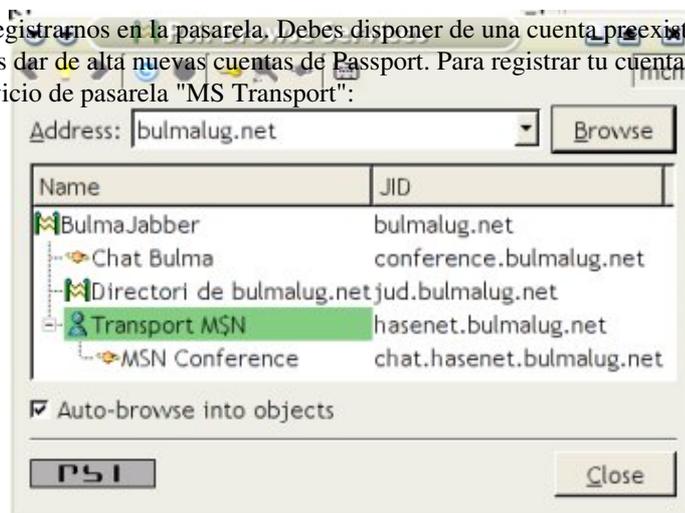
Posteriormente, para que la carga sea automática en el arranque del host, podemos ejecutar `update-rc.d msntrans defaults XX` donde XX es un número mayor que el asignado -por defecto 20- al demonio jabber. Lo anterior nos crea los enlaces necesarios dentro de cada `/etc/rcX.d/`.

Aunque lo anterior es para Debian y para cada distribución de linux el sistema varía ligeramente, la base es la misma. Además, si vais a configurar un servidor de jabber con pasarela incluida seguro que ya sabéis como arrancar automáticamente un proceso al arrancar la máquina ;-)

Darse de alta a la pasarela MSN

Si toda la parte de instalación y configuración ha salido como debía, ya está listo para que los clientes configuren sus cuentas MSN haciendo uso de la pasarela. Para ello debes disponer, primero, de una cuenta jabber. En `bulmalug.net` el registro de las cuentas es gratis, o sea que si no tienes ninguna puedes darte de alta allí :-D

A continuación debemos registrarnos en la pasarela. Debes disponer de una cuenta preexistente para acceder a MSN; desde la pasarela no puedes dar de alta nuevas cuentas de Passport. Para registrar tu cuenta MSN en el servidor jabber, haz doble clic sobre el servicio de pasarela "MS Transport":

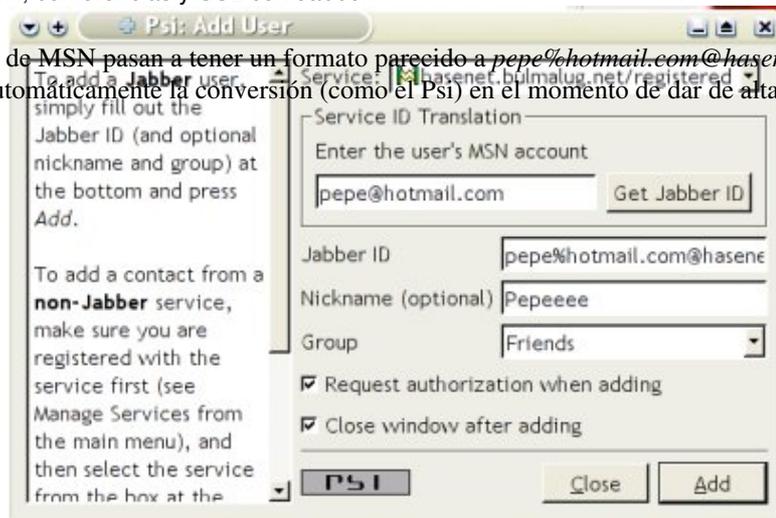


¿Cómo es eso? Hasenet disponible en bulmalug.net, conferencias incluidas!! Están locos estos linuxeros.

Acto seguido debes proporcionar al servidor la dirección de correo y el password **de la cuenta de MSN que ya teníamos** para que se pueda acceder de modo transparente a la red de MSN cada vez que te conectes. La primera vez se encargará de importar todos los contactos que tengamos dados de alta (será necesario verificar y dar permiso a cada uno de ellos para que estén en nuestra lista de contactos).



En general, las cuentas de MSN pasan a tener un formato parecido a `pepe@hotmail.com@hasenet.bulmalug.net`. Hay clientes que realizan automáticamente la conversión (como el Psi) en el momento de dar de alta un nuevo contacto:



La interficie del cliente Psi es muy cómoda. Basta introducir la dirección del messenger para que la convierta automáticamente al formato de "pasarela".

Y atención, porque esta pasarela **SOPORTA** (a su manera) el nuevo protocolo que Microsoft ha venido anunciando desde hace semanas como obligatorio a partir del 15 de octubre del 2003. Además, ya no os saldrá el molesto mensajito de actualización cada vez que os conectéis :-)



¿Lo echaremos de menos? Ná.

Y ahora un poco de rant: evidentemente nada impide que el protocolo vuelva a cambiarse en el futuro, o que microsoft definitivamente prohíba el acceso a clientes de terceros y sin hacer alquiler de su red. Al contrario que con los chats de la compañía (cerrados por ser "criadero de spam y prostitución" según ellos mismos), el messenger es el sistema de mensajería instantánea más usado en España -de momento...-, por lo que podría ser que en el futuro el servicio fuese de pago, o libre pero más limitado. O cualquier otra cosa. Ésto es un FUD (Fear, Uncertainty, Doubt), puesto que nadie (¿tal vez "ellos"?) sabe lo que pasará.

Si eres alguien a quien le gustan menos las sorpresas, ésta puede ser una buena oportunidad para cambiar de sistema de IM y adoptar uno que siga estándares libres.



Desuscribirse de la pasarela:

Puedes hacerlo de dos maneras. La primera y en caso de que exista, a través de las opciones que te proporciona la interficie de tu cliente de jabber. Normalmente basta con hacer click [con el botón derecho] en el agente **hasenet.bulmalug.net**, y seleccionar la opción de "borrar" o "desuscribir". La segunda manera es enviando directamente el código XML al servidor, haciendo uso de la consola XML de tu cliente. El código es el siguiente:

```
<iq type="set">
<query xmlns="jabber:iq:roster" >
<item subscription="remove" jid="hasenet.bulmalug.net/registered" />
</query></iq>
```

Salas de Chat

En este apartado vamos a configurar el soporte de salas de conferencia multiusuario. Tal como se ha visto, la componente *msn-trans* incluye soporte para chat. Sin embargo es muy recomendable instalar otra componente extra pensada para la conferencia entre usuarios del jabber. Aún a sabiendas de que *para realizar un chat con los contactos de MSN deberemos utilizar el servicio de conferencia configurado dentro de la componente que realiza de pasarela (chat.hasenet.bulmalug.net)*.

Por si todavía no habéis quedado convencidos, sabed que los mensajes enviados a la pasarela de Hasenet son reenviados, precisamente, hacia allíiii muy lejos, al otro lado del mar a los servidores de MSN y sin ningún tipo de encriptación ni protección. Mientras que los chats realizados a través del servicio de conferencias que vamos a configurar siguen un estándar bien fijado y no tienen por qué viajar a ningún lado si no hace falta (sobretudo con chats entre contactos del mismo servidor). Además, para algo nos hemos molestado en configurar la encriptación cliente-servidor por SSL, ¿verdad?

Estos son los pasos a seguir para conseguir el funcionamiento del servicio de conferencias:

NOTA: Al igual que con las pasarelas, también hay ya paquete debian oficial para el plugin de conferencias.

1. En primer lugar, y siguiendo el esquema explicado en la configuración de la pasarela, necesitamos el código fuente del jabber. Lo depositamos en `/var/cache/apt-build/build/jabber-1.4.2/` o donde aplique, si es que no lo teníamos ya.

2. Vamos a usar la componente **mu-conference**, que podemos conseguir [aquí](#)⁽⁹⁾.

El servidor jabber de bulma tiene instalada desde hace tiempo una componente para realizar conferencias bajo el servicio de nombre `conference.bulmalug.net`. No se trata, en este caso, de la componente *mu-conference* sino del *conference* (algo más antiguo), pero la funcionalidad es muy parecida.

A continuación descomprimos `mu-conference-0.5.2.tar.gz` dentro de `/var/cache/apt-build/build/jabber-1.4.2a/mu-conference-0.5.2/`

3. Entramos en `mu-conference-0.5.2/` y ejecutamos `make`

4. Copiamos el fichero `mu-conference.so` (situado dentro de `src/`) en un directorio donde haya las demás componentes. Por ejemplo, en el caso de debian en `/usr/lib/jabber/mu-conference/mu-conference.so`

Ahora debemos modificar la configuración para cargar la componente de conferencias. En este caso no configuraremos un proceso externo, sino que lo incluiremos dentro del demonio principal. Por lo tanto, las modificaciones deben realizarse en el fichero `jabber.xml`.

1. Dentro de la etiqueta `<browse>` debe incluirse el siguiente código para que los clientes puedan ver la existencia de servicio de conferencias indexada en la lista de servicios ofrecidos. En este caso vamos a dar la posibilidad de que cualquier usuario pueda crear una sala de chat.

```
<conference type="public" jid="conference.cantallops.dyndns.org"
```



```

    name="Chat Cantallops">
        <ns>jabber:iq:register</ns>
        <ns>gc-1.0</ns>
        <ns>http://jabber.org/protocol/muc</ns>
</conference>

```

2. Dentro de la etiqueta <jabber> incluimos la configuración general de las salas de conferencia:

```

<service id='conference.bulmalug.net'>
    <load>
        <!-- Path al módulo de conferencias, a partir de /usr/lib/jabber/ -->
        <conference>./mu-conference/mu-conference.so</conference>
    </load>
    <conference xmlns="jabber:config:conference">
        <!-- con la etiqueta public especificamos que cualquiera
             puede crear una sala de chat-->
    </public/>
    <vCard>
        <FN>Conferencias</FN>
        <DESC>This service is for public chat.</DESC>
        <URL>http://bulma.net/</URL>
    </vCard>
    <history>20</history>
    <logdir>./log/</logdir>
        <!-- el admin puede acudir a cualquier sala,
             en caso de establecer restricciones -->
    <sadmin>
        <user>admin@bulmalug.net</user>
    </sadmin>
    <notice>
        <join> has become available</join>
        <leave> has left</leave>
        <rename> is now known as </rename>
    </notice>
</conference>
</service>

```

Por último, para crear un chat sólo hay que acceder al servicio *conference.bulmalug.net* e introducir el nombre de la sala ("room") bajo el cual se identificará la sala de conferencias. A continuación se puede invitar a otros usuarios a participar en el chat (es **necesario** invitarlos previamente si la conferencia/chat es privada y no pública).

Referencias

- Jabber.org:
 - ◆ Documentación sobre la [administración](#)⁽¹⁰⁾ de jabber.
 - ◆ [Proyectos](#)⁽¹¹⁾ en Jabberstudio. Allí podeis encontrar muchas más cosas además de las componentes de conferencia y pasarelas a varios protocolos de mensajería.
 - ◇ Componente [mu-conference](#)⁽⁹⁾.
 - ◇ Componente [msn-transport](#)⁽⁸⁾. La última versión soporta el nuevo protocolo del 15 de octubre.
 - ◇ Futura componente [msn-tng](#)⁽¹²⁾ con soporte para el protocolo de MSN a partir del 15 de octubre de 2003. Todavía está en desarrollo, pero el desarrollo del proyecto es activo y en cuanto esté listo intentaremos adaptarnos. Se supone que ha de substituir a la anterior.
- [Mensaje](#)⁽¹³⁾ sobre la configuración de msn-transport en la lista de correo de *jabbermanal*. (Se refiere a una versión antigua. No sirve para las últimas).

Agradecimientos

- A Guillermo Cantallops y a Ricardo Galli, por hacer de BOFH con el servidor. Afortunadamente para todos los usuarios, el invento funcionó a la primera ;-)
- A los desarrolladores del protocolo de jabber, a los autores de las componentes, y a toda la gente que hace posible la existencia de libertad entre las fauces de las grandes corporaciones, que cada vez quieren más, y



más, y más.

- A los que usan el servidor jabber *bulmero*.

Lista de enlaces de este artículo:

1. <http://bulma.net/body.phtml?nIdNoticia=1872&nIdPage=2>
2. <http://bulma.net/body.phtml?nIdNoticia=1872&nIdPage=3>
3. <http://bulma.net/body.phtml?nIdNoticia=1872&nIdPage=4>
4. <http://bulma.net/body.phtml?nIdNoticia=1872&nIdPage=5>
5. <http://bulma.net/body.phtml?nIdNoticia=1872&nIdPage=6>
6. <http://bulma.net/body.phtml?nIdNoticia=1872&nIdPage=7>
7. <http://jabberd.jabberstudio.org/>
8. <http://msn-transport.jabberstudio.org/>
9. <http://www.jabberstudio.org/projects/mu-conference/project/view.php>
10. <http://www.jabber.org/admin/>
11. <http://www.jabberstudio.org/>
12. <http://www.jabberstudio.org/projects/msn-tng/project/view.php>
13. <http://www.jabberstudio.org/pipermail/jabbermanual/2002-September/000006.html>

E-mail del autor: [miquel _ARROBA_ cantallops.net](mailto:miquel_ARROBA_cantallops.net)

Podrás encontrar este artículo e información adicional en: <http://bulma.net/body.phtml?nIdNoticia=1872>