

VoIP con FreeSWITCH

EN VOZ BAJA

FreeSWITCH es una potente y versátil plataforma de telefonía capaz de escalar desde un softphone hasta un PBX o incluso un switch de portadora por software. **POR MICHAEL S. COLLINS**

Los proyectos de código abierto han bajado la barrera de entrada a la telefonía hasta el punto de hacerla accesible para aficionados y pequeñas empresas. La popular herramienta de PBX Asterisk, por ejemplo, es una alternativa altamente funcional y de bajo coste para telefonía que ha demostrado ser disruptiva en el mundo de los sistemas comerciales de telefonía [1]. El que quizás sea el software más disruptivo de todos, sin embargo, aparece en forma de librería para switch por software llamada FreeSWITCH.

El sitio web de FreeSWITCH [2] asegura que puede usarse como “motor switch simple, PBX, pasarela de medios o servidor de medios para alojar aplicaciones IVR”.

Haciendo honor a esa descripción camaleónica, FreeSWITCH ha aparecido como switch de telecomunicaciones en entornos de producción como PBX en numerosos escenarios empresariales, como teléfono por software en portátiles de usuarios, e incluso como motor de redirección de voz en el proyecto OpenSim.

Además, FreeSWITCH se puede implantar como aplicación o usarlo como una librería (*libfreeswitch*) para utilizarla en software de terceros. Esta doble identidad es similar a la relación entre *cURL* (la aplicación) y *libcurl* (la librería).

FreeSWITCH se usa también en escenarios algo más exóticos, sin embargo, a aquellos que sean nuevos en materia de

aplicaciones de telefonía les conviene comenzar con la configuración predeterminada de FreeSWITCH, como PBX SOHO (es decir, para pequeña oficina o uso doméstico). Aunque en un primer momento la cantidad de archivos y valores de configuración pueda parecer desbordante, en poco tiempo se aprecian la potencia y la flexibilidad de esta versátil herramienta de telefonía.

Instalación de FreeSWITCH

En el momento de escribir estas líneas, la versión actual y estable de FreeSWITCH es la 1.0.4, que se puede descargar desde el sitio web del proyecto [3], aunque también se puede utilizar el repositorio de Subversion para instalar la última versión. A diferencia de otros muchos proyectos, la rama trunk del SVN es extremadamente estable, por lo que quizá lo más conveniente sea usar simplemente el último trunk.

En cualquier caso, los desarrolladores de FreeSWITCH han facilitado mucho la tarea de actualizar al último trunk, incluso aunque se hubiera instalado la última versión estable. Se sabe que existen binarios precompilados; pero lo más recomendable es compilar los fuentes.

Primero, descomprimos e instalamos los fuentes. Para este ejemplo los hemos descargado a `/usr/src` y desde allí los hemos instalado:

```
tar zxvf freeswitch-1.0.4.tar.gz
cd freeswitch-1.0.4
```

```
./configure
make install
```

Recomendamos obtener los archivos de sonido:

```
make cd-sounds-install
make cd-moh-install
```

FreeSWITCH incluye archivos de sonido y es capaz de manejar frecuencias de muestreo de 8, 16, 32 y 48kHz. Muy pocos, si acaso alguno, independientemente de que sean libres o privativos, son capaces de hacer el tipo de cosas que puede hacer FreeSWITCH, como manejar llamadas a diferentes frecuencias de muestreo.

La instalación llevará su tiempo, así que no pasa nada por encadenar los comandos con el operador `&&`.

Una vez finalizado el proceso, tendremos una instalación básica de FreeSWITCH. Debido a lo engorroso de tener dos archivos ejecutables principales de FreeSWITCH en el path, hemos usado enlaces simbólicos:

```
ln -s /usr/local/freeswitch/bin/
freeswitch /usr/local/bin/
freeswitch
ln -s /usr/local/freeswitch/
bin/fs_cli /usr/local/bin/
fs_cli
```

Para iniciar la aplicación se ha de introducir *freeswitch*. La utilidad *fs_cli* se emplea

para acceder al intérprete de cualquier servidor de FreeSWITCH al que se tenga acceso, incluso aunque éste se encuentre en otro servidor. Al principio veremos información de registro/depurado, pero en este punto no hay que preocuparse acerca de los mensajes de error o de advertencia que pudieran aparecer.

Nuestro Nuevo Softswitch

El primero de los comandos básicos es *version*, que muestra la versión de FreeSWITCH que se está ejecutando, incluida la revisión de SVN. El comando *status* muestra información sobre la instancia de FreeSWITCH, junto con el tiempo de ejecución total y el número de sesiones activas. Otro comando básico es *help*, aunque éste no necesita mucha explicación.

Para salir de FreeSWITCH se ha de introducir *fsctl shutdown* o el comando de atajo ... (tres puntos).

En caso de contar con una tarjeta de sonido en el sistema, se puede probar a hacer

```
load mod_portaudio
```

Después de varias líneas de depurado, se puede ver un +OK siempre que el módulo se haya cargado correctamente.

Luego, listamos los dispositivos de PortAudio disponibles en el sistema:

```
pa devlist
API CALL [pa(devlist)] output:
0;Built-in Microphone;2;0;i
1;Built-in Input;2;0;
2;Built-in Output;0;2;r,o
3;Logitech USB Headset;0;2;
4;Logitech USB Headset;1;0;
```

Pueden verse una *i* y una *o* junto a los dispositivos de entrada y salida, respectiva-

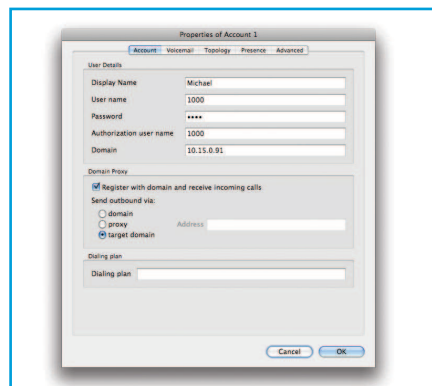


Figura 1: Pantalla de configuración de la cuenta SIP en X-Lite.

mente. Para cambiar de dispositivo de entrada podremos usar el comando *pa indev* seguido del número de dispositivo,

```
pa indev #x
```

donde *x* es el número de dispositivo. El dispositivo de salida se cambia con el comando *pa outdev*. En la configuración de ejemplo propuesta hemos usado el dispositivo 4 como dispositivo de salida y el dispositivo 5 como dispositivo de entrada.

Si los archivos de MOH (*music on hold*) están instalados, el siguiente comando provoca una llamada a la extensión MOH y envía el MOH al dispositivo PortAudio de salida:

```
pa call 9999
```

Si se oye música es porque la llamada ha tenido éxito. En caso contrario, hay que ver si los archivos MOH están correctamente instalados. Para colgar, se utiliza el comando *pa hangup*.

Podemos comprobar que tanto el dispositivo de entrada como el de salida funcionan correctamente haciendo una prueba de echo,

```
pa call 9996
```

con lo que al hablar por el dispositivo de entrada, debería oírse la voz en el de salida. Con el comando *pa hangup*, de nuevo, se desconecta.

Como puede apreciarse, es posible usar FreeSWITCH y PortAudio para crear un teléfono software simple. De este modo, cualquier codec soportado por la instalación de FreeSWITCH es válido; así, disponemos de codecs de muy alta calidad sin necesidad de instalar un cliente de teléfono por software.

Conectando el Teléfono SIP

La forma más común de usar FreeSWITCH es con un teléfono SIP. Al terminal físico se le suele llamar teléfono hardware, mientras que al teléfono software se le suele llamar *softphone*. Existen muchos tipos de teléfonos hardware disponibles en el mercado. Los softphones los hay tanto privativos como de tipo FLOSS. Un famoso softphone gratuito (pero no de código libre) es *X-Lite*, de CounterPath [4], que dispone de versiones para Linux, Mac OS X y Windows. Incluso cuando el servidor de FreeSWITCH se ejecuta bajo

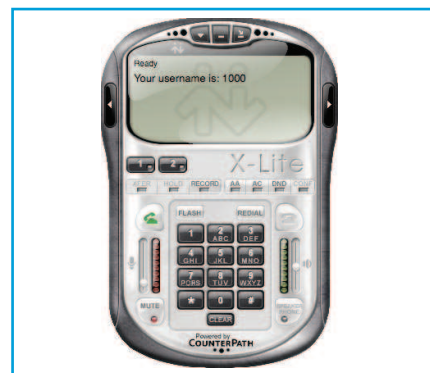


Figura 2: Cliente X-Lite registrado correctamente.

Linux, los softphones se podrán ejecutar en cualquier sistema operativo, siempre que cumplan con el estándar SIP.

Para empezar, descargamos *X-Lite* y realizamos la instalación predeterminada (no se recomienda instalar un softphone en el mismo sistema que alberga a FreeSWITCH, pero hemos hecho una excepción sólo para las pruebas).

Luego, iniciamos *X-Lite* y abrimos la configuración de la cuenta SIP pulsando sobre el botón *Show Menu* (el botón es la pequeña flecha que hay señalando hacia abajo, en la parte superior del teléfono, sobre el centro o la izquierda).

En *X-Lite* sólo hay disponible una cuenta SIP, por lo que pulsaremos sobre ella y elegiremos el botón *Properties*. Rellenaremos los campos como se muestra en la Figura 1, especificando *1234* como contraseña y *1000* como extensión.

Lo siguiente que haremos será desactivar el campo *Dialing plan*, ya que no será necesario con FreeSWITCH. Luego, pulsamos sobre *OK* y sobre *Close*, y el teléfono tratará de registrarse en el servicio. La Figura 2 muestra un registro correcto.

Para hacer una llamada de prueba marcamos *9999* y pulsamos el botón verde de enviar. En unos momentos deberíamos oír la música de espera. La Figura 3 mues-

Session Initiation Protocol

SIP es un protocolo de la capa de aplicación basada en TCP/IP usado para sesiones multimedia a través de Internet. El protocolo, basado en texto, es similar a HTTP, con peticiones y respuestas, campos de cabecera, reglas de codificación y códigos de estado. Además de para VoIP, SIP se utiliza en mensajería instantánea, juegos en línea y distribución de flujos multimedia.

tra una llamada en progreso a la extensión 9999.

Para cerrar FreeSWITCH, usamos el comando `fsctl shutdown (...)`.

Configuración

La configuración predeterminada de FreeSWITCH incluye muchas funciones. La Tabla 1 muestra servicios predeterminados de Dialplan, y la Tabla 2 algunos de los archivos y subdirectorios más importantes en la configuración de FreeSWITCH.

La instalación predeterminada crea un directorio llamado `/usr/local/freeswitch/conf`, al que nos referiremos a partir de ahora como `$CONF`. Todos los archivos de configuración son XML, que pueden incluir a su vez otros archivos XML.

La cima de este árbol XML es `$CONF/freeswitch.xml`, que incluye varios archivos XML que a su vez incluyen a otros. Este método de organización permite realizar cambios sobre la configuración de FreeSWITCH sin tener que andar trasteando un solo y enorme archivo de configuración.

FreeSWITCH como Demonio

Para iniciar FreeSWITCH como demonio usamos el argumento `-nc` (*no console*) desde la línea de comandos:

```
freeswitch -nc
```

FreeSWITCH informará de que se está ejecutando en segundo plano y proporcionará su ID de proceso (PID). Para parar FreeSWITCH, basta con ejecutar `freeswitch -stop` desde un intérprete de comandos de Linux. Por ahora dejaremos a FreeSWITCH en funcionamiento y usaremos el comando `fs_cli` para conectarnos: Nos recibirá con un breve mensaje



Figura 3: Cliente X-Lite llamando a la extensión 9999.

de bienvenida y una interfaz de línea de comandos, y puede que también con algunos mensajes de depuración en pantalla. Para deshabilitar el nivel de depurado, introducimos

```
/log info
```

Con eso, se limitan los mensajes de registro al nivel INFO, evitando montones de mensajes con información de depurado.

Todos los comandos del programa `fs_cli` comienzan con una barra, por lo que para consultar la lista de comandos disponibles se habrá de introducir `/help`. Cualquier otra cosa que escribamos en el intérprete de `fs_cli` se enviará directamente al servidor de FreeSWITCH, con una sola excepción: Los tres puntos (...) no apagarán el servidor de FreeSWITCH desde `fs_cli`; en vez de eso, cierran el programa `fs_cli`. En lugar de apagar FreeSWITCH desde `fs_cli`, hay que cerrar el programa e introducir el comando `freeswitch -stop` desde el intérprete de Linux.

Dejaremos a FreeSWITCH ejecutándose en segundo plano mientras creamos un nuevo usuario y modificamos el Dialplan.

Añadir un Usuario para SIP

FreeSWITCH viene ya con 20 usuarios predefinidos, pero si se quieren añadir más, sólo hay que llevar a cabo dos pasos para modificar la configuración:

- Añadir una entrada de usuario en `$CONF/directory/default/`.
- Modificar el Dialplan para permitir al nuevo usuario recibir llamadas.

Para añadir un nuevo usuario llamado *Catalina* con el número de extensión 1500, vamos al directorio `$CONF/directory/default` y hacemos una copia del archivo `1000.xml`:

```
cp 1000.xml 1500.xml
```

Luego, abrimos `1500.xml` en un editor, cambiamos `1000` por `1500`, y ponemos el valor de `effective_caller_id_name` a *Catalina*. El nuevo archivo debería ser similar al mostrado en el Listado 1.

Entonces abrimos `$CONF/dialplan/default.xml` y localizamos la entrada de

Tabla 1: Servicios de Dialplan

Extensión	Función
1000-1019	Extensiones predeterminadas
2000-2002	Grupos de marcado de prueba
30xx	Conferencias de banda ligera (8kHz)
31xx	Conferencias de banda media (16kHz)
32xx	Conferencias de banda ultra ancha (32kHz)
33xx	Conferencias en calidad de CD (48kHz)
4000	Recuperación de correo de voz
5000	IVR de ejemplo
5900	Parque de llamadas
5901	Recuperación del parque de llamadas
9888	Conferencia de FreeSWITCH
9992	Aplicación de Información
9996	Prueba de Echo
9999	Música de espera

Dialplan (o *extension*) llamada *Local_Extension*. En la línea

```
<condition field="destination_number"
expression="^(10[01][0-9])$">
```

aquellos que estén familiarizados con expresiones regulares reconocerán el patrón `^(10[01][0-9])$`, que significa “coincide con exactamente cuatro dígitos, de los cuales los dos primeros son 10, el siguiente es 0 ó 1, y el último es cualquiera entre 0 y 9, y guarda los cuatro dígitos en la variable \$1”. Dicho de un modo simple, coincide con cualquier número desde el 1000 hasta el 1019, ambos incluidos.

Para añadir el nuevo usuario *Catalina* a la extensión 1500 se usa el carácter de tubería, que es un OR lógico:

```
<condition field="destination_number"
expression="^(10[01][0-9]|1500)$">
```

Ahora iniciamos el programa `fs_cli` e introducimos `reloadxml` o pulsamos F6. La nueva extensión ya está lista para ser usada.

La mejor forma de probarla es configurar un softphone (o bien un teléfono hardware, en caso de disponer de uno) como 1500 con FreeSWITCH. En el cliente *X-Lite*, abrimos la configuración de las cuentas SIP y cambiamos el nombre a *Catalina*. Cambiaremos también *User* y *Authorization* a 1500 antes de guardar la configuración. El cliente debería mostrar *registering* y luego *Ready*. Desde ese momento ya se podrán hacer llamadas

Tabla 2: Archivos de Configuración Destacables

Subdirectorio/Archivo	Descripción
\$CONF/vars.xml	Define una serie de variables de sistema configurables, como las contraseñas predeterminadas y las listas de codecs preferidos.
\$CONF/dialplan/default.xml	La mayor parte del Dialplan predeterminado se configura aquí.
\$CONF/directory/default/*.xml	Aquí se definen los usuarios individuales, como por ejemplo <i>1000.xml</i>
\$CONF/sip_profiles/internal.xml	Un "perfil SIP", o UA SIP para servir un puerto de IP (normalmente el puerto 5060)
\$CONF/sip_profiles/external.xml	Otro UA SIP para manejar los registros salientes a proveedores SIP (puerto 5080)
\$CONF/autoload_configs/modules.conf.xml	Módulos cargados automáticamente al iniciar FreeSWITCH.

usando la nueva extensión. En caso de disponer de un segundo teléfono, se le puede registrar con la *1000* y usarlo para llamar a la *1500*.

Fuera del Switch Local

Una de las ventajas de VoIP es que se pueden hacer llamadas gratis, o a precios muy bajos, a través de Internet. FreeSWITCH es capaz de enrutar llamadas desde y hacia Internet, incluso desde detrás de un dispositivo de NAT. Los dispositivos de NAT pueden ocasionar todo tipo de trastornos en las llamadas SIP; aún con todo, FreeSWITCH hace un buen trabajo solucionando dichos problemas.

En muchos casos, la configuración predeterminada funcionará sin cambiar nada. Una prueba simple consistiría en llamar al servidor de conferencia pública de FreeSWITCH. La URI SIP de este servidor es *sip:888@conference.freeswitch.org*, y el Dialplan predeterminado conectará con nuestro teléfono con sólo marcar *9888*.

Si la llamada tiene éxito, se oirá una voz TTS (*text-to-speech*) anunciando el ID del llamante. En caso de ser la única persona presente en la conferencia, oiremos también una grabación informándonos sobre ello.

Para llamar a la conferencia con Port-Audio introducimos:

```
pa call >
sip:888@conference.freeswitch>
.org
```

Son varios los proveedores de servicios que ofrecen conectividad hacia y desde redes PSTNs (*public switched telephone network*) o cuentas SIP simples. FreeSWITCH hace uso del concepto de "pasarela" para establecer conexiones con estos

proveedores de servicios. Las pasarelas se usan como registros de salida a los proveedores. Quien ya disponga de un proveedor de SIP, tendrá que usar la configuración proporcionada por el mismo. La wiki de FreeSWITCH ofrece configuraciones de ejemplo para varios proveedores [5]. En este ejemplo mostraremos cómo configurar una pasarela para *iptel.org*, que ofrece cuentas SIP gratuitas.

Primero, creamos una cuenta en el sitio web *iptel.org* [6] y anotamos los datos de acceso. Luego, creamos un nuevo archivo llamado *iptel.org.xml* en *\$CONF/sip_profiles/external*, asegurándonos de poner el nombre de usuario y la contraseña en los campos correspondientes. El archivo debería tener un aspecto como el mostrado en el Listado 2.

Luego, guardamos el archivo y, desde el intérprete de FreeSWITCH, introducimos el comando:

```
sipia profile external >
restart reloadxml
```

Sofia es el nombre de la librería SIP [7] empleada por FreeSWITCH y es también el nombre del comando para la utilización de funciones relacionadas con SIP (para consultar la sintaxis adecuada, ejecutar el comando *sofia* sin argumentos). El comando anterior indica a FreeSWITCH que debe reiniciar el perfil externo después de recargar los datos XML. La opción *reloadxml* es opcional; sin embargo, si se han hecho cambios a los datos XML (creando o editando un archivo), se necesitará cargar dichos cambios al reiniciar el perfil.

Para comprobar si el sistema se ha registrado con *iptel.org*, ejecutamos

```
sofia status
```

Si el registro se llevó a cabo con éxito, debe aparecer algo como

```
iptel gateway >
sip:myusername@sip.iptel.org >
REGED
```

El estado *REGED* significa que la pasarela se ha registrado correctamente en el proveedor. Si apareciese *FAIL_WAIT*, entonces lo más probable es que haya algún problema con la configuración.

El último paso necesario para poder recibir llamadas es permitir su entrada en el Dialplan. La configuración predeterminada tiene un contexto de Dialplan "public" que trata a las llamadas entrantes desde Internet como no confiables.

Primero hay que permitir las llamadas al nuevo usuario *iptel.org* y luego hay que especificar a dónde se quieren enviar esas

Listado 1: Nuevo Usuario 1500.xml

```
01 <include>
02   <user id="1500" mailbox="1500">
03     <params>
04       <param name="password" value="${default_password}"/>
05       <param name="vm-password" value="1500"/>
06     </params>
07     <variables>
08       <variable name="toll_allow"
09         value="domestic,international,local"/>
10       <variable name="accountcode" value="1500"/>
11       <variable name="user_context" value="default"/>
12       <variable name="effective_caller_id_name" value="Catalina"/>
13       <variable name="effective_caller_id_number" value="1500"/>
14       <variable name="outbound_caller_id_name"
15         value="${outbound_caller_name}"/>
16       <variable name="outbound_caller_id_number"
17         value="${outbound_caller_id}"/>
18       <variable name="callgroup" value="techsupport"/>
19     </variables>
20   </user>
21 </include>
```

llamadas. Para ello, creamos un nuevo archivo `$CONF/dialplan/public/00_inbound_ip.tel.xml` (o hacer una copia del archivo de ejemplo llamado `00_inbound_did.xml`). Debería ser como el del Listado 3.

Hay que estar seguros de especificar el nombre de usuario `iptel.org` en la expresión regular. El nombre de usuario será el número de destino (los números de destino no están restringidos a caracteres numéricos).

La llamada se enruta a la extensión especificada en los parámetros proporcionados a la aplicación de transferencia.

En nuestro caso, la hemos dirigido hacia la extensión MOH, para simplificar las pruebas. Se puede enrutar también hacia un teléfono SIP. Si se está registrado como `1000`, entonces el destino de la transferencia es `1000 XML default`. Guardamos el archivo, nos dirigimos al intérprete de FreeSWITCH y pulsamos F6 (o introducimos `reloadxml` en el intérprete de FreeSWITCH).

Para recibir una llamada en la nueva cuenta, alguien deberá hacer una llamada SIP a la URI SIP del nuevo `iptel.org`. Introducimos, en el intérprete de FreeSWITCH:

```
pa call sip:myusername@sip.
.iptel.org
```

Si la llamada tiene éxito, sonará la extensión de destino.

Codex de Alta Calidad

Las llamadas telefónicas tradicionales tienen una calidad de sonido pésima, y el de las de telefonía móvil es aún peor. Free-

SWITCH permite llamadas de alta-calidad, calidad-de-voz, mediante el uso de codecs gratuitos y de código abierto. El codec de código abierto CELT [8] ofrece una gran calidad y unas pérdidas mínimas.

También están incluidos con FreeSWITCH los codecs Siren [9] de Polycom. Estos codecs de alta calidad son de código cerrado, pero están libres de regalías. La configuración predeterminada carga los codecs *Siren*.

Para cargar el codec CELT ejecutamos el comando `mod_celt`.

Algo que hay que tener en cuenta es que unos codecs de mayor calidad podrían poner de manifiesto las limitaciones de los terminales. La mayoría de los terminales más modernos, así como aquellos listados como *Skype certified*, tendrán un buen rendimiento con codecs de alta calidad. En nuestras pruebas hemos usado un terminal *Logitech Clear Chat Pro USB* con buenos resultados.

Para las pruebas recomendamos realizar las llamadas a algún amigo que también tenga FreeSWITCH instalado.

Para comenzar, hacemos una llamada a PSTN o a móvil, luego, colgamos y hacemos una llamada usando un codec de alta calidad. Con PortAudio es fácil probarlo. Primero, editamos `$CONF/autoload_configs/portaudio.xml`. Localizamos la línea `<param name="sample-rate" value="8000"/>`, cambiamos el valor a `48000` y guardamos.

Desde el intérprete de FreeSWITCH introducimos los siguientes comandos:

```
reloadxml
reload mod_portaudio
```

Entonces hacemos una llamada a ese amigo o llamamos a la conferencia de FreeSWITCH:

```
pa call 9888
```

La calidad de audio debería ser mucho mayor, especialmente si se dispone de una conexión a Internet rápida. Las conferencias de FreeSWITCH pueden mezclar llamadas de cualquier codec y con cualquier frecuencia de muestreo. De hecho, podemos llamar a la conferencia de FreeSWITCH a través de PSTN, marcando `+1-213-799-1400`, y comparar la diferencia que hay entre los codecs de alta calidad de VoIP y los de baja calidad del PSTN.

¿Dónde Vamos Ahora?

Con esta breve introducción a FreeSWITCH apenas arañamos la superficie de lo que este software de código abierto es capaz de hacer. Una forma de poner FreeSWITCH en práctica es la presentada por Rob Smart, de IBM, quien escribió un tutorial llamado "Home VOIP system using FreeSwitch and a Linksys 3102 voice gateway" [10], sobre cómo configurar FreeSWITCH como servidor de comunicaciones doméstico en Reino Unido.

Una comunidad vibrante y creciente ofrece soporte a los recién llegados, y también a los veteranos, siendo un buen lugar en el que continuar aprendiendo sobre las increíbles posibilidades de VoIP. ■

Listado 2: Configuración Pasarela Cuenta SIP

```
01 <include>
02 <gateway name="iptel">
03 <param name="username" value="EL_NOMBRE_DE_USUARIO_AQUÍ"/>
04 <param name="realm" value="iptel.org"/>
05 <param name="password" value="LA_CONTRASEÑA_AQUÍ"/>
06 <param name="proxy" value="sip.iptel.org"/>
07 </gateway>
08 </include>
```

Listado 3: 00_inbound_ip.tel.xml

```
01 <include>
02 <extension name="iptel-inbound">
03 <condition field="destination_number"
04 expression="^(EL_NOMBRE_DE_USUARIO_DE_IPTEL.ORG)$">
05 <action application="set" data="domain_name=${domain}"/>
06 <action application="transfer" data="9999 XML default"/>
07 </condition>
08 </extension>
09 </include>
```

RECURSOS

- [1] "How does FreeSWITCH compare to Asterisk?": <http://www.freeswitch.org/node/117>
- [2] Sitio Web de FreeSWITCH: <http://www.freeswitch.org>
- [3] Descargas de FreeSWITCH: <http://files.freeswitch.org/>
- [4] X-Lite: <http://www.counterpath.com/x-lite.html>
- [5] Wiki de FreeSWITCH: http://wiki.freeswitch.org/wiki/SIP_Provider_Examples
- [6] Sitio Web iptel.org: <http://www.iptel.org>
- [7] Librería Sofia-SIP: <http://sofia-sip.sourceforge.net>
- [8] Codec CELT: <http://www.celt-codec.org>
- [9] Codecs Siren: http://www.polycom.com/company/about_us/technology/siren_g7221/index.html
- [10] Tutorial de FreeSWITCH: http://robsmart.co.uk/2009/06/02/freeswitch_linksys3102