

# **Comunicaciones Unificadas con Elastix**

**Volumen 2**

***Edgar Landívar***

Copyright (c) 2008-2009 Edgar Landívar

Este documento está permitido de copiar, distribuir y/o modificar bajo los términos de la licencia **GNU Free Documentation License, Versión 1.3**; sin Secciones Invariantes (*Invariant Sections*), Textos de Cubierta Frontal (*Front-Cover Texts*), y sin Textos de Cubierta Posterior (*Back-Cover Texts*).

El autor puede en cualquier momento decidir actualizar la licencia de la presente obra a una versión más reciente de la **GNU Free Documentation License**.

Una copia de esta licencia se incluye en el Apéndice A: *GNU Free Documentation License*.

**Segunda Edición, versión Beta**

*A los que siempre creyeron y a otros gigantes que nos  
mostraron el camino*



# Reconocimiento

La elaboración de este volumen 2 involucró un enorme trabajo que fue facilitado gracias a la generosa ayuda de algunas personas con las cuales me encuentro agradecido. Mis más sinceros agradecimientos a:

- *A la comunidad Elastix*, por su valioso *feedback* del producto Elastix y de seguro por el que vendrá acerca de esta obra. Sin este componente este libro no podría evolucionar
- *Edwin Boza*, por su ayuda en el capítulo de “Email System”
- *Bruno Macías*, por aportar con contenido para el apartado “Extendiendo la Interfase Web de Elastix”
- *Alex Villacís*, por su aporte en el capítulo “Módulo Call Center”
- *A Rafael Bonifaz* por su ayuda en el apartado referente a VirtualBox

## Por qué comprar este libro?

Comprar una versión impresa de este libro es una manera de colaborar con el proyecto Elastix. Sin embargo, si al lector no le gustan los libros impresos pero desea colaborar con este proyecto de software libre, todavía puede realizar una donación a través del siguiente enlace:

[http://www.elastix.org/index.php?option=com\\_content&task=view&id=128&Itemid=54](http://www.elastix.org/index.php?option=com_content&task=view&id=128&Itemid=54)

## Qué significa “versión Beta”?

Significa que se trata de una edición preliminar que pudiera estar sujeta a errores menores y por lo tanto cualquier retroalimentación de los lectores es bienvenida.

## *Feedback*

Cualquier sugerencia o colaboración será bien recibida. Que el lector no dude en escribir un email a [elastixbook@palosanto.com](mailto:elastixbook@palosanto.com)



# Indice

<b>Indice .....</b>	<b>7</b>
<b>Introducción.....</b>	<b>13</b>
<b>17. Plan de marcado avanzado .....</b>	<b>15</b>
17.1 Introducción.....	15
17.2 Manipulación de variables.....	15
Eliminando dígitos al inicio y final de una variable.....	15
Concatenación de variables .....	16
Expresiones.....	16
Sintaxis condicional.....	18
17.3 Macros .....	18
Variables disponibles en la macro .....	19
Otro ejemplo de macro .....	19
17.4 AstDB .....	20
Aplicaciones para administrar AstDB .....	20
17.5 El plan de marcado de freePBX .....	21
extensions.conf .....	21
extensions_additional.conf .....	22
17.6 Algunos conocimientos útiles.....	23
Cómo verificar si una variable se encuentra vacía? .....	23
Cómo obtener la fecha actual en el plan de marcado? .....	24
Cómo proceder si quiero realizar cambios en el extensions.conf? .....	24
<b>18. Elastix Email System .....</b>	<b>25</b>
18.1 Arquitectura del sistema de email .....	25
18.2 Configurando el sistema de correo electrónico de Elastix .....	26
18.3 Configuraciones a bajo nivel .....	32
18.4 Luchando contra el SPAM .....	34
18.5 Instalando antivirus .....	35
<b>19. Mensajería Instantánea.....</b>	<b>39</b>
19.1 Protocolo XMPP (Jabber).....	39
19.2 Openfire.....	40
Características destacables .....	40
19.3 Usando Openfire en Elastix .....	40
Activación de Openfire.....	41
Interfaz Web de administración de Openfire.....	44
Integración de Openfire con Asterisk.....	46
<b>20. Fax en Elastix .....</b>	<b>51</b>

20.1 Hylafax e IAXModem .....	51
Hylafax .....	51
IAXModem.....	52
20.2 Configuración de Fax en la interfase Web de Elastix .....	53
Listado de Fax Virtual .....	53
Nuevo Fax Virtual .....	54
Fax Master .....	55
Clientes de Fax .....	56
Visor de Fax .....	56
20.3 Enviando un FAX desde una estación de trabajo .....	57
20.4 Procesamiento interno de un documento de FAX en Elastix .....	60
20.5 Protocolo T.38 .....	61
20.6 AvantFAX, un GUI alternativo .....	62
<b>21. Sintetización y reconocimiento de voz .....</b>	<b>63</b>
21.1 Sintetización de voz.....	63
Festival .....	65
Cómo interactuar con Festival desde el plan de marcado .....	67
Sintetizando voz desde la interfase Web de Elastix .....	69
21.2 Reconocimiento de voz .....	70
Lumenvox.....	70
Instalando LumenVox en Elastix .....	71
Probando LumenVox en Asterisk .....	73
<b>22. Desempeño y dimensionamiento .....</b>	<b>75</b>
22.1 Desempeño de Elastix .....	75
Factores que afectan el desempeño de Elastix.....	75
Herramientas.....	77
22.2 Dimensionamiento telefónico.....	78
Determinando el número de líneas telefónicas necesarias .....	78
Agner Krarup Erlang .....	78
La unidad Erlang .....	78
Modelos Erlang .....	79
22.3 Cuando un sólo servidor no alcanza .....	81
Distribuir el trabajo en equipos especializados .....	81
Distribuir el plan de marcado .....	82
<b>23. Seguridad.....</b>	<b>83</b>
23.1 Algo de historia .....	83
23.2 Seguridad física .....	85
Limitación del acceso al servidor .....	86
No usar el usuario root .....	86



NO dejar sesiones de consola abiertas.....	86
Bloquear el reinicio .....	87
Cambiar las claves por omisión en Elastix.....	87
23.3 Ataques y vulnerabilidades a nivel de red IP .....	88
Denegación de servicio.....	88
Ataques de inundación SIP (SIP <i>Flooding</i> ).....	88
<i>Sniffing</i> .....	89
23.4 Encriptando la VoIP .....	91
SRTP.....	91
VPNs.....	92
Encriptación IAX.....	93
23.5 Auditorías de seguridad.....	94
23.6 Algunas soluciones a los problemas de seguridad .....	97
Apagando los servicios que no vamos a usar .....	97
Firewalls .....	98
IDSs e IPSs.....	100
Actualizaciones.....	101
Monitoreo .....	101
Usar un Switch en lugar de un HUB .....	102
<b>24. Debugging Avanzado.....</b>	<b>105</b>
24.1 Introducción.....	105
Qué hacer cuando Asterisk se comporta de mal? .....	106
Errores de programación que provocan caídas, colgados o comportamientos indeseados.....	108
24.2 strace.....	109
Un ejemplo sencillo de depuración de Asterisk .....	111
24.3 GDB.....	113
Ejemplos simples de uso de GDB .....	113
Comandos básicos .....	114
24.4 Archivos core y <i>backtraces</i> .....	115
Cómo reportar un <i>backtrace</i> a Digium? .....	118
<b>25. DUNDI.....</b>	<b>119</b>
25.1 Introducción.....	119
25.2 Ejemplo de configuración.....	120
Escenario .....	120
Certificados de encriptación .....	120
dundi.conf.....	121
iax_custom.conf.....	127
extensions.conf .....	128

extensions_custom.conf.....	128
Aplicar la configuración .....	130
Conclusión.....	130
25.3 Integrando IVRs con DUNDi .....	130
<b>26. Módulo de Call Center .....</b>	<b>133</b>
26.1 Introducción.....	133
26.2 Funcionalidad básica .....	133
Modelo de funcionamiento.....	133
Funcionalidad implementada.....	134
Limitaciones conocidas .....	135
26.3 Componentes del Elastix Dialer .....	135
Marcador predictivo .....	135
Consola de agente.....	137
Administración de campañas.....	137
Reportes.....	137
26.4 Procedimientos para el uso del Elastix Dialer .....	138
Administrador - Configuración de agentes.....	138
Administrador - Creación de campañas .....	138
Administrador - Reportes .....	138
Operador - interacción con consola de agente.....	139
26.5 Resolución de problemas.....	139
Marcador no marca (ningún agente recibe llamadas):.....	139
Marcador sólo marca a un número pequeño de agentes (subutilización):.....	140
Marcador deja de generar llamadas luego de funcionar correctamente por un intervalo:.....	141
26.6 Usando la interfase Web de administración de campañas.....	141
Administración de Agentes .....	142
Administración de Formularios.....	144
Administración de campañas salientes.....	148
Administración de llamadas entrantes.....	154
Administración de <i>Breaks</i> .....	157
<b>27. Extendiendo la interfase Web de Elastix.....</b>	<b>161</b>
27.1 El <i>framework</i> Neo.....	161
Arquitectura Web basada en MVC2.....	162
Desarrollo POO .....	164
Implementación modular.....	164
27.2 Creación de un módulo de Elastix .....	165
Creando el Menú .....	165
Definición de lenguajes .....	168

Visualizando el nuevo módulo .....	169
27.3 Programación de un Módulo .....	170
27.4 Arquitectura Modular .....	171
Archivo index.php .....	171
27.5 “Elastix Developer Tools” al rescate.....	172
Instalando Elastix Developer Tools.....	172
Construyendo un módulo de Elastix con el <i>Developer</i> .....	173
Eliminación de un módulo a través del <i>Developer</i> .....	178
Administración de lenguajes desde el “Developer” .....	179
27.6 Estándar de codificación PHP en Elastix .....	179
Indentación .....	180
Estructuras de control .....	180
Llamadas a funciones .....	181
Definición de funciones.....	181
Comentarios.....	182
Código de inclusión.....	183
Comentarios de cabecera .....	183
URL de ejemplo .....	184
Convención de nombres .....	185
<b>28. Virtualización.....</b>	<b>187</b>
28.1 Opciones de virtualización .....	187
28.2 Elastix sobre VMWare .....	188
28.3 Elastix sobre Virtualbox .....	188
Virtualbox y Redes .....	189
Configuración Interfase Anfitrión .....	189
Crear la máquina virtual .....	191
Configuración Previa de la Máquina Virtual.....	192
Arrancar la Máquina Virtual por Primera Vez .....	192
Opciones Avanzadas .....	193
28.4 Elastix sobre XEN .....	194
Breve introducción a XEN .....	195
Algunos comandos XEN básicos .....	196
Instalación de Elastix virtualizado utilizando XEN .....	197
<b>29. Compilando Asterisk.....</b>	<b>201</b>
29.1 Recompilando los SRPMs.....	201
Instalar un Elastix con los requisitos para construir el RPM.....	202
Instalar el SRPM de Asterisk y modificar el SPEC de acuerdo a nuestras necesidades .....	203
Construir el RPM de Asterisk.....	203

<b>30. La Certificación Elastix .....</b>	<b>205</b>
30.1 Introducción.....	205
30.2 El Curso de Certificación Elastix .....	205
30.3 El Examen de Certificación Elastix.....	206
Tópicos y Pesos .....	206
30.4 Preguntas típicas de examen.....	207
<b>Apéndice A. GNU Free Documentation License .....</b>	<b>211</b>

# Introducción

Ya existían otros libros o manuales de Elastix cuando me vi en la misión de Escribir la presente obra. Había leído ya *Elastix Without Tears*, un gran trabajo de Ben Shariff y me parecía que ya estaban cubiertos los temas más importantes cuando nos reunimos en mi oficina con otros colaboradores a planificar lo que sería nuestro nuevo producto de entrenamiento para la certificación de Elastix.

Casi de inmediato nos dimos cuenta que pese a que las obras existentes cubrían muchos tópicos importantes no funcionarían del todo como material de estudio para un curso de certificación. La obra que necesitábamos tenía que cubrir aspectos internos del funcionamiento de Elastix y sus componentes, así como también cubrir posibles vacíos de los alumnos en tópicos diversos como Asterisk y protocolos de comunicación como SIP y RTP. No existía nada parecido en ese momento, una alternativa era usar diferentes obras para cubrir la totalidad del entrenamiento pero eso sería un poco engorroso y se vería algo desorganizado.

Salimos de la reunión con el sinsabor de no tener material de estudio.

Semanas después me di a la tarea de escribir un pénsum para nuestros cursos olvidándome un poco del tema del libro faltante. El pénsum fue del agrado de todos así que decidí bosquejar brevemente lo que debería contener cada tópico y ese fue el inicio del libro que tienen en sus manos (o en su pantalla)... Sin darme cuenta lo había comenzado a escribir. Una noche en mi casa vi que ya tenía cierta forma así que decidí comentarlo al siguiente día en la oficina y todos aprobaron la idea. Decidimos hacer dos tomos o volúmenes, el primero cubriría los tópicos requeridos para nuestro primer curso y un segundo cubriría los tópicos de nuestro segundo y final curso.

Ya iba por la mitad del primer volumen cuando surgió la pregunta acerca del licenciamiento de la obra. Elastix es software libre y yo no había pensado aún en el licenciamiento del libro. Algunos pensaban en que no debería ser libre debido a que era material de entrenamiento que de alguna manera nos otorgaba ventaja competitiva en el mercado. En este punto debo de agradecer la ayuda de algunos integrantes del equipo de PaloSanto Solutions por sus acertados argumentos en pro de FDL (*Free Documentation License*).

Finalmente la presente obra se termino con la escritura de esta introducción el 29 de Octubre de 2008. Espero que sea de la utilidad y agrado de todos vosotros.

-- Edgar Landivar  
Elastix Project Leader



# 17

## Plan de marcado avanzado

*Olvidar lo malo también es tener memoria*

-- José Hernandez en su obra “Martín Fierro”

### 17.1 Introducción

En el volumen 1 exploramos brevemente el plan de marcado (*dialplan*) de Asterisk explicando conceptos básicos como contextos, aplicaciones o variables. En este capítulo avanzaremos más y explicaremos conceptos adicionales que nos permitirán leer y escribir plan de marcado con más facilidad.

### 17.2 Manipulación de variables

#### Eliminando dígitos al inicio y final de una variable

En ocasiones necesitaremos modificar una variable como `${EXTEN}` y eliminar algunos dígitos al principio del número ya que estos muchas veces son prefijos telefónicos. Por ejemplo, si queremos marcar por un proveedor VoIP que termina llamadas exclusivamente en USA puede ser que él necesite solamente que se le envíe el número en sí, sin el prefijo internacional 001.

Este tipo de cosas se logran de manera sencilla usando cierta sintaxis que nos permite eliminar estos dígitos iniciales.

La sintaxis usada para eliminar dígitos al inicio y/o al final de una variable es **`${EXTEN:x:y}`**

Donde “x” representa el número de dígitos que se eliminarán al principio de la variable y “y” representa el número de dígitos que se extraerán de dicha variable.

Por ejemplo, supongamos que originalmente la variable `${EXTEN}` contiene el número telefónico 123456789. Por lo tanto la expresión `${EXTEN:2:4}` nos devolverá el número 3456. Es decir que se eliminaron 2 (x) dígitos al principio y el número devuelto debe contener 4 (y) dígitos.

Si se omite “y” en la expresión (lo cual es el caso más común) entonces se asume que se devolverán todos los restantes dígitos contenidos en la variable. En el ejemplo del número telefónico anterior con la expresión `${EXTEN:2}` obtendríamos 3456789.

## Concatenación de variables

Para concatenar dos o mas variables simplemente debemos escribirlas juntas. Por ejemplo, para concatenar las variables VAR1, VAR2 y VAR3 podríamos hacer algo como lo siguiente.

```
exten => s,n,SetVar(VAR_CONCAT=${VAR1}${VAR2}${VAR3})
```

## Expresiones

Una expresión es una combinación de variables y operadores que producen un resultado. Un uso común de una expresión es el de realizar operaciones matemáticas como por ejemplo sumar dos valores. Una expresión tiene la siguiente sintaxis.

`${expresión}`

Por ejemplo, para sumar dos variables llamadas VAR1 Y VAR2 podemos utilizar la expresión `${ ${VAR1} + ${VAR2} }`



## Operadores

Los operadores los hay de diferentes tipos como *booleanos*, matemáticos, para comparar y para manipular expresiones regulares.

Operador	Descripción
<b>Booleanos</b>	
<code>expr1   expr2</code>	Operador OR. Retorna <code>expr1</code> en caso de que <code>expr1</code> contenga una cadena no-vacía o diferente de cero. Caso contrario retorna <code>expr2</code> .
<code>expr1 &amp; expr2</code>	Operador AND. Retorna <code>expr1</code> siempre y cuando tanto <code>expr1</code> como <code>expr2</code> contengan cadenas no-vacías o diferentes de cero. Caso contrario retorna cero.
<code>!expr</code>	Operador de complemento.
<b>Para comparar</b>	
<code>expr1 = expr2</code>	Devuelve 1 si la comparación es cierta y 0 si es falsa
<code>expr1 != expr2</code>	Devuelve 1 si la comparación es cierta y 0 si es falsa
<code>expr1 &lt; expr2</code>	Devuelve 1 si la comparación es cierta y 0 si es falsa
<code>expr1 &gt; expr2</code>	Devuelve 1 si la comparación es cierta y 0 si es falsa
<code>expr1 &lt;= expr2</code>	Devuelve 1 si la comparación es cierta y 0 si es falsa
<code>expr1 &gt;= expr2</code>	Devuelve 1 si la comparación es cierta y 0 si es falsa
<b>Matemáticos</b>	
<code>expr1 + expr2</code>	Suma dos valores
<code>expr1 - expr2</code>	Resta dos valores
<code>- expr</code>	Niega un valor
<code>expr1 * expr2</code>	Multiplica dos valores
<code>expr1 / expr2</code>	Divide un valor para otro
<code>expr1 % expr2</code>	Devuelve el módulo de un valor para otro
<b>Para manipular expresiones regulares</b>	
<code>expr1 : regexp</code>	Compara <code>expr1</code> contra una expresión regular contenida en <code>regexp</code> . La comparación asume un carácter de inicio de cadena <code>^</code> implícito al inicio de <code>regexp</code> .

<code>expr1 =~ expr2</code>	Similar al operador anterior excepto que no se asume un carácter de inicio de cadena implícito.
-----------------------------	---

## Sintaxis condicional

Cuando se necesita evaluar una expresión y tomar una decisión como resultado de aquello podemos usar lo que se llama sintaxis condicional y que mostraremos a continuación.

```
expresión?destino1:destino2
```

Es decir que si la expresión es verdadera se retorna destino1 y si es falsa se retorna destino2.

Se podría decir que los caracteres “?” y “:” son operadores pero es mejor estudiarlos por separado.

Esta sintaxis es la usada por aplicaciones como GotoIf.

## 17.3 Macros

Las macros no son otra cosa que aplicaciones definidas por el usuario. Una de las grandes ventajas de esto es que nos permiten reutilizar código de plan de marcado.

Para definir una macro simplemente escribimos el código de plan de marcado como si fuera un contexto cualquiera con el cuidado de que el nombre de este contexto debe empezar con la palabra “macro” seguida de un guión y del nombre de nuestra macro. Por ejemplo:

```
[macro-nohacenada]
exten => s,1,NoOp("No se hace nada")
```

Para invocar la macro lo hacemos a través de una aplicación especial llamada también Macro, la cual tiene la siguiente sintaxis.

```
Macro(nombremacro,arg1,arg2...)
```

Como vemos anterior a la aplicación Macro debemos pasarle el nombre de la macro y los argumentos que toma la macro en caso de que hayan. Hay que tener cuidado de que el nombre de la macro constituye todo lo que hemos escrito después del prefijo “macro-” cuando la definimos.

Si queremos que la macro manipule ciertos argumentos podemos acceder a ellos automáticamente a través de las variables `${ARG1}`, `${ARG2}`, `${ARG3}`, etc

Por ejemplo, para invocar la macro que creamos antes podemos proceder de la siguiente manera.

```
exten => s,n,Macro(nohacenada)
```

Veamos a continuación la definición de una macro muy usada en el plan de marcado propuesto por freePBX y que sirve para colgar una llamada.

```
; What to do on hangup.  
[macro-hangupcall]  
exten => s,1,ResetCDR(w)  
exten => s,n,NoCDR()
```

## Variables disponibles en la macro

Aparte de los argumentos del tipo `${ARGn}` existen otras variables disponibles en el contexto de una macro. Estas variables son las siguientes.

Nombre variable	Descripción
<code>\${MACRO_CONTEXT}</code>	El contexto de la extensión desde donde se llamó a la macro
<code>\${MACRO_EXTEN}</code>	La extensión desde donde se llamó a la macro
<code>\${MACRO_PRIORITY}</code>	La prioridad en la extensión donde se llamó a la macro

## Otro ejemplo de macro

Un ejemplo de macro muy didáctico y sencillo, aunque probablemente poco útil en la vida práctica, es el de un sumador que nos permita reproducir en audio la suma de dos números. A continuación la macro.

```
[macro-suma]
exten => s,1,SetVar(TOTAL=[ ${ARG1} + ${ARG2} ])
exten => s,n,SayNumber(${TOTAL})
```

## 17.4 AstDB

Asterisk cuenta con una pequeña base de datos a la que se puede acceder desde el plan de marcado a través de algunas aplicaciones destinadas a este fin. Esta pequeña base de datos está basada en el motor Berkeley DB versión 1 que básicamente es una implementación muy sencilla de base de datos con reducida funcionalidad.

AstDB reside en `/var/lib/asterisk/astdb` y se usa para almacenar información de cualquier tipo a la cual se quiera tener acceso desde el dialplan.

En esencia AstDB permite agrupar información en “familias” que no son otra cosa sino contenedores que nos servirán para categorizar la información. En cada familia tendremos pares conformados por dos elementos llamados clave y valor. Donde cada clave es un nombre único dentro de cada familia. Dos familias importantes que son mantenidas por freePBX son AMPUSER y DEVICE.

En el plan de marcado provisto por Elastix se usa AstDB debido a que freePBX también la usa así que es buena idea revisar algunas aplicaciones que nos servirán para manipular esta base de datos.

### Aplicaciones para administrar AstDB

A continuación algunas aplicaciones que nos servirán para administrar AstDB desde el plan de marcado.

Aplicación	Descripción
DBdel	Elimina una clave de la base de datos
DBdeltree	Elimina una familia de la base de datos
DBget	Obtiene un valor de la base de datos. Para esto debemos proporcionarle la clave correspondiente.
DBput	Almacena un valor en la base de datos

## 17.5 El plan de marcado de freePBX

El plan de marcado escrito por freePBX reside principalmente en los archivos `extensions.conf` y `extensions_additional.conf`. El primero contiene información fija que no cambia cuando se hacen cambios desde la interfase Web de freePBX y el segundo contiene información variable que sí cambia cuando se hacen cambios desde la interfase Web.

### **extensions.conf**

El archivo `extensions.conf` es donde Asterisk buscará primero el plan de marcado. Este archivo a su vez incluye el `extensions_additional.conf`.

Podríamos decir a nivel general que este archivo se encuentra dividido en dos tipos de contextos: macros y puntos de acceso.

#### Macros de apoyo

El primer tipo constituye un conjunto de macros que sirven de apoyo en la codificación del plan de marcado y que permiten la reutilización de código y organización del mismo. Algunas de las macros más importantes que podemos encontrar aquí son:

Macro	Descripción
[macro-dial]	Se usa para timbrar a una o más extensiones. Toma como argumentos las variables <code>\$DIAL_TIMER</code> , <code>\$DIAL_OPTIONS</code> , <code>\$EXT1</code> , <code>\$EXT2</code> , <code>\$EXT3</code> , etc
[macro-exten-vm]	Timbra a una extensión y si dicha extensión está ocupada o no contesta envía la llamada al casillero de voz ( <i>voicemail</i> )
[macro-vm]	Llama sistema de casillero de voz y permite acceder a un IVR personal
[macro-rg-group]	Timbra grupos de teléfonos ( <i>ring groups</i> )
[macro-outisbusy]	Informa al cliente que los canales de salida están ocupados
[macro-hangupcall]	Cuelga una llamada. Básicamente <i>resetea</i> los CDRs antes de colgar.
[macro-dialout]	Permite marcar hacia afuera y elimina el prefijo.
[macro-record-enable]	Revisa si una llamada tiene habilitada la grabación de llamadas y procede a la grabación de ser el caso

## Puntos de acceso al *dialplan*

El segundo tipo de contextos está constituido por lo que podríamos llamar puntos de acceso al plan de marcado o contextos “from”. Es decir los lugares por donde la comunicación se inicia, ya que una llamada podría originarse en diferentes lugares, como extensiones internas, troncales SIP, dispositivos Zaptel, etc.

Contexto “from”	Descripción
[from-pstn]	Por aquí ingresan las llamadas desde la PSTN o también desde una troncal VoIP. Este contexto es equivalente a <code>from-trunk</code> ; de hecho este último es un alias de <code>from-pstn</code> .
[from-sip-external]	Responde a usuarios SIP externos un mensaje de congestión y luego cuelga
[from-internal]	Por aquí ingresan las llamadas realizadas desde extensiones internas
[from-zaptel]	Para llamadas que ingresan desde dispositivos zaptel

## **extensions\_additional.conf**

Prácticamente cada entidad o característica configurable desde la interfase Web de freePBX está especificada aquí. Por ejemplo, en este archivo se definen los servicios, las salas de conferencia, las extensiones mismas, los IVRs, los accesos DISA, las rutas salientes, las rutas entrantes, las colas, los ring-groups, etc.

En conclusión se podría decir que este archivo contiene los parámetros configurables por el usuario. Es el archivo donde freePBX hace su mayor trabajo.

Trataremos de dar una idea al lector de qué significan los diferentes contextos que aquí se pueden encontrar. En ocasiones especificaremos sólo el prefijo con el que empieza el nombre del contexto para agrupar mejor.

Contexto	Descripción
globals	En el contexto <code>globals</code> (recordemos que se trata de un contexto de nombre reservado) se enlistan algunas variables que define internamente freePBX para su correcto funcionamiento.
app-*	Este tipo de contextos corresponde a un servicio o aplicación como por ejemplo el servicio de dictado, el servicio de directorio telefónico, el servicio de acceso al <i>voicemail</i> , entre otros. Se encuentra en este archivo ya

	que el usuario en cualquier momento puede cambiar el número de acceso al servicio desde el menú “Feature Codes” de la interfase Web del freePBX
<code>ivr-*</code>	Contiene la definición de un IVR. Al prefijo “ivr-“ se le agrega un número que identifica internamente al IVR
<code>macro-*</code>	Obviamente las macros de apoyo necesarias para simplificar el código. Una macro muy importante es la macro <code>dialout-trunk</code> que es donde se decide por cuál ruta saldrá la llamada.
<code>ext-*</code>	<p>En estos contextos encontraremos cualquier cosa que pueda ser considerada como un número de extensión. Tanto extensiones que correspondan a teléfonos o <i>endpoints</i>, como extensiones que podríamos denominar virtuales ya que no corresponden a teléfonos directamente relacionados con un usuario; como es el caso de cuartos de conferencia, colas, grupos de <i>paging</i>, entre otros, que son accesibles como si fueran números de extensión.</p> <p>Un contexto de especial interés es <code>ext-local</code> que es donde se encuentran las extensiones internas o locales.</p>
<code>from-internal-additional</code>	Este contexto es referenciado desde el contexto <code>from-internal</code> que es un punto de acceso para extensiones locales definido en el archivo <code>extensions.conf</code> . Es realmente quien realiza la mayor parte del trabajo al incluir a vez otros contextos donde podrían ser ubicadas las extensiones destino.
<code>from-trunk-*</code>	Aquí se definen las rutas entrantes.
<code>outrt-*</code>	Aquí se definen las rutas salientes
<code>outbound-allroutes</code>	Aquí se enlistan todos los contextos que corresponden a rutas salientes. Es decir los que empiezan con “outrt-”

## 17.6 Algunos conocimientos útiles

A continuación algunos ejemplos de preguntas comunes o trucos que siempre es interesante conocer.

### Cómo verificar si una variable se encuentra vacía?

No existe en el plan de marcado una función como `isNull`, `isEmpty` o cosa por el estilo que nos permita verificar si una variable se encuentra vacía, pero podemos valernos de la aplicación `LEN` para este fin y comparar su salida contra cero. Veamos un ejemplo de esto.

```
exten => s,1,GotoIf(${LEN(${VARIABLE_TEST})} > 0)?destino)
```

## Cómo obtener la fecha actual en el plan de marcado?

Para obtener la fecha actual podemos hacer uso de la variable `${EPOCH}` que contiene un *timestamp* del instante actual. Para traducir este *timestamp* a una fecha más humanamente legible podemos utilizar la función `STRFTIME`. Veamos un ejemplo de esto tomado del archivo `extensions.conf`

```
[macro-record-enable]
exten => s,1,GotoIf(${LEN(${BLINDTRANSFER})} > 0)?2:4)
exten => s,2,ResetCDR(w)
exten => s,3,StopMonitor()
exten => s,4,AGI(recordingcheck,${STRFTIME(${EPOCH},,%Y%m%d-%H%M%S)},${UNIQUEID})
exten => s,5,Noop(No recording needed)
exten => s,999,MixMonitor(${CALLFILENAME}.wav)
```

## Cómo proceder si quiero realizar cambios en el `extensions.conf`?

No es recomendado cambiar el archivo `extensions.conf` debido a que freePBX podría sobrescribirlo durante una operación de actualización. Qué hacer entonces?

Existe un archivo llamado `extensions_override_freepbx.conf` que puede ser utilizado para este fin. Si un nombre de contexto dentro de este archivo coincide con uno definido en el `extensions.conf` entonces dicho contexto será sobrescrito por el contenido en `extensions_override_freepbx.conf`.

Por lo tanto si se quiere modificar un determinado contexto del archivo `extensions.conf` lo recomendable es escribir nuestra nueva versión de dicho contexto en el `extensions_override_freepbx.conf`.



# 18

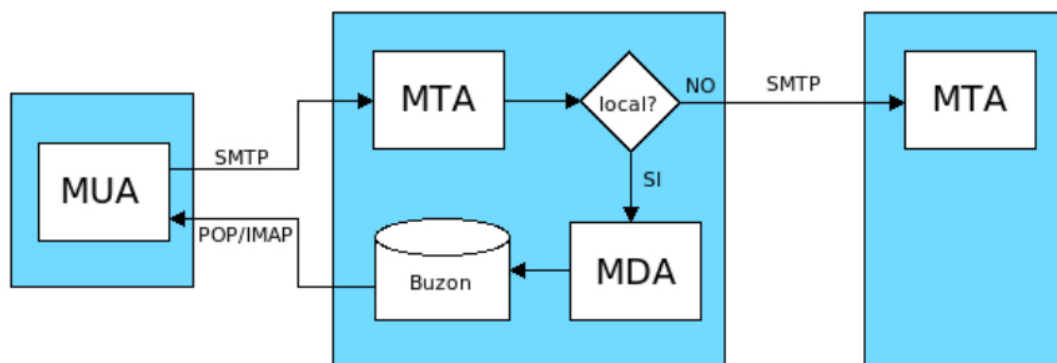
## Elastix Email System

*Todo derecho que no lleve consigo un deber, no merece que se luche por defenderlo*

-- Mahatma Gandhi

### 18.1 Arquitectura del sistema de email

Un sistema de correo electrónico está conformado por varios componentes. Los principales son el MUA (Mail User Agent), el MTA (Mail Transfer Agent) y el MDA (Mail Delivery Agent)



Componentes de un sistema de correo electrónico

El MUA es el programa con el que se genera el correo electrónico y generalmente reside en el computador del cliente, por ejemplo el Microsoft Outlook, el Mozilla Thunderbird, entre otros. También existen situaciones en las que el MUA reside en el mismo servidor, como es el caso de los servicios de Webmail u otros programas que generan correos electrónicos como los sistemas de Voicemail y de Fax.

El MTA es quién, mediante protocolo SMTP, recibe el mensaje de parte del MUA y decide a quién entregarlo. Si el correo está destinado a otro dominio, el MTA tratará de contactarse con el servidor de correo electrónico de dicho dominio y le transferirá el mensaje mediante el mismo protocolo SMTP. Si el correo está destinado para un dominio local, el MTA transferirá internamente el mensaje al MDA quien a su vez será el encargado de colocarlo en el buzón respectivo.

Los usuarios pueden acceder a sus buzones de correo utilizando el MUA, pero mediante los protocolos POP y/o IMAP, los cuales deben estar activos en el servidor de correos.

Elastix utiliza como MTA al Postfix y como MDA al Cyrus IMAP, quien también provee los servicios de POP e IMAP. Así mismo, con el Elastix viene por defecto una interfaz de webmail llamada RoundCube Mail, la misma que puede ser usada como MUA.

## **18.2 Configurando el sistema de correo electrónico de Elastix**

Elastix facilita y simplifica las tareas de configuración de un servidor de correos. Utilizando la interfase Web administrativa se puede tener el servicio de correo electrónico completamente funcional en unos cuantos minutos siguiendo el procedimiento detallado a continuación.

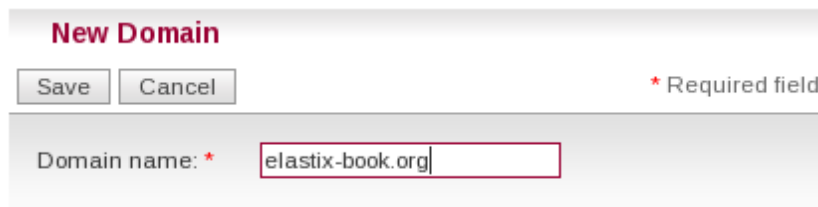
Primero se debe elegir el menú “Email” y luego la opción “Domains”, a continuación aparecerá un listado de los dominios locales configurados. En Elastix el servicio de correo electrónico es multi-dominio, por lo que se lo puede utilizar para recibir mensajes para varios dominios diferentes. Inicialmente la lista estará en blanco.



Elastix is licensed under [GPL](#) by [PaloSanto Solutions](#). 2006 - 2008.

### Listado de dominios

Para crear un nuevo dominio se utiliza el botón “Create Domain”, con lo cual aparece una nueva pantalla en la que se debe ingresar el nombre del nuevo dominio.



### Creando un nuevo dominio

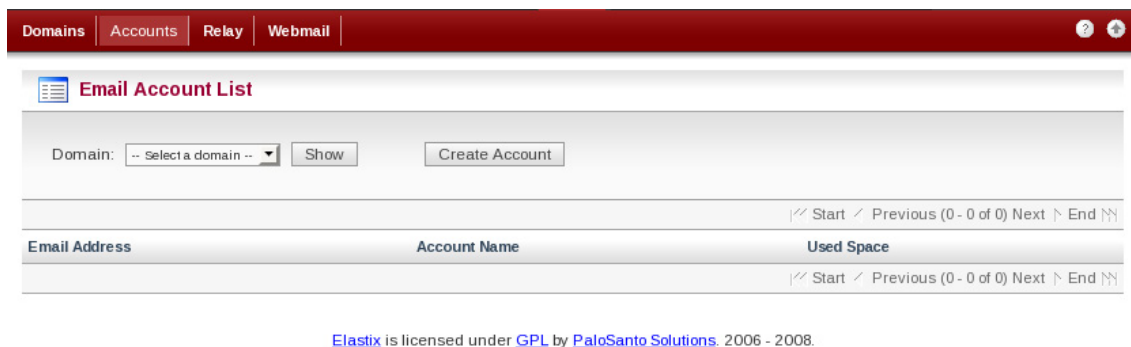
Luego se guarda la acción utilizando el botón “Save” y se retornará al listado de dominios donde aparecerá el que se acaba de crear.



Domain	Number of Accounts
<a href="#">elastix-book.org</a>	0

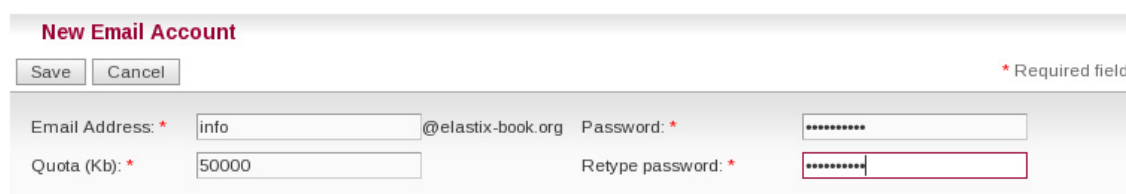
El listado de dominios muestra ahora el dominio creado

El siguiente paso es crear las cuentas de correo electrónico. Para ello, en el mismo menú “Email” se debe elegir la opción “Accounts”. En esta sección aparecerá una lista desplegable donde se debe seleccionar el dominio con cuyas cuentas se trabajará y se debe presionar el botón “Show” para mostrar las cuentas existentes en dicho dominio.



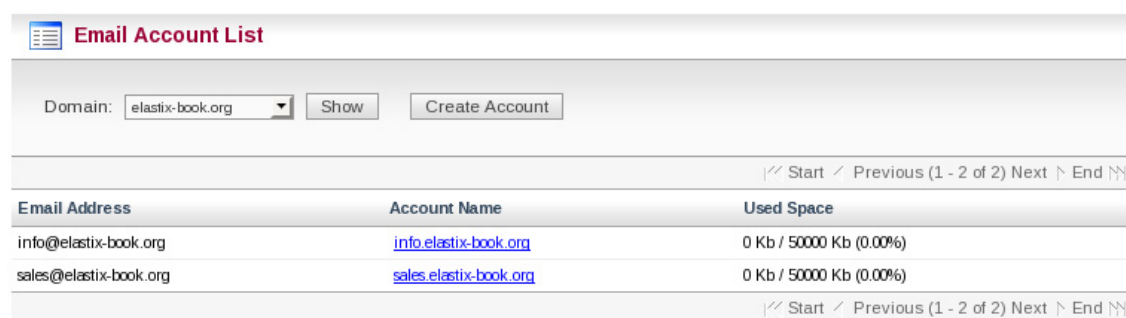
Listado de cuentas de email. Aún no se selecciona un dominio de la lista.

Para crear una cuenta nueva, se debe seleccionar el dominio en la lista desplegable y luego utilizar el botón “Create Account”. Aparecerá una pantalla donde se deben ingresar los datos básicos de la cuenta de correo.



Creando una nueva cuenta de correo electrónico

Una vez llenados todos los campos se procede a guardar la cuenta con el botón “Save” e inmediatamente se retornará al listado de las cuentas creadas para el dominio seleccionado anteriormente. Se debe repetir este procedimiento para crear todas las cuentas de correo necesarias.

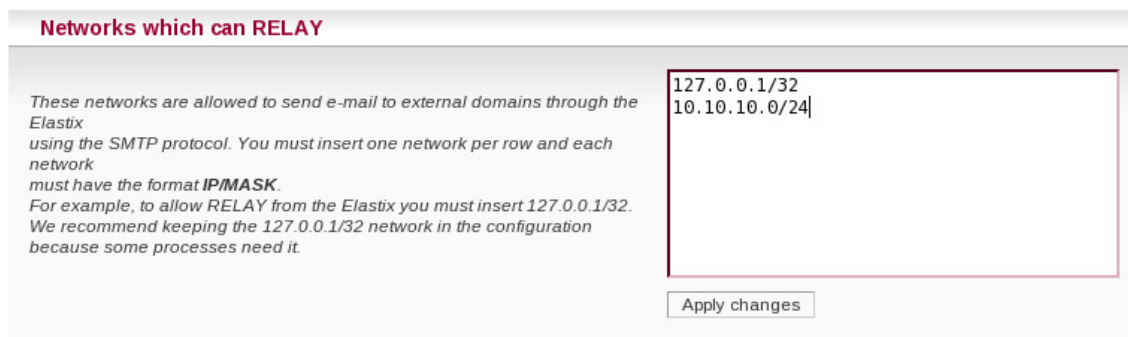


Cuentas de email para el dominio creado

Es importante resaltar el verdadero nombre de la cuenta creada. En el ejemplo anterior se creó la dirección de correo “info@elastix-book.org”, sin embargo, como se puede ver en el listado de las cuentas, el nombre real es “info.elastix-book.org”. Éste es el nombre de usuario o *login* que deberá ser usado para acceder al buzón de la cuenta.

Este esquema se ha utilizado para poder implementar la característica de multi-dominio, puesto que de esta manera se permite que en el mismo servidor existan direcciones como “info@elastix-book.org” e “info@otro-dominio.com”.

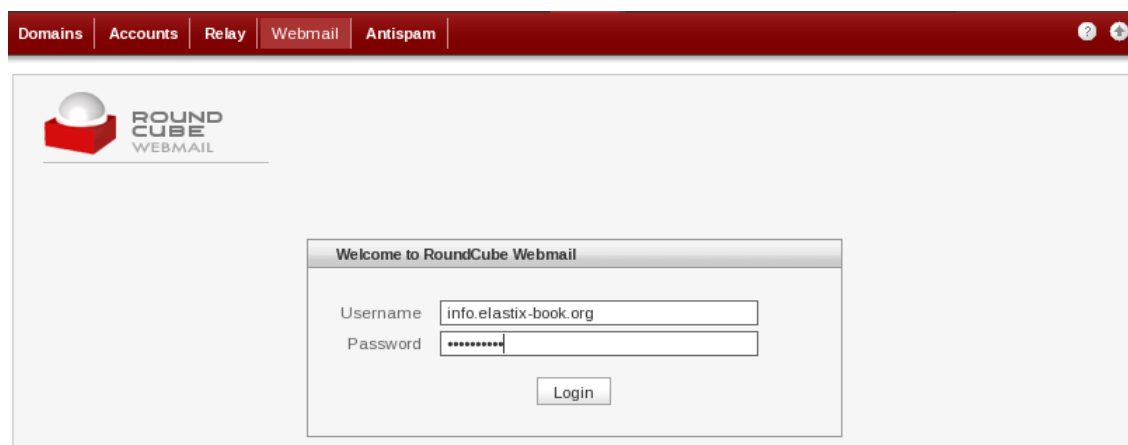
Si alguna de las cuentas de correo que han sido creadas será utilizada en un cliente de correo como Microsoft Outlook, Mozilla Thunderbird u otro similar, se le deberá indicar al servidor cuales son las direcciones de red a las que se les permitirá enviar correos hacia dominios que no han sido configurados como locales. Este proceso de envío de correos a dominios externos se conoce como RELAY y para permitir nuevas direcciones de red se debe elegir la opción “Relay” en el menú “Email”, aparecerá un listado de las direcciones que actualmente tienen el permiso mencionado.



Configuración de las redes que pueden hacer RELAY

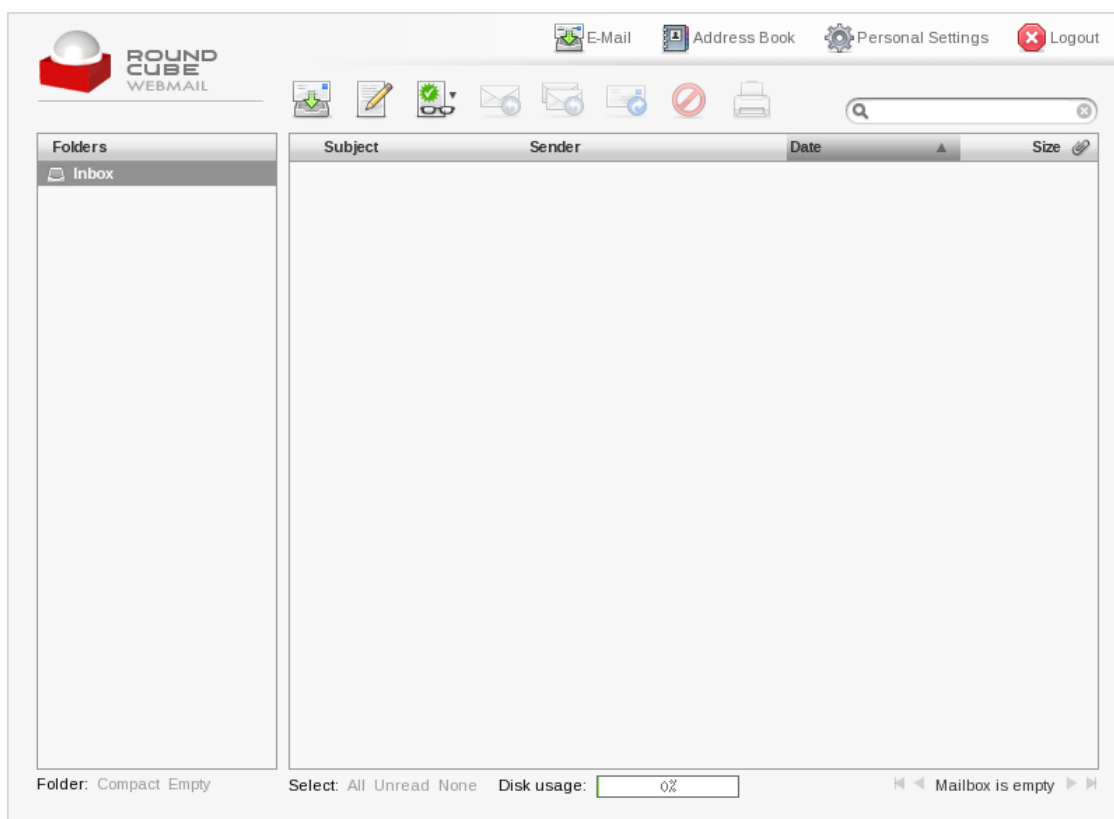
En esta lista se puede agregar las nuevas direcciones de red, una por línea, asegurándose de que se especifique la respectiva máscara de red. También es importante recordar que la dirección de localhost (127.0.0.1/32) deberá estar siempre listada, de lo contrario se pueden presentar problemas con ciertas funcionalidades de Elastix como *voicemail*, *fax-to-email*, entre otras.

El siguiente paso es comprobar el funcionamiento del servicio, para ello se puede utilizar una de las cuentas creadas y configurarla en un cliente de correo electrónico, sin embargo la manera más sencilla es utilizar la interface de webmail que viene incluida con el Elastix llamada RoundCube WebMail. Para acceder a este servicio debemos elegir la opción “Webmail” del menú “Email” e ingresar los datos de acceso de una de las cuentas creadas anteriormente.



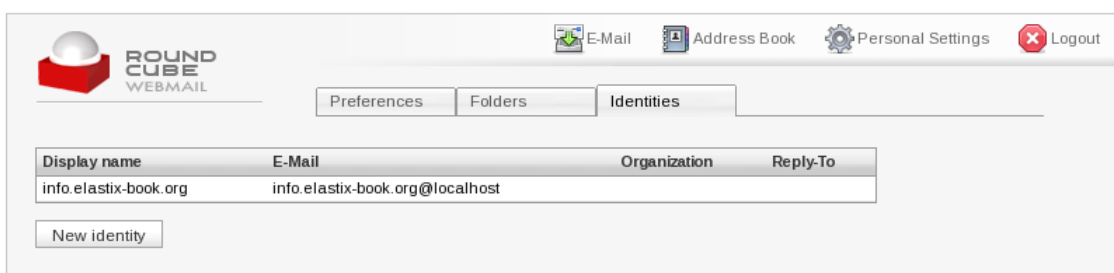
Ingreso al Webmail provisto a través de RoundCube

Se debe tener cuidado respecto con el nombre de la cuenta y la respectiva clave. A continuación se debe usar el botón “Login”, con lo cual se accederá al buzón de entrada de dicha cuenta.



Pantalla principal del Webmail

Por defecto, el RoundCube configura de manera incorrecta la identidad de la cuenta y debe ser corregida antes de enviar mensajes, para esto se debe usar la opción “Personal Settings” del menú que está situado en la parte superior derecha de la pantalla y luego seleccionar la pestaña “Identities”, con lo cual aparecerá lo siguiente:



Pantalla de configuración de Identidades

Se puede observar que la dirección está en un formato incorrecto que incluye la palabra “@localhost” lo cual debe ser cambiado. Al seleccionar la identidad actual automáticamente aparecerá un formulario de edición en el que podremos cambiar los valores y escribir adecuadamente la dirección de correo electrónico.

ROUND CUBE WEBMAIL

E-Mail Address Book Personal Settings Logout

Preferences Folders **Identities**

Display name	E-Mail	Organization	Reply-To
info.elastix-book.org	info.elastix-book.org@localhost		

New identity

Edit item

Display name: info@elastix-book.org

E-Mail: info@elastix-book.org

Organization:

Reply-To:

Bcc:

Signature:

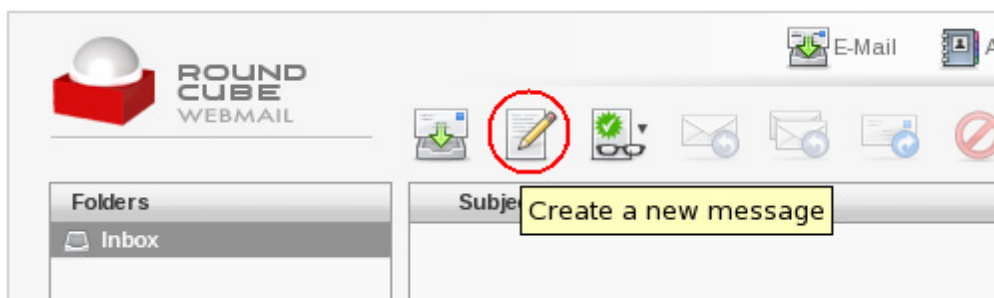
HTML signature: ☐

Set default: ☒

Delete Save

Configuración de parámetros

Se finaliza el procedimiento presionando el botón “Save”. Luego de esto se puede usar la opción “E-Mail” del menú superior para regresar al buzón de entrada de la cuenta. Para redactar un mensaje nuevo, se debe utilizar el ícono de una hoja de papel con un lápiz amarillo.



Para redactar un nuevo mensaje

El formulario par escribir un mensaje es bastante sencillo e intuitivo. Luego de redactar el correo, se lo envía con el botón “*Send the message now*” que está en la parte inferior.

Escribiendo un nuevo email

Si se ha enviado el mensaje a otra cuenta del mismo dominio, se puede cerrar la sesión actual e ingresar con la otra cuenta. Los mensajes nuevos en la bandeja de entrada lucen así:

Subject	Sender	Date	Size
Email 1	info@elastix-book.org	Today 17:38	701 B

Nuevo mensaje en la bandeja de entrada

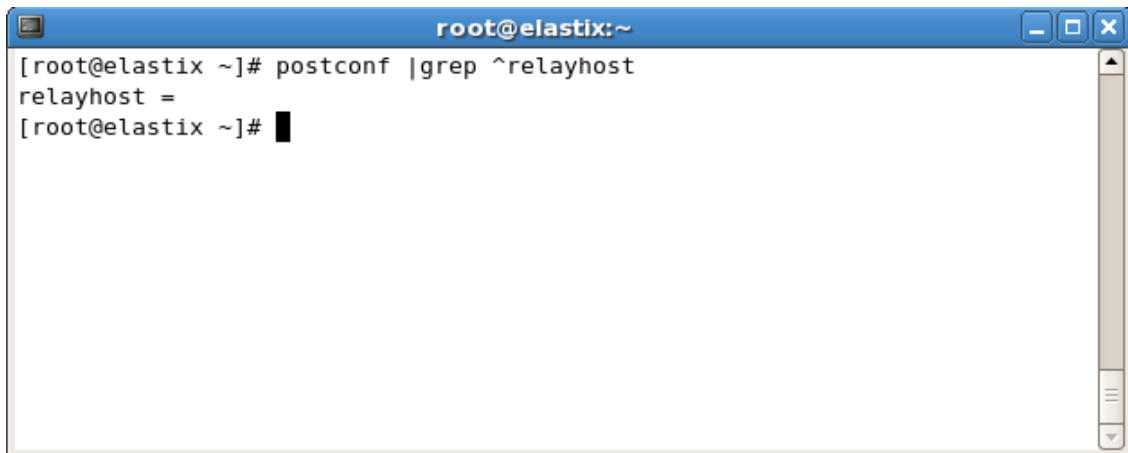
## 18.3 Configuraciones a bajo nivel

Como se mencionó anteriormente, Elastix usa como MTA a Postfix que es un sistema de correo electrónico sumamente potente, seguro y por sobre todo flexible para permitir innumerables configuraciones avanzadas. Se puede obtener más información acerca de Postfix en el sitio Web oficial [www.postfix.com](http://www.postfix.com), así mismo es común utilizar un buscador de Internet para encontrar documentación respecto a configuraciones específicas.



Una necesidad recurrente en sistemas de correo electrónico es la configuración de un Smart Relay, que consiste en indicarle al MTA (Postfix) que todos los mensajes que no están destinados a dominios locales se los envíe a un servidor específico que se encargará a su vez de transferirlo a los servidores de destino. Esto es de gran utilidad cuando el ISP (Proveedor de Servicios de Internet) ha restringido las conexiones SMTP salientes y solo permite envíos a través de su propio servidor SMTP, algunas veces llamado SMTP Gateway. En Postfix esta opción se llama “relayhost”.

Postfix posee un archivo de configuración principal llamado “/etc/postfix/main.cf”, en el cual se especifican las diferentes opciones del mismo. Este archivo no muestra todas las opciones, sino solamente las más usuales y aquellas que han sido modificadas de su valor por defecto. Para ver todas las opciones de configuración actuales de Postfix se debe utilizar el comando `postconf`, sin embargo, la salida es muy larga puesto que tiene cerca de 525 opciones, por lo que se recomienda utilizarlo en conjunto con el comando `grep`. Por ejemplo, para el caso del Smart Relay, se quiere verificar cual es el valor actual de la opción “relayhost”:

A terminal window titled 'root@elastix:~' showing the command '[root@elastix ~]# postconf | grep ^relayhost' and its output 'relayhost ='. The prompt '[root@elastix ~]#' is shown again on the next line.

```
root@elastix:~  
[root@elastix ~]# postconf | grep ^relayhost  
relayhost =  
[root@elastix ~]#
```

Ejecución del comando `postconf`

Para cambiar una opción del Postfix se debe editar el archivo “/etc/postfix/main.cf” y buscar la opción requerida, en caso de que no exista se la puede agregar. Se recomienda que cuando se agrega una línea se lo haga en la parte final del archivo pero antes de los comentarios que dicen:

```
#####  
#Ingresado por yb-webadmin
```

En el caso del ejemplo del Smart Relay, se configurará la opción “relayhost” para que tenga el valor “smtp-gw.myisp.net”, para ello se agregará la línea:

```
relayhost = [smtp-gw.myisp.net]
```

```
root@elastix:~  
# sample_directory: The location of the Postfix sample configuration files.  
# This parameter is obsolete as of Postfix 2.1.  
#  
sample_directory = /usr/share/doc/postfix-2.2.2/samples  
  
# readme_directory: The location of the Postfix README files.  
#  
readme_directory = /usr/share/doc/postfix-2.2.2/README_FILES  
  
virtual_alias_maps = hash:/etc/postfix/virtual  
  
relayhost = [smtp-gw.myisp.net]  
  
#####  
#Ingresado por yb-webadmin  
mydomain = example.com  
myhostname = yb-server.example.com  
  
-- INSERT --
```

Archivo /etc/postfix/main.cf

Luego de editado el archivo “/etc/postfix/main.cf” se debe aplicar los cambios utilizando uno de los siguientes comandos:

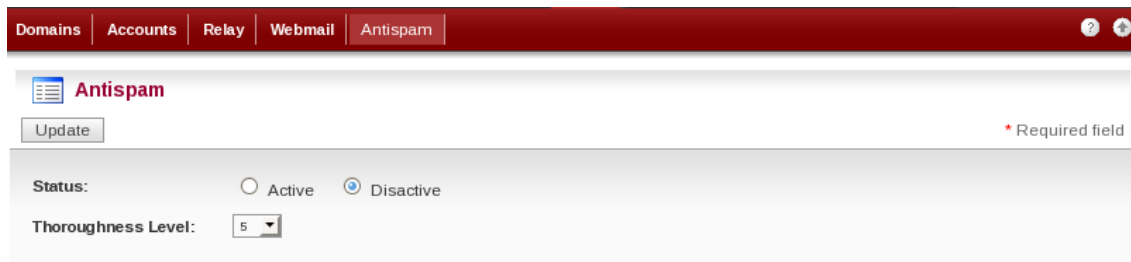
```
# service postfix reload  
# service postfix restart
```

## 18.4 Luchando contra el SPAM

Elastix también provee un filtro contra SPAM que, cuando está activado, verifica todos los mensajes que ingresan al servidor de correos y les asigna un puntaje dependiendo de ciertas características. Si el mensaje obtiene un puntaje superior a lo que se conoce como “Nivel de Rigurosidad”, entonces el mensaje es marcado, es decir que se agregan cabeceras indicando que es SPAM y también se le agrega en el asunto la palabra “[SPAM]”

Es muy fácil configurar los clientes de correo para aplicar filtros y reglas que redireccionen los SPAM a otras carpetas o incluso los eliminen.

Para activar esta característica en el Elastix se debe elegir la opción “Antispam” en el menú “Email” y seleccionar “Active”. En esta pantalla también se puede cambiar el nivel de rigurosidad, se debe recordar que mientras más bajo es el número escogido, más alta es la rigurosidad del filtro.



Configuración de la rigurosidad del filtro anti-SPAM

No es recomendable especificar un nivel de rigurosidad menor a 3, puesto que se incrementarían los casos de falsos positivos, que no son otra cosa que mensajes válidos que han sido catalogados como SPAM de manera errónea.

## 18.5 Instalando antivirus

En Elastix se podría instalar cualquier antivirus que trabaje con Postfix, para lo cual habrá que seguir las instrucciones provistas por el respectivo fabricante. A continuación se detalla el procedimiento para instalar ClamAV que es un antivirus gratuito. Se utilizará el paquete ClamSMTP para integrarlo con el Postfix.

Lo primero que hay que hacer es instalar el paquete rpm-build que permitirá generar el RPM del clamsmtp:

```
# yum install rpm-build
```

Luego se debe descargar la versión más reciente del SRPM del clamsmtp desde el sitio web <http://www.inet.lt/clamsmtp/>

```
# wget http://www.inet.lt/clamsmtp/clamsmtp-1.10-1.src.rpm
```

Generar el respectivo RPM, el paquete resultante será almacenado en la carpeta `/usr/src/redhat/RPMS/i386/`.

```
# rpmbuild --rebuild clamsmtp-1.10-1.src.rpm
```

Desde el sitio Web “<http://dag.wieers.com/rpm/packages/clamav/>” se debe descargar las versiones más recientes de clamav, clamd y clamav-db para CentOS 5:

```
# wget http://dag.wieers.com/rpm/packages/clamav/clamav-0.92.1-1.el5.rf.i386.rpm

# wget http://dag.wieers.com/rpm/packages/clamav/clamd-0.92.1-1.el5.rf.i386.rpm

# wget http://dag.wieers.com/rpm/packages/clamav/clamav-db-0.92.1-1.el5.rf.i386.rpm
```

Finalmente se deben instalar los 3 paquetes:

```
rpm -Uvh clamav-0.92.1-1.el5.rf.i386.rpm clamav-db-0.92.1-1.el5.rf.i386.rpm /usr/src/redhat/RPMS/i386/clamsmtp-1.10-1.i386.rpm
```

Se debe crear manualmente la carpeta “/var/run/clamav/”:

```
mkdir /var/run/clamav/
```

Editar el archivo de configuración de “/etc/clamsmtpd.conf” y cambiar las siguientes líneas:

```
ClamAddress: /var/run/clamav/clamd.sock
#User: clamav
```

Por estas dos:

```
ClamAddress: /tmp/clamd.sock et
User: clamav
```

Una vez instalado se debe iniciar los servicios del clamsmtp y clamd :

```
# service clamd start
# service clamsmtp start
# chkconfig --level 3 clamd off
# chkconfig --level 3 clamsmtp off
```

Luego hay que configurar el Postfix para que utilice el paquete instalado. Se debe editar el archivo “/etc/postfix/main.cf” y agregar la siguiente línea:

```
content_filter = scan:[127.0.0.1]:10025
```

También se debe editar el archivo “/etc/postfix/master.cf” y agregar las siguientes líneas:

```
# AV scan filter (used by content_filter)
scan      unix   -       -       n       -       16       smtp
           -o smtp_send_xforward_command=yes
           -o smtp_enforce_tls=no
# For injecting mail back into postfix from the filter
127.0.0.1:10026 inet  n       -       n       -       16       smtpd
           -o content_filter=
           -o
           receive_override_options=no_unknown_recipient_checks,no_header_b
ody_checks
           -o smtpd_helo_restrictions=
           -o smtpd_client_restrictions=
           -o smtpd_sender_restrictions=
           -o smtpd_recipient_restrictions=permit_mynetworks,reject
           -o mynetworks_style=host
           -o smtpd_authorized_xforward_hosts=127.0.0.0/8
```

Y finalmente se debe reiniciar el servicio del Postfix:

```
# service postfix restart
```



# 19

## Mensajería Instantánea

*Nunca he encontrado una persona tan ignorante que no se pueda aprender algo de ella*  
-- Galileo Galilei

Elastix basa su funcionalidad de mensajería instantánea en el software Openfire que funciona en base al protocolo XMPP (originalmente conocido como Jabber).

### 19.1 Protocolo XMPP (Jabber)

El hecho de que se utilice XMPP en Elastix nos da una gran ventaja, debido a que XMPP es un estándar y actualmente existen muchas aplicaciones cliente compatibles con este protocolo. En resumen, tenemos muchas opciones al momento de escoger un cliente de mensajería interna.

XMPP es un protocolo probado, abierto, extensible, generalmente aceptado, seguro y basado en XML. Estas son algunas de las razones por las que fue escogido para formar parte de Elastix.

No es el objetivo de este libro explicar el protocolo XMPP. Sin embargo, es recomendable conocer un poco más de él por lo que dejamos al lector con algunos vínculos interesantes.

- <http://www.jabber.org> Sitio Web oficial del protocolo XMPP
- <http://en.wikipedia.org/wiki/Jabber> Página de Jabber en Wikipedia
- <http://www.jabberes.org> Sitio Web en español

## 19.2 Openfire

Openfire es un completo sistema de mensajería instantánea que nos permite disponer de un servicio de mensajería instantánea rico en características como transferencia de archivos, mensajes *broadcast*, integración con telefonía, integración con otros *gateways* IM, entre otras.

Openfire se encuentra escrito en Java y su funcionalidad puede ser extendida en base a *plugins*, los cuales pueden ser descargados del sitio Web del fabricante ([www.igniterealtime.org](http://www.igniterealtime.org)). Además cuenta con una amigable interfase Web de administración que nos permite realizar las tareas administrativas de una manera intuitiva.

### Características destacables

A continuación algunas de las características más interesantes.

- Interfase amigable de administración Web
- Funcionalidad expandible a través de *plugins*
- Posibilidad de integración con Asterisk
- Interacción con otras redes de mensajería instantánea como Gtalk, Yahoo Messenger, ICQ, entre otros
- Soporte para mensajes *broadcast*
- Transferencia de archivos
- Puede guardar histórico de conversaciones
- Soporte para *clustering*
- Los usuarios pueden ser obtenidos de una base de datos LDAP
- Permite crear diferentes grupos de trabajo

## 19.3 Usando Openfire en Elastix

Openfire viene por omisión desactivado en Elastix. Esto se hace así debido a que si el usuario no lo necesita activar explícitamente es preferible que quede apagado para que no consuma recursos innecesariamente.



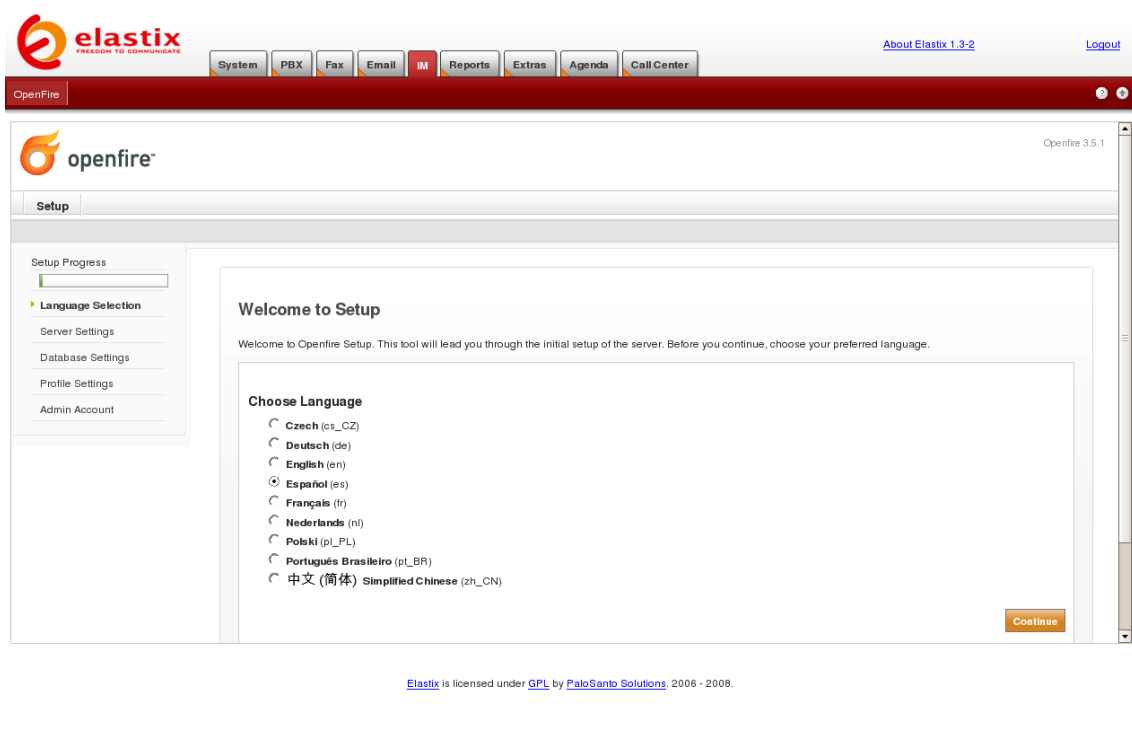
## Activación de Openfire

Si lo queremos activar debemos dar clic en el vínculo que se muestra en la siguiente figura.



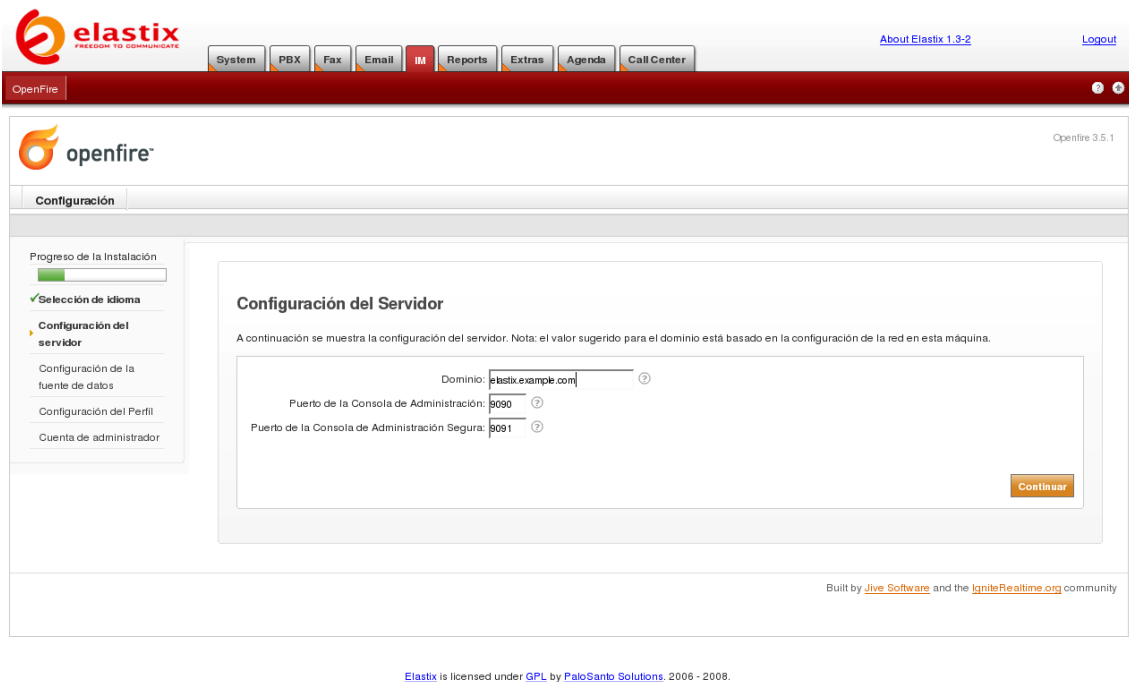
Pantalla de activación de Openfire

Lo primero que veremos es una pantalla donde se nos pide escoger un idioma.



Pantalla de selección de idioma en configurador de Openfire

Luego debemos especificar el nombre de dominio del servidor. Esto debería ser lo mismo que se configuró en los parámetros de red.



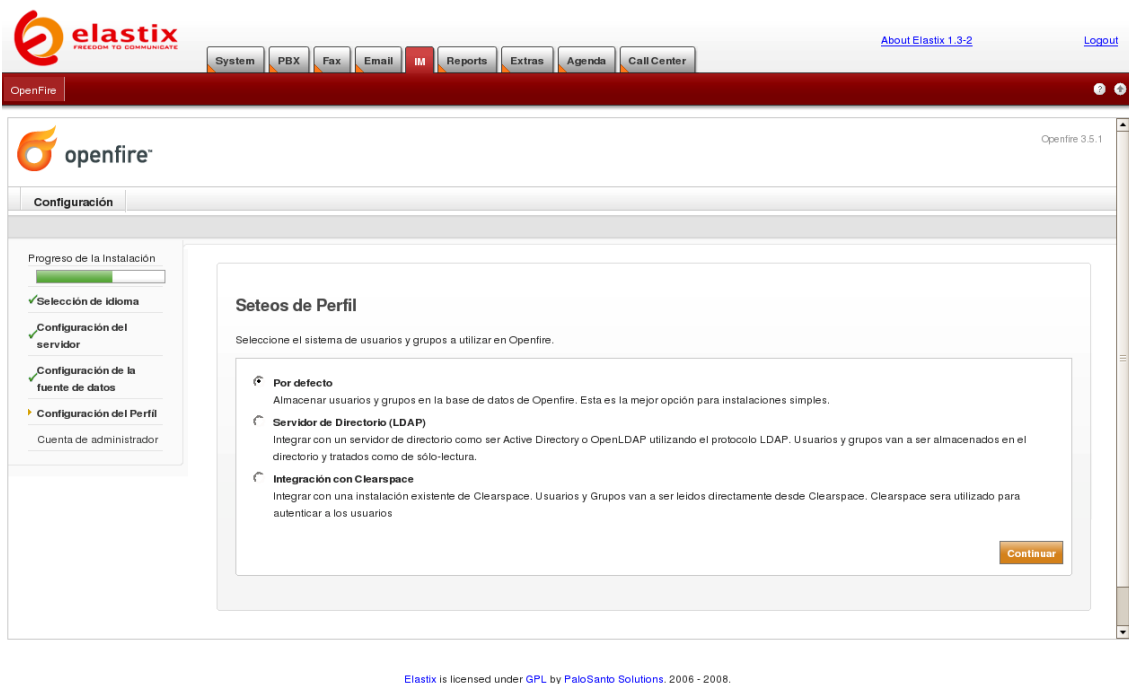
### Configuración de nombre de dominio en el configurador de Openfire

A continuación escogemos la base de datos donde Openfire almacenará la configuración y la información que debe ir almacenando conforme su funcionamiento. Podemos usar un motor de base de datos Standard (Ej. MySQL) o podemos usar una base de datos embebida que trae Openfire y que nos facilita las tareas de configuración si es que no vemos ninguna ventaja en usar un motor Standard.

En la mayoría de los casos es conveniente elegir la base de datos embebida.



A continuación configuramos dónde se almacenarán los usuarios de Openfire. Una opción muy interesante aquí es el soporte LDAP.



### Configuración del sistema de almacenamiento de usuarios de Openfire

Para terminar configuramos la cuenta de administración de Openfire. Con esta cuenta podremos ingresar a la interfaz Web de Openfire y administrar sus parámetros de funcionamiento.

The screenshot shows the Elastix OpenFire 3.5.1 web interface. At the top, there's a navigation bar with links like System, PBX, Fax, Email, Reports, Extras, Agenda, and Call Center. The main content area is titled 'Configuración' (Configuration) and shows a progress bar for the installation. The current step is 'Cuenta de administrador' (Administrator Account). The page contains a form for creating an administrator account with fields for 'Email del Administrador' (pre-filled with 'admin@example.com'), 'Nueva Contraseña' (New Password), and 'Confirme la Contraseña' (Confirm Password). There are 'Saltar este paso' (Skip this step) and 'Continuar' (Continue) buttons. The footer mentions 'Built by Jive Software and the IgniteRealtime.org community' and 'Elastix is licensed under GPL by Palo Santo Solutions, 2006 - 2008'.

Configuración de cuenta de administrador

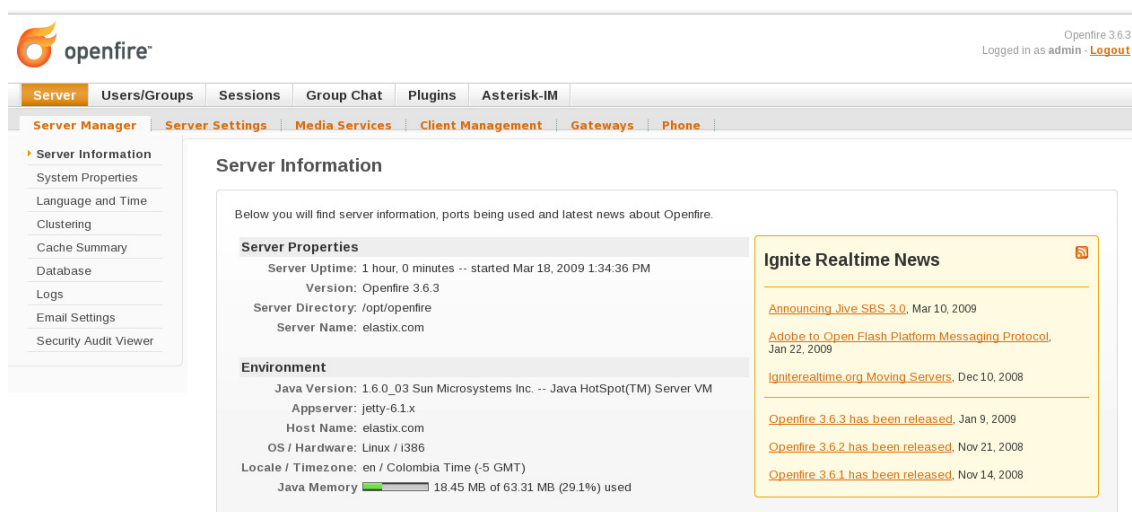
## Interfaz Web de administración de Openfire

Una vez activado Openfire ya podemos ingresar a la interfase de administración, crear algunos usuarios y comenzar a usar mensajería instantánea en Elastix. Lo primero que haremos será ingresar a la interfase como administrador usando la clave configurada previamente.

The screenshot shows the Openfire Administration Console login page. It features the Openfire logo and the text 'Administration Console'. Below this is a login form with two input fields labeled 'username' and 'password', and a 'Login' button. The version 'Openfire, Version: 3.6.3' is displayed at the bottom right.

Ingreso a la consola de administración de Openfire

Una vez dentro lo primero que veremos será una pantalla similar a la siguiente.



Pantalla inicial en la interfase Web de administración de Openfire

Casi lo primero que queremos hacer será crear un nuevo usuario de mensajería instantánea. Esto lo podemos hacer en el menú “*Users/Groups -> Create New User*”. Los parámetros a ingresar son bastante intuitivos y se muestra un ejemplo en la siguiente figura.

Server	<b>Users/Groups</b>	Sessions	Group Chat	Plugins	Asterisk-IM
--------	---------------------	----------	------------	---------	-------------

<b>Users</b>	Groups
--------------	--------

[User Summary](#)  
[▶ Create New User](#)  
[User Search](#)  
[MotD Properties](#)  
[Registration Properties](#)

### Create User

Use the form below to create a new user.

**Create New User**

Username: *	<input type="text" value="testuser"/>
Name:	<input type="text" value="Test User"/>
Email:	<input type="text" value="test@example.com"/>
Password: *	<input type="password" value="....."/>
Confirm Password: *	<input type="password" value="....."/>
Is Administrator?	<input type="checkbox"/> (Grants admin access to Openfire)

\* Required fields

Creando un nuevo usuario de mensajería instantánea

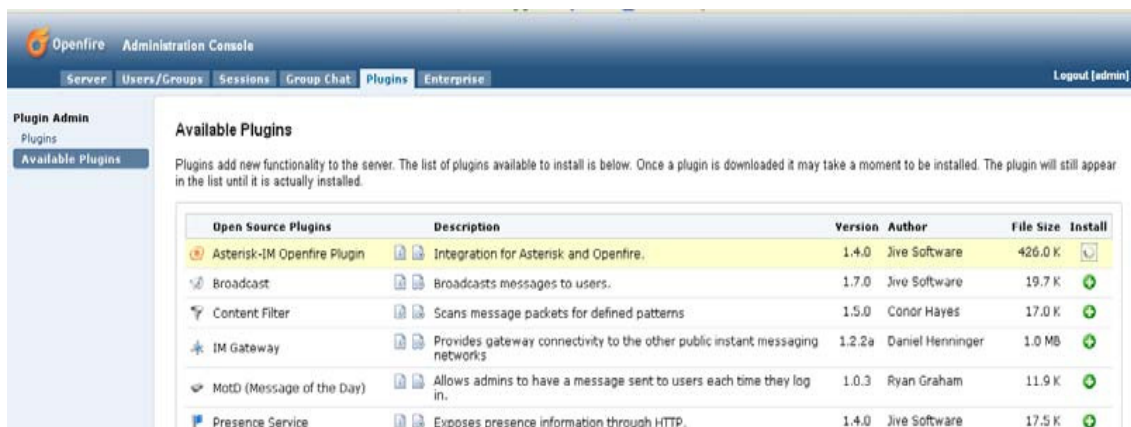
Una vez creado el usuario ya éste se puede registrar desde su cliente de IM.

## Integración de Openfire con Asterisk

Openfire cuenta con un *plugin* que nos permite integración con Asterisk.

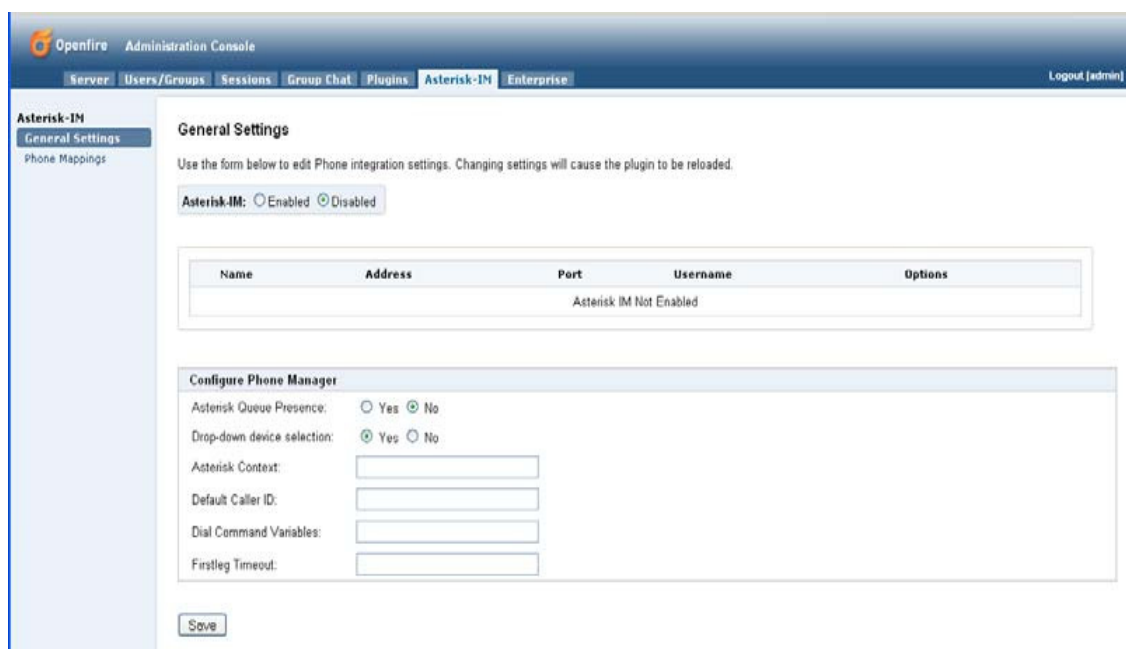
Gracias a este *plugin* podemos efectuar algunas tareas de interacción telefónica desde el cliente de mensajería como por ejemplo generar una llamada, ser notificado cuando una llamada llega al teléfono o incluso poder ver si otro contacto se encuentra en medio de una llamada telefónica. Es por esto que el cliente de mensajería no puede ser cualquiera para que la integración sea visible por el usuario. El *plug-in* funciona con el software de mensajería llamado Spark, el cual también puede ser descargado gratuitamente desde el sitio Web de Openfire.

El *plugin* para integración con Asterisk recibe el nombre de Asterisk-IM y su instalación es muy sencilla. Nos dirigimos al menú “*Plugins -> Available Plugins*” y damos clic sobre el botón verde a la derecha de la fila correspondiente al *plugin* *Asterisk-IM*. En unos segundos estará activo.



Instalación del plugin Asterisk-IM desde la interfase de administración de Openfire

Si el *plugin* se instaló correctamente veremos aparecer automáticamente un nuevo menú llamado Asterisk-IM. Desde aquí podremos configurar la integración de Openfire con Asterisk.



Configuración del plugin Asterisk-IM

Openfire se integra con Asterisk a través de AMI. Por lo tanto, antes de configurar el *plugin* Asterisk-IM debemos haber creado un usuario AMI para este fin, pues en algún momento de la configuración vamos a necesitar ingresar estos datos. Recordemos que los usuarios AMI se crean en el archivo `/etc/asterisk/manager.conf`. También podríamos utilizar el usuario que viene creado por omisión en Elastix. Es decir, el usuario **admin**, con clave **elastix456**

En todo caso para crear un nuevo usuario AMI podríamos agregar las siguientes líneas al `manager.conf`

```
[openfire]
secret = clavesecreta
displayconnects = yes
read = system,call,log,verbose,agent,user,config,dtmf
write = system,call,agent,user,config,command
```

Bueno, dicho lo anterior, iniciemos la configuración del *plugin*. Comencemos por habilitar el *plugin* seleccionando la opción “Enabled”.

En la sección “Configure Phone Manager” encontraremos algunos campos para configurar. No es necesario llenarlos todos, sin embargo es importante llenar el campo “Asterisk Context” con la palabra “default”. Este campo indica el contexto desde el cual se generarán las llamadas telefónicas desde el cliente de mensajería instantánea.

Hasta ahora nuestra configuración debe estar como la mostrada en la siguiente figura.

The screenshot shows the Openfire Administration Console interface. The top navigation bar includes links for Server, Users/Groups, Sessions, Group Chat, Plugins, Asterisk-IM, and Enterprise. The left sidebar shows the Asterisk-IM section with sub-links for General Settings and Phone Mappings. The main content area is titled 'General Settings' and contains a message: 'Use the form below to edit Phone integration settings. Changing settings will cause the plugin to be reloaded.' Below this message, there is a toggle for 'Asterisk-IM' which is currently set to 'Enabled'. A table below the toggle shows a single row with the text 'Asterisk IM Not Enabled'. Further down, the 'Configure Phone Manager' section contains several settings: 'Asterisk Queue Presence' (Yes/No), 'Drop-down device selection' (Yes/No), 'Asterisk Context' (a text field containing 'default'), 'Default Caller ID' (an empty text field), 'Dial Command Variables' (an empty text field), and 'Firstleg Timeout' (an empty text field). A 'Save' button is located at the bottom of this section. At the very bottom of the page, a small footer reads: 'Openfire 3.4.2, built by [Jive Software](#) and the [openfire@realtime.org](#) community.'

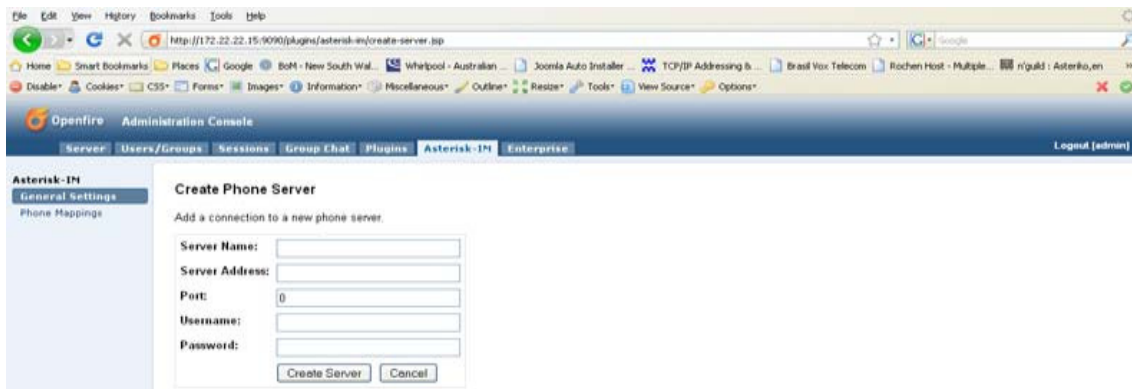
Avance de la configuración del *plugin* Asterisk-IM

Ahora damos clic donde dice “Add Server” para añadir a nuestro servidor Asterisk.

Aquí es donde ingresamos los datos de la cuenta AMI de la que hablamos anteriormente, es decir usuario y clave.

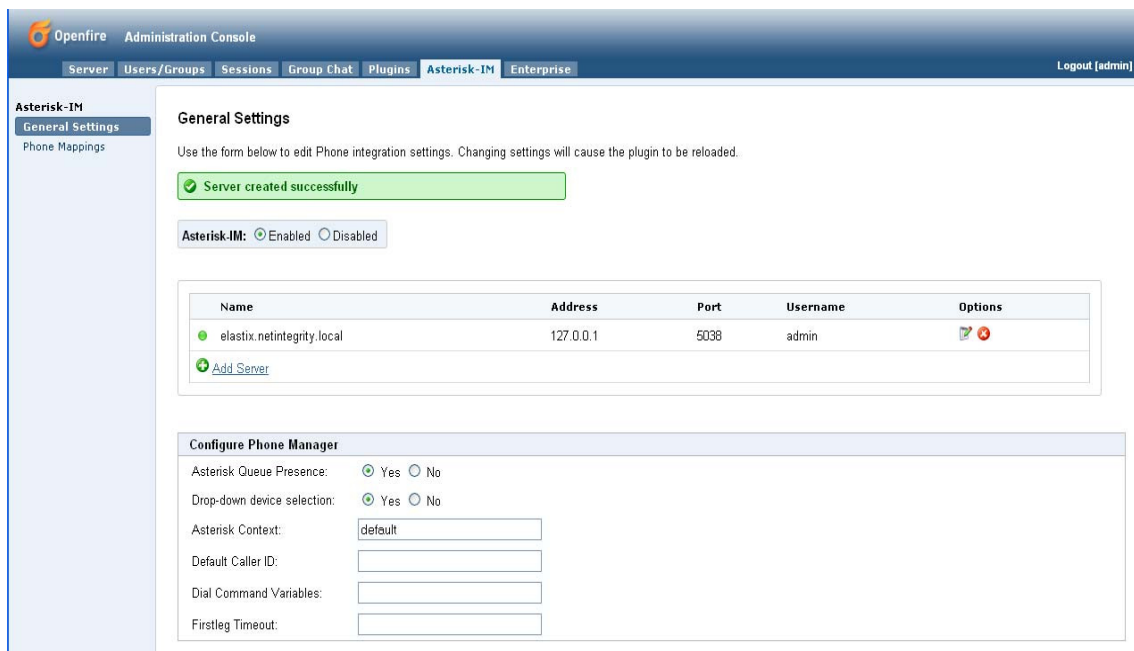
El parámetro **Port** debe llenarse con **5038**, el cual es el puerto AMI por omisión. Supongo que se acordaban, verdad?. En **Server Name** podemos poner cualquier nombre que nos sirva para recordar al servidor y los demás parámetros son algo obvios.





Configuración de servidor AMI

Finalmente tendremos nuestra configuración debe lucir parecida a esta:



Final de la configuración general del *plugin* Asterisk-IM

Bueno, hemos terminado de realizar la configuración general del *plugin* pero nuestro trabajo no ha terminado aquí, no no. Aun debemos mapear o relacionar los usuarios de Openfire con las extensiones telefónicas creadas en Elastix.

Esto es lógico porque no existe manera de que Openfire sepa qué usuario de mensajería instantánea usa una extensión telefónica determinada. Por esto se lo debemos decir explícitamente en el menú “*Asterisk IM -> Phone Mappings*”.

Openfire Administration Console

Server Users/Groups Sessions Group Chat Plugins Asterisk-IM Enterprise

**Asterisk-IM**  
General Settings  
Phone Mappings

### Phone Mappings

Total Users: 0 -- Sorted by Username - Users per Page: 15

Username	Device	Extension	Caller ID	Options
No User/Device Mappings				

**Add User/Asterisk Phone mapping**

\* Username:

\* Device:  or

\* Extension:

Caller ID:

Primary: ☐

\* Required fields

Mapeo de usuarios de Openfire a extensiones de telefonía

Listo, una vez mapeadas las cuentas necesarias ya tendremos mensajería instantánea integrada con telefonía!

# 20

## Fax en Elastix

*El aspecto más triste de la vida actual es que la ciencia gana en conocimiento más rápidamente que la sociedad en sabiduría*

-- Isaac Asimov

### 20.1 Hylafax e IAXModem

Elastix usa una particular combinación de software para el manejo de documentos de fax. En esta combinación hay dos paquetes de software que destacan como Hylafax e IAXModem. Los dos interactúan entre sí de tal manera que Asterisk puede recibir un documento de FAX y enviarlo a una extensión IAX virtual donde reside IAXModem, quien a su vez lo remite a Hylafax quien finalmente convierte el audio correspondiente al FAX en una imagen.

Elastix añade a esta combinación una intuitiva interfase Web de administración donde es posible configurar la característica fax2email que nos permite que los documentos de fax lleguen a direcciones de correo electrónico en formato PDF.

#### Hylafax

Hylafax es un popular servidor de fax que funciona sobre Linux. Según su propio sitio Web “*HylaFAX is an enterprise-class system for sending and receiving facsimiles as well as for sending alpha-numeric pages.*”

Con Hylafax podemos recibir un alto volumen de documentos de fax sin problemas.

La arquitectura del software es cliente-servidor. Es decir que Hylafax abre un puerto de red por donde escucha peticiones de clientes. Estos clientes pueden residir en máquinas remotas desde donde pueden enviar faxes. Suena bien no?

Quiere decir que desde un lugar remoto me puedo conectar a mi servidor Elastix y enviar como fax un documento de texto, PDF, imagen, etc, desde una aplicación cliente.

Hay algunas aplicaciones cliente interesantes entre ellas se encuentra *Winprint HylaFAX* que añade soporte para enviar faxes a través del cuadro de diálogo de impresión que usan las aplicaciones Windows. De esta manera enviar un fax es tan sencillo como imprimir un documento.

## **IAXModem**

Debido a que HylaFax se puede comunicar con *modems* y Asterisk con extensiones IAX, necesitamos algún pegamento para unir ambas cosas y que puedan funcionar juntas. Ese pegamento es IAXModem.

IAXModem es un paquete de software que simula ser una extensión IAX por el lado de Asterisk y por el lado de HylaFax simula ser un *modem* serial. Es decir que incluso se registra como una extensión IAX con su respectivo usuario y clave.

Como es de suponer, este teléfono virtual, que simula ser el IAXModem, se ejecuta en el mismo servidor Elastix y por tanto se conectará desde `localhost` (la dirección IP 127.0.0.1). Es fácil verificar si tenemos una extensión IAXModem conectada desde la consola del CLI pues basta con examinar si existe alguna extensión IAX ejecutándose desde `localhost`.

```
elastix*CLI> iax2 show peers
```

Name/Username	Host	Mask	Port	Status
620	127.0.0.1	(D) 255.255.255.255	40001	OK (2 ms)
660	(Unspecified)	(D) 255.255.255.255	0	UNKNOWN
661	192.168.1.136	(D) 255.255.255.255	4569	OK (2 ms)
662	127.0.0.1	(D) 255.255.255.255	40000	OK (3 ms)
709	(Unspecified)	(D) 255.255.255.255	0	UNKNOWN
712	192.168.1.126	(D) 255.255.255.255	4569	OK (5 ms)
7214	(Unspecified)	(D) 255.255.255.255	0	UNKNOWN
7213	(Unspecified)	(D) 255.255.255.255	0	UNKNOWN
723	(Unspecified)	(D) 255.255.255.255	0	UNKNOWN
869	(Unspecified)	(D) 255.255.255.255	0	UNKNOWN
966	(Unspecified)	(D) 255.255.255.255	0	OK (9 ms)
964	(Unspecified)	(D) 255.255.255.255	0	UNKNOWN
863	192.168.1.233	(D) 255.255.255.255	4569	OK (9 ms)

De la salida anterior podríamos concluir que las extensiones 620 y 662 son extensiones de FAX manejadas por IAXModem.

## 20.2 Configuración de Fax en la interfase Web de Elastix

### Listado de Fax Virtual

La opción “Listado de Fax Virtual” del Menú “FAX” del Elastix nos permite verificar el listado de todos los faxes virtuales, donde se muestra el estado de cada uno.



Virtual Fax List						
<a href="#">Start</a> / <a href="#">Previous (1 - 1 of 1)</a> <a href="#">Next</a> <a href="#">End</a>						
Virtual Fax Name	Fax Extension	Secret	Destination Email	Caller ID Name	Caller ID Number	Status
<a href="#">Desarrollo Fax Virtual</a>	511	1234	afgueroa@palosanto.com	Palosanto	2294440	Waiting for modem to come free
<a href="#">Start</a> / <a href="#">Previous (1 - 1 of 1)</a> <a href="#">Next</a> <a href="#">End</a>						

Listado de faxes virtuales

Además al dar clic sobre el nombre del fax virtual podemos ver los datos del mismo:

**View Virtual Fax**

Edit Delete \* Required field

Virtual Fax Name: *	Desarrollo Fax Virtual	Fax Extension (IAX): *	511
Destination Email: *	afigueroa@palosanto.com	Secret (IAX): *	1234
Caller ID Name:	Palosanto		
Caller ID Number:	2294440		

Visualización de nueva extensión virtual de fax

En la parte superior se encuentran 2 botones, el primer botón “Editar” nos permite editar los datos del fax virtual.

Y el segundo botón “Eliminar” nos permite eliminar el fax virtual.

Si procedemos a editar tendremos lo siguiente:

**Edit Virtual Fax**

Apply changes Cancel \* Required field

Virtual Fax Name: *	Desarrollo Fax Virtual	Fax Extension (IAX): *	511
Destination Email: *	afigueroa@palosanto.com	Secret (IAX): *	1234
Caller ID Name:	Palosanto		
Caller ID Number:	2294440		

Edición de fax virtual

Una vez modifiquemos los valores, damos clic sobre el botón “Aplicar Cambios” para guardar los cambios o sobre “Cancelar” para regresar sin guardar.

## Nuevo Fax Virtual

La opción “Nuevo Fax Virtual” del Menú “FAX” del Elastix nos permite crear un nuevo fax virtual. Previamente debe haberse creado la extensión de fax de tipo IAX, para mayor información acerca de como crear una extensión diríjase a Creación de nueva extensión.

Formulario de creación de una nueva extensión de fax virtual

Para crear un nuevo fax virtual ingrese nombre, *email*, extensión y clave secreta para el fax virtual, estos son campos requeridos. Además de esto existen 2 campos adicionales que son el nombre y número del *Caller ID*. Una vez ingresados estos datos de clic en el botón “Guardar” para ingresar el nuevo fax virtual o “Cancelar” para salir sin guardar.

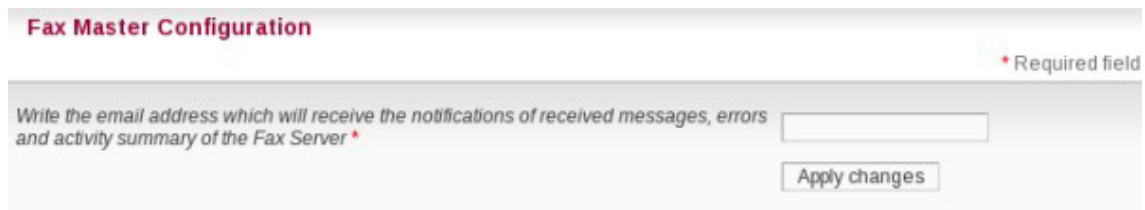
Note que la extensión del fax virtual será usada para recibir un fax.

Los datos a ingresar se describen a continuación:

Nombre	Descripción
Nombre de Fax Virtual	Nombre para identificar el fax virtual
Extensión Fax (IAX)	Extensión usada como servicio de fax. Esta extensión debe haber sido previamente creada como tipo IAX en el módulo “Extensiones” del Menú “Configuración PBX”
Email Destino	Email al cual se enviarán notificaciones por parte de la central de los faxes recibidos y de los errores en caso de haberlos
Secreto (IAX)	Debe ser la misma que la ingresada al crear la extensión de fax en el menú de “Configuración PBX”
Nombre Caller ID	El nombre de quien llama. Este campo no es obligatorio, si no se llena quien reciba un fax de esta extensión no podrá ver el nombre
Número Caller ID	Número de quien llama. Al igual que el anterior no es obligatorio, y si no se llena quien reciba el fax no podrá ver el número.

## Fax Master

La opción “Fax Master” del Menú “FAX” del Elastix nos permite ingresar la dirección de email del administrador del Fax al cual le llegarán las notificaciones de los mensajes recibidos, errores y la actividad del Servidor de Fax.



Configuración del Fax Master

Proceda a ingresar el email del administrador de fax y luego de clic en el botón “Aplicar Cambios”.

## Clientes de Fax

La opción “Clientes de Fax” del Menú “FAX” del Elastix nos permite ingresar las IPs que tendrán permiso para enviar faxes por medio de Elastix.



Configuración de IPs de clientes de fax

Proceda a ingresar las IPs, una IP por fila y luego de clic en el botón “Aplicar Cambios”.

Se recomienda ingresar la IP 127.0.0.1 y localhost en la configuración debido a que algunos procesos pueden necesitarlas.

## Visor de Fax

La opción “Visor de Fax” del Menú “Fax” del Elastix nos permite visualizar un listado con el detalle de las faxes recibidos.



File	Company Name	Company Fax	Fax Destiny	Fax Date
<a href="#">fax00000008.pdf</a>	58942 29024400000	58942 29024400000	Desarrollo Fax Virtual - 511	2007-09-24 18:30:27
<a href="#">fax00000009.pdf</a>	58942 29024400000	58942 29024400000	Desarrollo Fax Virtual - 511	2007-09-24 18:44:26
<a href="#">fax00000010.pdf</a>	58942 29024400000	58942 29024400000	Desarrollo Fax Virtual - 511	2007-09-24 18:48:51
<a href="#">fax00000011.pdf</a>	58942 29024400000	58942 29024400000	Desarrollo Fax Virtual - 511	2007-09-24 18:50:43

Visor de faxes. Si se da clic en alguno se lo visualiza en formato PDF.

Además tenemos la opción de descargar el fax en formato PDF, dando clic sobre el archivo deseado de la lista de faxes recibidos.

Nombre	Descripción
Nombre de Compañía	Nombre de la compañía que envía el fax
Fax de Compañía	Número de Fax de la compañía que envía el fax
Fecha de Fax	Fecha de envío del fax

## 20.3 Enviando un FAX desde una estación de trabajo

Una gran ventaja del esquema de FAX de Elastix es que permitirá a los usuarios enviar documentos de FAX desde la misma computadora, eliminando la necesidad de imprimir el documento y pasarlo a través de la maquina de FAX.

Lo anterior es posible hacerlo incluso desde la misma aplicación en la que se genera el documento, como el software de procesador de texto u hoja de cálculo. Para esto usaremos un programa que simulará ser una impresora. Es decir que podemos enviar a imprimir a esta impresora virtual cualquier documento y este en lugar de imprimirse se enviará por FAX. Sin duda esto añadirá comodidad a los usuarios y nos proporcionará un inherente ahorro de papel que siempre viene bien.

Veamos qué hacer para estaciones de trabajo con sistema operativo Windows que puede ser el caso más común.

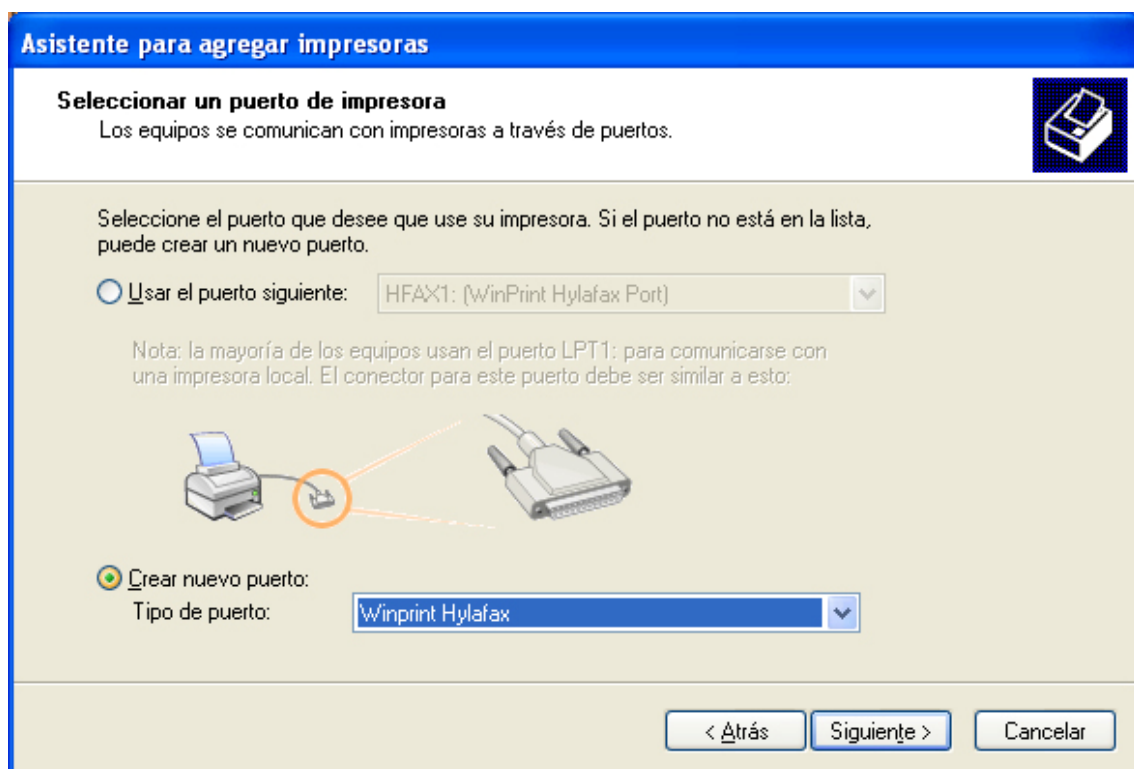
Lo primero que debemos hacer es descargar el software llamado WinPrint HylaFAX y esto lo podemos hacer desde la página del proyecto ubicada en <http://winprinthylafax.sourceforge.net>

Configurar WinPrint HylaFAX es una tarea que podríamos dividir en dos partes. La primera es añadir una nueva impresora y la segunda es configurar los parámetros del servidor HylaFAX.

Comencemos por añadir la nueva impresora. Para esto debemos seguir una serie de pasos que se relatan a continuación.

1) Debemos ir al panel de control y añadir una nueva impresora. En el primer cuadro de diálogo hay que escoger que se trata de una impresora local pero des-seleccionar donde dice que se detecte la impresora con plug-n-play ya que no se trata de una impresora real.

2) Luego aparecerá una nueva ventana que nos preguntará a qué puerto se encuentra conectada la impresora y debemos crear un nuevo puerto del tipo “Winprint Hylafax”



Configurando WinPrint HylaFAX

3) A continuación se nos preguntará el nombre del puerto. Podemos poner cualquier identificador aquí pero el nombre por omisión está bien.

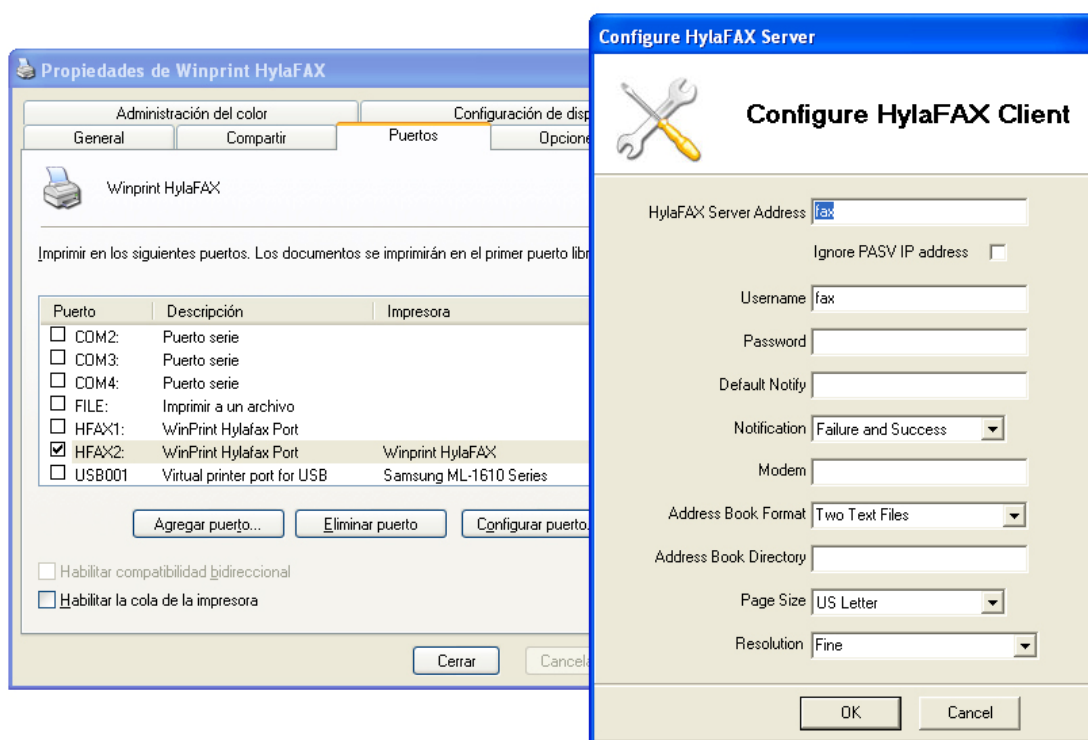
4) Una ventana aparecerá pidiéndonos especificar el tipo de impresora. Podemos escoger aquí cualquier impresora Postscript. El fabricante recomienda escoger “Apple LaserWriter 12/640 PS”

5) El resto de pasos son triviales y terminaremos con la configuración de la nueva impresora. Lo único importante es no elegir compartir esta impresora pues ya sabemos que no se trata de una impresora.

Hemos terminado con la primera parte, ahora configuremos los parámetros del servidor HylaFAX que reside en nuestro Elastix.

1) En el panel de control damos clic derecho en la impresora que acabamos de crear y escogemos “Propiedades”. Allí nos aparecerá una ventana con varias pestañas, debemos elegir la que dice “Puertos”.

2) En esta nueva ventana debemos configurar el puerto que aparece señalado por omisión. Aparecerá un cuadro de diálogo donde debemos ingresar los datos de nuestro servidor HylaFAX. Debemos configurar aquí la dirección IP de nuestro servidor Elastix.



Configuración de los parámetros del servidor HylaFAX

3) No debemos olvidar que para que HylaFAX permita a este computador enviar un FAX el IP del computador debe estar autorizada por HylaFAX y esto lo podemos configurar desde la interfase Web del Elastix en el menú “FAX -> Clientes de FAX”. Ya vimos cómo configurar esto antes, en este mismo capítulo.

Finalmente nos encontramos listos para enviar un FAX desde cualquier aplicación como si enviáramos el documento a imprimir. Al enviar el trabajo a la impresora virtual nos aparecerá un cuadro de dialogo donde debemos ingresar el número de teléfono al que queremos enviar el FAX.

## 20.4 Procesamiento interno de un documento de FAX en Elastix

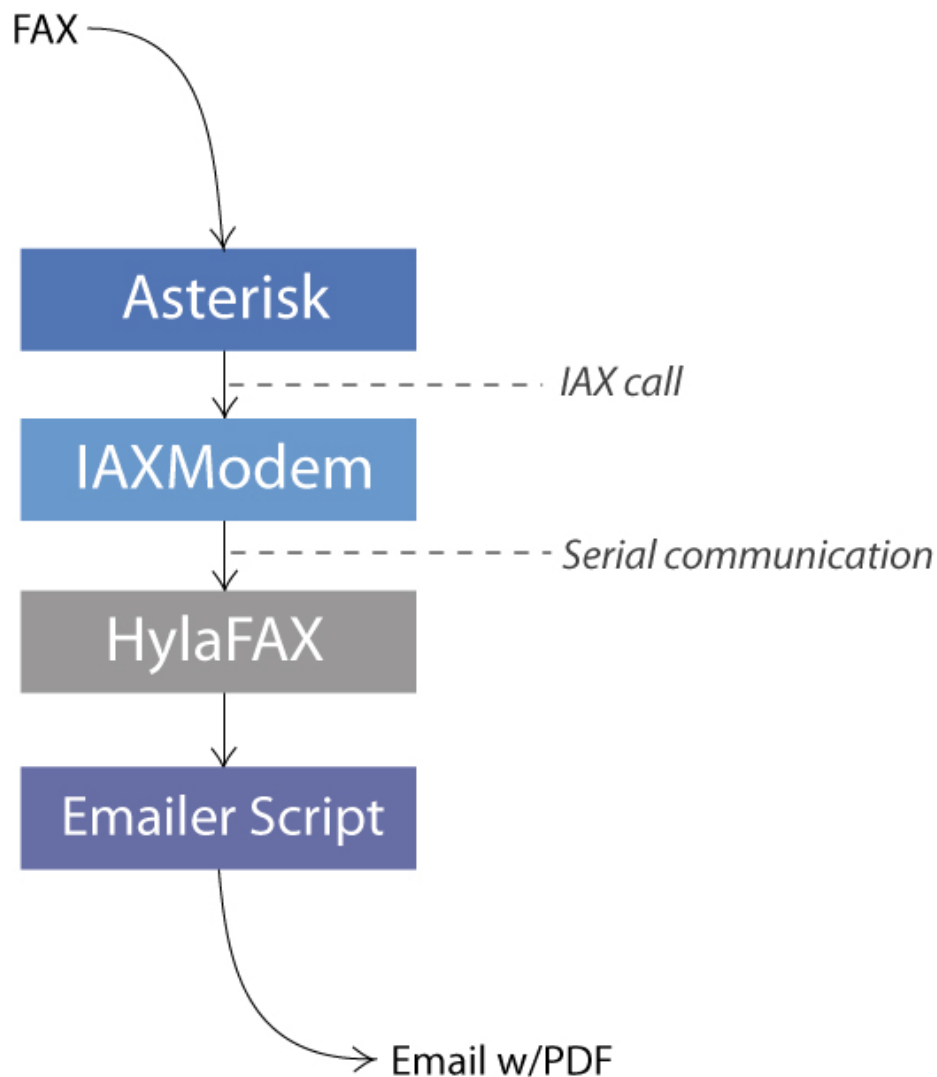
Algunas cosas interesantes suceden internamente en Elastix desde el momento en que llega una transmisión de FAX. El esquema de manejo de FAX en Elastix ha demostrado ser una solución robusta y escalable pero que involucra algunos componentes y es importante conocer como se relacionan entre sí.

Desde el momento en que un usuario marca a nuestro Elastix con la intención de enviarnos un FAX sucede más o menos lo siguiente.

- 1) Asterisk contesta como si fuera una llamada normal y el usuario selecciona la extensión de FAX
- 2) El usuario sin saberlo está marcando una extensión IAX controlada por el software IAXmodem que simula ser un teléfono IAX y contesta la llamada
- 3) IAXmodem mantiene la llamada y todo el *stream* de FAX lo escribe en un dispositivo virtual ubicado en `/var/spool/hylafax/FIFO.tty*` que simula ser un MODEM
- 4) Por otro lado Hylafax se encuentra escuchando el mismo dispositivo pensando que en realidad se trata de un MODEM y de esta manera recibe la señal de FAX
- 5) Internamente Hylafax transforma este *stream* a un formato humanamente legible llamado PostScript
- 6) Este archivo PostScript es pasado a una aplicación que llamaremos *Emailer Script* quien a su vez transforma este archivo a un formato más legible, como el formato PDF. Luego construye un mensaje de correo electrónico y envía el PDF adjunto.

Antes de enviar el email el *Emailer Script* consulta una base de datos interna de Elastix para recabar información del cuerpo del mensaje y el destinatario del email ya que como recordaremos esto fue configurado desde la interfase Web y fue almacenado en esta base de datos interna.

Para hacer más gráfico el procesamiento interno del FAX observemos la siguiente figura.



Comunicación entre los componentes del sistema de FAX en Elastix

## 20.5 Protocolo T.38

Si vamos a trabajar con tecnologías de FAX en algún momento nos toparemos con el término T.38. El protocolo T.38 es un protocolo recomendado por la ITU que sirve para transportar FAX sobre redes IP.

Si nos ponemos a pensar, una manera de transportar fax sobre redes IP sería transportar la señal de FAX como si fuera audio o voz a través de los protocolos de Voz sobre IP ya existentes en la actualidad, y en efecto esta es una manera de hacerlo. Sin embargo, esta no es la manera más adecuada en algunos casos pues, a diferencia de la voz, el FAX es mucho más sensible a errores de transmisión y por ejemplo una red con un nivel de pérdida de paquetes que pueda sonar aceptable para la transmisión de voz es probable que resulte inaceptable para la transmisión de FAX pues el documento podría llegar ilegible o simplemente no llegar.

El protocolo T.38 es quien resuelve estos problemas usando algunas técnicas interesantes como la redundancia de paquetes. Es decir que por cada paquete enviado a través de la red es probable que se envíen algunas copias por si alguno se pierde en el camino.

No entraremos en más detalles acerca de T.38 sino más bien nos dedicaremos a explicar el soporte actual que existe de T.38 en Asterisk ya que es un tema frecuente de discusión. En caso de que el lector sienta curiosidad por conocer más detalles de este protocolo le recomendamos visitar <http://en.wikipedia.org/wiki/T.38>

A partir de Asterisk 1.4 se introdujo soporte limitado para el protocolo T.38. Decimos limitado porque Asterisk no es capaz de terminar una transmisión T.38, es más no esta en capacidad de entender el protocolo. Lo único que puede hacer Asterisk con T.38 es pasar la comunicación de un lado a otro. O sea que T.38 puede pasar a través de Asterisk y a esta modalidad de funcionamiento se le llama *passthrough*.

En su versión 1.6, Asterisk añadirá más soporte para T.38 y estará en la capacidad de entender el protocolo un poco más y por lo tanto hacer algunas cosas interesantes. Sin embargo el soporte no será completo aun.

Para aquellos que requieren soporte T.38 y no pueden esperar a nuevas versiones de Asterisk les dejamos con la noticia de que existen soluciones comerciales para añadir soporte T.38 como la ofrecida por la empresa <http://www.attractel.com>

## 20.6 AvantFAX, un GUI alternativo

Muchas sugerencias han existido en los foros de Elastix acerca de soportar AvantFAX en la distro. AvantFAX nos ofrece una interfase Web para administrar faxes usando HylaFAX.

En busca de una solución recientemente hemos lanzado un instalador RPM de AvantFAX que puede ser descargado del repositorio de Extras de Elastix.

Para más información de AvantFAX visite <http://www.avantfax.com>

# 21

## Sintetización y reconocimiento de voz

*Si he logrado ver más lejos, ha sido porque he subido a hombros de gigantes*

-- Isaac Newton

Las tecnologías de sintetización y reconocimiento de voz han sido ideas algo futuristas hasta hace relativamente poco tiempo. En la actualidad es posible usar estas tecnologías en la práctica con resultados muy interesantes.

En Elastix se puede habilitar el soporte para ambos conceptos permitiéndonos con esto obtener un Servidor de Comunicaciones de vanguardia.

### 21.1 Sintetización de voz

Sintetización de voz es la tecnología que permite convertir el texto a audio. También se conoce como TTS debido al término Text-To-Speech (texto-a-habla en inglés).

Aunque parezca insólito su historia se remonta a algunos siglos atrás cuando se intentó sintetizar voz mecánicamente con aparatos que simulaban el aparato vocal humano. En ese entonces no fueron más que juguetes. Ya en la década de los 30 laboratorios Bell

desarrollaron el famoso VOCODER, que era al mismo tiempo un aparato de reconocimiento de voz que utilizaba electrónica y que podía producir sonido inteligible. Su finalidad original era la de reconocer voz para luego poder transmitirla de manera codificada por un medio y sintetizarla en el otro extremo.

El uso de la computación de allí en adelante hizo que fuera posible desarrollar tecnología de sintetización de voz que fuera práctica y que pudiera estar al alcance de cualquiera que pudiera pagar por ella. Debido a que en un principio muchas de estas tecnologías desarrolladas fueron propietarias y su oferta limitada sus precios fueron elevados y orientados a grandes empresas o bancos.

Actualmente ya existen alternativas de sintetización de voz de relativo bajo costo e inclusive de libre uso como Festival o FreeTTS. En breve explicaremos cómo hacer uso del motor de sintetización de voz Festival desde nuestro servidor Elastix.

Veamos un poco de cómo funciona la técnica de sintetización de voz moderna como la usada por Festival.

Podemos dividir al proceso de convertir el texto a voz en 3 partes

### Análisis de texto

Esta etapa es la que se encarga de normalizar el texto, es decir traducirlo a un formato de palabras estándar y por lo tanto más entendible para ser manipulado en la siguiente etapa. El texto crudo o sin manipular puede contener palabras mezcladas con números o abreviaciones que no son aptas para la conversión a audio. Por ejemplo, una frase como “Me debes \$100 mil dólares y debes pagarme el 10% ahora” se podría escribir como “Me debes cien mil dólares y debes pagarme el diez por ciento ahora” para fines de sintetizarla a voz.

*Nota: El ejemplo anterior tiene la finalidad de ilustrar el concepto de normalización del texto, no necesariamente muestra el lenguaje real de un texto normalizado.*

### Análisis lingüístico

Esta etapa es la que se encarga de convertir el texto normalizado en fonemas. Los fonemas son las unidades fonéticas que componen un lenguaje hablado. Son sonidos únicos y diferentes entre sí.

Una palabra está compuesta por uno o varios fonemas.



Para traducir el texto normalizado en fonemas hace falta bastante trabajo pues se requiere entre otras cosas conocer la entonación de la palabra pues no es lo mismo si se trata de un contexto de pregunta o exclamación.

*Nota: En el caso de Festival se trabaja con difonos más que con fonemas.*

### Generación de audio

Una vez que tenemos el texto en una representación de fonemas nos falta traducir esos fonemas a audio. Habrá que unir o concatenar el audio correspondiente a cada fonema e intercalarlo con los silencios que van entre las palabras.

Sin embargo, no es tan sencillo como suena pues el simple hecho de unir los fonemas audibles puede resultar en un audio de baja calidad que suene robótico o inhumano.

Existen algunas técnicas o algoritmos que se pueden usar para resolver este problema y hacer que la transición entre fonemas sea lo más suave posible. El algoritmo por omisión usado por Festival se llama *Residual Excited Linear Prediction* o simplemente RELP o RE-LPC. Sin embargo existen otros algoritmos que pueden ser utilizados por Festival como es el caso de MBROLA.

*Nota: La técnica de concatenar fonemas no es la única para generar audio a partir de fonemas pero es la que usa Festival. Para ser más específicos Festival usa una técnica de concatenación de difonos.*



## **Festival**

Ya hemos visto algo del funcionamiento de Festival pues ha sido tomada como ejemplo para ilustrar el proceso de conversión de texto a voz en el apartado pasado.

Festival es un sistema de sintetización de voz desarrollado por la Universidad de Edimburgo utilizando lenguaje de programación C++.

Festival puede soportar varios lenguajes, viene instalado por omisión en Elastix y podemos utilizarlo desde Asterisk; lo que lo convierte en una alternativa atractiva si estamos pensando en hacer sintetización de voz.

Festival puede trabajar como cliente y como servidor. Como cliente puede trabajar en dos modos: línea de comandos y `tts` (*text-to-speech*).

### Modo línea de comandos

En el modo línea de comandos Festival ejecutará un intérprete de comandos. Para entrar a este modo basta ejecutar el comando `festival` sin opciones.

```
[root@elastix ~]# festival
Festival Speech Synthesis System 1.95:beta July 2004
Copyright (C) University of Edinburgh, 1996-2004. All rights
reserved.
For details type `(festival_warranty)'
festival>
```

Como vemos obtenemos un *prompt* que dice **festival>** y desde el cual se pueden ejecutar comandos de Festival en formato similar al del editor de texto Emacs. Para más información se puede tipear la palabra `help`.

Bueno, ya que estamos en la línea de comandos aprovechemos para reproducir nuestro primer audio sintetizado en la tarjeta de sonidos. Obviamente tendremos que conectar unos audífonos allí para escuchar algo.

```
festival> (SayText "Hola Mundo")
#<Utterance 0xb7132578>
```

También podríamos sintetizar el contenido entero de un archivo de texto desde la línea de comandos.

```
festival> (tts "content.txt" nil)
nil
festival> (quit)
```

Finalmente salimos de la línea de comandos con el comando `(quit)`. También podemos salir con la combinación de teclas `Ctrl+D`.

### Modo `tts`

En el modo `tts` podemos sintetizar texto a audio sin entrar en la línea de comandos. Para activar el modo `tts` simplemente hay que pasar el parámetro `--tts` cuando se ejecuta el comando `festival`.

Por ejemplo, para sintetizar un archivo de texto podemos ejecutar el siguiente comando en modo `tts`:

```
[root@elastix ~]# festival --tts archiv01.txt
```

Si queremos ejecutar un texto arbitrario sin utilizar un archivo de por medio podemos hacerlo mediante una invocación con una tubería (*pipe*).

```
[root@elastix ~]# echo "Hola Mundo, este es un ejemplo" |  
festival --tts
```

Por omisión Festival tratará de enviar el audio al dispositivo o tarjeta de sonido, pero muchas veces no es práctico hacerlo de esta manera por lo que veremos cómo enviar la salida de audio a un archivo para escucharlo después.

Para realizar lo anterior Festival nos proporciona un comando para facilitar las cosas llamado `text2wave`. El uso del comando `text2wave` es sumamente sencillo.

```
[root@elastix ~]# text2wave archivo.txt -o archivo.wav
```

Donde `archivo.txt` contiene el texto a ser sintetizado y `archivo.wav` representa el archivo de audio donde se guardará la salida.

## **Cómo interactuar con Festival desde el plan de marcado**

Asterisk nos provee de una aplicación de plan de marcado llamada Festival. Esta aplicación funciona internamente conectándose a un servidor Festival a través de la red y reproduciendo la salida de audio en el canal en uso (o corriente) en el plan de marcado.

Esto nos brinda la flexibilidad de ejecutar un servidor Festival inclusive en un equipo diferente al servidor Elastix. En algunos casos esto es conveniente si nos encontramos en un ambiente con alto volumen de llamadas y queremos descongestionar la carga del servidor. Explicaremos más adelante cómo ejecutar Festival como servidor de red.

La sintaxis de la aplicación Festival es la siguiente.

```
Festival(text[|intkeys])
```

Donde `text` es el texto que se desea sintetizar (y reproducir) e `intkeys` es un parámetro opcional que especifica las teclas telefónicas que pueden interrumpir la ejecución del comando. Si el parámetro `intkeys` toma el valor de “any” quiere decir que el usuario puede interrumpir la ejecución del comando con cualquier tecla.

Veamos un ejemplo de utilización de la aplicación Festival en el plan de marcado.

```
exten => 5333,1,Answer
exten => 5333,n,Festival('Hola, esto es un ejemplo de conversión
de texto a audio','any');
exten => 5333,n,Hangup()
```

Con el ejemplo anterior hemos creado una extensión 5333. Si se llama a esa extensión escucharemos el mensaje tipeado.

### Ejecutando Festival como servidor de red

Festival viene con un *script* llamado `festival_server` que sirve para arrancar Festival como servidor de red. Por omisión levantará un servicio de red en el puerto 1314.

Para arrancar el servidor Festival debemos ejecutar el siguiente comando.

```
[root@elastix ~]# festival_server &
```

***Nota:*** Una manera sencilla de hacer que este comando se ejecute cada vez que el servidor Elastix se reinicie es incluyéndolo en el archivo `/etc/rc.local`

Sin embargo, antes de iniciar el servidor Festival debemos hacer unas modificaciones a la configuración de Festival pues Asterisk necesita que el servidor Festival entienda cierto comando y que le envíe audio en un formato particular.

Para configurar Festival para que funcione adecuadamente con Asterisk debemos editar el archivo `festival.scm` ubicado en `/usr/share/festival/`

```
(define (tts_textasterisk string mode)
  "(tts_textasterisk STRING MODE)
  Apply tts to STRING. This function is specifically designed for
  use in server mode so a single function call may synthesize the
  string.
  This function name may be added to the server safe functions."
  (let ((wholeutt (utt.synth (eval (list 'Utterance 'Text string)))))
    (utt.wave.resample wholeutt 8000)
    (utt.wave.rescale wholeutt 5)
    (utt.send.wave.client wholeutt)))
```

Las podemos añadir cerca del inicio del archivo luego de los comentarios iniciales.

Luego de esto podemos arrancar Festival como servidor (como ya explicamos antes) y reiniciar Asterisk. Finalmente marcamos la extensión 5333 y listo!

En caso de que el archivo `festival.scm` no haya sido modificado veremos un error como el siguiente en la salida del CLI.

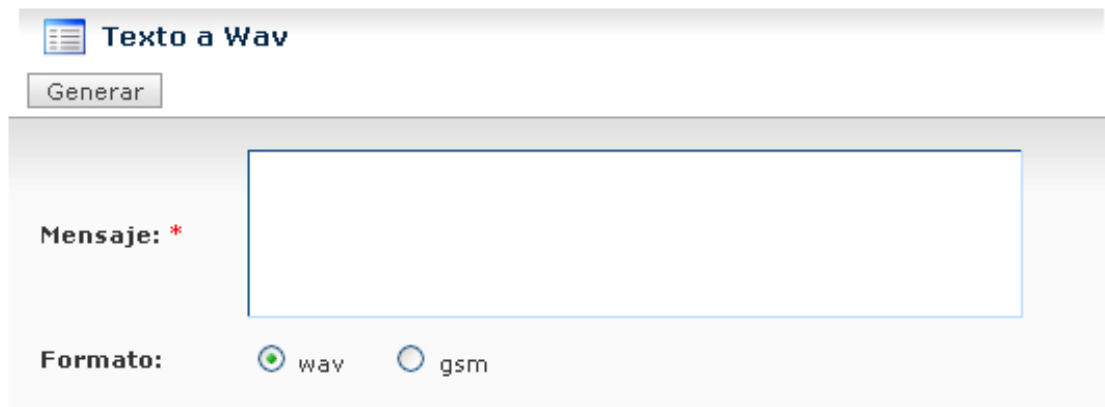
```
elastix*CLI> SIOD ERROR: unbound variable : tts_textasterisk
```

Por omisión todo debe funcionar bien. Sin embargo, es necesario explicar que los parámetros de conexión de Asterisk con el servidor Festival se pueden modificar en el archivo `/etc/asterisk/festival.conf`. Un cambio común en este archivo es cuando se desea ejecutar el servidor Festival en un servidor separado. Para esto deberemos cambiar el parámetro “host” por el IP del equipo donde se encuentra ejecutándose el servidor Festival.

## Sintetizando voz desde la interfase Web de Elastix

Elastix viene equipado con una bonita interfase Web que nos permite sintetizar texto a voz. Está basada en Festival y se puede encontrar en el menú de “Extras -> Texto a Wav”.

La salida puede obtenerse en formato wav o gsm y podría incluso servir para crear *promts* de voz en caso.

The image shows a web application window titled "Texto a Wav". At the top left is a small icon of a document with lines. Below the title is a button labeled "Generar". The main area contains a text input field with the label "Mensaje: \*" to its left. Below the input field is a "Formato:" label followed by two radio buttons: "wav" (which is selected, indicated by a green dot) and "gsm" (which is unselected, indicated by a blue dot).

Interfase Web para conversión de texto a wav

## 21.2 Reconocimiento de voz

Existen algunas definiciones de lo que es el reconocimiento de voz, por ejemplo se define como el proceso en el un sistema computacional reconoce lo que un ser humano le dice. Generalmente, este reconocimiento representa una orden que deberá ejecutar el computador.

El reconocimiento de voz, hoy en día tiene muchas aplicaciones que las vemos en los teléfonos celulares, en la domótica (o control de artefactos domésticos por medio de la voz), procesadores de texto que reciben “dictado”, en los ámbitos de la salud existen aplicaciones más específicas para personas discapacitadas que le permiten abrir y cerrar puertas, así como también en el mundo de la telefonía hay implementaciones en las que cuando se llama a una empresa, el sistema telefónico le pedirá decir el nombre de la persona con la que se desea hablar y el sistema reconocerá el nombre que usted ha dicho y lo redireccionará a la extensión deseada.

El término reconocimiento de voz es muchas veces utilizado para referirse al reconocimiento del habla. El reconocimiento de voz se entiende como identificar una voz en particular para un individuo. Esto se aplica a sistemas de seguridad que asocian la voz con una persona.

En un sistema telefónico lo que se usa es el reconocimiento del habla. Un sistema muy utilizado para sistemas telefónicos IP basados en Asterisk es LUMENVOX.

### Lumenvox

LUMENVOX aunque usa algunos de los fundamentos del reconocimiento de voz, es diferente ya que no trata de identificar a individuos, más bien trata de identificar lo que éste dice. La mayoría de los sistemas de reconocimiento de frases ó palabras siguen el siguiente procedimiento:

1. Cargar una lista de palabras para ser reconocidas. Esta lista de palabras se llama “gramática”.
2. El SRH carga el audio del altavoz. Este audio se representa como una forma de onda, esencialmente la representación matemática del sonido.
3. El SRH compara la forma de onda a sus propios modelos acústicos. Estas son las bases de datos que contienen información sobre las formas de onda de cada uno de los sonidos y es lo que permite al sistema reconocer del habla.
4. El SRH comparan las palabras en la gramática a los resultados obtenidos a partir de su búsqueda de modelos acústicos.
5. Por último, determina qué palabras en la gramática de audio concuerda y devuelve un resultado.

## Instalando LumenVox en Elastix

Dividiremos el proceso de instalación en algunas etapas.

### Descargar e instalar LumenVox

1. El primer paso en el proceso de instalación es bajar la llave GPG desde:  
<http://www.lumenvox.com/packages/FC6/i386/RPM-GPG-KEY-LumenVox.gpg>
2. Crear el archivo llamado LumenVox.repo en /etc/yum.repos.d/
3. Editar el archivo e ingresar la siguiente información:

```
#####  
[LumenVox]  
name=LumenVox Products $basearch  
baseurl=http://www.LumenVox.com/packages/EL5/i386/  
#baseurl=http://www.LumenVox.com/packages/EL4/i386/  
enabled=1  
gpgcheck=0  
#####
```

4. Grabar y cerrar el archivo

5. Desde la línea de comandos tipear “yum install LumenVoxLicenseServer”

6. Luego tipear “yum install LumenVoxSRE”

### Instalar la Licencia

1. Lo primero que hay que hacer es generar un archivo llamado Info.bts que guarda información acerca de su sistema.

Para realizar esto es necesario ejecutar el siguiente comando:

```
/usr/bin/lv_license_manager -g path donde path es la ruta completa del archivo server ID. Por ejemplo, si deseo crear el archivo server ID en el directorio root/LumenVox, debería tipear: /usr/bin/lv_license_manager -g /root/LumenVox/Info.bts
```

2. Cargar el archivo Info.bts y bajar la licencia

Para esto es necesario ir a <http://www.lumenvox.com/customers/>. Ingresar con el usuario y clave respectivo.

Luego ir al Link “View Licenses” y veremos un listado de las licencias disponibles inicialmente en estado “Upload License”. Hay que hacer clic en “Upload License” y cargar el archivo Info.bts que generamos.

Una vez hecho esto el estado de licencia cambiará a “Download License”. Hacer clic allí y obtener la licencia que será un archivo con nombre LicenseXXXX.bts donde XXXX son cuatro dígitos.

3. Instalar la licencia.

Para esto hay que tipear el comando **/usr/bin/lv\_license\_manager -m Licensefile** donde *Licensefile* es el archivo de la licencia que nos descargamos en el paso anterior.

### Verificar que la instalacion fue exitosa



1. Para verificar que la instalación fue exitosa es necesario correr el comando `ps ax|grep -i "LV"`, la salida de este comando nos debe mostrar si los procesos `lv_license_server` y `lv_sre_server` está ejecutándose tal como se muestra a continuación:

```
[root@elastix yum.repos.d]# ps ax|grep -i "LV"
2023 ?          Ssl      0:07 lv_license_server -console
2118 ?          Ssl      0:35 lv_sre_server -console
13956 pts/5      S+       0:00 grep -i lv
[root@elastix yum.repos.d]#
```

### Descargar e instalar el asterisk/lumenvox connection

1. Ir a <http://www.lumenvox.com/customers/> e ingresar con el usuario y clave respectivo.
2. Hacer click en “Download”.
3. Descargar el archivo etiquetado como Asterisk 1.4 Connector.
4. Extraer el archivo tar ejecutando `tar -xzf` y el nombre del archivo descargado.
5. Esto crea un nuevo directorio, entrar en ese directorio y copiar el archivo llamado `res_speech_lumenvox.so` en `/usr/lib/asterisk/modules/`.
6. Copie el archivo llamado `lumenvox.conf` en `/etc/asterisk/`
7. Modificar `/etc/asterisk/modules.conf` y asegurarse de que la línea de `preload => res_speech.so` esté allí y añadir la línea de `load => res_speech_lumenvox.so`

## **Probando LumenVox en Asterisk**

1. Editar el archivo `extensions_custom.conf` y adicionar el siguiente contexto:

```
[lumenvox-test]
exten => 1234,1,Answer
exten => 1234,n,Wait(1)
exten => 1234,n,SpeechCreate
exten => 1234,n,SpeechActivateGrammar(llamar_extension)
exten => 1234,n,Set(TIMEOUT(response)=8)
exten => 1234,n,Background(custom/ivrventas3)
exten => 1234,n,SpeechBackground(beep)
exten => 1234,n,Verbose(1|Result was ${SPEECH_TEXT(0)})
exten => 1234,n,Macro(user-callerid,)
exten => 1234,n,Dial(SIP/${SPEECH_TEXT(0)})
exten => 1234,n,Hangup()
```

2. Adicionar la línea include => lumenvox-test debajo del contexto from-internal-custom.
3. Conectarse a Asterisk con asterisk -r
4. Típear el comando reload
5. Probar marcando 1234

# 22

## Desempeño y dimensionamiento

*Yo no hablo de venganzas ni perdones, el olvido es la única venganza y  
el único perdón*

-- Jorge Luis Borges

### 22.1 Desempeño de Elastix

El desempeño de un servidor Elastix es un asunto muy complejo de analizar. En muchos casos interesa conocer los resultados y recomendaciones principalmente desde el punto de vista de una PBX, es decir de desempeño de Asterisk, puesto que es uno de los componentes más populares.

En este capítulo no haremos recomendaciones de hardware a utilizar pues esto haría que la presente obra quedara desactualizada en muy corto tiempo ya que el hardware evoluciona rápidamente. Más bien explicaremos los factores que afectan el desempeño en un servidor Elastix a nivel general.

#### Factores que afectan el desempeño de Elastix

Hay muchos factores que pueden afectar el desempeño de Elastix de alguna manera. Aquí explicaremos los más comunes.

### Número de llamadas concurrentes

Sin duda este es un parámetro muy importante, el número de llamadas concurrentes afecta directamente el consumo de CPU del servidor.

El siguiente es un gráfico de pruebas realizadas (sin *transcoding*) en un computador con un CPU de 2.2 GHz con 512 MB de RAM. Vemos que se han superado las 200 llamadas concurrentes con menos del 50% de CPU. En el computador se usó Elastix 1.3.

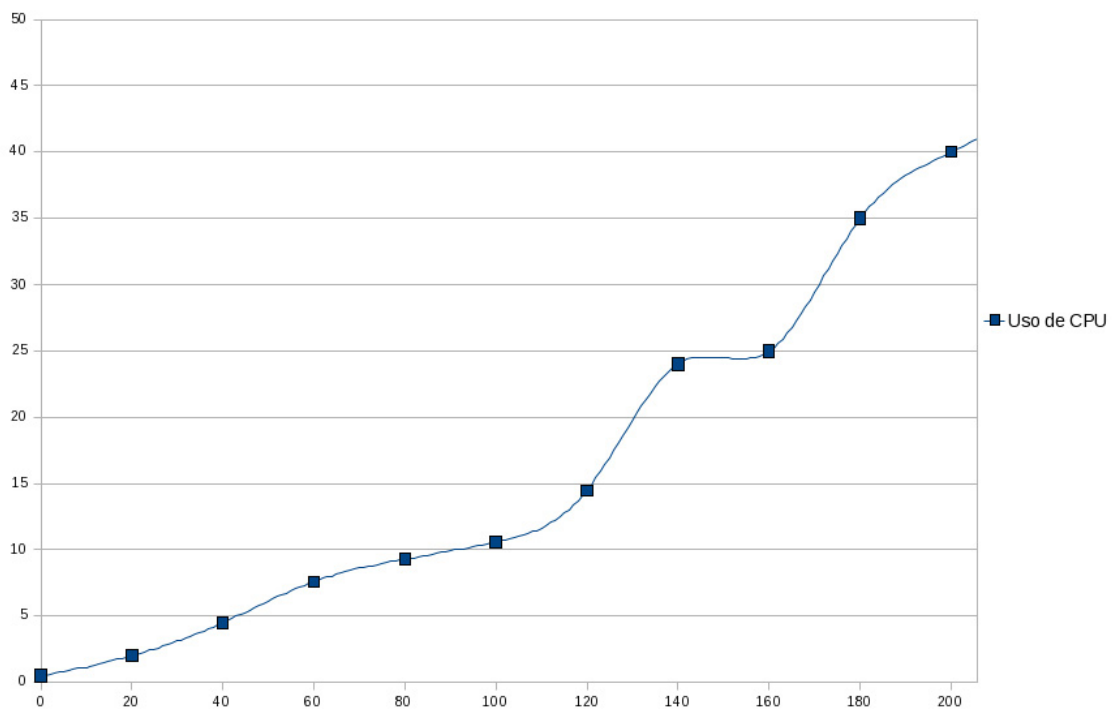


Gráfico CPU vs. Llamadas concurrentes

### Transcoding

El Transcoding se da cuando por un lado la llamada llega con un *codec* determinado y sucede que la llamada tiene que ir hacia un punto donde se utiliza otro *codec* distinto. Por lo tanto, Asterisk que está en el medio tiene que hacer las veces de traductor para que los dos puntos se entiendan. Esto es lo que se denomina *transcoding*.

El *transcoding* requiere de muchos cálculos matemáticos pues se está decodificando y codificando información con algoritmos complejos y por lo tanto esta operación requiere mucho CPU.

Una llamada con *transcoding* puede tomar 4, 5, 10 o incluso más veces más CPU que una llamada regular por lo que en escenarios con *transcoding* el desempeño se reduce significativamente. Por lo tanto, siempre que se pueda se debe evitar usar *codecs* distintos.

### Openfire

Openfire es un software escrito en Java que hace un muy buen trabajo como un servidor de Mensajería Instantánea. Sin embargo también es un gran consumidor de memoria. Por lo tanto si no es realmente necesario y el servidor está limitado de recursos es mejor dejarlo apagado.

Por omisión Openfire inicia apagado en Elastix.

### Flash Operator Panel

En instalaciones grandes, cuando existe mucha actividad telefónica el FOP puede comenzar a consumir recursos fuera de los niveles deseados. Esto es porque el FOP se comunica con Asterisk a través de ARI y por lo tanto se encuentra escuchando toda la actividad telefónica a través del puerto de red que levanta Asterisk para este fin. Cuanto más actividad, más información tiene que procesar el FOP.

En instalaciones grandes, en caso de que el uso del FOP no sea requerido es mejor apagarlo.

## **Herramientas**

Hay algunas herramientas que nos permitirán analizar el rendimiento de nuestro servidor Elastix.

### top

Para hacer un análisis breve de los servicios que nos están consumiendo recursos basta con ejecutar un comando `top` y ordenar su salida por consumo de memoria y también por consumo de CPU. Para hacer lo primero debemos topear la letra M (Mayúscula) y para lo segundo la letra P.

En algunas ocasiones podemos observar el proceso `java` liderando las estadísticas de consumo. Normalmente esto tiene que ver con Openfire debido a que se encuentra implementado sobre java.

sipp

Sipp es una herramienta de código abierto que sirve para generar tráfico SIP. En otras palabras nos servirá para generar llamadas SIP hacia nuestro servidor Elastix mientras tomamos muestras para entender cómo reaccionan los recursos del sistema (CPU, memoria) en escenarios de altos volúmenes de llamadas.

## 22.2 Dimensionamiento telefónico

### Determinando el número de líneas telefónicas necesarias

Cuando a un ingeniero le encomiendan la tarea de planificar un nuevo proyecto telefónico casi siempre el primer parámetro que necesita calcular es cuántas líneas se necesitarán para satisfacer la demanda prevista?

En instalaciones grandes esto es importante puesto que no se quiere tener muchas más líneas de las necesarias pues esto podría significar un desperdicio de recursos y por otro lado no se quiere pedir tan pocas que quedemos cortos y los usuarios consigan tono de ocupado muy seguido.

El cálculo por tanto debe basarse en criterios técnicos y no ser tomado a la ligera.

En este apartado estudiaremos una aproximación estadística llamada modelo Erlang para calcular cuántas líneas telefónicas necesitaremos en una instalación determinada para satisfacer una demanda prevista con cierta calidad.

### Agner Krarup Erlang

Erlang fue un ingeniero y matemático danés quien desarrolló a principios del siglo 20 las técnicas de Ingeniería de Tráfico y Teoría de Colas. No estudiaremos estos conceptos aquí en detalle pero los exploraremos vagamente para darle al lector cierto conocimiento básico y práctico de estas técnicas sin llegar al rigor matemático que ellas demandan. Estas importantes técnicas nos permitirán dimensionar cuántas líneas telefónicas necesitaremos en una instalación.

### La unidad Erlang

Antes de ir a las fórmulas es necesario introducir un nuevo término llamado la unidad Erlang. Un Erlang es una unidad adimensional comúnmente utilizada para medir el tráfico telefónico por hora.

También se puede decir que un Erlang representa la utilización continua de un circuito. Es decir que 1 Erlang equivaldría a la utilización de un circuito continuamente durante una hora en un lapso de una hora de tiempo o lo que es lo mismo que el circuito estuvo siempre en uso. Así mismo 0.5 Erlangs quiere decir que en el lapso de una hora el circuito se utilizó solo 30 minutos.

Para calcular el tráfico en Erlangs se puede proceder de la siguiente manera:

1. **Encontrar el tráfico total en horas:** Es decir, sumar el tiempo de todas las llamadas durante un lapso de tiempo cualquiera y convertir este valor a horas
2. **Encontrar el tráfico por unidad de tiempo (en horas):** Es decir, dividir el total anterior para el lapso de tiempo en horas

Algo confundido? Veamos un ejemplo para clarificar un poco más las cosas:

Supongamos que en una empresa cualquiera entre las 9H00 y 12H00 de un día determinado se realizaron 300 llamadas con un promedio de 6 minutos por llamada y se quiere medir el tráfico en Erlangs.

1. **Tráfico Total** = Tráfico en 3 horas =  $300 \times 6 = 1800$  minutos =  $1800 \text{ minutos} / 60 \text{ minutos} = 30$  horas (quiere decir que se hablaron 30 horas en ese lapso de tiempo)
2. **Tráfico por hora** = Tráfico Total / Lapso de Tiempo =  $30 \text{ horas} / 3 \text{ horas} = 10$  Erlangs

Hemos obtenido una medida de tráfico de **10 Erlangs**.

## Modelos Erlang

Ahora sí vamos a lo que nos interesa, o sea el cálculo del dimensionamiento telefónico.

En esencia existen dos modelos en las teorías de Erlang que se utilizan mucho en telefonía y son el modelo B y el modelo C. En la presente obra estudiaremos el modelo B pero recomendamos al lector recabar información del modelo C. El modelo C se utiliza para calcular probabilidades en llamadas que entran a colas así que si el lector está pensando en administrar un *call center* sería útil que indagara en el modelo Erlang C.

### Erlang B

El modelo Erlang B es el más común y es el que usaremos en la mayoría de escenarios pues es el que se usa para determinar el número de líneas necesario. Básicamente necesitamos alimentar a la fórmula con un par de parámetros y podemos obtener el número de líneas que necesitamos. Estos parámetros son: el tráfico pico (el tráfico en la hora de mayor actividad) y el Grado de Servicio (o GOS por sus siglas en inglés).

***Nota:** La fórmula Erlang B asume que una llamada bloqueada es realmente bloqueada, es decir que no entra a una cola. O sea que si todas las líneas se encuentran ocupadas el llamante obtendrá tono de ocupado en lugar de ingresar a una cola. Esto es verdad para la mayoría de PBXs.*

El **Tráfico pico** se obtiene sumando todo el tráfico en la hora de más actividad telefónica. Es útil tomar muestras no solo de un día sino de algunos días o semanas para que la información sea más exacta.

El **Grado de Servicio** representa la probabilidad de que una llamada sea rechazada. Su valor va de 0 a 1. Donde 1 significa el 100% de rechazo de llamadas. Algunos criterios comunes que he observado en la práctica son.

GoS	Percepción del servicio por parte del llamante
De 0 a 0.02	De excelente a muy bueno, casi no se rechazan llamadas. El valor 0 no es posible.
De 0.03 a 0.06	De normal a aceptable. Un valor de 0.03 a 0.04 es el más comúnmente usado.
De 0.07 a 0.10	Malo
De 0.10 en adelante	Pésimo, Terrible. Quiere decir que el 10% de llamadas obtendrán ocupado.

Los valores anteriores de ninguna manera constituyen una regla sino más bien una guía aproximada basada en la mayoría de los casos. El Grado de Servicio va en función de la importancia de perder una llamada y esto muchas veces va en función del tipo de negocio. Por ejemplo, la importancia de perder una llamada es diferente en una clínica que en una compañía regular.

También podría depender del departamento o unidad de negocios de una empresa. En general, el cliente, que conoce muy bien su negocio será el termómetro que nos permitirá determinar este parámetro.

A continuación la fórmula propuesta por el modelo Erlang B.



$$GoS = \frac{\frac{E^N}{N!}}{\sum_{i=0}^N \frac{E^i}{i!}}$$

Donde,

$E$  es el valor del tráfico pico (en Erlangs)

$N$  es el número de líneas telefónicas

Como vemos no es algo precisamente sencillo calcular el valor de GoS y peor aún despejar  $N$  que es lo que realmente buscamos para calcular el número de líneas necesarias. Pero no hay por qué preocuparnos porque actualmente hay muchas herramientas que nos permiten calcular el valor de  $N$  de manera sencilla, desde hojas de cálculo hasta calculadoras gratuitas online. La idea de presentar la fórmula aquí es para que el lector sepa los fundamentos del cálculo de la misma lo cual siempre es útil.

Así que bueno, ya que sabemos los conceptos básicos al menos sabremos lo que hacen internamente estas herramientas que vamos a usar. Veamos algunas de ellas disponibles de manera gratuita a través del Web.

- <http://www.erlang.com/calculator/erlb/>
- <http://personal.telefonica.terra.es/web/vr/erlang/>
- <http://www.cs.usyd.edu.au/~dcorbett/erlang.cgi>

## 22.3 Cuando un sólo servidor no alcanza

Muchas veces una instalación es tan grande que rebasa la capacidad de un solo equipo. Para estos casos también existe una solución que nos permite dividir un Elastix en uno o más servidores. Es lo que se llama balanceo de carga.

Existen dos esquemas que se usan en la práctica.

### Distribuir el trabajo en equipos especializados

Este podría ser un primer acercamiento para liberar de carga al servidor Elastix. Este esquema funciona siempre y cuando la carga no sea excesiva puesto que si lo es entonces vamos a tener que irremediablemente dividir el plan de marcado en otros equipos.

La idea consiste en migrar ciertos servicios a otros servidores. Algunos de los servicios que se pueden migrar son los siguientes.

- **MySQL:** Esto incluye los CDRs ya que estos se almacenan también en MySQL
- **Antispam:** Se puede levantar el servidor antispam en otro equipo y dejar el servidor de correo en el servidor Elastix principal si el volumen de emails es tolerable
- **Postfix:** Si el volumen de correos electrónicos es elevado incluso se puede mover Postfix
- **Openfire:** Incluso con Openfire en otro equipo es posible integrarlo con Asterisk con el *plugin* Asterisk-IM

La migración de estos servicios a otros equipos requiere trabajo y entendimiento de los componentes utilizados. Por ejemplo, para migrar el *antispam* deberemos hacer cambios al archivo `master.cf` que es el archivo de configuración de Postfix y migrar MySQL requerirá incluso modificar algunos archivos PHP usados por la interfase Web de Elastix. En caso de que se decida por este esquema asegúrese de entender bien el funcionamiento de los componentes o servicios involucrados.

## Distribuir el plan de marcado

En ocasiones, cuando la instalación es bastante grande, es deseable dividir el plan de marcado en más de un servidor. Para lograr esto nos podemos valer del protocolo DUNDi.

# 23

## Seguridad

*Las revoluciones se producen, generalmente, en los  
callejones sin salida*

-- Bertolt Brecht

La disponibilidad de Elastix así como la privacidad de la información que maneja como llamadas, *emails*, faxes, etc, son aspectos cruciales para su correcto funcionamiento. Estos aspectos pueden ser afectados por personajes inescrupulosos con fines contrarios a los nuestros.

Por estas razones la seguridad de este tipo de equipos se vuelve un aspecto fundamental que debemos estudiar en detalle. En este capítulo revisaremos los tópicos más importantes.

### 23.1 Algo de historia

Allá por los años 60s unas cuantas personas, entre ellas unos famosos *hackers*, se dieron cuenta que si generaban una señal audible a la frecuencia exacta de 2,600 Hertz producían que las llamadas telefónicas se colgaran. Se cuenta que uno de ellos, conocido bajo el pseudónimo de Joybubbles, descubrió el hecho a la edad de siete años mientras silbaba, y era capaz de reproducir esta frecuencia a su antojo solo silbando con la boca.

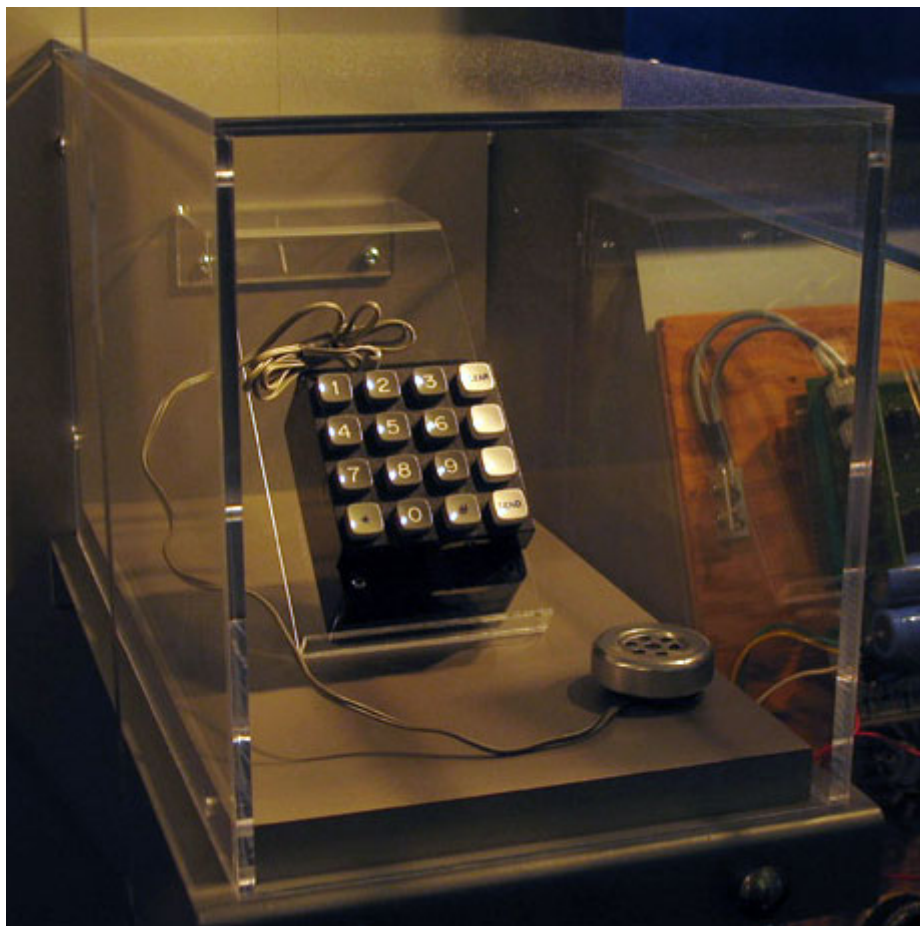
Años después se conoció que esta frecuencia era parte de la señalización que usaba Bell para “resetear” las llamadas telefónicas. Algunos hackers se pusieron a estudiar y entender la señalización telefónica y no pasó mucho tiempo antes de que alguien se diera cuenta que podía utilizar este hecho para realizar llamadas de larga distancia de manera gratuita lo cual era como haber descubierto oro pues en ese entonces las llamadas de larga distancia eran mucho más caras que en la actualidad.

Una de estas personas fue conocida como “John Capitán Crunch” pues descubrió que un silbato que venía como obsequio dentro de una caja de cereal de una conocida marca llamada *Captain Crunch* producía coincidentalmente la dichosa frecuencia de 2,600 Hz.

El truco consistía en soplar el silbato para resetear la línea y luego reproducir silbidos cortos para marcar el número telefónico. Así un silbido corto significaba uno, dos silbidos el número dos y así. De repente muchas personas conocían como realizar este fraude telefónico y no tuvo que pasar mucho para que algunos tecnificaran este principio y fabricaran cajas electrónicas que producían las señales audibles con solo marcar un teclado.

A estas cajas electrónicas se las denominó Bluebox en relación al color de la primera caja encontrada.

El hecho cobró tanto vuelo que en algún momento se comenta que Steve Wozniak y Steve Jobs, quienes después fundaran Apple Computers, se dieron a la tarea de fabricar Blueboxes. Vaya detalle! En la siguiente figura se muestra el modelo de Wozniak, el cual se encuentra en exposición en el museo de ciencias de la computación en USA.



Bluebox fabricado por Steve Wozniak expuesto en el Computer History Museum

Incluso hay una leyenda que cuenta que Wozniak en una ocasión llamó al Vaticano haciéndose pasar por Henry Kissinger y pidió hablar con el Papa.

Sucedió en verdad o no? Habría que preguntarle a Wozniak. En todo caso así fue como comenzaron los primeros trucos para burlar los sistemas telefónicos. No es de extrañarse que en la actualidad existan miles de maneras de burlar un sistema de comunicaciones y por lo tanto debemos estar muy entrenados para cuidarnos de que esto no suceda.

## 23.2 Seguridad física

La seguridad física tiene que ver con aspectos tangibles como por ejemplo si alguien desconecta el cable de poder del servidor o alguien derrama agua sobre el equipo, deliberadamente o no.

Muchos pensarán que esto es algo obvio a lo cual no hay que dedicarle un apartado especial, pero es importante recordar que muchos de los problemas de seguridad caen en este tipo de categoría ya que no se necesita ser un experto para vulnerar el equipo, basta con la mala intención o peor aún con el error humano involuntario.

## **Limitación del acceso al servidor**

Para evitar en gran medida los problemas de seguridad física es muy importante limitar el acceso físico al servidor Elastix. Esto se logra principalmente aislando el servidor en un ambiente controlado y limitando su acceso a unas poquísimas personas. Normalmente esto se logra mediante llave o un sistema de autenticación electrónico con acceso a una o dos personas.

Es muy importante complementar esto con un procedimiento riguroso de uso del acceso pues de nada sirve que la llave la tenga una sola persona si esta persona deja la llave siempre olvidada en el comedor de la empresa.

En algunos casos también puede ser útil llevar una bitácora de accesos y los tiempos de mantenimiento para de esta manera llevar un mejor control y esforzar los procedimientos. Una manera automática de hacer esto es utilizar sistemas electrónicos que cuentan el tiempo desde la autenticación al ingreso hasta la salida.

## **No usar el usuario root**

El uso habitual del usuario root puede constituir un problema potencial, aparte de los problemas de seguridad nos hace más proclives a que un error humano termine dañando nuestro computador o al menos el software instalado, inclusive mucha información puede quedar irrecuperable.

## **NO dejar sesiones de consola abiertas**

Es muy común ver que hasta los mismos administradores de sistemas suelen dejar las sesiones de consola abiertas incluso durante días. En ocasiones hasta con información crítica visible en la pantalla del computador (por ejemplo el comando `top` ejecutándose) o con una consola del CLI abierta. Esto es análogo a dejar las puertas del automóvil abiertas de par en par.

Por más confiado que uno se encuentre en principio nunca debe dejar una consola abierta si se va a alejar del servidor peor como “root” porque le estamos dando total acceso a cualquiera que se siente frente al computador.

Si se nos hace muy incómodo cerrar todo lo que estamos haciendo y salir de la consola solo porque vamos a una pequeña reunión de 10 minutos entonces podemos hacer uso de la aplicación `screen` que nos permite cerrar la consola pero conservar nuestra sesión activa. Así que no hay excusa señores!

Para instalar screen basta con ejecutar el comando yum.

```
# yum install screen
```

## Bloquear el reinicio

Algunos BIOS nos permiten configurar una contraseña de arranque. En muchas ocasiones esto puede frenar intentos de *hackers* que podrían querer reiniciar el computador para usar nuevos kernels, entre otras cosas.

***Nota:** Esta medida puede ser útil en la mayoría de los casos, pero hay que tener en cuenta que esto también nos impedirá reiniciar el equipo remotamente a nosotros también. Claro está, a menos que tengamos un equipo KVM sobre IP que nos permita ingresar la contraseña del BIOS remotamente.*

## Cambiar las claves por omisión en Elastix

Es muy importante cambiar las claves por omisión en Elastix pues en caso de no hacerlo cualquiera puede acceder a nuestro equipo libremente y eso sin duda es altamente peligroso.

Algunas claves muy importantes de cambiar son las siguientes.

Sistema	Valor por omisión	Dónde cambiar
Administrador interfase Web Elastix	admin/palosanto	Interfase Web menú Sistema ->Usuarios
freePBX	admin/admin	Interfase Web de freePBX
FOP	admin/eLaStIx.2007	/etc/amportal.conf
MySQL	root/eLaStIx.2007	Comando mysqladmin
SugarCRM	admin/password	Interfase Web de SugarCRM
vTiger	admin/admin	Interfase Web de vTiger

## 23.3 Ataques y vulnerabilidades a nivel de red IP

### Denegación de servicio

En seguridad informática, un ataque de denegación de servicio, también llamado ataque DoS (de las siglas en inglés Denial of Service), es un ataque a un sistema de computadoras o red que causa que un servicio o recurso sea inaccesible a los usuarios legítimos. Normalmente provoca la pérdida de la conectividad de la red por el consumo del ancho de banda de la red de la víctima o sobrecarga de los recursos computacionales del sistema de la víctima.

Se genera mediante la saturación de los puertos con flujo de información, haciendo que el servidor se sobrecargue y no pueda seguir prestando servicios, por eso se le dice "denegación", pues hace que el servidor no de abasto a la cantidad de usuarios. Esta técnica es usada por los llamados crackers para dejar fuera de servicio a servidores objetivo.

El llamado DDoS (siglas en inglés de *Distributed Denial of Service*, denegación de servicio distribuida) es una ampliación del ataque DoS, se efectúa con la instalación de varios agentes remotos en muchas computadoras que pueden estar localizadas en diferentes puntos. El invasor consigue coordinar esos agentes para así, de forma masiva, amplificar el volumen del flood o saturación de información, pudiendo darse casos de un ataque de cientos o millares de computadoras dirigido a una máquina o red objetivo. Esta técnica se ha revelado como una de las más eficaces y sencillas a la hora de colapsar servidores, la tecnología distribuida ha ido sofisticándose hasta el punto de otorgar poder de causar daños serios a personas con escaso conocimiento técnico.

En ocasiones, esta herramienta ha sido utilizada como un notable método para comprobar la capacidad de tráfico que un ordenador puede soportar sin volverse inestable y perjudicar los servicios que desempeña. Un administrador de redes puede así conocer la capacidad real de cada máquina.

### Ataques de inundación SIP (*SIP Flooding*)

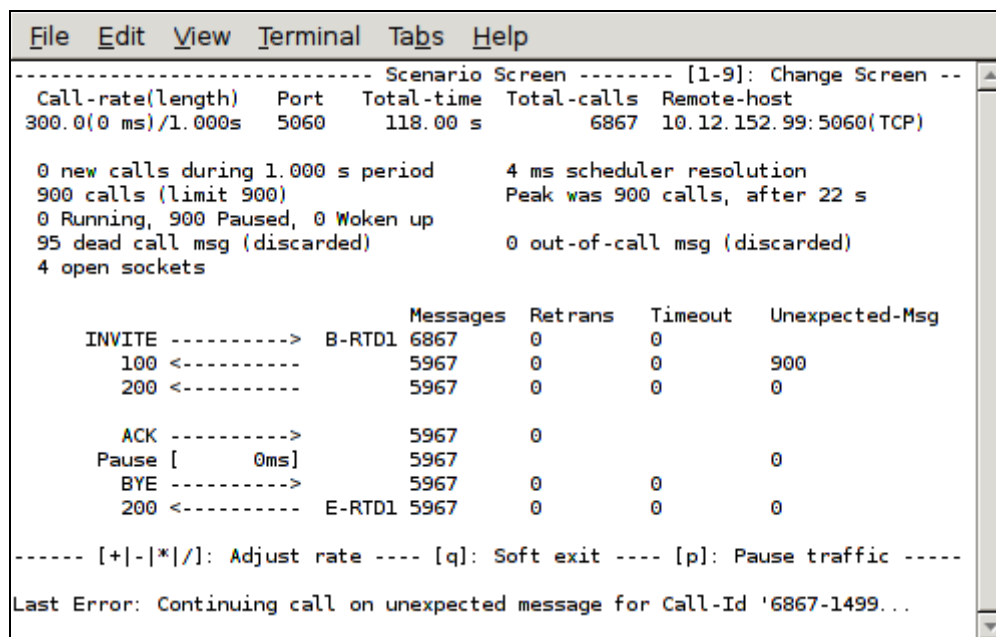
SIP Flooding o inundación de SIP es un tipo de ataque de denegación de servicio que consiste en bombardear un servidor SIP como Asterisk de paquetes INVITE. Como ya vimos antes estos paquetes sirven para iniciar una llamada y por lo tanto el servidor SIP, al recibir uno de estos paquetes, se quedará esperando respuesta del supuesto usuario. Mientras el servidor espera recibir una respuesta que nunca llegará muchas más falsas llamadas intentan hacer más conexiones.



Si un alto volumen de estas falsas llamadas ingresa al equipo pueden ocasionar una Denegación de Servicio.

Hay que tener mucho cuidado con esto pues generar un ataque de SIP *flooding* no es tan difícil. Existen herramientas que fueron creadas con buenas intenciones (ej. generadores de tráfico para pruebas o para hacer análisis de carga) que pueden ser utilizadas con fines equivocados. Una de estas herramientas es SIPP que se encuentra disponible gratuitamente desde el sitio <http://sipp.sourceforge.net> y que normalmente se usa para generar tráfico SIP para pruebas.

SIPP puede configurarse para generar miles de llamadas hacia un servidor SIP.



```
File Edit View Terminal Tabs Help
----- Scenario Screen ----- [1-9]: Change Screen --
Call-rate(length) Port Total-time Total-calls Remote-host
300.0(0 ms)/1.000s 5060 118.00 s 6867 10.12.152.99:5060(TCP)

0 new calls during 1.000 s period 4 ms scheduler resolution
900 calls (limit 900) Peak was 900 calls, after 22 s
0 Running, 900 Paused, 0 Woken up
95 dead call msg (discarded) 0 out-of-call msg (discarded)
4 open sockets

      Messages Retrans Timeout Unexpected-Msg
INVITE -----> B-RTD1 6867 0 0
100 <----- 5967 0 0 900
200 <----- 5967 0 0 0

      ACK -----> 5967 0
Pause [ 0ms] 5967 0
      BYE -----> 5967 0 0
200 <----- E-RTD1 5967 0 0

----- [+|-|*|/]: Adjust rate ---- [q]: Soft exit ---- [p]: Pause traffic -----
Last Error: Continuing call on unexpected message for Call-Id '6867-1499...
```

Aplicación SIPP en acción

## Sniffing

*Sniffing* es una práctica que consiste en escuchar los paquetes que circulan por la red. En algunos casos (por ejemplo si usamos un Hub en lugar de un *switch*) un atacante puede incluso escuchar el tráfico entre otros equipos ajenos al equipo donde se tiene acceso.

Para poder capturar el tráfico se utiliza una aplicación llamada *Sniffer* y existen algunas alternativas disponibles en Internet por lo que esta práctica se ha convertido en una tarea no tan difícil de llevar a cabo por *hackers*.

Luego de capturar el tráfico el atacando lo analizará dependiendo de lo que quiera hacer. Por ejemplo si el tráfico corresponde a una llamada telefónica SIP es probable que el atacante analice la información de señalización para recabar información de

Caller IDs principalmente. También es posible que decodifique las tramas RTP y con esto logre inclusive reproducir la llamada y escucharla.

***Nota:** Una manera de verificar si existe un sniffer ejecutándose en nuestro servidor Elastix es verificar si la tarjeta de red se encuentra en modo “promiscuo”. Esto se puede hacer de manera sencilla analizando la salida del comando ifconfig.*

### Escuchas de llamadas a nivel de VoIP

En algunos casos es posible escuchar tráfico VoIP que circula por la red con las herramientas adecuadas. Después de todo, los paquetes RTP se encuentran codificados con *codecs* conocidos y por lo tanto la decodificación de los mismos no es una tarea que complique la vida a un *hacker* experimentado. Además, los *codecs* de audio no fueron concebidos con el objetivo de encriptar la comunicación sino más bien de comprimir la voz y evitar errores de transmisión.

Si podemos capturar los paquetes RTP de una llamada podemos utilizar herramientas para convertir estos paquetes en audio que podamos escuchar.

Adicionalmente recordemos que el protocolo SIP es un protocolo de texto o sea que si logramos capturar paquetes SIP de la red podremos ver información como Caller IDs, comandos y respuestas SIP.

Existen aplicaciones especializadas que facilitan la vida de los *hackers* cuando se pretenden hacer escuchas de llamadas VoIP desde equipos intrusos. Citaremos las más conocidas según mi punto de vista.

- VoIPong
- Wireshark (con el plugin de análisis VoIP)
- Oreka

**VoIPong** rastrea la red en busca de llamadas VoIP y una vez detectadas permite guardar las llamadas en formato wav. VoIPong soporta g.711. El sitio oficial de VoIPong es <http://www.enderunix.org/voipong>

**Wireshark** (el antiguo proyecto Ethereal) ahora trae soporte para análisis de llamadas VoIP y nos permite grabar el *payload* (la voz) de los paquetes RTP en formato de audio crudo. Vale decir que Wireshark también constituye una herramienta de análisis poderosa y muy útil que no debe faltar en el maletín del Ingeniero de VoIP.

Finalmente **Oreka** es un proyecto algo más complejo pero que nos permite incluso decodificar *codecs* como GSM e iLBC. Oreka nos puede proveer incluso de una

interfase Web para administrar las llamadas. Para más información el sitio de Oreka se encuentra en <http://oreka.sourceforge.net/>

En todos estos casos estamos suponiendo que se quieren capturar llamadas externas al equipo donde se encuentran estas herramientas. Muchas veces esto es posible debido a que se utilizan HUBs en lugar de Switches para conectar los equipos de voz o a malas configuraciones del Switch.

## 23.4 Encriptando la VoIP

Para evitar escuchas o pichazos telefónicos a nivel de VoIP tenemos la alternativa de encriptar la comunicación. Existen algunas opciones que explicaremos a continuación.

### SRTP

El protocolo SRTP (*Secure RTP*) no se encuentra soportado oficialmente en Asterisk 1.4. Es decir la versión más reciente de Asterisk que trae Elastix. Sin embargo, el código de Asterisk puede ser parchado para añadirle este soporte. Adicionalmente existen planes de incorporar soporte SRTP en versiones futuras de Asterisk (parece que puede ser posible en la rama 1.6) por lo que en un futuro SRTP se convertirá en una alternativa viable y por lo tanto merece una explicación en este apartado.

#### Detalles del protocolo

El *Secure Real-time Transport Protocol* (o SRTP) define un perfil de RTP (*Real-time Transport Protocol*), con la intención de proporcionar cifrado, autenticación del mensaje e integridad, y protección contra reenvíos a los datos RTP en aplicaciones unicast y multicast. Fue desarrollado por un pequeño grupo del protocolo IP y expertos criptográficos de Cisco y Ericsson incluyendo a David Oran, David McGrew, Mark Baugher, Mats Naslund, Elisabetta Carrara, Karl Norman, y Rolf Blom. Fue publicado por primera vez por el IETF en marzo de 2004 como el RFC 3711.

Dado que RTP está muy relacionado con RTCP (RTP control protocol), que puede ser usado para controlar una sesión RTP, SRTP también tiene un protocolo hermano llamado Secure RTCP (or SRTCP). SRTCP proporciona las mismas características relacionadas con la seguridad a RTCP, al igual que hace SRTP con RTP.

El empleo de SRTP o SRTCP es opcional al empleo de RTP o RTCP; pero incluso utilizando SRTP/SRTCP, todas las características que estos protocolos proporcionan (tales como cifrado y autenticación) son opcionales y pueden ser habilitadas o deshabilitadas por separado. La única excepción a esto último es la autenticación de los mensajes, que es obligatoria cuando se está usando SRTCP.

## Activando SRTP en Asterisk

Si se desea activar soporte SRTP en Asterisk debemos tener en cuenta dos escenarios: comunicaciones entre dos servidores y comunicaciones entre servidor y endpoint.

En el último caso deberemos asegurarnos que el teléfono elegido también soporte SRTP ya que de nada nos servirá activar soporte SRTP en Asterisk si el teléfono no entiende el protocolo.

En fin, esta tarea no es trivial y demanda muchos conocimientos de desarrollador puesto que la compilación de Asterisk podría fallar en nuevas versiones de la rama 1.4 si el parche no se encuentra actualizado, por lo que habría que hacer correcciones manuales.

Para activar el soporte SRTP en Elastix podemos seguir los siguientes pasos:

- Obtener el SRPM de Asterisk de los repositorios de Elastix e instalarlo
- Obtener el código de libSRP e instalarlo. El proyecto se encuentra alojado en <http://srtp.sourceforge.net/download.html>
- Parchar Asterisk. Existen algunas guías útiles en estos enlaces:
  - o <http://www.voip-info.org/wiki/view/Asterisk+SRTP>
  - o <http://www.e164.org/wiki/AsteriskSRTP>
- Hacer las modificaciones pertinentes en el plan de marcado ya que se necesitan añadir ciertas variables antes de la llamada a la aplicación Dial. Ver enlaces de arriba.
- Compilar Asterisk vía el comando “rpmbuild”. Previamente incrementar la versión del RPM desde el archivo asterisk.spec
- Actualizar el rpm en Elastix

***Nota:** Este hilo en la herramienta de seguimiento de bugs de Digium muestra el estatus actual del soporte SRTP en Asterisk.*

## **VPNs**

Al momento esta es mi manera preferida de encriptar la comunicación IP, al menos hasta observar la evolución de SRTP sobre Asterisk.

Las VPNs son ampliamente usadas y existe un proyecto de código abierto a mi parecer muy bueno llamado OpenVPN que funciona sobre Linux y por ende sobre Elastix.

Una VPN (*Virtual Private Network*) o Red Privada Virtual es un concepto que nos permite simular una red privada sobre una red pública como el Internet. Para poner un ejemplo una Empresa X tiene oficinas en varias ciudades llamadas A, B y C, y estas ciudades se encuentran conectadas entre sí a través de Internet. Sucede que esta empresa desea que todos los computadores y equipos de red de todas estas ciudades se encuentren dentro del mismo segmento de red privada 192.168.100.X. De esta manera se podrán compartir impresoras entre ciudades pues VIRTUALMENTE estas impresoras se encontrarían en la misma ubicación o segmento de red. Adicionalmente las ciudades B y C podrán alcanzar el servidor Elastix que se encuentra en la ciudad A desde un IP interna (no pública).

Todo suena muy bonito pero enviar las comunicaciones internas o privadas a través de un medio público como el Internet suena algo riesgoso verdad?. Bueno, para que esto sea posible de manera segura existe un concepto llamado *tunneling* que nos permite que estas comunicaciones transiten por la red pública de manera encriptada de tal forma que se asocia esta tecnología con un túnel virtual ya que se encuentra aislado del exterior. Para establecer el túnel virtual se utilizan protocolos seguros.

En concreto se puede establecer una VPN para meter en dicha red todos los dispositivos de voz y de esta manera hacer que las comunicaciones de voz viajen seguras. Existen en la actualidad algunos modelos de teléfono que soportan incluso VPNs por lo que los podríamos conectar directamente con el servidor Elastix. Por ejemplo:

- Snom 370
- ATCOM 530

En caso de que usemos *softphones* podemos establecer una VPN entre el PC donde se encuentra el *softphone* y el servidor Elastix.

## Encriptación IAX

Esta es muy probablemente la manera más sencilla de encriptar la comunicación entre dos servidores Asterisk.

Cuando IAX fue concebido se pensó en estos problemas de violación a la privacidad y se añadió soporte nativo para encriptación de llamadas. Habilitar este soporte es sumamente sencillo pues consiste en añadir un parámetro en la configuración IAX.

```
; El siguiente parámetro habilita encriptación IAX.  
; También se puede usar el parámetro encryption=yes  
encryption=aes128
```

***Nota:** Pese a que este método activa encriptación IAX y por ende puede ser útil en comunicaciones entre 2 o más servidores Asterisk; no conozco aún de teléfonos que implementen esta característica. Si el lector conoce de alguno le agradecería que me lo haga saber para actualizar esta información.*

## 23.5 Auditorías de seguridad

Las auditorías de seguridad se suelen realizar cuando se sospecha que un servidor ha sido hacheado. Sin embargo, lo recomendable es que sea una práctica constante que se realice periódicamente para detectar a tiempo problemas de seguridad que podrían poner en riesgo nuestros equipos.

Existen algunas prácticas comunes indispensables en una auditoría de seguridad. Explicaremos algunas de ellas que considero importantes.

### Escaneo de puertos de red

En muchas ocasiones un equipo atacado es instalado con software que ejecuta servicios de red. Es útil verificar esto y se puede realizar fácilmente desde el mismo equipo atacado mediante el comando `netstat`. Sin embargo en algunos casos este comando no nos ayudará mucho pues el atacante puede haber modificado el comando para que nos muestre lo que a él le interesa que pensemos... o sea “que el equipo está limpio”. Por lo tanto es necesario también inspeccionar el equipo desde fuera y para esto podemos usar el comando `nmap` ejecutado desde un equipo remoto.

Un puerto sospechoso en equipos vulnerados suele ser el 6666 o 6667 que son puertos relacionados con servidores IRC pues sucede que una costumbre de los *hackers* es instalar un servidor IRC para, a través de una simple aplicación de chat IRC, controlar el servidor mediante comandos.

Veamos cómo luce la salida de un `nmap` en un equipo Elastix sano desde la red interna y no protegido por firewall.

```
[root@remote ~]# nmap elastix34.example.com

Starting nmap 3.81 ( http://www.insecure.org/nmap/ ) at 2008-11-16
23:18 ECT
Interesting ports on elastix34.example.com:
(The 1647 ports scanned but not shown below are in state: closed)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
25/tcp    open  smtp
80/tcp    open  http
110/tcp   open  pop3
143/tcp   open  imap
443/tcp   open  https
993/tcp   open  imaps
995/tcp   open  pop3s
2000/tcp  open  callbook
3306/tcp  open  mysql
4559/tcp  open  hylafax
7070/tcp  open  realserver
9090/tcp  open  zeus-admin

Nmap finished: 1 IP address (1 host up) scanned in 4.556 seconds
```

### Revisión de logs del sistema

Revisar los *logs* del sistema siempre es útil aunque en muchas ocasiones el *hacker* tratará de borrarlos para que no veamos su actividad. En todo caso ver un *log* borrado también puede ser útil pues puede confirmar nuestras sospechas de que el equipo ha sido vulnerado.

Algunos logs que son interesantes de revisar:

- **/var/log/messages** – Por información general como nuevos usuarios creados por el hacker o mensajes de ERROR inusuales
- **/var/log/maillog** – Por si el hacker intentó ingresar por el servidor de correos
- **/var/log/httpd/error\_log\*** - Por si el hacker intentó ingresar por el servidor Apache
- La salida del comando **dmesg**
- El historial de comandos ejecutados. Para el usuario actual se puede ejecutar el comando **history**

## Revisión de rootkits

Muchas veces es difícil darnos cuenta si un *hacker* ha ingresado al equipo pues estas personas suelen instalar un conjunto de herramientas solo con el fin de hacerlos más invisibles a nuestras inspecciones. Entre esas herramientas constan versiones alteradas de comandos del sistema que muestran salidas modificadas para que los procesos ejecutados por el *hacker* queden ocultos a nuestra vista.

También suelen instalar programas para inspeccionar nuestro equipo o para controlarlo (como los servidores IRC de los que hablamos antes).

En general a este conjunto de utilerías y herramientas que el *hacker* instala en nuestro servidor se le denomina *rootkit*.

Por suerte existe un antídoto para este mal y se llaman buscadores de *rootkits*.

Un buscador de *rootkits* es un paquete de software que ejecuta una serie de revisiones por *rootkits* conocidos o patrones típicos de incursiones de *hackers*, permitiéndonos descubrir el tipo de ataque realizado y en ocasiones hasta la manera de eliminar al *hacker* del sistema.

Normalmente un buscador de *rootkit* contiene su propio set de comandos de sistema para no utilizar los comandos disponibles en el servidor porque estos pueden estar adulterados.

A continuación un par de buscadores de rootkits conocidos y de libre uso:

- Chkrootkit (<http://www.chkrootkit.org>)
- Rkhunter (<http://www.rootkit.nl>)

## Auditoría avanzada con Nessus

Nessus es un *scanner* que rastrea un equipo en busca de vulnerabilidades. Existen listas públicas de vulnerabilidades mantenidas por algunas organizaciones y Nessus se basa en ellas.

Primero trata de determinar a través de la red los servicios que ejecuta el equipo examinado abriendo conexiones a sus puertos (de manera similar a nmap) y si encuentra uno trata de determinar el software que se encuentra corriendo detrás y su versión. Para esto usa una base de datos interna de lo que llama “huellas” puesto que la manera como un software responde por la red puede darnos información de la versión del mismo. Por ejemplo tratemos de iniciar una sesión al puerto 25 de un servidor Elastix.



```
[root@apps ~]# telnet elastix.example.com 25
Trying elastix.example.com...
Connected to elastix.example.com.
Escape character is '^]'.
220 elastix.example.com ESMTP Postfix
```

Como podemos ver solo con el simple hecho de abrir una conexión ya nos hemos enterado de que Postfix se encuentra allí. Sin embargo, esto es algo evidente. Nessus conoce hechos menos obvios que le permite conocer en muchos de los casos la versión que estamos ejecutando.

Finalmente, conociendo el software y su versión, Nessus tratará de buscar en su base de datos de vulnerabilidades si existen vulnerabilidades para estas versiones de software. En caso de encontrarlas nos alertará.

Este proceso toma su tiempo pero al final Nessus nos preparará de manera automática un reporte de los paquetes de software que presentan problemas y que deberíamos actualizar. Tener a Nessus trabajando para nosotros es como haber contratado un experto de seguridad en nuestra organización.

Para más información de Nessus el lector puede seguir este link <http://www.nessus.org>

## 23.6 Algunas soluciones a los problemas de seguridad

### Apagando los servicios que no vamos a usar

Elastix es una distro rica en funcionalidad. Para proveer toda esta funcionalidad muchos paquetes de software son empaquetados juntos con la distro.

Sin embargo, en muchos casos no necesitamos de todos ellos para lograr lo que queremos con Elastix. Es por esto que una de las primeras tareas luego de instalar Elastix consiste en apagar los servicios que no vamos a usar. Algunos servicios típicos que podríamos apagar podrían ser los siguientes.

Si no se va a usar servidor de Email:

- postfix
- cyrus-imapd

- spamassassin
- mailman
- saslauthd

Si no se va a usar mensajería instantánea:

- Openfire

Otros servicios:

- Vsftpd (Servidor FTP)
- Tftp (este servicio trabaja bajo xinetd y es necesario en caso de utilizar provisionamiento en algunos modelos de teléfonos como Polycom)

## Firewalls

Un *firewall* básicamente es un componente de software que nos permite establecer ciertas políticas de bloqueo a puertos de red que no queremos que sean visibles del otro lado, normalmente desde Internet. Bueno... realmente un *firewall* es mucho más que eso, pero lo vamos a mantener simple aquí.

Elastix trae incorporado un poderoso *firewall* llamado **iptables**. Iptables se ha vuelto el *firewall* de facto en la mayoría de distribuciones Linux ya que se encuentra incorporado en el kernel mismo.

En el presente apartado configuraremos un *firewall* básico residente en el mismo servidor Elastix. En realidad siempre es recomendable que el *firewall* sea un componente aparte del servidor Elastix, pero si no tenemos esa opción instalarlo en el propio Elastix es un excelente plan B.

### Ejecutando reglas de Iptables

No entraremos en detalles del uso del comando Iptables. Dejamos al lector la importante tarea de adquirir esos conocimientos adicionales. Lo que sí haremos es copiar algunas reglas básicas que se pueden usar para proteger nuestro servidor de comunicaciones. Recordar que el orden de las reglas es importante así como también ejecutar todas las reglas, en especial la última que es la regla que bloquea todos los demás puertos.

***Nota:** Si no hemos configurado un firewall a través de Iptables antes es recomendable que nos encontremos frente al servidor. Si lo hacemos remotamente y ejecutamos las reglas incorrectamente podríamos perder conexión con el equipo.*

En fin, vamos a mostrar las reglas. En los siguientes ejemplos se ha supuesto que la tarjeta de red pública es la eth0. Si esto no es así en el caso del lector se deberá modificar este parámetro.

Aceptando el tráfico SIP:

```
# iptables -A INPUT -p udp -m udp -i eth0 --dport 5060 -j ACCEPT
```

Aceptando el tráfico IAX2:

```
# iptables -A INPUT -p udp -m udp -i eth0 --dport 4569 -j ACCEPT
```

Aceptando el tráfico RTP (Suponiendo que no se ha alterado el archivo rtp.conf):

```
# iptables -A INPUT -p udp -m udp -i eth0 --dport 10000:20000 -j ACCEPT
```

Aceptando tráfico MGCP (solo aplicar si es que se va a usar. En la mayoría de los casos no es necesario):

```
# iptables -A INPUT -p udp -m udp -i eth0 --dport 2727 -j ACCEPT
```

Aceptando el tráfico de mensajería instantánea (si es que se va a acceder desde fuera):

```
# iptables -A INPUT -p tcp -i eth0 --dport 9090 -j ACCEPT
```

Aceptando el tráfico del servidor de correo y POP/IMAP:

```
# iptables -A INPUT -p tcp -i eth0 --dport 25 -j ACCEPT
# iptables -A INPUT -p tcp -i eth0 --dport 110 -j ACCEPT
# iptables -A INPUT -p tcp -i eth0 --dport 143 -j ACCEPT
```

Aceptando tráfico Web (HTTP y HTTPS) para poder visitar la interfase administrativa de Elastix:

```
# iptables -A INPUT -p tcp -i eth0 --dport 80 -j ACCEPT
# iptables -A INPUT -p tcp -i eth0 --dport 443 -j ACCEPT
```

Finalmente denegando el acceso a todo lo demás:

```
iptables -A INPUT -p all -i eth0 -j DROP
```

Ya habiendo visto el patrón de las reglas de Iptables dejamos al lector la tarea de predecir cómo abrir otros puertos como SSH (puerto 22), MySQL (puerto 3306), entre otros servicios que podrían ser útiles. Claro está, la premisa es que mientras menos puertos expuestos a los *hackers* mejor.

Al final es deseable verificar si las reglas se aplicaron bien. Para hacer esto podemos ejecutar el comando iptables con la opción `-L`.

```
[root@localhost ~]# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source                destination
ACCEPT     udp  --  anywhere              anywhere             udp dpt:sip
ACCEPT     udp  --  anywhere              anywhere             udp dpt:iax
ACCEPT     udp  --  anywhere              anywhere             udp dpts:ndmp:dnf
ACCEPT     tcp  --  anywhere              anywhere             tcp dpt:websm
ACCEPT     tcp  --  anywhere              anywhere             tcp dpt:smtp
ACCEPT     tcp  --  anywhere              anywhere             tcp dpt:pop3
ACCEPT     tcp  --  anywhere              anywhere             tcp dpt:imap
ACCEPT     tcp  --  anywhere              anywhere             tcp dpt:http
ACCEPT     tcp  --  anywhere              anywhere             tcp dpt:https
DROP       all  --  anywhere              anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
```

La salida que obtengamos debería verse algo similar a la anterior. Si vemos que algo anda mal podemos evacuar todas las reglas con el comando “`iptables -F`” y empezar de nuevo.

## IDSs e IPSs

Un tipo más avanzado de estrategia para bloquear ataques de *hackers* es lo que se llama IDS o *Intrusión Detection System*. Un concepto similar es el de IPS o *Intrusión Prevention System*. La diferencia entre los dos es que mientras el IDS solo detecta intrusos el segundo toma medidas para evitar que el ataque continúe.

Básicamente estas tecnologías tratan de ir más allá de un simple bloqueo de puertos como vimos en el apartado de *firewalls* y se dedican a observar la actividad de red en busca de patrones de ataques conocidos. Estos patrones han sido almacenados

previamente en lo que se conoce como bases de datos de patrones (fingerprint databases). Como los patrones cambian día a día se deben actualizar de cuando en cuando para que estas aplicaciones estén al tanto de patrones nuevos.

Un IDS nos notificará cuando encuentre un patrón de ataque conocido mientras que un IPS irá más allá y podría ejecutar automáticamente una regla de *firewall* específica para el ataque recibido con el objetivo de bloquear al posible intruso.

Estos sistemas eran normalmente utilizados por grandes empresas pues las pequeñas no podían costear su precio. Todo esto cambió con la aparición de una alternativa libre llamada Snort.

Snort funciona sobre Linux y puede ser instalado sobre Elastix para monitorear la red en busca de patrones de ataques. Snort puede detectar algunos tipos de trucos o intentos de ataques a nivel de VoIP tanto como a nivel de otros servicios o protocolos que podrían comprometer la seguridad del servidor.

Al lector interesado le recomendamos visitar <http://www.snort.org>

## Actualizaciones

Cuando una nueva versión de un paquete de software es lanzada al público se pone al análisis de personas que tratan de vulnerarla. Más aún si se trata de una aplicación que puede ser alcanzable a nivel de red como es el caso de Asterisk, Postfix, Cyrus, etc.

Cuando se descubre una de estas vulnerabilidades los desarrolladores sacan nuevas versiones y recomiendan la actualización del software. Por lo general los bugs que comprometen la seguridad del equipo se marcan como críticos y conllevan a que se disponga de la corrección de manera expedita.

Es por esta razón que es muy importante actualizar los servicios de red de cuando en cuando y de esta manera estar prevenido de posibles fallos de seguridad que puedan contener los paquetes de software. También es muy útil estar al tanto de las alertas de seguridad que publican los distintos fabricantes de software. Por ejemplo, Asterisk publica las suyas en <http://www.asterisk.org/security>

No entraremos en mucho detalle aquí de cómo proceder con las actualizaciones pero les recomendamos utilizar la interfase Web para actualización de paquetes que viene con Elastix ó en su defecto utilizar los comandos de YUM explicados en el capítulo 3 de la presente obra.

## Monitoreo

El monitoreo constante de un servidor Elastix puede marcar la diferencia en cuanto a seguridad. Es una acción preventiva que nos puede ahorrar un dolor de cabeza.

Existe una aplicación bastante útil para automatizar este proceso y su nombre es Nagios. Nagios es un proyecto de código abierto que puede configurarse para monitorear diversos parámetros como uso de ancho de banda, uso de memoria, carga de CPU, temperatura del CPU, uso de disco duro, carga de la cola de emails e inclusive, con la configuración adecuada, parámetros de Asterisk como número de llamadas concurrentes, etc.

De qué me sirve esto?

Bueno, muchas veces los ataques de *hackers* van acompañados del uso excesivo de recursos como por ejemplo el ancho de banda.

En ocasiones a los *hackers* no les interesa si el equipo es un servidor de VoIP, base de datos o cualquier otra cosa. Basta con que tenga un buen acceso Internet para que el fulano lo vea atractivo pues lo puede utilizar de plataforma para vulnerar a su vez a otros equipos. Recordemos que muchos ataques DoS se consiguen enviando grandes cantidades de información.

Por lo dicho, en muchos casos resulta útil monitorear algunos recursos del servidor y en caso de que el uso de estos recursos crezca de manera inusual se puede configurar Nagios para que nos envíe una alerta vía email y poder tomar medidas correctivas para evitar daños mayores.

## Usar un Switch en lugar de un HUB

A simple vista un HUB y un Switch son dispositivos bastante parecidos pero esa no es la realidad.

Un HUB es un dispositivo de red bastante sencillo cuya misión es repetir cada paquete de datos que llega por un puerto determinado en todos los demás puertos. Por esta razón muchas personas también lo llaman repetidor. Como es de esperar este comportamiento genera muchas colisiones por lo que adicionalmente un HUB es capaz de enviar una señal de colisión a sus puertos para indicarles que dejen de transmitir por un momento.

Un Switch en cambio es un dispositivo un poco más inteligente y cuando un equipo conectado en uno de sus puertos trata de enviar paquetes a otro, en lugar de repetir el paquete en todos los demás puertos crea una conexión únicamente entre los dos equipos involucrados en la comunicación. Esto parece algo simple pero es una gran ventaja pues mejora enormemente el rendimiento de la red pues ya no es necesario enviar información innecesaria. Evidentemente se consigue una reducción en el consumo de ancho de banda de la red.



Foto de HUB marca NETGEAR de 4 puertos

Sin embargo la ventaja de usar un Switch no solo radica en el ahorro de ancho de banda. Por su naturaleza un HUB envía información adicional no necesaria a todos los equipos de la red, o sea comunicaciones de otros equipos de red que no deberían caer en manos de otros equipos no involucrados en la conversación. Normalmente un equipo determinado desecha o filtra esa información innecesaria y allí acaba el problema PERO cuando los *hackers* entran en escena esa información que antes era indeseable se vuelve valiosa información en manos de estas personas ya que de allí pueden tratar de obtener textos de email, mensajes de IM e incluso pueden grabar llamadas telefónicas de terceros!

En resumen, la recomendación aquí es no utilizar HUBs en implantaciones de VoIP!





# 24

## ***Debugging Avanzado***

*Yo he preferido hablar de cosas imposibles, por que de lo posible se sabe demasiado*

-- Silvio Rodriguez

### **24.1 Introducción**

Muchas veces encontramos un *bug* o comportamiento poco usual en Asterisk (o cualquier otro paquete de software) y luego de tratar de rastrear el problema por mucho tiempo sin encontrar la respuesta caemos en la desazón.

En ese momento tal vez quisiéramos meternos dentro del software y entender qué está ocurriendo. La buena noticia es que en cierto modo eso es posible ya que existen herramientas de software que nos permiten lograr en parte ese cometido. En el presente capítulo estudiaremos estas herramientas llamadas depuradores.

Un depurador es un programa que nos permite examinar la ejecución de un programa a un nivel de detalle mayor que el habitual, permitiéndonos con esto encontrar errores que de otro modo resultarían difíciles de hallar. Inclusive nos brindan la posibilidad de examinar qué estuvo ocurriendo en el momento en que un programa se colgó.

En el presente capítulo presentaremos básicamente dos alternativas de herramientas de depuración como lo son `strace` y `gdb`. La primera es una herramienta ligera de uso relativamente fácil que puede ser usada en primera instancia mientras que la segunda es una herramienta mucho más completa cuyo uso requiere un poco más de estudio.

A pesar de que el objetivo primario de este capítulo es la depuración de Asterisk los conceptos aquí descritos pueden ser aplicados a cualquier programa.

## Qué hacer cuando Asterisk se comporta de mal?

Cuando ocurren problemas con un servidor de comunicaciones en producción podemos caer en la desesperación pues se trata de una aplicación crítica en una empresa. Trataremos de explicar aquí una serie de pasos para que los podamos seguir en esos momentos en los que la adrenalina corre por nuestras venas.

### Identificar el componente que está causando problemas

Es probable que muchas veces no sea un problema de Asterisk sino de sistema operativo o de otros componentes de software o incluso hardware así que es importante diferenciar si lo que tenemos es un *bug* de Asterisk. Este apartado se enfocará en explicar un procedimiento para tratar de solucionar un *bug* de Asterisk.

También es importante notar que un *bug* es un error en el código de un programa. Muchas personas creen haber encontrado un *bug* de Asterisk para luego darse cuenta que borraron sin querer un archivo importante de configuración o cosa por el estilo y por esa ocasión Asterisk no arrancaba.

### Reproducir el bug

Uno de los problemas más inquietantes es cuando un programa se cae inesperadamente y no sabemos por qué.

Imagínense que un cliente nos llama y nos dice que Asterisk se colgó durante la noche y amaneció apagado. El problema dice que se presenta un par de veces a la semana y ha ocurrido incluso durante el día.

Qué hacer en estos casos donde no hay pistas de qué pueda estar ocurriendo?

Bueno, no hay una solución ideal pero al menos la siguiente guía podría servir.

- Examinar los *logs* en busca de la actividad minutos antes de que ocurriera el problema y tratar de reproducir la misma actividad o al menos la inusual para ver si así gatillamos el *bug*. En ocasiones cosas como la simple recepción de un fax podría gatillar un *bug*.
- Consultar a los usuarios telefónicos si tienen alguna idea de lo que podría estar ocasionando el problema. Nunca hay que subestimar el poder de los usuarios pues se encuentran viviendo la situación telefónica muy de cerca.
- Aunque suene drástico en ocasiones es necesario montar guardia en las instalaciones del cliente y recorrer las instalaciones tratando de esperar a que el problema ocurra. Es posible que veamos a alguien tratando de enviar un *paging* justo antes de que la central se cuelgue.

A pesar de todo lo que podamos hacer hay ocasiones en donde no podremos encontrar la situación que genera el *bug*. En ese caso debemos seguir con los siguientes pasos.

#### Buscar si se trata de un *bug* ya se encuentra identificado

Hay tantos usuarios utilizando Asterisk en el mundo que es probable que si encontramos un *bug* no seamos los únicos que lo hayamos notado.

Por lo tanto si encontramos la situación que gatilla el *bug* lo siguiente que podemos hacer son dos cosas:

1. **Verificar si ya fue solucionado** en recientes versiones de Asterisk. Esto lo podemos hacer examinando el archivo Changelog (o log de cambios) que se incluye con cada lanzamiento de Asterisk. Podemos descargar la versión más reciente de Asterisk para ver el log de cambios más actualizado o podemos consultarlo en el repositorio SVN. Por ejemplo, para la versión 1.4.22 el Changelog se encuentra en la ruta <http://svn.digium.com/view/asterisk/tags/1.4.22/ChangeLog>
2. **Verificar si al menos ha sido reportado** en la interfase de seguimiento de bugs de Asterisk. La cual se encuentra disponible en <http://bugs.digium.com>. Tomémonos un buen tiempo para buscar.

#### Depurar

Esto se explicará con más detalle en el transcurso de este capítulo.

## Reportar el bug

Esto se explicará al final del presente capítulo.

## **Errores de programación que provocan caídas, colgados o comportamientos indeseados**

Es útil conocer estos conceptos para hacer una mejor depuración e inclusive estar más capacitados para corregir el código de darse el caso.

Un programa se puede caer, colgar o funcionar erráticamente por diversas causas, aquí les dejamos con algunas de las más comunes.

### Falla de segmentación (*Segmentation Fault*)

Esto sucede cuando un programa utiliza mal la memoria y pretende ocupar un segmento de memoria que no le corresponde o lo pretende utilizar de una manera no permitida.

En ese momento el kernel, que está vigilante, se da cuenta de ello y para preservar la estabilidad del sistema remueve al programa de memoria y termina su ejecución.

Si no existiera este control de parte del kernel el programa mal escrito podría sobrescribir el espacio de memoria de otras aplicaciones o inclusive del sistema operativo.

### Deadlock

Un *deadlock* o bloqueo mutuo ocurre cuando dos procesos (o hilos) de un programa pretenden utilizar un mismo recurso. El primer proceso adquiere permisos sobre el recurso y lo bloquea de tal manera que el segundo proceso no puede accederlo y queda esperando a que se libere. Del mismo modo el segundo proceso ha bloqueado un recurso que necesita el primer proceso de tal modo que los dos procesos quedan bloqueados esperando que los recursos se liberen, cosa que no sucederá.

Esto provoca que estos procesos queden en un estado de espera sin poder ejecutar más acciones.

### División para cero

Este es un error común. Cuando esto ocurre se puede observar un error del tipo “Floating exception”. El programa se detiene.

### Buffer overflow

Ocorre cuando el programa trata de escribir en búferes de memoria adyacentes. Esto provoca un comportamiento errático de la aplicación.

Cuando se encuentra un problema de *buffer overflow* los fabricantes tienden a solucionarlo rápidamente pues aparte de los problemas obvios de inestabilidad que esto representa existen *hackers* que se encuentran en busca de estos errores para explotarlos en busca de poder ejecutar código propio malicioso en el equipo atacado mediante varias técnicas.

## 24.2 strace

strace es un depurador para Linux muy fácil de usar que monitorea las llamadas a sistema que realiza un programa determinado y todas las señales que este recibe.

Una llamada a sistema es una petición que realiza un programa al kernel para que éste realice una tarea. Esto funciona así porque existen algunas tareas que por cuestión de privilegios una aplicación no puede realizar directamente, entonces tiene que solicitarle dicha tarea al kernel. Un ejemplo es el uso de dispositivos de sistema como la tarjeta de red o manipulación de archivos.

Algunas llamadas a sistema comunes en Linux son `open`, `read`, `write`, `close`, `wait`, `exec`, `fork`, `exit`, `kill`, entre otras.

El trabajo de strace es posible gracias a una característica del kernel llamada *ptrace* que le permite a un programa cualquiera monitorear las llamadas a sistema de cualquier programa hijo. Para más información se puede ejecutar el comando “man ptrace” en el servidor Elastix.

La manera más sencilla de usar strace es ejecutar el comando seguido por el programa que se desea monitorear. Veamos el siguiente ejemplo sencillo monitoreando el comando “cat /proc/cpuinfo”

```
[root@elastix ~]# strace cat /proc/cpuinfo
execve("/bin/cat", ["cat", "/proc/cpuinfo"], [/* 20 vars */]) = 0
brk(0)                                = 0x975c000
access("/etc/ld.so.preload", R_OK)    = -1 ENOENT (No such file or
directory)
open("/etc/ld.so.cache", O_RDONLY)    = 3
fstat64(3, {st_mode=S_IFREG|0644, st_size=37776, ...}) = 0
mmap2(NULL, 37776, PROT_READ, MAP_PRIVATE, 3, 0) = 0xb7fe9000
close(3)                              = 0
open("/lib/libc.so.6", O_RDONLY)     = 3
[...]
open("/usr/lib/locale/locale-archive", O_RDONLY|O_LARGEFILE) = 3
fstat64(3, {st_mode=S_IFREG|0644, st_size=56417648, ...}) = 0
mmap2(NULL, 2097152, PROT_READ, MAP_PRIVATE, 3, 0) = 0xb7de7000
close(3)                              = 0
fstat64(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(136, 4), ...}) = 0
open("/proc/cpuinfo", O_RDONLY|O_LARGEFILE) = 3
fstat64(3, {st_mode=S_IFREG|0444, st_size=0, ...}) = 0
read(3, "processor\t: 0\nvendor_id\t: Genuin...", 4096) = 1076
write(1, "processor\t: 0\nvendor_id\t: Genuin...", 1076) = 1076
read(3, "", 4096)                    = 0
close(3)                             = 0
close(1)                             = 0
exit_group(0)                        = ?
```

La salida anterior puede confundirnos un poco al principio pero si prestamos atención veremos que cada línea representa una llamada de sistema. Podemos ver entre otras cosas que se está tratando de abrir el archivo `/proc/cpuinfo` en modo “sólo lectura”.

```
open("/proc/cpuinfo", O_RDONLY|O_LARGEFILE) = 3
```

El número tres al final significa el código de salida de la llamada de sistema. Salidas negativas como -1 generalmente corresponden a errores. Se puede observar esto en el ejemplo anterior cuando se trata de acceder al archivo `/etc/ld.so.preload`.

En general la sintaxis de uso de `strace` es sencilla.

```
strace [opciones] programa [argumentos]
```

Algunas opciones interesantes son las siguientes.

Parámetro	Descripción
-o	Escribe la salida a un archivo
-p	Monitorea un proceso especificado por su PID. De esta manera se puede monitorear

	un programa en ejecución
-c	Muestra un reporte donde se puede observar un resumen del número de apariciones de las diversas llamadas de sistema
-e	Nos permite filtrar la salida por tipo de llamada de sistema o señal

## Un ejemplo sencillo de depuración de Asterisk

Hay ocasiones donde es importante monitorear un proceso de Asterisk en ejecución. Por ejemplo en el caso de que nos demos cuenta que nuestro servidor se encuentra tomando mucho CPU y queramos averiguar qué está ocurriendo.

***Nota:** En el caso anterior es probable que GDB sea una mejor opción para averiguar qué está ocurriendo.*

Para depurar un programa en ejecución con strace primero debemos averiguar el PID del proceso que queremos monitorear. En nuestro caso monitorearemos Asterisk así que primero enlistaremos los procesos que contengan la palabra Asterisk y tomaremos el menor PID.

```
[root@elastix ~]# ps aux | grep asterisk
asterisk 1619 0.5 1.9 290568 19564 ? S1 Dec10 3:13
/usr/sbin/asterisk -f -U asterisk -G asterisk -vvvg -c
asterisk 4298 0.0 1.4 297492 15024 ? S Dec09 0:00
/usr/sbin/asterisk -f -U asterisk -G asterisk -vvvg -c
root 8656 0.0 0.0 4472 700 ? S Nov07 0:00
/bin/sh /usr/sbin/safe_asterisk -U asterisk -G asterisk
asterisk 17198 0.0 1.4 297332 14864 ? S Dec09 0:00
/usr/sbin/asterisk -f -U asterisk -G asterisk -vvvg -c
[...]
```

En el ejemplo anterior el menor PID es el 1619.

***Nota:** Una manera alternativa de averiguar este PID es usando el comando “pstree -p | grep asterisk”*

Ahora que sabemos el PID podemos monitorearlo con strace de manera sencilla.

```
[root@elastix ~]# strace -p 1619
Process 1619 attached - interrupt to quit
```

En caso de que estemos experimentando con strace sobre un Asterisk que no posea actividad podemos ejecutar el siguiente comando para probar la salida de strace.

```
[root@elastix ~]# asterisk -rx "module reload"
```

Veremos ahora lo siguiente.

```
[root@elastix ~]# strace -p 1619
Process 1619 attached - interrupt to quit
read(0, 0xbf83cc3b, 1)                = ? ERESTARTSYS (To be
restarted)
--- SIGURG (Urgent I/O condition) @ 0 (0) ---
rt_sigaction(SIGURG, {0x80652e0, [URG], SA_RESTART}, {0x80652e0,
[URG], SA_RESTART}, 8) = 0
sigreturn()                          = ? (mask now [])
--- SIGURG (Urgent I/O condition) @ 0 (0) ---
rt_sigaction(SIGURG, {0x80652e0, [URG], SA_RESTART}, {0x80652e0,
[URG], SA_RESTART}, 8) = 0
sigreturn()                          = ? (mask now [])
read(0, 0xbf83cc3b, 1)                = ? ERESTARTSYS (To be
restarted)
--- SIGURG (Urgent I/O condition) @ 0 (0) ---
rt_sigaction(SIGURG, {0x80652e0, [URG], SA_RESTART}, {0x80652e0,
[URG], SA_RESTART}, 8) = 0
sigreturn()                          = ? (mask now [])
--- SIGURG (Urgent I/O condition) @ 0 (0) ---
rt_sigaction(SIGURG, {0x80652e0, [URG], SA_RESTART}, {0x80652e0,
[URG], SA_RESTART}, 8) = 0
sigreturn()                          = ? (mask now [])
--- SIGURG (Urgent I/O condition) @ 0 (0) ---
rt_sigaction(SIGURG, {0x80652e0, [URG], SA_RESTART}, {0x80652e0,
[URG], SA_RESTART}, 8) = 0
sigreturn()                          = ? (mask now [])
--- SIGURG (Urgent I/O condition) @ 0 (0) ---
rt_sigaction(SIGURG, {0x80652e0, [URG], SA_RESTART}, {0x80652e0,
[URG], SA_RESTART}, 8) = 0
sigreturn()                          = ? (mask now [])
[...]
```

Para salir basta con ejecutar Ctrl+C



## 24.3 GDB

GDB es un completo depurador desarrollado y mantenido por la Free Software Foundation.

GNU nos permite hacer seguimiento de la ejecución de un programa insertando pausas cuando creamos necesario; también nos permite observar lo que un programa estaba haciendo cuando se colgó. Inclusive nos permite cambiar ciertas cosas dentro del programa para ver los efectos que se producen con dichos cambios.

GDB es una opción más completa que *strace* pero también su uso requiere un tanto más de conocimiento.

Con GDB podemos depurar programas escritos en C o C++ como es el caso de Asterisk. Sin embargo, GDB también soporta otros lenguajes.

### Ejemplos simples de uso de GDB

Existen tres maneras de usar gdb. Para depurar un programa ejecutándolo desde el propio depurador, para depurar un programa que ya se encuentra en ejecución o para hacer depuración de un archivo `core` dejado por una aplicación caída.

#### Depurando un programa desde el depurador

El depurador gdb puede encargarse de arrancar el programa y ejecutarlo. Para esto podemos utilizar la sintaxis.

```
gdb programa
```

#### Depuración sobre Asterisk en plena ejecución

Podemos depurar un programa en plena ejecución si conocemos su PID. Encontrar el PID no es muy difícil y podemos usar el comando `ps` para ello.

Para depurar un programa en plena ejecución debemos especificar primero el nombre del proceso seguido por el PID.

```
gdb programa PID
```

## Depuración haciendo backtrace sobre un Asterisk caído

También es posible depurar un programa después de caído siempre y cuando éste haya dejado un archivo `core`. Para depurar un archivo `core` debemos especificar el nombre del programa que generó dicho archivo y la ubicación del archivo `core`.

```
gdb programa core
```

## Comandos básicos

A continuación una lista con los comandos más básicos de GDB. Para más información puede recurrir al comando “`man gdb`”.

### run

Ejecuta un programa

### bt

Hace un *backtrace*, es decir muestra la pila de memoria del programa

### break número-línea

Inserta una parada en una línea dada del programa que se va a ejecutar. Cuando se llega a esa línea se detiene la ejecución del programa en ese punto.

### break nombre-función

Inserta una parada antes de la primera línea de la función especificada. Del mismo modo que el comando anterior, cuando se llega a esa línea se detiene la ejecución del programa en ese punto.

### cont

Si el programa se encuentra detenido, continúa con la ejecución del programa. Por ejemplo, cuando se ha llegado a una parada con el comando `break`.

list

Enlista la fuente del programa en las cercanías de donde se encuentra detenido.

step

Ejecuta el programa línea por línea, después de que ha sido detenido.

next

Igual que `step` pero no ingresa a las funciones.

status

Imprime una lista de las paradas (*breakpoints*)

delete

Elimina una parada (*breakpoint*)

quit

Nos permite salir de GDB

## 24.4 Archivos core y *backtraces*

Muchas aplicaciones dejan archivos core cuando terminan inesperadamente, lo que nos permite hacer una depuración post-mortem del software.

En el caso de Asterisk esto es posible cuando se lo inicia con la opción `-g`. En el caso de Elastix, Asterisk se inicia por defecto con dicha opción.

Si no se está seguro que Asterisk haya sido iniciado con la opción `-g` se puede verificar con el siguiente comando.

```
[root@elastix ~]# ps aux | grep asterisk
root      8656  0.0  0.0   4472   696 ?        S    Nov07   0:00
/bin/sh /usr/sbin/safe_asterisk -U asterisk -G asterisk
asterisk  8715  0.6  2.4 297488 25412 ?        Sl   Nov19 188:55
/usr/sbin/asterisk -f -U asterisk -G asterisk -vvvg -c
asterisk 17198  0.0  1.4 297332 14864 ?        S    Dec09   0:00
/usr/sbin/asterisk -f -U asterisk -G asterisk -vvvg -c
[...]
```

Un archivo `core` no es otra cosa que el contenido de memoria del programa en el momento en que se detuvo. Es el sistema operativo quien guarda ese contenido de memoria pues la aplicación no está en capacidad de hacerlo puesto se encuentra detenida.

Este contenido de memoria, combinado con la tabla de símbolos que se genera cuando el programa es compilado hacen que `gdb` nos pueda mostrar información más legible.

Un *backtrace* (o seguimiento hacia atrás) es un reporte de las llamadas a funciones que se encontraban en memoria en un momento dado.

Desde `gdb` es posible realizar *backtraces* sobre un archivo `core`. Es decir obtener un reporte de las llamadas a funciones en el momento en que el programa se cayó. Para hacerlo existen dos comandos, el `bt` y el `bt full`.

El `bt full` es un backtrace más detallado que el normal que se consigue con `bt` y muestra además parámetros y variables locales usados por las funciones que están siendo invocadas.

Veamos un ejemplo de esto. Primero identificamos el archivo `core` que necesitamos examinar, por lo general se trata del último.

```
[root@elastixtest ~]# ll /tmp/ | grep core
-rw----- 1 asterisk asterisk 301694976 Oct 17 14:02 core.34543
-rw----- 1 asterisk asterisk 280817664 Oct 23 10:07 core.45433
-rw----- 1 asterisk asterisk 323821568 Nov  7 16:44 core.54333
-rw----- 1 asterisk asterisk 323416064 Nov 14 10:30 core.54654
-rw----- 1 asterisk asterisk 292384768 Nov 19 16:28 core.63222
```

En nuestro caso examinaremos el archivo llamado `core.63222`. Iniciamos el depurador y realizamos un backtrace.

```

[root@elastix ~]# gdb asterisk /tmp/core.63222
[...]
Reading symbols from /usr/lib/asterisk/modules/chan_mgcp.so...done.
Loaded symbols for /usr/lib/asterisk/modules/chan_mgcp.so
Reading symbols from /usr/lib/asterisk/modules/format_gsm.so...done.
Loaded symbols for /usr/lib/asterisk/modules/format_gsm.so

Core was generated by `/usr/sbin/asterisk -f -U asterisk -G asterisk -vvvg -c'.
Program terminated with signal 11, Segmentation fault.
#0  0x080ecfd7 in c_hmatch ()
(gdb) set logging on
Copying output to gdb.txt.
(gdb) bt
#0  0x080ecfd7 in c_hmatch ()
#1  0x00000001 in ?? ()
#2  0x00000064 in ?? ()
#3  0xbfab4b80 in ?? ()
#4  0xa9610c08 in ?? ()
#5  0x09989cf0 in ?? ()
#6  0xbfab4b78 in ?? ()
#7  0x080ed371 in parse_line ()
Previous frame inner to this frame (corrupt stack?)
(gdb) bt full
#0  0x080ecfd7 in c_hmatch ()
No symbol table info available.
#1  0x00000001 in ?? ()
No symbol table info available.
#2  0x00000064 in ?? ()
No symbol table info available.
#3  0xbfab4b80 in ?? ()
No symbol table info available.
#4  0xa9610c08 in ?? ()
No symbol table info available.
#5  0x09989cf0 in ?? ()
No symbol table info available.
#6  0xbfab4b78 in ?? ()
No symbol table info available.
#7  0x080ed371 in parse_line ()
No symbol table info available.
(gdb) thread apply all bt
[...]

```

## Cómo reportar un *backtrace* a Digium?

Si determinamos que tenemos un *bug* de Asterisk en nuestras manos sea que lo resolvamos o no, el siguiente paso lógico es reportar ese *bug* al fabricante.

Digium dispone de una interfase Web para esta finalidad ubicada en:

<http://bugs.digium.com>

Cuando se reporta un *bug* es de mucha ayuda para los desarrolladores de Asterisk contar un *backtrace* para poder entender mejor lo que ocurrió a nivel interno.

Los lineamientos generales para reportar *backtraces* se encuentran en el siguiente sitio:

<http://www.asterisk.org/doxygen/trunk/AstDebug.html>

Básicamente se pide no subir archivos core sino solo la salida del comando `bt` y `bt full` conseguidos con `gdb`.

Adicionalmente es importante al reportar un *bug* de Asterisk mencionar que se está utilizando Elastix porque esto permitirá a los desarrolladores de Digium conocer que se trata de un Asterisk compilado con parches y funcionalidad adicional.

# 25

## DUNDI

*Afirmaciones extraordinarias requieren siempre de evidencia  
extraordinaria*

-- Carl Sagan

### 25.1 Introducción

DUNDi <sup>TM</sup> (Distributed Universal Number Discovery) es un sistema de localización de *gateways* para el acceso a los servicios de telefonía. Utiliza un esquema punto a punto y, a diferencia de ENUM que es estrictamente jerárquico por que se basa en DNS, DUNDi no requiere ninguna arquitectura de red en particular ni un esquema jerárquico cliente-servidor.

En otras palabras DUNDI es capaz de consultar los contextos de otros equipos para encontrar una ruta hacia determinado número.

DUNDi en realidad no realiza una llamada como tal, ya que no es un protocolo de señalización VoIP; mas bien el recibe y proporciona la información necesaria para poder contactar a los equipos independientemente del protocolo VoIP que sea usado.

DUNDi fue inventado por Mark Spencer, quien también desarrolló Asterisk, núcleo de la telefonía en Elastix, por lo tanto es de esperar que la integración con este último sea completamente transparente así como su compatibilidad impecable.

Para que DUNDi funcione se debe conocer al menos un *peer* quien a su vez, si no puede responder al requerimiento, puede delegar la consulta a un *peer* conocido por él, y así hasta que se encuentre a un *peer* que tenga una respuesta al número consultado (aunque este parámetro es configurable con TTL).

Un ejemplo típico de la utilidad de DUNDi es en una empresa multi-sede, en la cual cada sede tiene su propio equipo con Elastix. Cada equipo puede publicar sus extensiones y si uno de los equipos pregunta, por ejemplo, ¿donde está el número 456? la consulta se enviará directa o indirectamente a todos los servidores, y el servidor que tenga esa extensión responderá algo como `IAX2/usuario:clave@1.2.3.4/456`.

Aunque cada equipo tenga asignado un rango de extensiones o una serie de números, DUNDi resultará útil, ya que no intentará comunicarse con el equipo si la extensión no está creada, aunque esté en su rango de extensiones.

## 25.2 Ejemplo de configuración

La mejor manera de explicar DUNDi es ilustrando con un ejemplo, así que eso es lo que haremos a continuación. Mostraremos un ejemplo de cómo distribuir el plan de marcado en 3 equipos utilizando DUNDi.

### Escenario

En el presente ejemplo se configurarán tres equipos con los siguientes datos.

Nombre de equipo	Dirección IP	MAC
SRV01	192.168.255.201	00:21:5A:49:24:98
SRV02	192.168.255.202	00:21:5A:49:7E:9E
SRV02	192.168.255.203	00:22:64:08:F4:FE

### Certificados de encriptación

DUNDi usa certificados de encriptación RSA para compartir los planes de marcado, además por que las respuestas a una consulta incluyen las claves de las troncales (aunque es dinámico).



Usaremos la misma clave para la comunicación entre todos los servidores, así que el primer paso es generar los certificados de encriptación uno de los equipos (en este ejemplo en SRV01):

SRV01:

```
cd /var/lib/asterisk/keys
astgenkey -n SERVERS-DUNDI
```

Ahora necesitamos compartir los certificados entre los dos servidores; esto es para que cada uno pueda des-encriptar al otro.

Copiar hacia SRV02:

```
[root@elastix ~]# scp /var/lib/asterisk/keys/SERVERS-DUNDI.* \
192.168.255.202:/var/lib/asterisk/keys
```

Copiar hacia SRV03:

```
[root@elastix ~]# scp /var/lib/asterisk/keys/SERVERS-DUNDI.* \
192.168.255.203:/var/lib/asterisk/keys
```

## **dundi.conf**

Vamos a generar este archivo desde el principio, sección por sección:

### Sección [general]

En esta sección del archivo definimos nuestra propia identificación en la nube DUNDi, así como las opciones globales, debemos rellenar los campos con los datos reales de nuestros equipos (MAC, etc.):

Para SRV01:

```
[general]
;
department=SERVER01
organization=AlgunaOrganizacion
locality=Quito
stateprov=Pichincha
country=EC
email=info[at]elajonjoli.org
phone=+593-0000000
;
; Aqui podemos definir en que red va a escuchar dundi
;bindaddr=0.0.0.0
; El puerto en el que va a escuchar, por defecto 4520
port=4520
; Aqui nos identificamos con nuestra MAC address
entityid=00:21:5A:49:24:98
; El numero de consultas que hará DUNDI hasta alcanzar una extensión
ttl=12
; Finaliza las conexiones fallidas
autokill=yes
```

**Para SRV02:**

```
[general]
;
department=SERVER02
organization=AlgunaOrganizacion
locality=Quito
stateprov=Pichincha
country=EC
email=info[at]elajonjoli.org
phone=+593-0000000
;
;Aqui podemos definir en que red va a escuchar dundi
;bindaddr=0.0.0.0
; El puerto en el que va a escuchar, por defecto 4520
port=4520
; Aqui nos identificamos con nuestra MAC address
entityid=00:21:5A:49:7E:9E
; El numero de consultas que hará DUNDI hasta alcanzar una extensión
ttl=12;
; Finaliza las conexiones fallidas
autokill=yes
```

Para SRV03:

```
[general]
;
department=SERVER03
organization=AlgunaOrganizacion
locality=Quito
stateprov=Pichincha
country=EC
email=info[at]elajonjoli.org
phone=+593-0000000
;
;Aquí podemos definir en que red va a escuchar dundi
;bindaddr=0.0.0.0
; El puerto en el que va a escuchar, por defecto 4520
port=4520
; Aquí nos identificamos con nuestra MAC address
entityid=00:22:64:08:F4:FE
; El numero de consultas que hará DUNDI hasta alcanzar una extensión
ttl=12;
; Finaliza las conexiones fallidas
autokill=yes
```

### Sección [mappings]

Básicamente aquí definimos nuestra respuesta a una consulta hacia determinado número que tengamos localmente.

Para SRV01:

```
[mappings]
priv => dundi-priv-
canonical,0,IAX2,dundi:${SECRET}@192.168.251.251/${NUMBER},nopartial
priv => dundi-priv-
customers,100,IAX2,dundi:${SECRET}@192.168.251.251/${NUMBER},nopartial
priv => dundi-priv-via-
pstn,400,IAX2,dundi:${SECRET}@192.168.251.251/${NUMBER},nopartial
```

Para SRV02:

```
[mappings]
priv => dundi-priv-
canonical,0,IAX2,dundi:${SECRET}@192.168.251.252/${NUMBER},nopartial
priv => dundi-priv-
customers,100,IAX2,dundi:${SECRET}@192.168.251.252/${NUMBER},nopartial
priv => dundi-priv-via-
pstn,400,IAX2,dundi:${SECRET}@192.168.251.252/${NUMBER},nopartial
```

Para SRV03:

```
[mappings]
priv => dundi-priv-
canonical,0,IAX2,dundi:${SECRET}@192.168.251.253/${NUMBER},nopartial
priv => dundi-priv-
customers,100,IAX2,dundi:${SECRET}@192.168.251.253/${NUMBER},nopartial
priv => dundi-priv-via-
pstn,400,IAX2,dundi:${SECRET}@192.168.251.253/${NUMBER},nopartial
```

Los valores de `${NUMBER}` y `${SECRET}` se reemplazarán automáticamente, según el número que se consulte y la clave aleatoria que se haya generado en ese momento.

### Peers

Adicionalmente debemos definir los PEERS que consultaran y serán consultados (con la opción `symmetric`), esto lo hacemos así:

Para SRV01:

```
; Identificamos al servidor 2 (SRV02) por su entityid
[00:21:5A:49:7E:9E]
;permitir y realizar conexiones
model=symmetric
; IP ser servidor SRV02
host=192.168.255.202
inkey=SERVERS-DUNDI
outkey=SERVERS-DUNDI
; Incluimos el contexto antes definido en la sección mappings
include=priv
permit=priv
qualify=yes
order=primary

; Identificamos al servidor 3 (SRV03) por su entityid
[00:22:64:08:F4:FE]
;permitir y realizar conexiones
model=symmetric
; IP ser servidor SRV03
host=192.168.255.203
inkey=SERVERS-DUNDI
outkey=SERVERS-DUNDI
; Incluimos el contexto antes definido en la sección mappings
include=priv
permit=priv
qualify=yes
order=primary
```

Para SRV02:

```
; Identificamos al servidor 1 (SRV01) por su entityid
[00:21:5A:49:24:98]
;permitir y realizar conexiones
model=symmetric
; IP ser servidor SRV01
host=192.168.255.201
inkey=SERVERS-DUNDI
outkey=SERVERS-DUNDI
; Incluimos el contexto antes definido en la sección mappings
include=priv
permit=priv
qualify=yes
order=primary

; Identificamos al servidor 3 (SRV03) por su entityid
[00:22:64:08:F4:FE]
;permitir y realizar conexiones
model=symmetric
; IP ser servidor SRV03
host=192.168.255.203
inkey=SERVERS-DUNDI
outkey=SERVERS-DUNDI
; Incluimos el contexto antes definido en la sección mappings
include=priv
permit=priv
qualify=yes
order=primary
```

Para SRV03:

```

; Identificamos al servidor 1 (SRV01) por su entityid
[00:21:5A:49:24:98]
;permitir y realizar conexiones
model=symmetric
; IP ser servidor SRV01
host=192.168.255.201
inkey=SERVERS-DUNDI
outkey=SERVERS-DUNDI
; Incluimos el contexto antes definido en la sección mappings
include=priv
permit=priv
qualify=yes
order=primary

; Identificamos al servidor 1 (SRV01) por su entityid
[00:21:5A:49:24:98]
;permitir y realizar conexiones
model=symmetric
; IP ser servidor SRV01
host=192.168.255.201
inkey=SERVERS-DUNDI
outkey=SERVERS-DUNDI
; Incluimos el contexto antes definido en la sección mappings
include=priv
permit=priv
qualify=yes
order=primary

```

## **iax\_custom.conf**

Aquí debemos definir la configuración de la troncal; como DUNDI va a hacer todo el trabajo de enviar la cadena exacta de como contactarlos solo creamos un usuario que recibe las llamadas autenticadas de la nube DUNDi:

En SRV01 y en SRV02

```
[dundi]
type=user
dbsecret=dundi/secret
context=ext-local
; Podemos restringir ciertos codecs para mejorar la calidad
; de voz, o dar prioridad al ancho de banda (con gsm o g729)
; disallow=all
; allow=ulaw
```

## extensions.conf

Para lograr la integración con freePBX debemos modificar un poco el contexto "from-internal" que es el contexto que usa freePBX para las extensiones.

Teniendo en cuenta que cuando un número no es encontrado en el plan de marcado local se usa el contexto [bad-number] es allí donde debemos modificar y la configuración debe ser exactamente la misma en los tres servidores:

En el archivo `extensions.conf` buscamos la sección:

```
[from-internal]
include => from-internal-xfer
include => bad-number
```

Comentamos la línea **include => bad-number** (poniendo un punto y coma adelante) y agregamos otra que diga **include => dundi-priv-lookup**.

que es un contexto que crearemos luego en el archivo `extensions_custom.conf`, quedando así:

```
[from-internal]
include => from-internal-xfer
;include => bad-number
include => dundi-priv-lookup
```

## extensions\_custom.conf

Aquí definimos los contextos personalizados para que DUNDI mapee nuestra extensiones, así como una macro para hacer las búsquedas en otros equipos (lo que evita loops) y la sentencia para redirigir las llamadas (switch).



Debemos añadir las siguientes líneas al final del archivo `extensions_custom.conf` en los dos servidores:

```
; *****
; CONFIGURACION PARA DUNDi
[dundi-priv-canonical]
; Aqui incluimos el contexto que contiene las extensiones.
include => ext-local
; Aqui incluimos el contexto que contiene las colas de atención
include => ext-queues

[dundi-priv-customers]
; Si tenemos clientes (o revendemos servicios) podemos listarlos aqui

[dundi-priv-via-pstn]
; Aquí podemos incluir contexto con nuestras troncales hacia la PSTN,
; si queremos que los demás equipos puedan usar nuestras troncales
; include => outbound-allroutes

[dundi-priv-local]
; En este contexto unificamos los tres contextos, este lo podemos
; usar como contexto de la troncal iax de dundi
include => dundi-priv-canonical
include => dundi-priv-customers
include => dundi-priv-via-pstn

[dundi-priv-lookup]
; Este contexto se encarga de hacer la búsqueda de un numero por dundi
; Antes de hacer la búsqueda definimos apropiadamente nuestro
; caller id. ya que si no tendremos un caller id como "device<0000>".
exten => _X.,1,Macro(user-callerid)
exten => _X.,n,Macro(dundi-priv,${EXTEN})
exten => _X.,n,GotoIf("${DIALSTATUS}" = "BUSY"?100)
exten => _X.,n,Goto(bad-number,${EXTEN},1)
exten => _X.,100,Playtones(congestion)
exten => _X.,101,Congestion(10)

[macro-dundi-priv]
; Esta es la macro que llamamos desde el contexto [dundi-priv-lookup]
; Tambien evita que hayan loops en las consultas dundi.
exten => s,1,Goto(${ARG1},1)
switch => DUNDi/priv
; *****
```

## Aplicar la configuración

Para que los cambios se hagan efectivos podemos ejecutar el comando:

```
asterisk -rx "restart when convenient"
```

O si no queremos esperar:

```
asterisk -rx "restart now"
```

## Conclusión

Ahora cada extensión que creamos desde freePBX se añadirá al contexto [ext-local] el cual esta incluido en `dundi-priv-canonical` que va a usar DUNDi para indicar que extensiones tenemos localmente.

Si nosotros tenemos la extensión responderemos con los datos para la comunicación hacia esa extensión; esos datos a su vez los interpreta el servidor que esta preguntando y usa la función `switch` que pusimos en el archivo `extensions_custom.conf` para comunicarse con la extensión local.

## 25.3 Integrando IVRs con DUNDi

Cuando tenemos varios equipos y troncales de la PSTN que ingresan a uno de ellos, es muy posible que queramos que el IVR que recibe esas líneas pueda llamar directamente a las extensiones de la red DUNDi.

Para lograrlo, lo primero que hay que hacer es localizar el contexto de IVR que hemos creado desde freePBX, este último siempre los crea así con un formato [ivr-N], donde N es un numero incremental que asigna automáticamente.

Tenemos que revisar en el archivo `/etc/asterisk/extensions_additional.conf` y buscar el contexto adecuado.

Suponiendo que el contexto del IVR es [ivr-2] tendríamos que definir la siguiente configuración en el archivo `/etc/asterisk/extensions_additional.conf`:

```
[dundi-priv-lookup-ivr]
exten => _XXXX,1,Macro(dundi-priv,${EXTEN})
exten => _XXXX,n,GotoIf("${DIALSTATUS}" = "BUSY"?100)
exten => _XXXX,n,Goto(ivr-2,i,1)
exten => _XXXX,n,Hangup()
exten => _XXXX,100,Playtones(congestion)
exten => _XXXX,101,Congestion(10)
exten => _XXXX,102,Hangup()
[ivr-2-custom]
include => ext-local
include => ext-queues
include => dundi-priv-lookup-ivr
```

Esto teniendo en cuenta que nuestras extensiones internas tienen 4 dígitos, si no debemos modificar el contexto [dundi-priv-lookup-ivr] y en vez de \_XXXX poner el numero de dígitos que usamos.



# 26

## Módulo de Call Center

*Hace más ruido un sólo hombre gritando que cien mil callados*

-- José de San Martín

### 26.1 Introducción

El módulo de CallCenter de Elastix es un paquete de software diseñado para asistir en la implementación de centros de llamadas masivas con interacción entre operadores contratados por el administrador de la central telefónica (de aquí en adelante llamados "agentes") y personas contactadas a través de llamadas telefónicas (de aquí en adelante llamados "abonados"). En este capítulo, se usa el término "campana" para designar la ejecución de un conjunto de llamadas atendidas por un grupo de agentes, durante un intervalo de fechas específico.

### 26.2 Funcionalidad básica

#### Modelo de funcionamiento

El módulo del CallCenter (en su versión actual) está implementado alrededor del soporte de colas de Asterisk. El diseño del módulo asume que cada una de las colas alberga a uno o más agentes, los cuales pueden estar "registrados" (o también "logoneados") como presentes en una cola, o ausentes. Además se asume que cada

campaña utiliza exactamente una cola de Asterisk. Actualmente se reconocen dos modos principales de funcionamiento: campañas salientes, y campañas entrantes. Estos dos modos de funcionamiento se diferencian en quién origina las llamadas.

En una campaña saliente, al sistema se lo alimenta con un archivo que contiene los números telefónicos a marcar. Al activarse la campaña, y si hay agentes registrados como presentes en la cola asignada, el sistema marca los números telefónicos, tantos como agentes libres se encuentren en la cola. El método por medio del cual se coloca una llamada, une la cola (no un agente específico) con el número del abonado al cual se llama, y por lo tanto la llamada realizada, si es exitosa, entra a la cola y es atendida por el siguiente agente libre (según la política de asignación). Al conectarse la llamada, el sistema registra el agente que el Asterisk ha elegido para atender la llamada, y escribe esta información en la base de datos, para que sea recogida por la porción Web. Esta información también permite asociar una llamada con una respuesta a un formulario asociado a la campaña. Al terminar la llamada, el agente pasa al estado ocioso, y está de nuevo disponible para recibir una llamada. Este proceso continúa hasta recorrer toda la lista de números. Según el parámetro de reintentos especificado por la campaña, se vuelven a intentar marcar números que fallaron en ser contactados, hasta terminar con el número de reintentos o cuando cada número haya sido correctamente contactado.

En una campaña entrante, se define una cola a la cual ingresan los agentes, pero el marcador no inicia llamada alguna. En lugar de ello, se debe organizar que los abonados llamen al sistema, y que resulten ruteados a la cola configurada para la campaña. Como en el caso de la campaña saliente, se registra a qué agente fue asignada la llamada. Opcionalmente se puede alimentar al sistema con datos sobre números específicos, de forma que cuando este número llama, el agente tenga la información correspondiente al abonado disponible en su pantalla de información.

## **Funcionalidad implementada**

La versión 1.3 del módulo CallCenter tiene las siguientes funcionalidades implementadas:

- Llamado automático a partir de una lista de números
- Ejecución de múltiples campañas simultáneas, cada una con su propia cola asignada
- Seguimiento de agente asignado para atender una llamada
- Marcador predictivo. Esta característica intenta predecir, en base a la duración observada de las llamadas realizadas durante una campaña, la probabilidad de que una llamada actualmente en curso sea terminada en un intervalo determinado. Si la llamada se considera a punto de terminarse, se colocan llamadas adicionales con la esperanza de que el agente está disponible cuando la llamada llegue a conectarse.
- Almacenamiento de guión de atención. Esta característica permite asociar la campaña con un discurso que el agente debe seguir para atender óptimamente al

abonado. Este discurso está disponible en la cejilla que aparece en la consola de agente.

- Ingreso de formulario. Si la campaña saliente es realizada para hacer una encuesta, el sistema puede almacenar un formulario que el agente llena con las respuestas del abonado. Las respuestas de todos los números marcados en una campaña son recuperables como un archivo CSV a través de la interfaz Web.
- Auditoría de tiempos de descanso
- Capacidad de poner una llamada en "hold"
- Reportes de actividad de llamadas

## Limitaciones conocidas

El ingreso de formulario no está implementado en la versión 1.3 del CallCenter para campañas entrantes. Esta funcionalidad se planea será implementada en versiones futuras del módulo CallCenter.

El marcador del módulo CallCenter no controla cuál agente termina atendiendo un número específico que ha sido marcado, o que ha sido recibido. Esto es una limitación fundamental del modelo de agentes de Asterisk como miembros de una cola de atención. Debido a que las llamadas son colocadas entre la cola y el número marcado, el marcador no puede decidir a cuál agente se le asignará la llamada colocada. Esto excluye el marcador para su uso en situaciones en las que un agente debe atender a un subconjunto definido de números. Esta característica podrá ser implementada en una versión futura de CallCenter, aunque requiere volver a plantear la interacción registro del agente con el sistema.

El marcador del módulo CallCenter no está diseñado para reproducir mensajes grabados en llamadas no atendidas. Esto no es exactamente una limitación, sino una consecuencia del objetivo para el cual fue escrito el módulo CallCenter. El módulo de CallCenter está pensado para mediar llamadas que luego son atendidas por agentes humanos. Por lo tanto, no coloca llamadas en una cola sin agentes. Es posible que se implemente esta característica en una versión futura, como un tipo especial de campaña que no requiere de agentes.

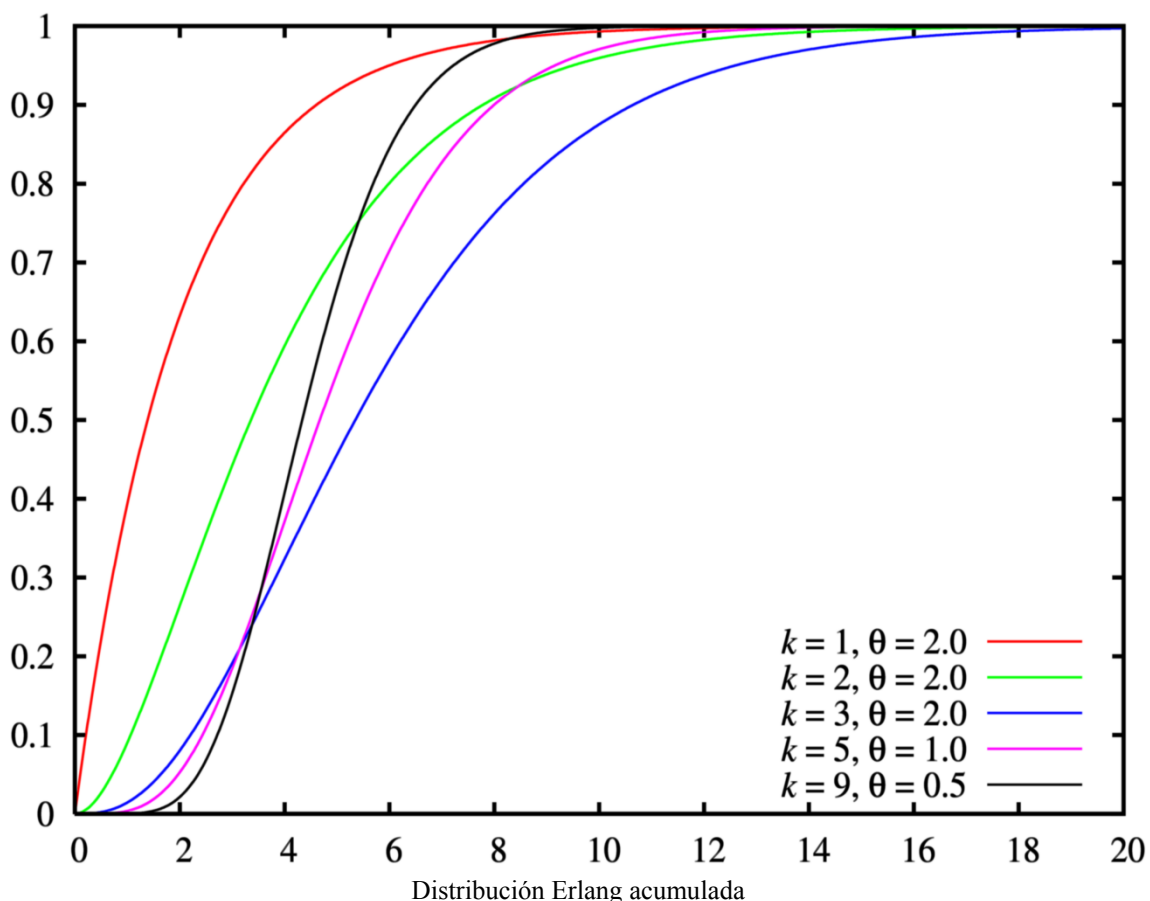
## 26.3 Componentes del Elastix Dialer

### Marcador predictivo

El marcador predictivo es la porción del CallCenter que interactúa directamente con las llamadas. El marcador está implementado como un servicio residente (daemon) escrito en PHP, e iniciable como un servicio a través del comando 'service elastixdialer stop|start'. Para las campañas salientes, el marcador tiene la tarea de iniciar cada una de

las llamadas hacia los abonados, mediante el uso del comando Originate de Asterisk. El marcador lleva la cuenta de las llamadas que han sido colocadas, pero no han sido informadas de éxito o fracaso a través de un evento OriginateResponse enviado por Asterisk. Cuando una llamada es respondida, el marcador recibe eventos OriginateResponse, Join y Link que informan características de la llamada colocada, en particular el agente al cual terminó siendo asignada. Al recibir estos eventos, el marcador escribe la información de la llamada en la base de datos para su uso por parte de la porción Web del módulo.

El marcador interroga regularmente (cada 3 segundos) el estado de la cola (para las campañas salientes), para averiguar cuántos agentes están logoneados y libres. El número de agentes libres (más el límite por campaña, véase Resolución de problemas) regula cuántas llamadas son iniciadas simultáneamente cada vez que se revisa el estado de la campaña. Si el número de llamadas colocadas en la historia de la campaña es suficientemente grande, el marcador intenta usar las estadísticas de duración media de la llamada, para intentar predecir si una llamada en curso está a punto de terminar, para colocar proactivamente llamadas adicionales. El modelo para predicción es una distribución Erlang acumulada.



La interfaz Web permite iniciar y detener el servicio de marcador, y además permite activar y desactivar la depuración de problemas en el marcador, incluso si el marcador está corriendo. La ruta por omisión del log del marcador, donde aparecen las operaciones realizadas por éste, es el archivo `/opt/elastic/dialer/dialerd.log`. Cuando la



depuración está desactivada, sólo los errores se escriben a este log. Al activar la depuración, también se escriben reportes regulares sobre el estado interno del marcador, así como los parámetros que se usan para generar llamadas, los eventos que recibe el marcador, y la interpretación de cuántos agentes están libres.

## Consola de agente

La consola de agente es el componente del CallCenter que administra la interacción del agente con el sistema. Esta consola de agente es esencialmente una página Web que revisa regularmente el estado de llamadas asignadas al agente, y consulta la información relevante a esta llamada para ser mostrada. Por medio de la consola de agente se informa cuál número está siendo atendido, la información de contacto para el número marcado (si está disponible), y además se interactúa con el formulario para la encuesta asociada al agente (si existiese alguna). La consola de agente tiene integrados comandos para iniciar y finalizar una pausa (*break*), para poner a la llamada en espera, y para realizar transferencias de llamadas.

## Administración de campañas

La administración de campañas permite crear nuevas campañas, para su ejecución al logonearse los agentes. Aquí se implementa también la creación de formularios, y la administración de *breaks*. Adicionalmente se proporciona una interfaz para la creación de agentes en el sistema que son capaces de recibir llamadas de campañas, y una pantalla de configuración que permite iniciar, detener y configurar el marcador predictivo, incluyendo la activación de depuración.

## Reportes

Los reportes implementados incluyen:

- Reporte de los descansos tomados por los agentes
- Detalle de las llamadas colocadas, filtrable por periodo de tiempo, cola y agentes
- Histograma de las llamadas colocadas en cada hora, por fecha y estado
- Reporte de las llamadas contestadas por cada agente, filtrable por periodo
- Histograma de tiempo de espera para las llamadas
- Intervalo total que cada agente pasa logoneado en el sistema
- Eficiencia de colas de llamadas entrantes, como porcentaje de llamadas entrantes que son abandonadas

## **26.4 Procedimientos para el uso del Elastix Dialer**

### **Administrador - Configuración de agentes**

Para configurar nuevos agentes para la realización de campañas, se debe de ir a la pantalla de CallCenter-->Agents. La página mostrada es un listado de los agentes existentes en el sistema (una lista vacía si no se ha creado ninguno). Al seleccionar el botón de Nuevo Agente, se muestra un diálogo que pide para el nuevo agente, el nombre, el número de agente, y una contraseña que se usa para el login del agente. Una vez ingresado esto, el agente se crea en el sistema y puede ser usado para nuevas campañas.

### **Administrador - Creación de campañas**

Para la creación de una campaña saliente, primero se debe crear, desde alguna fuente externa, un archivo de texto que contenga todos los números que serán contactados en una tabla. Este archivo debe de estar en formato CSV (archivo separado por comas) y debe de tener la columna de números de teléfono como primera columna. [TODO: cuál es la columna de los nombres?] A continuación se debe de elegir una cola que no está siendo todavía utilizada.

Una campaña debe tener además un formulario asociado para que sea llenado con las respuestas del abonado. Si no se ha creado un formulario todavía, se debe ir a la pantalla de Formularios, y pulsar en el botón de Nuevo Formulario. Esto inicia el editor de formularios, en el cual se especifica una serie de campos. Cada campo tiene un nombre, una etiqueta y un widget que se mostrará para el ingreso del campo. [TODO: confirmar esto] El formulario definido se guarda con un nombre simbólico, el cual se usará como referencia en pasos futuros. Es posible reusar el mismo formulario para varias campañas.

Una vez hecho esto, se debe ir a la pantalla de CallCenter-->Outgoing Calls, donde se muestra un listado de las campañas que se han creado en el sistema. Se pulsa el botón de Nueva Campaña, y aparece un diálogo en el que se ingresa el intervalo de fechas en que se realiza la campaña, el horario en que se realiza la campaña en cada día, el formulario que se usa para la campaña, la cola que se usará para la campaña, y el guión que el agente debe de usar para interactuar con el cliente, entre otros valores. También se requiere en el mismo formulario, la carga del archivo CSV creado como primer paso. Al pulsar el botón Guardar, se crea la nueva campaña.

### **Administrador - Reportes**

Para todos los reportes, se requiere elegir las opciones de filtrado requeridas.

## **Operador - interacción con consola de agente**

Todo agente debe tener a mano un teléfono IP configurado para que se conecte a la central Elastix usada para las campañas. Para esto se puede usar cualquier teléfono VoIP físico, o un programa de *softphone* que corra en la PC del agente.

Para empezar a atender llamadas, el agente debe ingresar a la página Web de Elastix con su usuario de operador, y luego elegir la opción CallCenter-->Agent Console. Esto muestra una segunda petición de login, que muestra una lista desplegable con los agentes creados, y otra lista desplegable con las extensiones disponibles. El agente debe elegir de las listas su propio código y extensión, y pulsar el botón de Aceptar. Entonces la interfaz Web ordena al Asterisk que inicie una llamada de login hacia la extensión del agente. El agente debe contestar el teléfono, y escuchará una grabación que indica que ingrese la clave de acceso del agente. El agente debe usar el teclado numérico del teléfono para ingresar la clave, y si es correcta, el sistema admite al agente como logoneado y disponible para recibir llamadas.

Si hay una campaña activa en la cola donde el agente está logoneado, el agente eventualmente recibirá una llamada a través de la cola. Entonces la interfaz Web muestra la información del número marcado, y posiblemente la información ingresada según la campaña. El agente debe entonces hablar con el abonado, usando si es necesario el guión disponible en la cejilla marcada como Script. Si la campaña se realiza para encuesta, y se tiene un formulario, el formulario aparece en otra cejilla, para que el agente lo llene con la información proporcionada con el abonado. Si el abonado cierra la llamada, la información de la llamada desaparece, y el agente queda disponible para atender la siguiente llamada.

## **26.5 Resolución de problemas**

A continuación se detallan algunos problemas que podrían ocurrir durante el funcionamiento del marcador, junto con sugerencias sobre posibles soluciones a tales problemas. Para cualquier problema relacionado con el marcador, es de suma utilidad examinar el archivo de log del marcador (/opt/elastix/dialer/dialerd.log). Se puede activar la emisión de mensajes de depuración al archivo de log, a través de la interfaz Web del Elastix CallCenter-->Configuration.

### **Marcador no marca (ningún agente recibe llamadas):**

Si ningún agente de una campaña recibe llamadas, pueden haber dos escenarios principales: el que no se estén generando llamadas en primer lugar, o el que todas las

llamadas estén fallando en ser colocadas. Si ninguna llamada está siendo generada, puede ocurrir que

- a) la campaña está en estado inactivo (verificable a través del listado de campañas en la página Web)
- b) la campaña está fuera de horario. Se puede verificar esto con la opción Ver (View) del listado de campañas. Versiones viejas de la interfaz Web tenían un problema de usabilidad en el cual la fecha de inicio de la campaña y la hora de inicio de la campaña en cada día se introducían con el mismo control, así como con la fecha final y la hora de fin de actividad diaria. Resultaba frecuente que se eligiera la hora igual para inicio y fin de la banda diaria de actividad por error, lo cual conducía a que la campaña nunca se activara. Aunque la interfaz actual fue rediseñada para evitar este error, hay que tenerlo en cuenta, especialmente con campañas creadas con versiones antiguas de la interfaz.

Si todas las llamadas fallan en ser colocadas (lo cual se puede averiguar examinando el *log* detallado), se debe verificar si se pueden colocar llamadas a través de un teléfono ordinario o un *softphone*, a través de la troncal que se debe usar para la campaña. Si se puede hacer esto, se debe verificar que la troncal está correctamente especificada. Las versiones recientes del marcador corrigen un problema con troncales que requieren prefijos antes del número marcado, cuando tal prefijo se guarda en el FreePBX. Si su central usa una troncal personalizada, verifique que la plantilla de marcado especifica un \$OUTNUM\$ (todo en mayúsculas) para que el marcador sepa cómo construir la cadena de marcado.

```
2008/10/03 13:14 : (DialerProcess) DEBUG: generando llamada
Destino..... 9832855147
Cola..... 3000
Contexto.... from-internal
Trunk..... SIP/IPCOM
Plantilla... SIP/IPCOM/84162352$OUTNUM$
Caller ID... (no definido)
Cadena de marcado SIP/IPCOM/841623529832855147
```

En el ejemplo de arriba, el programa de marcado emite (con depuración activada) el *trunk* usado, la plantilla que usa para marcar, y la cadena que se usa para especificar el marcado al Asterisk.

## **Marcador sólo marca a un número pequeño de agentes (subutilización):**

La causa más frecuente de la subutilización de agentes es que se haya dejado en su valor por omisión al máximo número de canales a usar para la campaña. Por omisión se usan

3 canales como máximo para el *trunk* de la campaña. Si usted tiene más agentes, y la troncal admite una capacidad mayor, se debe aumentar el valor a un número consistente con la naturaleza de la troncal. La interfaz ISDN soporta 23 canales de voz en T1 y 30 en E1, y las troncales VoIP como IAX y SIP no tienen límite superior, excepto por ancho de banda.

Si usted usa una troncal VoIP como SIP o IAX, y luego de aumentar el máximo número de canales los agentes siguen subutilizados, es posible que las llamadas estén tardando demasiado en ser colocadas. Esto puede deberse a congestión de la red o del proveedor del otro lado de la troncal SIP. La depuración en versiones recientes del CallCenter indica cuánto tiempo tardan las llamadas en ser colocadas:

```
2008/10/02 04:32 : (DialerProcess) DEBUG: (cola 501) todavia quedan 1
llamadas pendientes de OriginateResponse!
2008/10/02 04:32 : (DialerProcess) DEBUG:          llamada 32065-14-18429
espera respuesta desde hace 28 segundos.
```

Otro motivo frecuente por el que los agentes están subutilizados es que un porcentaje muy alto de las llamadas está fallando en la colocación. Esto también se puede revisar en el *log* del marcador.

## **Marcador deja de generar llamadas luego de funcionar correctamente por un intervalo:**

Se conoce de un bug en la versión de Asterisk 1.4.22 en la cual el estado de un agente (como lo reporta el comando 'queue show') es inconsistente con el estado verdadero del agente. Específicamente, 'queue show' reporta que el agente está en uso (estado InUse) cuando realmente está libre. Las versiones de CallCenter posteriores a 1.3 contienen un rodeo que consulta también el estado según el comando 'agent show' el cual no es afectado por el bug. Si no puede obtener una versión actualizada del CallCenter, se puede usar una versión anterior de Asterisk.

## **26.6 Usando la interfase Web de administración de campañas**

Bueno, vamos ahora a explicar brevemente cómo se usa la interfase Web de administración de campañas. Esto es muy importante de conocer para el administrador del call center.

Desde esta interfase podremos administrar Agentes, Formularios, Campañas y Breaks.

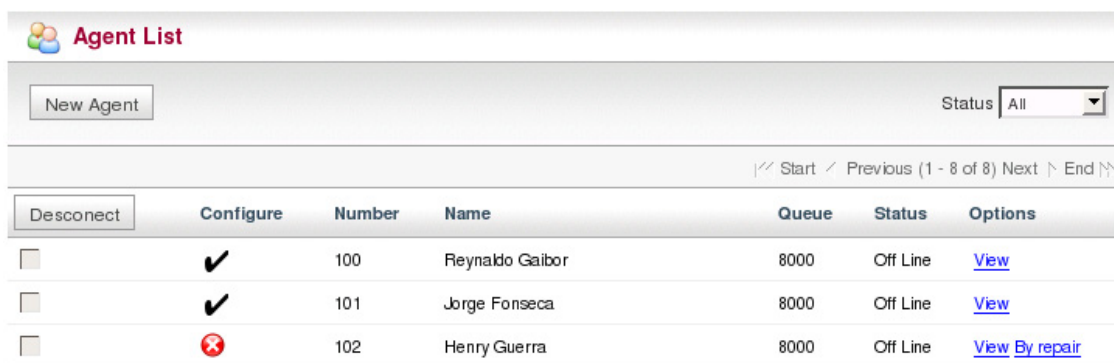
## Administración de Agentes

La administración de agentes se realiza desde el menú de Agentes. Aquí se nos permite ingresar los datos de las personas que van a operar el sistema y que se les ha denominado agentes. Cada agente debe tener un número y clave asignado, para que por medio de esta se registre y reciba o realice llamadas.

Dentro del menú de agente tenemos las siguientes pantallas o funcionalidad.

### Listado de Agentes

Muestra la lista de los agentes que han sido creados previamente.



Disconnect	Configure	Number	Name	Queue	Status	Options
<input type="checkbox"/>	✓	100	Reynaldo Gaibor	8000	Off Line	<a href="#">View</a>
<input type="checkbox"/>	✓	101	Jorge Fonseca	8000	Off Line	<a href="#">View</a>
<input type="checkbox"/>	✗	102	Henry Guerra	8000	Off Line	<a href="#">View By repair</a>

Pantall de listado de agentes

Para cada agente se muestra la siguiente información.

Nombre	Descripción
Desconectar	Desconecta a un agente, que posee la sesión activa en la consola de agente.
Configuración	<p><b>Si hay un visto:</b></p> <p>El agente se encuentra registrado en la base de datos y en el archivo de configuración de Asterisk</p> <p><b>Si hay una equis:</b></p> <p>Indica que hay un error con este agente y los errores pueden ser los siguientes:</p> <ul style="list-style-type: none"><li>• Registrado en la base de datos pero no en el archivo de configuración de Asterisk</li><li>• Registrado en el archivo de</li></ul>

	configuración de Asterisk pero no en la base de datos
Número	Número asignado al agente. Es con este número que deberá ingresar a la consola de agente.
Nombre	Nombre del Agente
Estado	Puede ser: Conectado o Desconectado
Opciones	<p><b>Ver:</b></p> <p>Muestra la información del agente, el agente podrá ser Modificado ó Eliminado.</p> <p><b>Reparar:</b></p> <p>Un agente será reparado si se encuentra en la siguientes situaciones:</p> <ol style="list-style-type: none"> <li>1. No se registró en el archivo de configuración de Asterisk, así que será tomado de la base de datos y agregado.</li> <li>2. No se registró en la Base de Datos en tal caso se elimina del archivo de configuración de Asterisk</li> </ol>

### Ingreso de Agentes

Para agregar un nuevo Agente, dar click en el botón “Nuevo Agente”. Se mostrará un formulario en cual se ingresarán los siguientes campos:

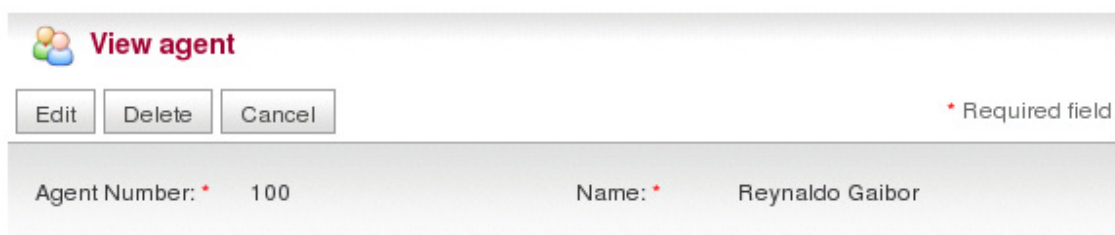
Formulario para la creación de nuevo agente

Nombre del campo	Descripción
Número de Agente	Número asignado a un agente. Tener en cuenta que este número es la

	identificación para ingresar a la interfase Web
Nombre	Nombre del Agente
Contraseña	Contraseña del Agente
Confirmar Contraseña	

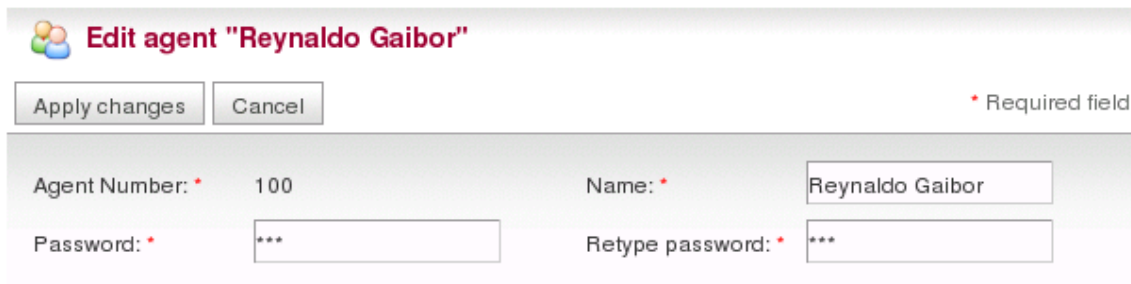
### Visualizar, Editar y Eliminar

Al dar clic sobre la palabra View se nos conducirá a una pantalla donde se mostrará los datos del agente.



Viendo los datos de un agente

Para eliminar un agente de clic en el botón Eliminar. Se puede además modificar los datos del agente en el botón Editar. Los datos que puede cambiar son: Nombre y Contraseña.



Editando un agente

## Administración de Formularios

Esta ventana nos permite la creación de formularios, los cuales se crean con el objetivo de recolectar datos al momento de ejecutar una campaña y realizar llamadas desde la consola de agente.

En este menú tenemos dos opciones Listado de Formularios y Diseño de Formularios.



## Listado de Formularios

Muestra la lista de formularios existentes, con el objetivo de hacer una vista preliminar de estos, antes de que sean usados en una campaña. Los datos que se pueden apreciar en el listado son los siguientes:



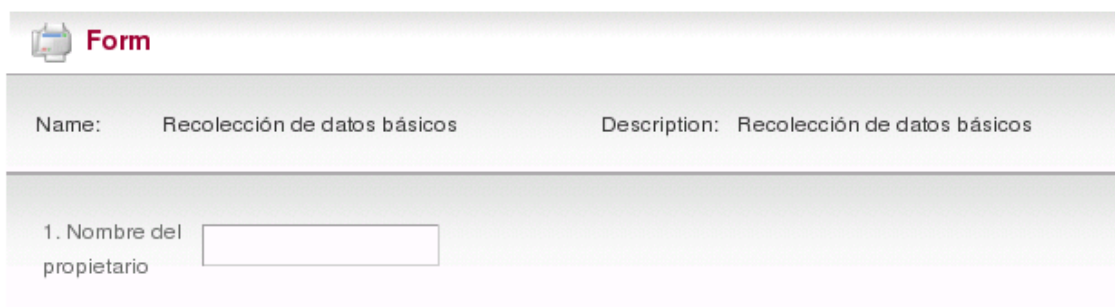
Name	Description	Status	Options
Recolección de datos básicos	Recolección de datos básicos	Active	<a href="#">Preview</a>

Listado de formularios

Nombre del campo	Descripción
Nombre	Nombre del Formulario
Descripción	Descripción del Formulario
Estado	Estado del Formulario, que puede ser Activo o Inactivo
Opciones	<b>Vista Previa:</b> Visualización de los campos que posee el formulario de forma Gráfica.

Para hacer una vista preliminar de los formularios se debe hacer clic sobre la palabra *preview*. Esto nos conducirá a una pantalla donde se mostrará: el Nombre, Descripción y los campos que contiene un formulario en forma gráfica.

Es importante tomar en cuenta que desde esta opción no se puede ingresar ningún dato.



1. Nombre del propietario

Vista preliminar del formulario

## Diseño de Formularios

Al igual que la opción anterior se muestra el listado de formularios existentes y las columnas que se muestran están explicadas en la tabla de más abajo:

The screenshot shows a web interface for 'Form Designer'. On the left is a sidebar with 'Form Designer' and 'Form List' options. The main area has a 'Create New Form' button and a dropdown menu for 'Estado' (Status) set to 'Todos'. Below this is a table with two rows of form data. Navigation links like 'Start', 'Previous', 'Next', and 'End' are visible at the top and bottom of the table.

Name	Description	Status	Options
Recolección de datos básicos	Recolección de datos básicos	Active	<a href="#">View</a>
Form 1	NN	Inactive	<a href="#">Activate</a>

Diseñador de formularios

Nombre del campo	Descripción
Nombre	Nombre del Formulario
Descripción	Descripción del Formulario
Estado	Estado del Formulario, puede ser Activo o Inactivo
Opciones	<b>Ver:</b> Visualización de los campos que posee el formulario de forma Textual. El formulario podrá ser: Editado, Desactivado, Eliminado.  <b>Activar:</b> Activa los formularios, cuyo estado es Inactivo.

Para agregar un nuevo Formulario se debe dar clic en el botón Crear Nuevo Formulario. Se mostrará una pantalla que se encuentra dividida en dos secciones. La sección superior o cabecera sirve para ingresar información general del formulario como el nombre del mismo y una descripción.

**New Form**

Save Cancel

Name: Description:

**New Field** Add Field

Field Name: Order:

Type: Type Label

Creación de nuevo formulario

La sección inferior sirve para definir los campos que conformarán el formulario. Se puede añadir un número arbitrario de campos.

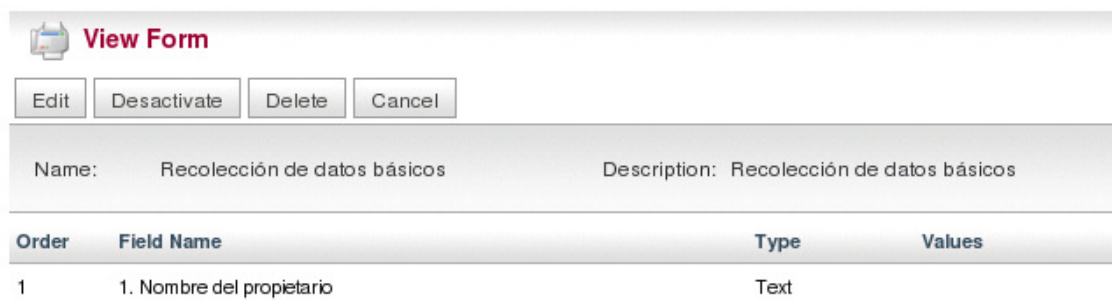
Estos campos son agregados desde la sección que dice Nuevo Campo. La información que se debe ingresar para cada campo es la siguiente.

Nombre	Descripción
Nombre del Campo	Nombre del campo
Orden	El orden con que se presentará el campo en el formulario Web
Tipo	<p>Los tipos de campos soportados hasta el momento son los siguientes :</p> <ul style="list-style-type: none"> <li>• <b>Label:</b> Etiqueta</li> <li>• <b>Text:</b> Caja de Texto</li> <li>• <b>List:</b> ComboBox</li> <li>• <b>Date:</b> Campo tipo fecha</li> <li>• <b>Text Area:</b> Ingreso de Texto extenso</li> </ul>

***Nota:** Si se selecciona el tipo List se debe indicar las posibles opciones separadas por coma en el cuadro de texto Opciones.*

### Visualizar, Editar y Eliminar formularios

Al dar clic sobre en la palabra *View*, se nos conducirá a una pantalla donde se mostrará los campos que contiene un formulario en forma textual.



**View Form**

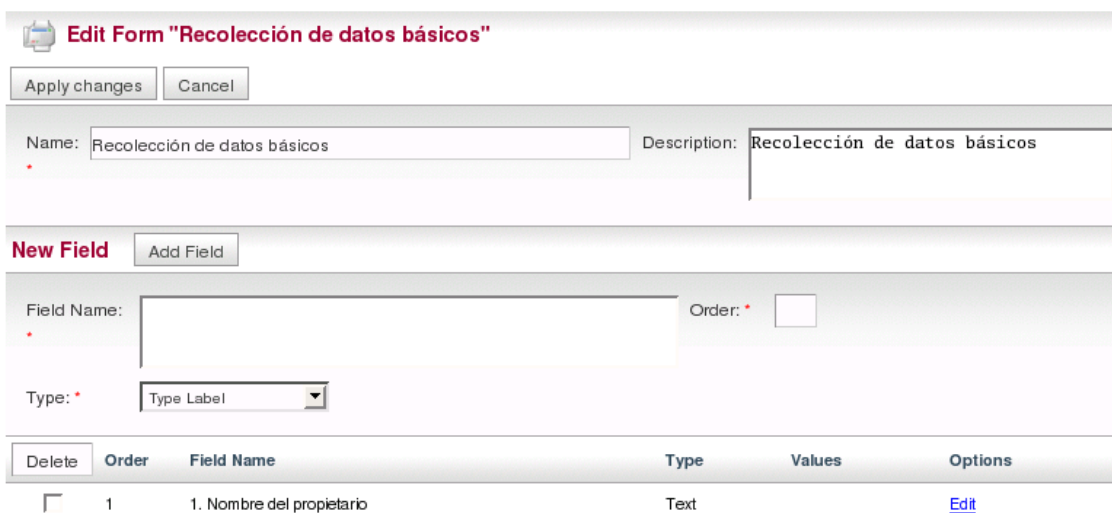
Name: Recolección de datos básicos Description: Recolección de datos básicos

Order	Field Name	Type	Values
1	1. Nombre del propietario	Text	

Visualización de formulario

Para Desactivar o Eliminar un formulario haga uso de los botones que se encuentran en la parte superior.

Puede además modificar el nombre, descripción del formulario así como también se puede añadir, editar y eliminar sus campos definidos.



**Edit Form "Recolección de datos básicos"**

Name:  Description:

**New Field**

Field Name:  Order:

Type:

Delete	Order	Field Name	Type	Values	Options
<input type="checkbox"/>	1	1. Nombre del propietario	Text		<a href="#">Edit</a>

Edición de formulario

## Administración de campañas salientes

En esta pantalla se puede crear lo que se conoce como campañas salientes, que son información de cuándo y cómo se va a generar una serie de llamadas de manera automática a números telefónicos que se deben subir en un archivo CSV desde esta misma opción.

### Listado de Campañas Salientes.

En esta opción se muestran todas las campañas salientes que se han creado en el sistema. Los datos que se visualizan en forma de columnas son:

Campaigns List

Create New Campaign

Estado 

All

Start / Previous (1 - 2 of 2) Next End

Name	Start Date	Start Time	End Date	Time end	Retries	Trunk	Queue	Completed calls	Average time	Status	Options
Campaña 1	2008-02-01	15:23:00	2008-02-29	15:23:00	2	ZAP/g0	7000			Active	<a href="#">View CSV Data</a>
Campaña 2	2008-02-05	15:26:00	2008-02-14	15:26:00	3	ZAP/g0	200			Inactive	<a href="#">Activate</a> <a href="#">View CSV Data</a>

Start / Previous (1 - 2 of 2) Next End

Listado de campañas salientes

Nombre	Descripción
Nombre	Nombre de la Campaña
Fecha Inicio	Fecha de inicio en que se va a encontrar vigente la Campaña
Hora Inicio	Hora de inicio en que se va a encontrar vigente la Campaña
Fecha Fin	Fecha de fin de la Campaña. Luego de esta fecha no se realizarán más llamadas aunque hayan quedado números telefónicos pendientes.
Hora Fin	Hora de fin de la Campaña. Luego de esta hora no se realizarán más llamadas aunque hayan quedado números telefónicos pendientes.
Intentos	Número de Intentos de las llamadas a realizar
Troncal	Salida de la llamadas de una campaña
Cola	Cola asignada para realizar la campaña
Llamadas Completadas	Número de llamadas completadas.
Promedio	Promedio de duración de las llamadas realizadas en una campaña
Estado	Estado de una campaña, puede estar Activa o Inactiva
Opciones	<b>Ver:</b> Datos de la Campaña. Además la campaña saliente podrá ser: Editada, Desactivada, Eliminada.  <b>Activar:</b>

	<p>Cuando la campaña está desactivada.</p> <p><b>Datos CSV:</b>  Datos en formato CSV de los formularios recolectados en las respectivas campañas.</p>
--	--

### Ingreso de Campañas Salientes

Desde esta pantalla se van a especificar todos los datos necesarios para la creación de una campaña. Hay que tomar en cuenta que estos datos son importantes al momento que se generan las llamadas. A continuación se describe cada uno de ellos:

Formulario de ingreso de nueva campaña

Nombre	Descripción
Nombre	Nombre de la Campaña
Fecha Inicio	Fecha de inicio en que se va a encontrar

	vigente la Campaña
Fecha Fin	Fecha de fin de la Campaña. Luego de esta fecha no se realizarán más llamadas aunque hayan quedado números telefónicos pendientes.
Formulario	Seleccionar los formularios de los cuales se va a hacer uso durante la campaña telefónica, para recolectar datos.
Troncal	Troncal por la que las llamadas telefónicas salen.
Contexto	Nombre del contexto a utilizar en una campaña. Por defecto se utiliza: <code>frominternal</code>
Cola	Cola que se ha creado en el IVR con los agentes que van a realizar las llamadas y que es asignada para realizar la campaña
Intentos	Número de veces que se reintenta ejecutar una llamada que no se ha ejecutado con éxito.
Archivo de Llamadas	Archivo formato CSV que contiene los números de teléfonos de las personas a las cuales se realizará las llamadas. A más de los teléfonos también se pueden cargar más datos.
<i>Script</i>	<i>Script</i> o Guión es un mensaje de guía para los agentes en el momento que están realizando las llamadas, es decir, que este podría servir en el momento que una llamada es realizada y el agente tiene un saludo cargado en este <i>script</i>

### Archivo de Llamadas

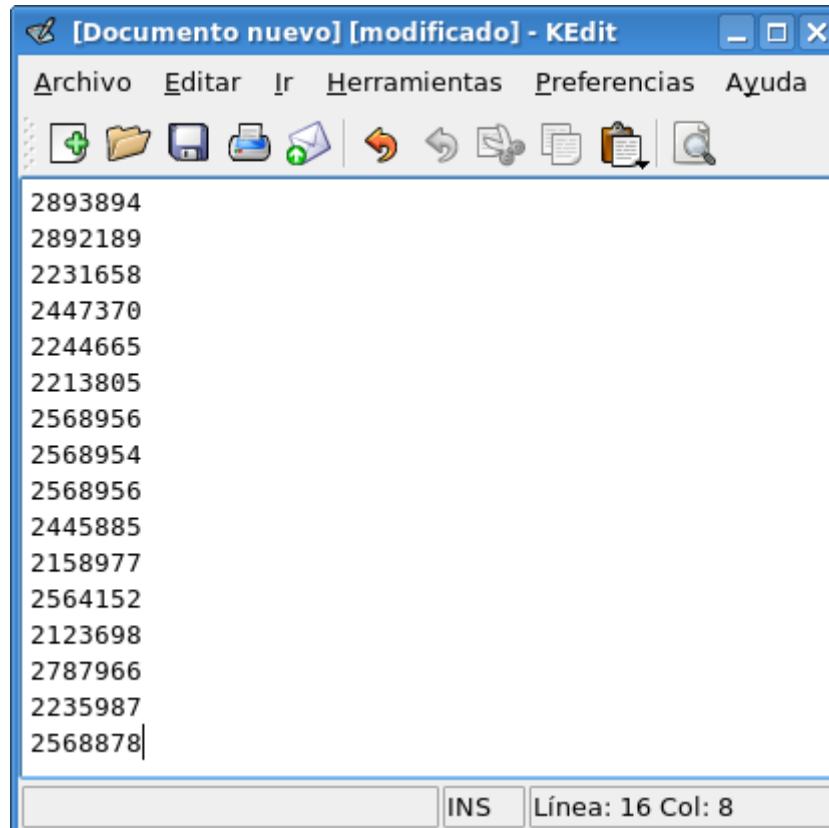
En el archivo de llamadas se colocarán los números de teléfonos, los cuales serán marcados por el call center para efectuar una campaña. También es posible especificar en el archivo más columnas que contengan datos como Nombre, Dirección, Cédula, etc. Estas columnas se pueden definir arbitrariamente y pueden ser referenciadas luego desde el guión.

Existen algunas restricciones de formato del archivo de llamadas:

- Debe estar en formato CSV
- La primera columna debe ser el número telefónico.

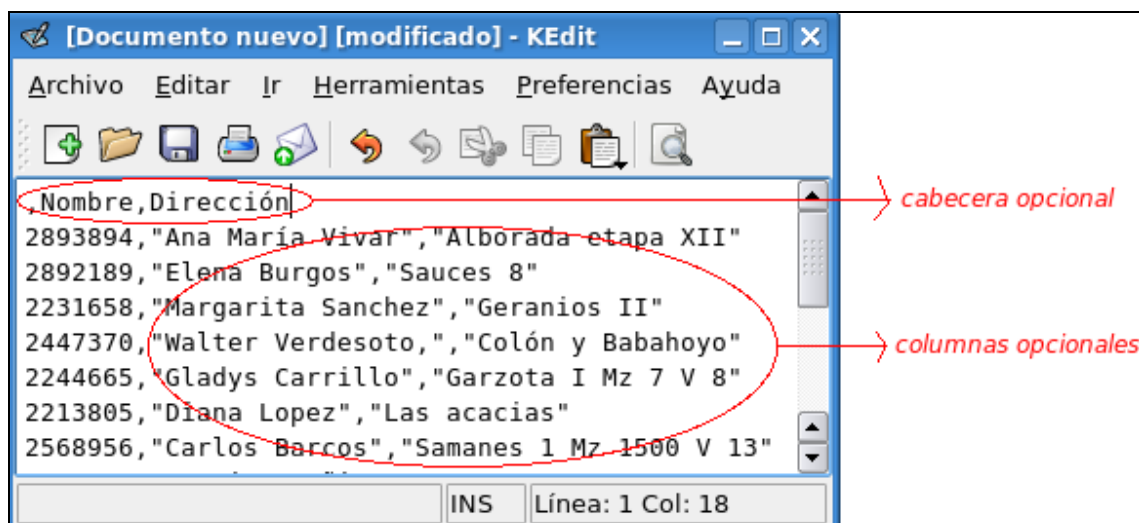
- Si se especifica en la primera fila las cabeceras, esta debe iniciar con una coma (.).

En general un archivo de llamadas puede estar constituido únicamente por los números telefónicos a los que van a llamar.



Formato básico del archivo de llamadas

Como explicamos antes también se puede incluir un número arbitrario de columnas separadas por comas conteniendo cualquier información que se necesite mostrar en el guión.





**Nota:** El número de teléfono, no tendrá cabecera alguna, los demás campos se colocarán separados por (,) comas.

### Visualizar, Editar y Eliminar campañas

Al dar clic sobre la palabra *View*, se nos conducirá a una pantalla donde se mostrará los datos de la campaña.




The screenshot shows a web interface titled "View Campaign" with a printer icon. Below the title are four buttons: "Edit", "Desactivate", "Delete", and "Cancel". The main content is a form with the following fields:

Name: *	Campaña 1
Start Date: *	01 Feb 2008 15:23
End Date: *	29 Feb 2008 15:23
Form: *	Recolección de datos básicos
Trunk: *	ZAP/g0
Context: *	from-internal
Queue: *	7000 Ventas
Retries: *	2
Script: *	Buenos Días, La Cuesta que usted.....































































Visualización de los datos de una campaña

Para editar, desactivar o eliminar una campaña haga uso de los botones que se encuentran en la parte de arriba de la información.

Los datos que puede cambiar son: Nombre, Fecha Inicial, Fecha Final, Formularios, Troncal, Contexto, Cola, Número de Intentos, *Script*.

 **Edit Campaign "Campaña 1"**

Apply changes Cancel

Name: *	Campaña 1
Start Date: *	01 Feb 2008 15:23 
End Date: *	29 Feb 2008 15:23 
Form: *	<div> <div> <div>&gt;&gt;</div> <div>&lt;&lt;</div> </div> <div>Recolección de datos básicos</div> </div>
Trunk: *	ZAP/g0
Context: *	from-internal
Queue: *	7000 Ventas
Retries: *	2
Script: *	<div> <div> <div>[Style]</div> <div>[Font]</div> <div>[Size]</div> </div> <div> <div><b>B</b></div> <div><i>I</i></div> <div><u>U</u></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> </div> <div>Buenos Dias, La Cuesta que usted.....</div> </div>


Editando una campaña

## Administración de llamadas entrantes

Aquí se nos permite asignar aquellas colas que serán destinadas para recepción de llamadas en un call center, facilitando realizar un monitoreo de la llamadas recibidas por un agente, para determinar cuales se realizaron con éxito y cuales no lograron contestación.

### Listado de Colas

Muestra la lista de las Llamadas Entrantes, con los siguientes datos.

 **Queue List**

Status All

|<< Start / Previous (1 - 2 of 2) Next >> End

Name Queue	Status	Options
8000	Active	<a href="#">View</a>
791	Inactive	<a href="#">Activate</a>

|<< Start / Previous (1 - 2 of 2) Next >> End

Listado de colas para manejar llamadas entrantes

Nombre	Descripción
Nombre de cola	Número de la Cola
Estado	Puede ser Activa o Inactiva
Opciones	<p><b>Ver:</b></p> <p>Muestra información de la cola, tal como: <i>Script</i> detallado, cuál es el mensaje observado por los agentes.</p> <p>La cola seleccionada podrá ser: Editada o Desactivada.</p> <p><b>Activar:</b></p> <p>Activa una cola posee el estado de Inactiva</p>

### Seleccionar una Cola

Para seleccionar una cola, dar clic en el botón de Seleccionar una cola. Se mostrará un formulario en cual se ingresarán los siguientes campos:

Formulario para seleccionar una cola y asociarle un gui3n

Nombre	Descripci3n
Seleccionar una cola	Selecciona de un combo una cola previamente creada
<i>Script</i>	Gui3n o mensaje visualizado por Agente

### Visualizar, Editar y Eliminar

Al dar clic sobre la palabra *View*, se nos llevar3 a una pantalla donde se mostrar3 los datos de la cola.

Queue: *	8000
Script: *	Buenos Dias Le Saluda.....

Visualizaci3n de una cola

Para eliminar una cola de clic en el bot3n Eliminar. Se puede adem3s modificar los datos de la cola clic en el bot3n Editar. El dato que puede cambiar es: el *Script*.

Edición de una cola

## Administración de *Breaks*

La opción Administrador de *Breaks* nos permite crear instantes de tiempos predefinidos en los cuales los agentes no recibirán o realizarán llamadas, un ejemplo de *Break*, es el tiempo de *lunch* de un agente, muy importante prefijar los tiempos de *break* para de esta forma poder determinar la productividad de los agentes y reconocer cual fue el tiempo en el que estuvo ausente del *call center*.

### Listado de Breaks

Muestra la lista de *Breaks*, con los siguientes datos:

Name	Description	Status	Options
Administración	Reunión sin ser planificada con Mandos Superiores	Active	<a href="#">View</a>
Lunch	Tiempo de 30 min	Active	<a href="#">View</a>
Reunión	Reunión previamente planificada	Inactive	<a href="#">Activate</a>

Pantalla de listado de breaks

Nombre	Descripción
Nombre	Nombre del Break
Descripción	Descripción del Break
Estado	Puede ser Activo o Inactivo
Opciones	<b>Ver:</b> Muestra información del <i>break</i> . El <i>break</i> podrá ser: Editado o Desactivado.  <b>Activar:</b> Activa un <i>break</i> cuyo estado es Inactivo

### Insertar nuevo *Break*

Para crear un *break*, dar clic en el botón de Crear Nuevo *Break*. Se mostrará un formulario como el de la siguiente figura.

Formulario de creación de nuevo *break*

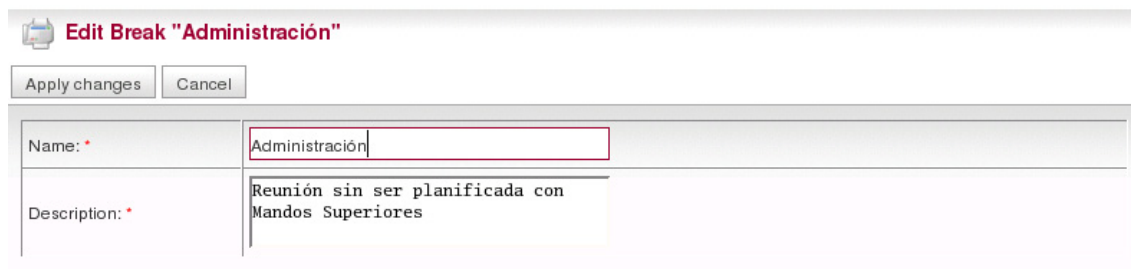
### Visualizar, Editar y Eliminar

Al dar clic sobre la palabra *View*, se nos llevará a una pantalla donde se mostrará los datos del *break*.

Visualización de los datos de un *break*

Para desactivar un *break* de clic en el botón desactivar.

Puede además modificar los datos del *break* clic en el botón Editar. Los datos que se pueden cambiar son: Nombre y Descripción.



The screenshot shows a dialog box titled "Edit Break 'Administración'". At the top, there is a small icon of a document and the title. Below the title, there are two buttons: "Apply changes" and "Cancel". The main area of the dialog is divided into two sections. The first section is labeled "Name: \*" and contains a text box with the word "Administración". The second section is labeled "Description: \*" and contains a text box with the text "Reunión sin ser planificada con Mandos Superiores".

Editando un *break*





# 27

## Extendiendo la interfase Web de Elastix

*Aunque supiera que el mundo se desintegrará mañana, igual plantaría  
mi manzano*

-- Martin Luther King

### 27.1 El *framework* Neo

El *framework* Elastix para la interfaz Web desde sus inicios ha estado en constantes mejoras tanto en estándares de desarrollo Web, codificación y arquitectura, con la finalidad de proveer la mejor interacción tanto a los usuarios y desarrolladores.

Entre las características tenemos:

- Arquitectura Web basada en MVC2
- Desarrollo POO
- Concepto e implementación modular
- Mantenimiento de lenguajes
- Mantenimiento de la ayuda embebida
- Patrones de desarrollo

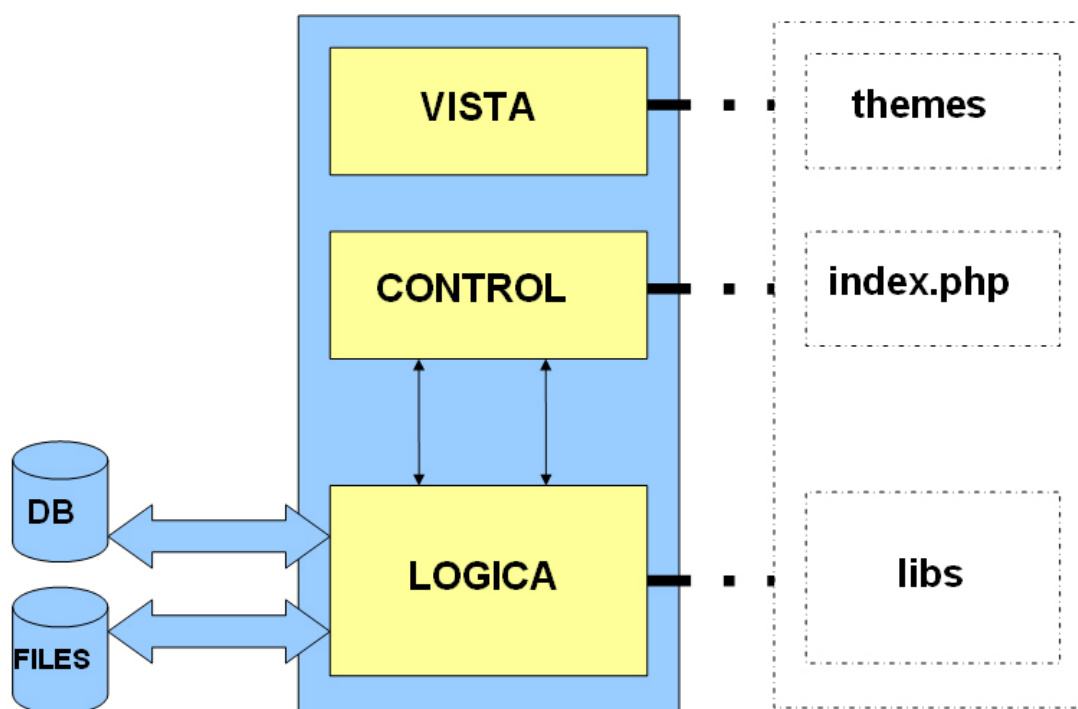
- Soporte ACLs

Para entender un poco de las premisas de diseño de la interfase de Elastix explicaremos un poco de qué se trata MVC2.

## Arquitectura Web basada en MVC2

El modelo MVC2 es una arquitectura ampliamente usada para Web, en donde “M” se refiere al modelo, “V” de vista y “C” de Control, el numero dos se refiere a la versión del mismo.

Esta arquitectura se encuentra compuesta por cuatro capas principales en donde se encapsula el concepto modular, obteniendo así tres características claves de diseño y desarrollo que son: Mantenibilidad, Modularidad y Reutilización; consiguiendo con esto un modelo escalable lo cual es sumamente útil para Web.



Framework Neo basado en arquitectura MVC2

### Capa de Vista

Esta capa esta compuesta por el código HTML (*tags*), por los estilos (CSS) y los *scripts* (como *javascript*).

El estándar recomienda que los estilos y *scripts* estén como archivos externos para que el recurso de los mismos (como archivos) se encuentren siempre en la caché del servidor.

Esto produce ahorro de espacio en disco al no repetir estilos y *scripts* por página.

***Nota:*** El estándar no acepta tablas para diseño, se deben usar capas (*layers*).

### Capa de Control

Esta capa es la encargada de recoger y entregar los datos a la capa de vista. Para esto se usan sesiones y una librería de manejo de plantillas llamada *Smarty*.

Para recoger los datos se usan los métodos GET y POST establecidos en los formularios HTML (etiquetas *form*). Otra opción de recoger datos es por sesiones.

Esta capa se encuentra mayormente constituida por archivos de clases.

### Capa de Modelo

La participación de esta capa es para realizar las transacciones y/o obtener y enviar datos a la capa de datos.

Hay que notar que aquí se ha dividido lo que es consultas (sentencias SELECT) y las inserciones, actualizaciones y eliminaciones. Esto se hace para poder hacer reportes bajo cualquier criterio.

Para las consultas no es necesario realizar *stored procedures*.

Una innovación es el concepto de *hibernate* que realiza consultas, eliminaciones, inserciones y actualizaciones bajo xml y programación orientada a objetos.

### Capa de Datos

En esta capa se encuentra la metadata y los datos de un modelo de negocios, encargada de dar soporte a múltiples usuarios concurrentes (dependiendo de la base de datos).

El esquema para MVC2 es la capa de repositorios de datos en donde se encuentran también los *stored procedures*.

Recomendación es realizar completamente bien el análisis del modelo de negocios para que la base de datos sea altamente flexible y robusta a cambios.

## Desarrollo POO

Este es uno de los mas recientes paradigmas para programación y sus siglas significan Programación Orientada a Objetos; su método de abstracción es elevada, muy organizador que la parte lógica de los desarrollos puede ser resuelto en gran parte con el diagramado de clases.

En Elastix esta parte está relacionada con los archivos dentro de las carpetas `libs` y estas pueden ser tanto para el *framework* como para un modulo.

La programación POO es usada en la Capa Modelo de MVC2.

## Implementación modular

Elastix es un software que se mantiene en constante innovación y tener el desarrollo en forma modular es la solución para esto. Actualmente un modulo de Elastix tiene aplicado en si MVC2 o sea una recursividad entre el *framework* y un modulo.

Se establece un estándar en el contenido o dentro de una carpeta:

Carpeta	Descripción
themes/	Están la definición de plantillas (html, javascript, css)
index.php	El punto de entrada común para todas las pantallas de la interface de Elastix. Es quien decide qué pantalla mostrar dependiendo de los parámetros recibidos
libs/	Aquí se encuentran las librerías (código reciclable)
configs/	Archivos de configuración
langs/	Definición de los lenguajes, actualmente se soportan 22 idiomas
images/	Imágenes que serán usadas tanto en la ayuda como en el modulo
help/	Ayuda del modulo

Podemos notar que la carpeta `themes` define la capa vista, la carpeta `libs` define la capa modelo y el archivo `index.php` define la capa control.

## 27.2 Creación de un módulo de Elastix

### Creando el Menú

Los menús en Elastix se dibujan automáticamente de la información extraída de la base de datos de menús. Por lo tanto para crear un nuevo menú debemos ingresar una nueva fila en la tabla menú de la base de datos `menu.db` en formato `sqlite3`.

```
$ sqlite3 /var/www/db/menu.db

sqlite> insert into menu values ('module_id', 'id_parent',
'link', 'module_name', 'module_type');
```

Donde,

Nombre Etiqueta	Descripción
<code>module_id</code>	Identificador del módulo. Normalmente se trata de un identificador de tipo texto.
<code>id_parent</code>	Id del módulo padre. En caso de que sea un menú de primer nivel se usa 0
<code>link</code>	Si el menú es de tipo <i>framed</i> este campo se llena con un link a la aplicación que se quiere vincular. Si el menú es de tipo <i>module</i> se deja vacío.
<code>module_name</code>	Nombre del módulo. Por ejemplo “Hardware Detector”
<code>module_type</code>	Un módulo puede ser de dos tipos: <i>framed</i> y <i>module</i> . El primero se usa cuando se quiere embeber en un <i>frame</i> otra aplicación. Por ejemplo este es el caso de VTigerCRM.

Ejemplo:

```
sqlite> insert into menu values ('example', 'extras','',
'Example', 'module');
```

Una vez creado el nuevo menú en la tabla `menu`, procedemos a hacer que éste menú sea accesible para los usuarios de Elastix.

Elastix maneja un sistema de ACLs (*Access Control List*) o Lista de Control de Acceso, el cual es un concepto que permite manejar permisos de acceso de una manera más organizada.

La información de ACLs se almacena en una base de datos SQLite llamada acl.db. Esta base de datos contiene algunas tablas que se enlistan a continuación.

```
[root@elastix ~]# sqlite3 /var/www/db/acl.db
SQLite version 3.3.6
Enter ".help" for instructions
sqlite> .tables
acl_action          acl_membership      acl_user
acl_group           acl_profile_properties  acl_user_permission
acl_group_permission  acl_resource        acl_user_profile
```

A continuación explicaremos para qué sirven las tablas más importantes.

Tabla	Descripción
acl_action	Pueden haber algunas acciones que se realicen sobre un recurso (en nuestro caso menú) como por ejemplo tener acceso de lectura, poderlo modificar, poderlo eliminar, etc. Esta tabla almacena las acciones posibles sobre un recurso.
acl_group	Los usuarios se pueden asociar en grupos para facilitar la asignación de permisos. De esta manera es más fácil modificar los permisos de un grupo de usuarios en lugar de estar cambiando permisos a cada usuario.
acl_group_permission	Matriz de permisos de un grupo dado. Básicamente aquí se almacenan qué acciones puede realizar un determinado grupo sobre un determinado recurso.
acl_membership	Esta tabla almacena la relación usuario vs grupo. Es decir que nos dice qué usuarios pertenecen a cada grupo.
acl_resource	En el caso de Elastix un recurso significa un menú. Un recurso es la entidad que es objeto de los permisos.
acl_user	Aquí se almacenan los usuarios de Elastix. Por omisión se encuentra aquí el usuario admin, caso contrario no se podría ingresar a la interfase de Elastix por primera vez.

acl_user_permission	Esta tabla contiene la matriz de permisos para cada usuario
---------------------	---

Veamos qué contiene por ejemplo la tabla `acl_resource` y veremos algunos menús conocidos allí. La salida de la tabla ha sido truncada pues no es necesario mostrar toda la salida.

```
[root@elastix ~]# sqlite3 /var/www/db/acl.db
SQLite version 3.3.6
Enter ".help" for instructions
sqlite> select * from acl_resource;
1|sysinfo|System Info
2|network|Network
3|network_parameters|Network Parameters
4|dhcp_server|DHCP Server
5|usermgr|User Management
6|grouplist|Group List
7|userlist|User List
8|group_permission|Group Permission
9|load_module|Load Module
10|shutdown|Shutdown
11|hardware_detection|Hardware Detection
12|updates|Updates
13|packages|Packages
14|repositories|Repositories
15|backup_restore|Backup/Restore
16|preferences|Preferences
17|language|Language
```

Ahora ingresemos nuestro menú en la tabla `acl_resource`.

```
$ sqlite3 /var/www/db/acl.db

sqlite> insert into acl_resource (name, description) values
('example', 'Example');
```

Si revisamos la interfaz Web en este punto ya veremos que un nuevo menú sale enlistado aunque sin nombre. Este es nuestro menú. Sale sin nombre por ahora puesto que no hemos modificado los archivos de lenguaje todavía pero al menos ya podemos asignarle permisos de manera más cómoda desde la interfase Web en lugar de manipular las tablas directamente.

Vayamos a *System --> User Management --> Group Permission*

Resource	Apply
Billing Report	<input checked="" type="checkbox"/>
Destination Distribution	<input checked="" type="checkbox"/>
Billing Setup	<input checked="" type="checkbox"/>
vTigerCRM	<input checked="" type="checkbox"/>
Calling Cards	<input checked="" type="checkbox"/>
Downloads	<input checked="" type="checkbox"/>
Softphones	<input checked="" type="checkbox"/>
Fax Utilities	<input checked="" type="checkbox"/>
Instant Messaging	<input checked="" type="checkbox"/>
SugarCRM	<input checked="" type="checkbox"/>
Nuevo módulo	<input checked="" type="checkbox"/>

Nuevo módulo sale enlistado al final de la lista de recursos

En la figura anterior se observa que la hemos seleccionado para el grupo administrador. Es decir que los usuarios de ese grupo ahora podrán tener acceso al nuevo menú.

Ahora definamos la cuestión de los lenguajes para poder ver el nombre de nuestro menú en la pantalla anterior.

## Definición de lenguajes

Los lenguajes en Elastix se definen en la carpeta `/var/www/html/lang`

Esta carpeta contiene algunos archivos, uno por cada lenguaje. Las dos primeras letras del nombre del archivo identifican el lenguaje. Por ejemplo, el archivo `en.lang` se usa para el idioma inglés (*english*) mientras que el archivo `es.lang` se usa para el idioma español.

Estos archivos son archivos de texto que podemos editar con el editor `vim` por ejemplo. Sin embargo es importante notar que se encuentran en lenguaje PHP por lo que



debemos tener cuidado de no eliminar caracteres especiales como el punto y coma o los paréntesis.

Bueno, editemos el archivo en.lang para incluir nuestro nuevo menú. Si su Elastix se encuentra configurado en otro idioma debe editar el archivo correspondiente.

Vaya al final de este archivo y antes del símbolo que cierra el arreglo “);” agregue el nombre de su módulo.

En este caso agregamos “Example” => “Example”, //nombre del módulo.

Grabe el archivo y listo, ya podremos ver el nombre de nuestro módulo en el listado anterior.

## Visualizando el nuevo módulo

Ahora en System --> User Management --> Group Permission, verá que el nombre de su modulo se visualiza, si no está marcado, márquelo y aplique los cambios.

System Info

Network

User Management

Load Module

Shutdown

Hardware Detection

Updates

Groups

Users

Group Permission

Group Permission

Group: administrator

Show

Apply

	Resource
<input checked="" type="checkbox"/>	Billing Report
<input checked="" type="checkbox"/>	Destination Distribution
<input checked="" type="checkbox"/>	Billing Setup
<input checked="" type="checkbox"/>	vTigerCRM
<input checked="" type="checkbox"/>	Calling Cards
<input checked="" type="checkbox"/>	Downloads
<input checked="" type="checkbox"/>	Softphones
<input checked="" type="checkbox"/>	Fax Utilities
<input checked="" type="checkbox"/>	Instant Messaging
<input checked="" type="checkbox"/>	SugarCRM
<input checked="" type="checkbox"/>	Example

Listado de permisos de grupo. Ahora si se muestra el nombre del nuevo módulo

Visitemos ahora nuestro nuevo módulo que se encuentra dentro del menú de Extras.

Veremos que al hacer clic en él, se presentará el siguiente error:

*Error: the module modules/examples/index.php could not be found.*



Visitando por primera vez nuestra nueva entrada en el menú

Para solucionar esto veamos cómo llenar nuestro módulo con código útil.

## 27.3 Programación de un Módulo

Bueno, ya hemos creado los permisos para poder acceder al nuevo módulo y aparece ya dibujado dentro del menú de Extras. Ahora comencemos a codificar algo para que nuestro módulo sea útil.

Cada módulo reside dentro de una carpeta dentro de la ruta `/var/www/html/modules/` y el nombre de esta carpeta tiene que ser igual al `module_id` que se ingresó en la tabla `menu`. En nuestro caso el id era “example” por lo que creamos la carpeta de la siguiente manera.

```
$ mkdir -p /var/www/html/modules/example/
```

Es de esta manera como el *framework* Neo ubica al módulo en el sistema de archivos en base a la información que encuentra en la base de datos de menús.

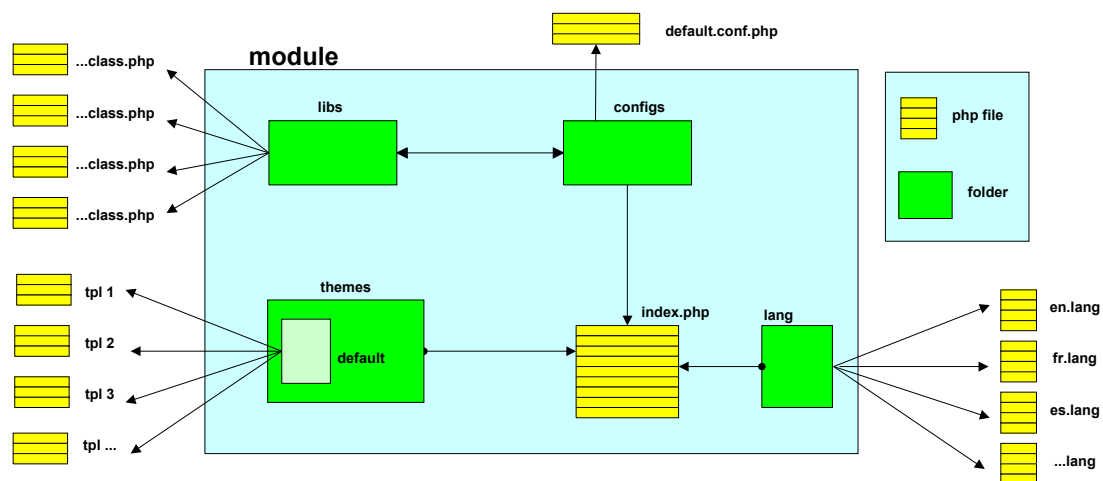
Para facilitar las cosas solicite a su instructor un archivo `tar.gz` que contiene un módulo de ejemplo el cual deberá copiar a la carpeta recién creada.

Es importante mantener los permisos de la carpeta creada bajo el usuario “asterisk” ya que por el momento el servidor Web de Elastix se ejecuta bajo este usuario

```
$ chown -R asterisk:asterisk /var/www/html/example/
```

## 27.4 Arquitectura Modular

A continuación un esquema general de los componentes de un módulo y cómo interactúan entre sí. Vemos que el archivo `index.php` es el director de orquesta. Explicaremos más de este archivo más adelante.



Componentes de un módulo de Elastix

### Archivo `index.php`

El archivo `index.php` dentro de la carpeta de un módulo es el punto de entrada al módulo. Es decir que es lo primero que el *framework* tratará de ejecutar para invocar al módulo.

Este archivo básicamente contiene una función llamada `_moduleContent`. Esta función debe retornar el contenido HTML a ser dibujado por el módulo.

Un ejemplo sencillo de módulo “hola mundo” sería el siguiente:

```
<?php
function _moduleContent($smarty, $module_name)
{
    return "hola mundo";
?>
```

## 27.5 “Elastix Developer Tools” al rescate

“Elastix Developer Tools” no es otra cosa que un módulo de Elastix que contiene una serie de herramientas, que entre otras cosas, nos permiten construir módulos. Suena confuso al principio verdad?

Bueno, en todo caso es una herramienta muy útil a la hora de construir un nuevo módulo pues nos permite crear el esqueleto inicial de código necesario para elaborar un módulo de Elastix, lo que puede constituir un ahorro de horas en muchos casos.

El módulo “Elastix Developer Tools”, también llamado simplemente **Developer**, se encuentra dividido en tres submódulos. Ha sido desarrollado para realizar acciones de creación de módulos, eliminación de módulos y mantenimiento de lenguajes. Su uso está dirigido a desarrolladores que tienen un conocimiento básico del *framework* Elastix (Neo), lenguaje de programación PHP y bases de datos MySQL y SQLite.

El uso de estas herramientas representan un ahorro en tiempo de codificación y nos otorga transparencia en procesos rutinarios al empezar a desarrollar un módulo para la GUI de Elastix.

### Instalando Elastix Developer Tools

La instalación de este módulo no es diferente a la de cualquier otro módulo de Elastix. Es decir que descargamos el módulo vía Internet y lo instalamos a través del menú “Sistema -> Cargar Módulo”.

Inmediatamente después de su instalación veremos un nuevo menú llamado “Developer”.

Este módulo *Developer* puede ser descargado del sitio de descargas de Elastix y se encuentra bajo el nombre de “Developer”. A la fecha de redacción de este capítulo, la versión más reciente se encuentra en la siguiente ruta:

[http://downloads.sourceforge.net/elastix/developer\\_1.0-2beta.tar.gz](http://downloads.sourceforge.net/elastix/developer_1.0-2beta.tar.gz)

## Construyendo un módulo de Elastix con el *Developer*

Comencemos dando clic en el primer submódulo del *Developer*, llamado el constructor de módulos.

Este es el principal componente del *Developer* y su función es generar el esqueleto de código del nuevo módulo que estamos creando, para luego continuar con el desarrollo de la lógica.

Al crear un modulo desde aquí, se generarán lo siguiente:

- El menú (o los menús) y su correspondiente información de ubicación dentro del árbol de menús de la interfase Web de Elastix
- El código esqueleto, el cual se encontrará dentro de una carpeta cuyo nombre será igual al ID del módulo. Esta carpeta será ubicada a su vez dentro de la carpeta `/var/www/html/modules/` que es donde residen todos los módulos de Elastix.
- Pantallas prototipo. Estas pantallas pueden ser de tres tipos: reporte (o grid), formulario y *framed*. Ya hablaremos de estos tres tipos de pantalla más adelante. Lo importante de conocer ahora es que esto facilita enormemente las tareas de codificación de formularios o reportes simples.

Veamos ahora cómo luce el constructor de módulos.

**Build Module**

Save \* Required field

**General Information**

Module Name: \*  Your Name: \*  Group Permission: administrator  
Module Id: \*  Your e-mail: \*  Operator  
extension

**Location**

Module Level: \* Level 2  
Level 1 Parent Exists: \* No  
Level 1 Parent Name:  Level 1 Parent Id:

**Module Description**

Module Type: \* Form Field Name:  >>   
Type Field: Type Text <<

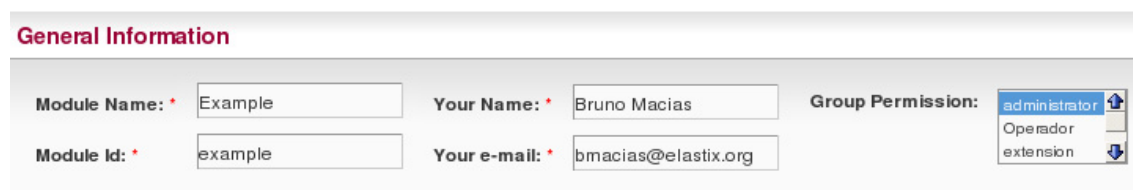
El constructor de módulos que forma parte de “Elastix Developer Tools”

Como podemos observar, el constructor de módulos se encuentra esta seccionado en tres partes:

- Información General
- Localización
- Descripción del módulo

### Información General

Esta sección nos permite definir el nombre e identificador único del módulo, así como también los datos de la persona que crea el modulo para comentarlas en las cabeceras de cada archivo de programación en PH. También nos permite configurar el nivel de accesabilidad que tendrá el módulo a través del parámetro “Group Permission”



**General Information**

Module Name: \* Example      Your Name: \* Bruno Macias      Group Permission: administrator

Module Id: \* example      Your e-mail: \* bmacias@elastix.org      Operador

extension

Sección de Información general dentro del constructor de módulos

Nombre campo	Descripción
Module Name	Nombre del Módulo. La primera letra de cada palabra con mayúscula.
Module Id	Identificador del Módulo. Debe ser en minúsculas y no se permiten espacios.
Group Permission	Esta opción permite elegir el grado de accesibilidad de los usuarios.
Your Name	El nombre de quien hizo el módulo.
Your e-mail	Dirección de correo electrónico de quien hizo el módulo.

### Localización

En esta sección se configura el lugar, dentro del menú, donde vamos a ubicar el nuevo módulo. Elastix maneja tres niveles de menús por ende debemos tener un poco de cuidado en el ingreso de los datos.

## Location

Module Level: *	<input type="text" value="Level 3"/>		
Level 1 Parent Exists: *	<input type="text" value="No"/>		
Level 1 Parent Name: *	<input type="text"/>	Level 1 Parent Id: *	<input type="text"/>
Level 2 Parent Name: *	<input type="text"/>	Level 2 Parent Id: *	<input type="text"/>

Sección donde se configura la ubicación del módulo

Nombre campo	Descripción
Module Level	Aquí definimos el nivel de jerarquía del modulo. Este puede ser 2 ó 3 ya que el nivel 1 simplemente es de etiquetamiento. Para clarificar, el nivel 1 es el conformado por las pestañas Sistema, PBX, Fax, Email, IM, Reportes, Extras, Agenda, etc.
Level 1 Parent Exists	<p>Aquí definimos si vamos a colocar el módulo bajo un menú de nivel 1 existente o se tendrá que crear uno nuevo. Para colocar el nuevo módulo bajo un menú previamente existente escogemos “<b>Sí</b>”.</p> <p>Si escogemos “<b>No</b>” crearemos un nuevo menú de nivel 1 sobre la marcha. Por esta razón aparecerán dos campos adicionales para especificar el nombre y el ID.</p>
Level 1 Parent Name	Definimos el nombre del "Modulo Padre". La primera letra con letra mayúscula.
Level 1 Parent Id	Definimos el identificador del "Modulo Padre". Todo con minúsculas y no se permiten espacios.
Level 2 Parent Exists	<p>Esta opción aparecerá si escogimos un nivel 3 en el campo “Module Level” ya que obviamente en este caso debemos especificar el menú padre de nivel 2.</p> <p>De manera similar a lo que ocurre con el campo “Level 1 Parent Exists”, si se escoge No aparecerán dos campos más.</p>
Level 2 Parent Name	Definimos el nombre del "Modulo Padre" de nivel 2. La primera letra con letra mayúscula.
Level 2 Parent Id	Definimos el identificador del "Modulo Padre" de nivel 2. Todo con minúsculas y no se permiten espacios.

A continuación algunas figuras de diversas disposiciones que puede tomar esta sección dependiendo de las opciones que vayamos eligiendo.

**Location**

**Module Level: \*** Level 2 ▾

**Level 1 Parent Exists: \*** Yes ▾

**Level 1 Parent: \*** System ▾

- System
- PBX
- Fax
- Email
- IM
- Reports
- Extras
- Agenda
- Developer
- Call Center
- Firewall

Creación de un módulo en un menú de segundo nivel con padre de nivel 1 existente.

**Location**

**Module Level: \*** Level 2 ▾

**Level 1 Parent Exists: \*** No ▾

**Level 1 Parent Name: \*** Parent 1

**Level 1 Parent Id: \*** parent\_1

Creación de un módulo en un menú de segundo nivel con padre no existente.



**Location**

<b>Module Level: *</b>	Level 3		
<b>Level 1 Parent Exists: *</b>	Yes	<b>Level 2 Parent Exists: *</b>	Yes
<b>Level 1 Parent: *</b>	System		
<b>Level 2 Parent: *</b>	<div>System Info</div> <div>System Info</div> <div>Network</div> <div>User Management</div> <div>Load Module</div> <div>Shutdown</div> <div>Hardware Detection</div> <div>Updates</div> <div>Backup/Restore</div> <div>Preferences</div> <div>Notas</div>		

Creación de un módulo en un menú de tercer nivel, tanto los padres de nivel 1 y nivel 2 existen.

**Location**

<b>Module Level: *</b>	Level 3		
<b>Level 1 Parent Exists: *</b>	Yes	<b>Level 2 Parent Exists: *</b>	No
<b>Level 1 Parent: *</b>	System		
<b>Level 2 Parent Name: *</b>	Parent 2	<b>Level 2 Parent Id: *</b>	parent_2

Creación de un módulo en un menú de tercer nivel. El padre de nivel 1 existe y el padre de nivel 2 no.

### Descripción del módulo

Esta última sección es la más interesante pues es aquí donde se crea la pantalla del módulo.

Como mencionamos antes, esta pantalla puede ser de tres tipos: reporte, formulario o *framed*.

- **Reporte:** Un reporte muestra datos simplemente. Estos datos se los puede visualizar como una tabla donde se incluyen automáticamente algunos controles útiles como botones para navegación. Luego de generar el código necesario con el constructor de módulos ya es más sencillo enlazar el reporte con una base de datos para mostrar información útil.

- **Formulario:** Sirve para recolectar datos del usuario. El constructor de módulos cuenta con soporte para los tipos de campos más comunes como Text, Select, Date, Textarea, Checkbox, Radio, Password, Hidden y File.
- **Framed:** Embebe un URL arbitrario en la pantalla. Muy útil cuando queremos integrar aplicaciones externas en la interfase Web de Elastix. Un ejemplo de esto es la integración de vtigerCRM.

Creación de una pantalla tipo Formulario

Con los botones ">>", "<<" podemos añadir o quitar los campos a insertar en nuestro formulario.

## Eliminación de un módulo a través del *Developer*

El submenú "Delete Module" del módulo "Developer" nos permite eliminar módulos en todos sus niveles. Primero se tendrán que eliminar los submódulos que existan en el módulo padre para poder eliminarlo por completo.

Eliminación de un módulo

**Nota:** Es muy importante estar seguro antes de seleccionar la opción "Delete Files" pues esto eliminará todos los archivos relacionados con el módulo en cuestión del disco duro.

## Administración de lenguajes desde el “Developer”

El submódulo "Language Admin" del módulo "Developer" nos permite administrar los diferentes idiomas de los módulos en Elastix.

Aquí podemos consultar si existe un lenguaje definido para un módulo, también para el *framework* y observar una lista de palabras con su respectiva traducción como lo podemos observar en la figura de abajo. Estos valores pueden ser modificados de acuerdo al idioma que desee traducir y guardar los cambios presionando el botón "Save all".

Clave	Value
System Resources	System Resources
CPU Info	CPU Info
Uptime	Uptime
CPU usage	CPU usage
Memory usage	Memory usage
Swap usage	Swap usage
Hard Drives	Hard Drives
Partition Name	Partition Name
Capacity	Capacity

Administración de lenguajes desde el módulo *Developer*

## 27.6 Estándar de codificación PHP en Elastix

Para mantener el código legible es necesario establecer un estándar. De esta manera los desarrolladores de Elastix no tendrán dificultad de entender el código de nuevos desarrolladores entusiastas en colaborar.

## Indentación

Usar una indentación de 4 espacios, “sin tabs”.

## Estructuras de control

Aquí se incluyen `if`, `for`, `while`, `switch`, etc. A continuación se muestra un ejemplo de la sentencia `if`:

```
<?php
if ((condicion1) || (condicion2)) {
    accion1;
} elseif ((condicion3) && (condicion4)) {
    accion2;
} else {
    accionpordefecto;
}
?>
```

Las sentencias de control deberían tener un espacio entre la sentencia misma y el paréntesis abierto, para distinguirlas de las llamadas a funciones.

El uso de llaves es estricto y obligatorio aun en situaciones donde ellas son técnicamente opcionales. El hecho de usarlas incrementa la legibilidad y disminuye probabilidad de errores de lógica introducidos cuando nuevas líneas son añadidas. Para la sentencia `switch` por ejemplo:

```

<?php
switch (condicion) {
case 1:
    accion1;
    break;

case 2:
    accion2;
    break;

default:
    accionpordefecto;
    break;

}
?>

```

## Llamadas a funciones

Las funciones deberían ser llamadas sin espacios entre el nombre de la función, el paréntesis de apertura y el primer parámetro; espacios entre comas y cada siguiente parámetro, y no espacio entre el último parámetro, el paréntesis de cierre y el punto y coma. A continuación se muestra un ejemplo:

```

<?php
$var = foo($bar, $baz, $quux);
?>

```

Tal como se muestra arriba, debería haber un espacio a ambos lados del símbolo de igual usado para asignar a una variable el valor retornado por la función. En el caso de un bloque de asignaciones relacionadas, más espacio puede ser insertado para promover la legibilidad:

```

<?php
$short           = foo($bar);
$long_variable = foo($baz);
?>

```

## Definición de funciones

La declaración de funciones sigue la convención “one true brace”:

```
<?php
function fooFunction($arg1, $arg2 = '')
{
    if (condicion) {
        statement;
    }
    return $val;
}
?>
```

Argumentos con valores por defecto van al final de la lista de argumentos. Siempre se debe intentar retornar un valor significativo de una función si existe algún valor apropiado. Aquí se muestra un ejemplo algo más elaborado que el anterior:

```
<?php
function connect(&$dsn, $persistent = false)
{
    if (is_array($dsn)) {
        $dsninfo = &$dsn;
    } else {
        $dsninfo = DB::parseDSN($dsn);
    }

    if (!$dsninfo || !$dsninfo['phptype']) {
        return $this->raiseError();
    }

    return true;
}
?>
```

## Comentarios

La documentación de clases debería seguir la convención PHPDoc, similar a Javadoc. Más información acerca de PHPDoc puede ser encontrada aquí:

<http://www.phpdoc.de>

Comentarios de No-documentación son permitidos y gratamente bienvenidos. Una regla general es que si revisas una sección de código y piensas “Uuuy, no tengo ganas de describir este alboroto ahora”, tu necesitas comentarlo antes de olvidar cómo funciona. Comentarios estilo C (`/* */`) y estandar C++ (`//`) están bien. No es bienvenido el uso de comentarios estilo Perl/shell (`#`).

## Código de inclusión

En cualquier parte del código en la que te encuentres incluyendo INCONDICIONALMENTE un archivo de clase, usa `require_once()` y en cualquier parte del código en la que te encuentres incluyendo CONDICIONALMENTE un archivo de clase (por ejemplo, el método de factoría), usa `include_once()`.

Cualquiera de estos dos asegurará que los archivos de clase sean incluidos solo una vez. Ellos comparten la misma lista de archivos, así que tu no necesitas preocuparte acerca de mezclarlos – un archivo incluido con `require_once()` no será incluido nuevamente por `include_once()`.

Nota: `include_once()` y `require_once()` son sentencias, no funciones. Tu no necesitas paréntesis alrededor de un nombre de archivo para ser incluido.

Etiquetas PHP

Siempre usa `<?php ?>` para delimitar código PHP, no uses la etiqueta corta `<? ?>`. Esto es requerido debido que hace el código más portable entre diferentes configuraciones y sistemas operativos.

## Comentarios de cabecera

Todos los archivos fuente concebidos por el grupo de desarrollo de PaloSanto Solutions deben contener el siguiente bloque de comentarios en la cabecera (excepto probablemente los archivos de configuración modificables por el usuario):

```

/* vim: set expandtab tabstop=4 softtabstop=4 shiftwidth=4:
Codificación: UTF-8
+-----+
| Elastix version 1.3                                     |
| http://www.elastix.org                                 |
+-----+
| Copyright (c) 2008 Palosanto Solutions S. A.           |
+-----+
| Cdla. Nueva Kennedy Calle E 222 y 9na. Este           |
| Telfs. 2283-268, 2294-440, 2284-356                   |
| Guayaquil - Ecuador                                   |
| http://www.palosanto.com                              |
+-----+
| The contents of this file are subject to the General Public License |
| (GPL) Version 2 (the "License"); you may not use this file except in |
| compliance with the License. You may obtain a copy of the License at |
| http://www.opensource.org/licenses/gpl-license.php      |
|                                                         |
| Software distributed under the License is distributed on an "AS IS" |
| basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See |
| the License for the specific language governing rights and |
| limitations under the License.                               |
+-----+
| The Original Code is: Elastix Open Source.             |
| The Initial Developer of the Original Code is PaloSanto Solutions |
+-----+
$Id$ */

```

No hay una regla fija para determinar cuándo un nuevo codificador debería añadir su nombre a la lista de autores de un archivo fuente. En general, sus cambios deberían caer en la categoría de “cambios sustanciales” (significando aproximadamente de 10% a 20% de cambios en el código). Excepciones podrían ser hechas al reescribir funciones o contribuciones de nueva lógica trascendente.

Simple reorganización de código o correcciones de bugs no deberían justificar la adición de un nuevo nombre a la lista de autores.

Archivos que no hayan sido concebidos por PaloSanto Solutions deberían tener un bloque similar mostrando sus licencias, copyright y autores. Todos los archivos deberían incluir los comentarios de “modo de línea” para forzar la consistencia.

## URL de ejemplo

Use “example.com” para todos los URLs de ejemplo, tal como lo sugerido por RFC 2606.



## Convención de nombres

Generalmente hablando, los nombres de clases, funciones y variables deberían ser siempre lo suficientemente descriptivos para que el lector entienda fácilmente para que están siendo usadas (estas clases, funciones o variables) en el código.

### Clases

Las clases deben tener nombres descriptivos. Evita usar abreviaciones siempre que sea posible. Los nombres de clases deben siempre comenzar con una letra mayúscula. La jerarquía debería estar también reflejada en el nombre de la clase, cada nivel de la jerarquía separado con un underscore.

Ejemplos de buenos nombres de clases se muestran a continuación:

- Log
- Net\_Finger
- HTML\_Upload\_Error

### Funciones y métodos

Las funciones y métodos deberían ser nombradas usando el estilo “Minúsculas-Mayúsculas” (también conocido como “studly caps”, “bumpy case” o “camel caps”). Las funciones deberían adicionalmente tener el nombre del paquete como prefijo, para evitar colisiones de nombres entre paquetes. La letra inicial del nombre es minúscula, y cada letra que comienza una nueva palabra es debe ser mayúscula. Algunos ejemplos son:

- Connect()
- getData()
- buildSomeWidget()
- XML\_RPC\_serializeData()

Miembros privados de una clase y propiedades (significando miembros de clase y propiedades que están creadas para ser usadas sólo desde dentro de la misma clase en la cual fueron declaradas; PHP aún no soporta funciones o métodos privados) deben estar precedidas por un underscore. Por ejemplo:

- \_sort()
- \_initTree()
- \$this->\_status

## Constantes

Nombres de constantes deben siempre estar escritos totalmente en mayúsculas, con underscores para separar las palabras. Constantes dentro de una clase (esto es sólo una sugerencia) pueden tener como prefijo el nombre de la clase o paquete. Por ejemplo, las constantes usadas por el paquete DB:: comienzan todas con “DB\_”.

## Variables Globales

Si tu paquete necesita definir variables globales, su nombre debería comenzar con un underscore seguido del nombre del paquete y otro underscore. Por ejemplo:

```
$ _PEAR_destructor_object_list.
```

## Valores predefinidos true, false, null

Los valores predefinidos de PHP true, false y null deben ser escritos en minúscula.

# 28

## Virtualización

*El hombre nunca debe avergonzarse de reconocer que se ha equivocado,  
puesto que hacerlo corresponde a decir que hoy sabe más que lo que  
sabía ayer*

-- Alexander Pope

Existen situaciones donde resulta interesante virtualizar Elastix. Una de ellas, que se ha vuelto algo popular en los últimos tiempos, es la de permitir a los proveedores de servicios de centros de datos el incluir un producto más a su lista: Hosted PBX.

Sin duda resulta más rentable ofrecer este servicio si se puede virtualizar Elastix en lugar de disponer de un servidor por cliente.

Sin embargo, esta no es la única situación interesante. Existen algunos otros escenarios donde el cliente puede encontrar productivo dividir un mismo hardware entre varias máquinas virtuales, entre ellas Elastix.

### 28.1 Opciones de virtualización

En la actualidad existen algunas opciones de virtualización en el mercado, pero nos limitaremos a explicar las siguientes.

- VMWare
- Virtualbox
- XEN

## 28.2 Elastix sobre VMWare

Para facilitar las cosas el proyecto Elastix distribuye de cuando en cuando una imagen para VMWare que se puede descargar del sitio Web oficial del proyecto.

La instalación de Elastix sobre VMWare es sumamente sencilla y no necesita que profundicemos en el tema.

Para más información sobre VMWare visite <http://www.vmware.com>

## 28.3 Elastix sobre Virtualbox

Virtualbox es una herramienta de virtualización que permite ejecutar diversos sistemas operativos en máquinas virtuales. De esta manera podríamos utilizar para correr una máquina de pruebas con Elastix o un servidor virtual con Elastix.

Existen 2 versiones de Virtualbox que podemos utilizar par virtualizar Elastix:

- La versión libre que viene incorporada en la mayoría de distribuciones Linux
- La no libre pero que se puede descargar sin costo de su página Web y funciona también para algunas plataformas, inclusive para Windows y Mac OS

La principal diferencia entre la versión libre y la no libre es el soporte para dispositivos USB en las máquinas virtuales. Esto es algo que no vamos a necesitar para nuestros propósitos por lo que la versión libre funcionará de excelente manera.

Por ejemplo en caso de utilizar una distribución basada en Debian como por ejemplo Ubuntu se debe ejecutar el siguiente comando:

```
sudo aptitude install virtualbox-ose virtualbox-ose-modules-  
`uname -r`
```

Para el caso de Windows o Mac OS o distribuciones Linux que no tengan VirtualBox en sus repositorios se puede descargar el instalador desde <http://virtualbox.org> y seguir el *wizard* para instalación

## Virtualbox y Redes

Antes de poder utilizar Virtualbox para crear nuestra máquina virtual debemos configurar la forma en la que Virtualbox manejará las redes. Virtualbox puede manejar hasta cuatro interfaces de redes virtuales por cada máquina virtual. Cada una de estas tarjetas de redes puede tener una de las siguientes configuraciones:

- **No Conectado:** en este caso la máquina virtual tiene una tarjeta de red pero funcionará como si no tiene un cable de red conectado. Esta modalidad no será útil para instalar Elastix ya que Elastix no tiene mayor uso si no esta conectado a una red.
- **NAT:** Esta es la configuración por defecto y permite que nuestra máquina virtual pueda ver nuestra red o salir a Internet, pero no puede ser vista por otras máquinas en la red. Esta función tampoco nos sirve para Elastix ya que necesitamos que nuestra máquina virtual sea vista por toda la red
- **Red Local:** La configuración de red local crea una red virtual entre nuestras máquinas virtuales. De esta forma las máquinas virtuales solo podrán ver a máquinas virtuales dentro de esta red virtual. Esto es útil si se quiere poner a servidores virtuales atrás de un *firewall* virtual. En el caso de Elastix se podría utilizar pero para este caso particular queremos que nuestra máquina virtual sea una más en la red.
- **Interfase Anfitrión:** Esta opción permite que nuestra máquina virtual sea vista por toda la red como si fuera una máquina real. Esta es la opción que utilizaremos para instalar la máquina virtual de Elastix.

*Nota: La máquina virtual puede tener hasta 4 interfaces virtuales por lo que se podría tener una interfase de red interna y otra interfase anfitrión en una sola máquina virtual.*

## Configuración Interfase Anfitrión

### Configuración de Red Sistema Operativo MS Windows

Para crear una interfase de red virtual en Windows es necesario ejecutar el siguiente comando:

```
VBoxManage createhostif "VM1 external"
```

## Configuración de Red en Sistema Operativo Linux

Para configurar la tarjeta de red de nuestra máquina virtual debemos crear un *bridge* a nivel del sistema operativo anfitrión. Este *bridge* se podría decir que será nuestro switch virtual a donde se conectarán las máquinas virtuales.

Para crear el *bridge* se requiere instalar el paquete *bridge* útil. Para distribuciones basadas en Debian se debe ejecutar el siguiente comando para instalar:

```
sudo aptitude install bridge-utils
```

Ahora podremos crear nuestro *bridge* y utilizaremos la interfase *eth0* para esto.

1. Configuramos la interfase *eth0* para que funcione en modo promiscuo

```
sudo ifconfig eth0 0.0.0.0 promisc
```

2. Creamos el *bridge* y añadimos la interfase *eth0* al mismo

```
sudo brctl addbr br0  
sudo brctl addif br0 eth0
```

3. Asignamos el IP de nuestra máquina al *bridge* y ya no será utilizado más por la interfase *eth0*

Con IP dinámico

```
sudo dhclient br0
```

Con ip estático

```
sudo ifconfig br0 <mi ip> netmask <mascara de red>  
sudo route add default gw <ip de mi gateway>
```

Con esto hemos creado el *bridge* y nos hace falta crear las interfaces virtuales para las instalaciones de Elastix. Para esto vamos a utilizar el utilitario que trae virtualbox

```
sudo VBoxAddIF vbox0 rafael br0
```

Con este comando hemos creado la interfase vbox0 que ha sido añadida al *brige* br0. Si quisiéramos añadir más interfaces ejecutamos el mismo comando pero cambiamos vbox0 por vbox1, vbox2, etc.

### Mantener la configuración de manera permanente

De lo antes visto, lo único que se conserva de manera permanente es la creación de la interfase vbox0, la configuración del *bridge* y la tarjeta de red se perderán por lo que sería muy útil crear un *script* para esto que contenga la siguiente información

```
#!/bin/bash
sudo ifconfig eth0 0.0.0.0 promisc
sudo brctl addbr br0
sudo brctl addif br0 eth0
```

Para que el *script* sea accesible desde cualquier ruta es recomendable guardarlo en /usr/local/sbin y darle permisos de ejecución. Si vamos a crear máquinas virtuales cada vez que arrancamos nuestro PC podríamos pensar en ejecutar este *script* de manera automática cada vez que iniciamos nuestro computador.

## Crear la máquina virtual

Tenemos todo el ambiente listo para crear nuestra máquina virtual y poder ejecutarla. Para hacer la instalación se puede usar un cdrom de Elastix o simplemente la imagen iso. En este caso utilizaremos la imagen .iso.

Ejecutamos Virtualbox y seleccionamos la opción de crear una nueva máquina virtual. Aquí usted deberá contestar algunas preguntas.

En la primera ventana se pedirá el nombre de la máquina virtual y el sistema operativo que se instalará en este. Ponga el nombre que usted desee y seleccione Red Hat como sistema operativo.

En la segunda ventana se debe seleccionar la cantidad de memoria RAM que nuestra máquina virtual utilizará. Para motivos de aprendizaje seleccionaremos 256 Mb pero si estuviéramos creando un servidor que utilizaremos deberíamos seleccionar más memoria dependiendo de las necesidades.

En la tercera ventana se nos solicita seleccionar el disco duro. Lo más probable es que se tenga que crear un disco duro virtual.

Al crear el disco duro virtual se nos dará 2 posibilidades. Utilizar una imagen de tamaño fijo o utilizar una de tamaño dinámico. La diferencia es que la de tamaño fijo ocupará todo el espacio asignado en la máquina real mientras que la dinámica ira creciendo en el tiempo. En este caso seleccionaremos tamaño fijo.

Se asigna el nombre del disco duro y el tamaño deseado. Para fines didácticos 4 GB puede ser un buen tamaño, pero para producción necesitaremos mucho más. Damos aceptar y hemos creado el disco duro virtual para nuestra máquina.

Con esta información ya estamos listos para terminar la creación de nuestra máquina virtual.

## Configuración Previa de la Máquina Virtual

Antes de encender nuestra máquina virtual, debemos decir a nuestro a Virtualbox que configuración de red utilizar. Para esto seleccionamos la máquina virtual creada y seleccionamos configuración. En la ventana de configuración se debe seleccionar la configuración red.

Una vez en la ventana de configuración de red configuraremos el adaptador 1 de red para que funcione como interfase de anfitrión. Para esto se deben modificar 2 cosas:

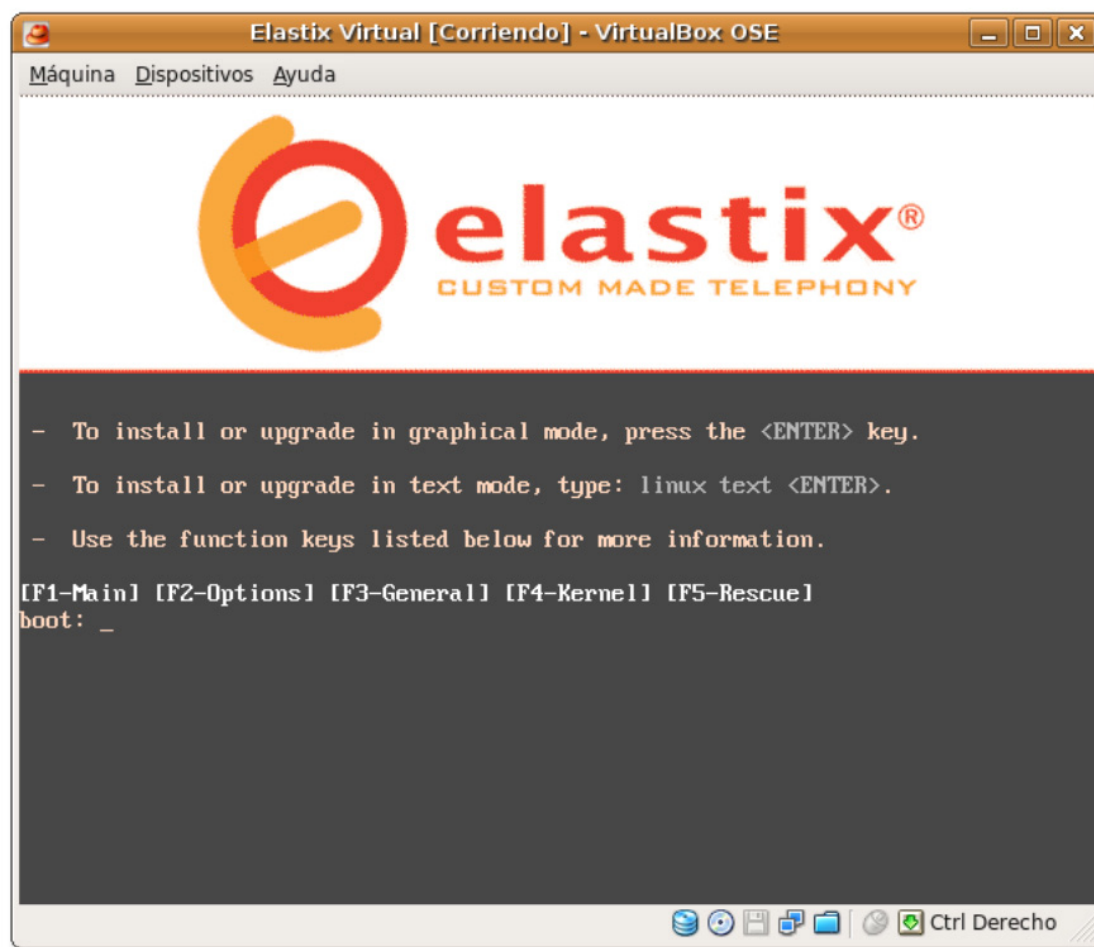
- Donde dice “*attached to*” se debe seleccionar Interfase Anfitrión.
- Donde dice “*Interface Name*” debemos escribir `vbox0` en el caso de Linux o seleccionar de la lista desplegable la interfase en el caso de Windows.

***Nota:** pusimos el nombre `vbox0` ya que anteriormente creamos esta interfase de red. Si creáramos más máquinas virtuales deberíamos crear `vbox1`, `vbox3`, etc.*

## Arrancar la Máquina Virtual por Primera Vez

La primera vez que arrancamos la máquina virtual se solicitará seleccionar el medio de arranque. En en el caso de este ejemplo se utilizará una imagen iso guardada en el disco duro. Para esto seleccionamos utilizar cdrom y utilizar archivo de imagen iso. Se selecciona el archivo y damos siguiente. A continuación seguiremos el procedimiento normal de instalación de Elastix.





Instalación de Elastix en VirtualBox

Una vez terminada la instalación podremos acceder a la interfase web administrativa de Elastix desde cualquier pc en nuestra red, añadir teléfonos ips, conectar troncales SIP o IAX, etc.... como si fuera una máquina real. Lo único que no podremos hacer es acceder a las tarjetas de red pci.

## Opciones Avanzadas

### Clonar Máquinas Virtuales

Una de las ventajas de la virtualización es que se puede clonar un disco duro de una máquina virtual y tener un respaldo completo de todo el sistema operativo. Si alguna vez la máquina que esta en producción deja de funcionar bastaría con reiniciar nuestra máquina virtual. Otro caso útil sería si tenemos un servidor pre-configurado que necesitamos replicar en varias instalaciones. Entonces se puede clonar varias veces esta máquina virtual.

Los discos duros de las máquinas virtuales de VirtualBox se almacenan en archivos con extensión .vdi. En el caso particular de Linux estos archivos se guardan en la siguiente ruta:

/home/<usuario>/VirtualBox/VDI

Donde <usuario> debe ser remplazado con el nombre de mi usuario.

Para clonar una imagen de disco duro se debe apagar la máquina virtual y ejecutar el siguiente comando en la ruta donde se encuentran los archivos VDI:

```
VBoxManage clonevdi Elastix_Virtual.vdi Elastix_Virtual-clone.vdi
```

En este ejemplo se ha clonado la imagen Elastix\_Virtual.vdi al archivo Elastix\_Virtual-clone.vdi. Si queremos utilizar el nuevo disco duro bastara con crear una nueva máquina virtual y seleccionar este archivo como disco duro.

### Arrancar Máquina Virtual sin Interfase Gráfica

Podría darse el caso de que necesitemos iniciar VirtualBox sin Interfase Gráfico, esto puede ser útil si se desea tener servidores cuya única función es la de correr máquinas virtuales por lo que no necesitan una interfase gráfica. Al ejecutar Elastix sin interfase gráfica solo será accesible a través de la interfase Web o ssh. Existen casos en lo que esto suele ser muy útil para compañías de *hosting* que venden servidores virtuales de Voz/IP con Elastix.

Para arrancar la máquina virtual sin interfase gráfica basta con ejecutar el siguiente comando:

```
VBoxHeadless --startvm "Elastix Virtual"
```

Donde “Elastix Virtual” es el nombre que hemos puesto a nuestra máquina virtual, si deseáramos arrancar una o varias máquinas virtuales al encender nuestro servidor, bastaría con ejecutar este comando cuando la computadora arranca.

## 28.4 Elastix sobre XEN

Desde mi punto de vista esta es la manera donde más provecho le podemos sacar al hardware de un servidor y por lo tanto ha sido la opción más popular para grandes

proyectos. Sin embargo, también es la alternativa que más dolores de cabeza nos puede ocasionar y requiere cierto nivel de conocimientos técnicos por lo que para proyectos pequeños muchas veces se elige otra opción.

## Breve introducción a XEN

Con XEN existen dos tipos de virtualización: virtualización completa (*full*) y paravirtualización.

La diferencia radica que en un esquema totalmente virtualizado se requiere un CPU con soporte nativo para virtualización. Por lo tanto, si queremos usar este esquema debemos revisar por las especificaciones de nuestro CPU. Si es Intel debemos revisar si su arquitectura tiene soporte VT y si es AMD el soporte se llama AMD-V.

En cualquier caso no es difícil conseguir en la actualidad procesadores con soporte nativo para virtualización.

En un esquema paravirtualizado, debido a que el CPU no nos ayuda, se debe usar una versión modificada del sistema operativo “Invitado” (*Guest*).

Algunas ventajas que XEN nos provee son:

- Gran desempeño
- Migraciones en vivo. Es decir que no se requiere apagar equipos y la interrupción del servicio es casi imperceptible.
- Es de licencia libre. Se puede descargar y usar sin costo.

### Arquitectura de XEN

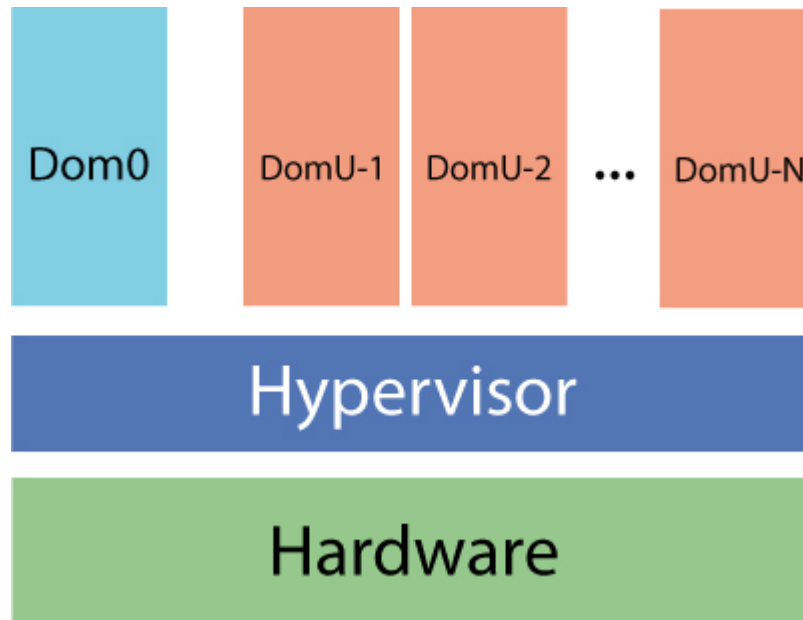
XEN se comunica directamente con el hardware de nuestro computador (CPU) a través de un componente llamado hipervisor (*hypervisor*). El hipervisor es quien coordina las acciones entre las máquinas virtuales (también llamadas dominios) y el CPU.

Cuando XEN arranca se inicia el hipervisor y automáticamente se arranca el primer dominio llamado “dominio cero” o simplemente Dom0. Este es un dominio especial de carácter administrativo. Es decir que si queremos iniciar o administrar otras máquinas virtuales (otros dominios) debemos acceder a Dom0 primero.

Dom0 por tanto es un dominio Privilegiado ya que desde allí se pueden administrar a los otros dominios.

Los demás dominios que vayamos creando reciben el nombre de DomU, donde la letra U viene de la palabra *Unprivileged* (no privilegiado, en inglés).

Explicaremos mejor estos conceptos con la siguiente figura.



Arquitectura de XEN

## Algunos comandos XEN básicos

Aunque existen productos que permiten administrar servidores virtuales XEN desde interfaz gráfica, la manera de hacerlo por omisión en Elastix es desde la línea de comandos.

Comando	Descripción
<code>xm list</code>	Muestra un listado con las maquinas virtuales que se están ejecutando en ese momento
<code>xm create &lt;maquina_virtual&gt;</code>	Inicia una maquina virtual. Uno puede confundirse y pensar que este comando crea una nueva máquina virtual
<code>xm shutdown &lt;maquina_virtual&gt;</code>	Apaga una maquina virtual determinada de manera controlada
<code>xm reboot &lt;maquina_virtual&gt;</code>	Reinicia una máquina virtual determinada de manera controlada
<code>xm destroy &lt;maquina_virtual&gt;</code>	Apaga una maquina virtual a la fuerza

**Nota:** En la anterior tabla de comandos la cadena `maquina_virtual` puede ser el identificador numérico o el nombre de la maquina virtual

### Ejemplo de salida del comando `xm list`

```
[root@virtual ~]# xm list
```

Name	ID	Mem (MiB)	VCPUs	State	Time (s)
Domain-0	0	386	2	r-----	78773.9
AntiSpam1-Centos51	5	920	2	-b-----	13953.1
Srv01-Centos51-TEST_234567	7	512	1	-b-----	797689.1
Srv02-Elastix-1.4beta_678	10	192	1	-b-----	25246.6
Srv03-Centos51-Test2	11	1024	1	-b-----	85855.3
Srv04-Fedora-Core9-test	12	256	1	-----	2106.7

## Instalación de Elastix virtualizado utilizando XEN

En este apartado vamos a explorar la instalación de uno o más servidores Elastix, funcionando bajo un esquema virtualizado, sobre un mismo hardware. De esta manera podemos optimizar los recursos al máximo.

**Nota:** Vamos a suponer para efectos de los ejemplos aquí mostrados que no necesitamos troncales Zap (o DAHDI).

### Configuración del servidor *Host*

Para comenzar debemos configurar el sistema operativo del servidor que servirá de *Host*. Por razones de desempeño y estabilidad es preferible que este sistema operativo base contenga un mínimo número de aplicaciones instaladas así que hagamos la instalación más simple posible.

Podemos utilizar CentOS o inclusive el mismo CD de Elastix para esta instalación. Lo importante es, como ya hemos mencionado, mantener los paquetes mínimos necesarios para su funcionamiento.

Sobre esta instalación base debemos instalar el soporte Xen a través de los siguientes comandos:

```
# yum install xen
# yum install kernel-xen
```

una vez finalizado esto debemos levantar el hypervisor con el comando:

```
# service xend start
```

En este punto el servidor *Host* se encuentra listo para recibir a nuestro invitado Elastix.

### Configuración del disco duro virtual

Antes de instalar al Elastix invitado vamos a crearle un disco duro virtual. Esto no es otra cosa que una imagen de disco duro contenida en un solo archivo. Como ejemplo vamos a suponer que necesitamos un disco duro de 15Gb.

El comando para realizar esta tarea en el servidor *Host* es el siguiente.

```
# dd if=/dev/zero of=<archivo_salida> bs=1M count=15360
```

Debemos reemplazar <archivo\_salida> por el nombre que le queramos dar al archivo. Por lo general estos archivos de disco duro se ubican en la ruta `/xen/`

### Instalación de Elastix como invitado

En este punto ya tenemos todo listo para que el invitado se instale sin problemas en el *Host* que hemos creado.

Para instalar no lo haremos con un CD como se instala normalmente un servidor Elastix sino que en este caso lo haremos a través de una unidad de CDROM virtual. Entonces, basta con copiar la imagen ISO del instalador de Elastix en algún lugar del disco duro del *Host* en lugar de quemarla físicamente en un CD.

Luego de copiar el ISO en el disco duro creamos un archivo de configuración correspondiente a la máquina virtual que vamos a instalar. Este archivo por lo general se copia en la ruta `/etc/xen/`

```

name = "<Nombre del domU>"
maxmem = 512
vcpus = 1
builder = "hvm"
kernel = "/usr/lib/xen/boot/hvmloader"
boot = "c"
pae = 1
acpi = 1
apic = 1
on_poweroff = "destroy"
on_reboot = "restart"
on_crash = "restart"
device_model = "/usr/lib64/xen/bin/qemu-dm"
extra = "3"
sdl = 0
vnc = 1
vncunused = 1
vncdisplay = "-5900"
disk = [ 'file:<ruta a disco duro>,hda,w' , 'file:<ruta a imagen
iso>,hdc:cdrom,r' ]
vif = [ "mac=<mac address>,bridge=xenbr0,type=ioemu" ]
serial = "pty"

```

Como podemos observar en el archivo anterior existen algunas variables que reemplazar como por ejemplo <ruta a imagen iso> que debe ser reemplazada por la imagen ISO de Elastix. Podemos guardar este archivo como /etc/xen/elastix\_ejemplo.cfg

Una vez modificado el archivo de configuración de la máquina virtual de acuerdo a nuestras necesidades, procedemos a crear la máquina virtual con el siguiente comando.

```
# xm create -c /etc/xen/elastix_ejemplo.cfg
```

Esto nos dará una consola, ya que especificamos la opción “c”, y a través de esta consola podemos comenzar el proceso de instalación de nuestra máquina virtual. Las pantallas que irán apareciendo lucirán como en una instalación normal de Elastix.





# 29

## Compilando Asterisk

*La vida no es la que uno vivió, sino la que uno recuerda*

-- Gabriel García Márquez

Son muchas las situaciones en las que es útil recompilar Asterisk. Por ejemplo si se quiere parchar el código para agregar funcionalidad o si se quiere optimizar Asterisk para eliminar funcionalidad no necesaria y mejorar el rendimiento.

Como en cualquier compilación de un paquete de software complejo esta es una tarea que requiere relativo conocimiento para resolver cualquier posible problema que se pueda presentar. Aquí trataremos de explicar brevemente los conocimientos básicos.

### 29.1 Recompilando los SRPMs

La manera relativamente más fácil de recompilar Asterisk si se está usando Elastix es utilizando los SRPMs provistos por el proyecto Elastix. Estos se pueden descargar de <http://repo.elastix.org/centos/5/updates/SRPMS/repodata/>

Un SRPM (o *Source RPM*) es un paquete que contiene la información necesaria para construir un RPM como el código fuente del paquete y las directrices usadas para compilarlo e instalarlo en el sistema.

Estas directrices se especifican en un archivo llamado SPEC, que usualmente se distingue por su extensión \*.spec

No explicaremos aquí todos los detalles relacionados con la construcción de RPMs pero explicaremos los pasos básicos a continuación.

## **Instalar un Elastix con los requisitos para construir el RPM**

Este es el primer paso. También uno pudiese utilizar un CentOS existente y adecuarlo para la construcción del RPM de Asterisk.

En cualquier caso vamos a necesitar algunos RPMs necesarios para poder construir el RPM de Asterisk a partir del SRPM.

### Requerimientos

Según el último archivo `asterisk.spec` los siguientes RPMs son requeridos para construir el RPM de Asterisk. Por lo tanto asegurémonos que tenemos esos paquetes instalados antes de empezar.

- gcc-c++
- m4
- openssl-devel
- newt-devel
- libtermcap-devel, ncurses-devel
- postgresql-devel, zlib-devel
- libpri-devel >= 1.4.0
- zaptel-devel >= 1.4.0, libtiff-devel >= 3.5.7
- libogg-devel, libvorbis-devel
- unixODBC-devel, libtool, sqlite-devel
- pwlib-devel > 1.10.5-2, opal-devel
- alsa-lib-devel
- gsm-devel
- net-snmp-devel
- spandsp-devel
- curl-devel, wget
- speex-devel >= 1.2

Algunos de ellos ya vienen instalados en una instalación de Elastix normal, otros habrá que instalarlos. Se pueden instalar a través del comando Yum.

## Instalar el SRPM de Asterisk y modificar el SPEC de acuerdo a nuestras necesidades

Luego de descargar el SRPM de Asterisk lo instalamos en la máquina concebida para este fin. Esto se hace de manera sencilla con el comando rpm.

```
[root@elastix ~]# rpm -ivh asterisk-1.4*.src.rpm
```

Esta instalación no instalará Asterisk como pudiera parecer sino que más bien poblará el directorio `/usr/src/redhat` que es donde se copiará el código fuente de Asterisk, el archivo SPEC y cualquier otra información nacida del SRPM.

A continuación deberemos hacer los cambios en el archivo SPEC para personalizar nuestro Asterisk. Aquí podremos hacer casi cualquier cosa como por ejemplo:

- Eliminar parches para conseguir un Asterisk mínimo
- Añadir funcionalidad nueva a Asterisk
- Parchar contra *bugs*
- Actualizar la versión de Asterisk

En este libro no explicaremos el formato del archivo SPEC y cómo se encuentra este dividido. Si el lector se encuentra interesado le recomendamos el siguiente vínculo <http://www.rpm.org/max-rpm/>

***Nota:** Es muy importante que la versión del RPM que generemos siempre sea mayor que la que tenemos instalada en el sistema donde queremos instalar dicho RPM puesto que si el RPM generado posee una versión menor o igual se rehusará a instalarse.*

## Construir el RPM de Asterisk

Una vez realizados los cambios en el SPEC nos encontramos listos para recompilar Asterisk o siendo exactos construir un nuevo RPM para Asterisk. Para lograr esto nos valemos del comando `rpmbuild`, pasándole como parámetro la ruta del archivo SPEC que hemos modificado.

```
[root@elastix ~]# rpmbuild -ba /usr/src/redhat/SPECS/asterisk.spec
```

Al final del proceso obtendremos los siguientes RPMs en la carpeta `/usr/src/redhat/RPMS/`

- asterisk
- asterisk-devel
- asterisk-sounds-es
- asterisk-sounds-fr

Además obtendremos un nuevo SRPM en la carpeta `/usr/src/redhat/SRPMS/`

# 30

## La Certificación Elastix

*Aprender sin pensar es esfuerzo perdido; pensar sin aprender,  
peligroso.*

-- Confusio

### 30.1 Introducción

Con la difusión de Elastix como plataforma de comunicaciones y su aceptación y uso en empresas alrededor del mundo, surgió el negocio de soporte técnico relacionado con dicha distribución. Hoy en día muchas empresas o personas brindan soporte relacionado con Elastix a empresas o terceros, e inclusive casa adentro; y junto con esto creció el interés de validar esos conocimientos técnicos de manera oficial.

Con el fin de proveer esta necesaria validación nació el Programa de Certificación Elastix, el mismo que tiene como eje central el Examen de Certificación Elastix. Este examen certifica a la persona que lo aprueba que tiene los conocimientos necesarios para implementar soluciones basadas en Elastix así como también resolver problemas técnicos relacionados con la *distro*.

### 30.2 El Curso de Certificación Elastix

El Curso de Certificación Elastix no es otra cosa que un programa de entrenamiento oficial de preparación para quien esté interesado en rendir el Examen de Certificación Elastix.

La capacitación dura 40 horas y es impartida durante 5 días, al final de los cuales el alumno debería estar apto para rendir el Examen de Certificación.

## 30.3 El Examen de Certificación Elastix

El Examen de Certificación Elastix es un examen objetivo, es su mayoría de alternativas múltiples, que tiene como objetivo calificar la idoneidad de un profesional Elastix.

El Examen se rinde a través de un sistema especializado y únicamente puede ser tomado en un Testing Center Elastix oficial.

Su duración es de 2 horas y contiene 100 preguntas.

### Tópicos y Pesos

Se han dividido los tópicos en tres grandes grupos. Cada grupo tiene un peso equivalente a la tercera parte del peso total. Dentro de cada grupo se enlistan los tópicos del examen ordenados de mayor a menor peso.

#### Linux, VoIP y Redes: Peso de 33.33%

- Introducción a la VoIP
- Protocolos SIP y RTP
- Introducción a la telefonía
- Calidad de voz
- Linux para administradores de Elastix
- Configuración de Hardware
- Configuración de teléfono IP
- Virtualización

#### Asterisk: Peso de 33.33%

- Fundamentos de Asterisk
- Plan de marcado avanzado

- Diagnóstico de problemas
- AMI y AGI
- Debugging
- Seguridad
- Desempeño
- DUNDi
- Compilación de Asterisk
- Sintetización y reconocimiento de voz

#### Elastix: Peso de 33.33%

- Configuración Básica de Elastix (Extensiones, colas, IVR, conferencias, FOP)
- Ruteo de llamadas
- funcionamiento interno de freePBX
- Interconexión con Elastix remotos
- Elastix Overview
- Fax
- Instalación de Elastix
- Tarifación y Reportación en Elastix
- Email System
- Módulo Call Center
- Mensajería Instantánea
- Extendiendo Elastix

## 30.4 Preguntas típicas de examen

A continuación algunas preguntas típicas de examen.

**Para qué se utiliza el cable MOLEX de suministro de poder en algunas tarjetas analógicas?**

- a) No es necesario conectar las tarjetas a un cable de poder adicional. La tarjeta ya se alimenta de poder a través del bus PCI.
- b) Para suministrar energía eléctrica adicional a los módulos FXS
- c) Para suministrar energía eléctrica adicional a los módulos FXO
- d) Esto sólo es necesario en antiguas tarjetas ISA
- e) Ninguna de las anteriores

**En qué carpeta se almacenan los voicemails?**

- a) /usr/lib/asterisk/voicemail
- b) /var/lib/asterisk/voicemails
- c) /var/spool/asterisk/voicemail
- d) En ninguna carpeta, los voicemails se almacenan en una base de datos interna de Asterisk para facilitar su acceso
- e) Ninguna de las anteriores

**Cual es el rango de la escala MOS?**

- a) 1-5
- b) 1-10
- c) 0-100%
- d) Empieza en el 0 pero no tiene tope
- e) Ninguna de las anteriores

**A través de que comando inicializamos el marcador predictivo?**

- a) service elastix start
- b) service dialer start
- c) service dialerelastix start
- d) service elastixdialer start
- e) Ninguna de las anteriores

**En qué se diferencian el módulo “PBX -> Conferencias” que provee directamente Elastix del módulo “PBX -> PBX Configuration -> Conferences” provisto a través de freePBX?**

- a) No existen dos módulos para administrar conferencias, sólo uno
- b) Existen dos módulos pero en realidad son la misma cosa, ya que el módulo provisto por Elastix es solo un *wrapper* del módulo de freePBX
- c) El módulo de freePBX administra cuartos de conferencia permanentes mientras que el provisto por Elastix administra cuartos temporales
- d) Los dos hacen prácticamente lo mismo pero son dos implementaciones diferentes
- e) Ninguna de las anteriores

**Cuáles contextos se encuentran reservados en el plan de marcado de Asterisk? Puede seleccionar más de uno.**



- a) globals
- b) general
- c) default
- d) reserved
- e) Todos los anteriores

**Cual de las siguientes NO es una estrategia de colas válida?. Puede elegir más de una respuesta.**

- a) ringall
- b) roundrobin
- c) leastrecent
- d) random
- e) leastactive

**Cuál es la extensión de los archivos de temas que se encuentran dentro de la carpeta “themes” y que sirven para dar el aspecto o *look* a la interfase Web de Elastix?**

- a) .ini
- b) .tpl
- c) .htm
- d) .conf
- e) Ninguna de las anteriores





# GNU Free Documentation License

*Version 1.3, 3 November 2008*

*Copyright (C) 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.*

*<<http://fsf.org/>>*

*Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.*

## 0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals

providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

## **1. APPLICABILITY AND DEFINITIONS**

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been

arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

## **2. VERBATIM COPYING**

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

### **3. COPYING IN QUANTITY**

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

### **4. MODIFICATIONS**

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at

your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

## **5. COMBINING DOCUMENTS**

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".



## **6. COLLECTIONS OF DOCUMENTS**

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

## **7. AGGREGATION WITH INDEPENDENT WORKS**

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

## **8. TRANSLATION**

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

## **9. TERMINATION**

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

## **10. FUTURE REVISIONS OF THIS LICENSE**

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

## **11. RELICENSING**

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

