



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN

INFORME DE MATERIA DE GRADUACIÓN

**“IMPLEMENTACIÓN DE UN SISTEMA DE VIDEO
VIGILANCIA UTILIZANDO UNA WEB CAM,
ASTERISK, MOTION Y CHAN MOBILE”**

Previa a la obtención del Título de:

**INGENIERO EN ELECTRÓNICA Y
TELECOMUNICACIONES**

Presentada por:

RODRIGO DANIEL GUAYAQUIL LOOR

JOSÉ LUIS SILVA PALMA

GUAYAQUIL – ECUADOR

2010

AGRADECIMIENTO

RODRIGO GUAYAQUIL LOOR

Agradezco en primer lugar a Dios por haberme dado Fuerza, Voluntad, Paciencia y Salud, y por haberme permitido alcanzar esta meta profesional;

En segundo lugar a mis Queridos y Grandiosos Padres LEON RODRIGO GUAYAQUIL CARVAJAL Y MARIA DEL PILAR LOOR GOMEZ, por su grandioso esfuerzo y apoyo constante cada día.

También a mis Hermanos, familiares y a mi adorada Guisella por haberme brindado su apoyo incondicional en los momentos difíciles de mi carrera.

JOSÉ LUIS SILVA PALMA

Agradezco principalmente a mis padres, MARÍA PALMA MENDOZA Y JOSÉ SILVA VALLE, por su apoyo constante e incondicional durante toda mi vida académica, por la motivación, por sus consejos y por todo el cariño que me dieron, guiándome siempre por el camino correcto hacia el éxito.

También agradezco a aquellos miembros de mi familia, a mis amigos y compañeros, quienes me brindaron sus consejos y confiaron en mí durante mi formación académica.

DEDICATORIA

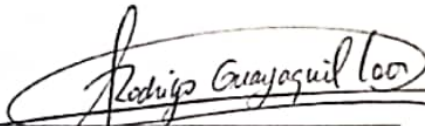
*Dedicamos este trabajo a nuestros
PADRES.*

*Y a todas las personas que aportaron a
la culminación de nuestra carrera
universitaria.*

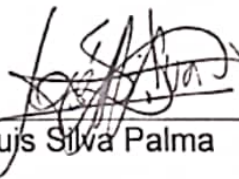
DECLARACIÓN EXPRESA

"La responsabilidad por los hechos, ideas y doctrinas expuestas en este trabajo, me corresponden exclusivamente; y, el patrimonio intelectual de la misma, a la Escuela Superior Politécnica del Litoral"

(Reglamento de exámenes y títulos profesionales de la ESPOL)



Rodrigo Guayaquil Loo



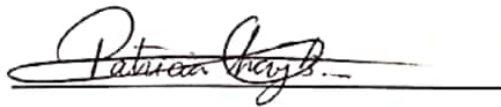
José Luis Silva Palma

TRIBUNAL DE SUSTENTACIÓN

A handwritten signature in black ink, reading "Gabriel Astudillo B.", written over a horizontal line.

Ing. Gabriel Astudillo B.

PROFESOR DE LA MATERIA DE GRADUACIÓN

A handwritten signature in black ink, reading "Patricia Chávez", written over a horizontal line.

Ing. Patricia Chávez

PROFESOR DELEGADO POR EL DECANO DE LA FACULTAD

RESUMEN

Actualmente en el Ecuador aun se encuentran muchas dificultades al momento de adquirir un Sistema de Video Vigilancia, ya sea por el alto costo de los equipos y/o por la complejidad del funcionamiento de los mismos, esto se debe en gran parte a que todo este tipo de soluciones de vigilancia se distribuyen bajo Licencias.

El presente trabajo ofrece una alternativa a esto, al implementar un Sistema de Video Vigilancia con Software de Código Abierto implementado totalmente bajo una plataforma de Linux. El procedimiento utiliza las bondades de un servidor ASTERISK vinculado con un programa de control de cámaras web, llamado MOTION, y un módulo incorporado en Asterisk llamado CHAN-MOBILE, el cual permite establecer comunicación con dispositivos móviles utilizando celulares como puertas de salida hacia la red GSM a través de un enlace Bluetooth.

El sistema detecta movimiento a través de una cámara web, luego ejecuta un sonido de ALARMA y realiza una llamada a un teléfono CELULAR, donde comunica la presencia de algún intruso y brinda la opción de apagar el sonido de alarma a través del celular.

INDICE GENERAL

AGRADECIMIENTO	ii
DEDICATORIA	iii
RESUMEN	vi
INDICE GENERAL	vii
INDICE DE FIGURAS	ix
INDICE DE TABLAS	xi
INTRODUCCIÓN.....	1
ANTECEDENTES Y JUSTIFICACIÓN	2
1.1 Antecedentes	3
1.2 Descripción del Proyecto	4
1.2.1 Objetivo General.....	6
1.2.2 Objetivos Específicos.....	6
1.3 Justificación	7
1.4 Metodología	7
1.5 Perfil de la Tesis.....	8
FUNDAMENTOS TEÓRICOS	9
2.1 Sistemas de Vigilancia	10
2.2 Asterisk	11
2.3 Cámaras Web.....	13
2.4 Motion	14
2.5 Bluetooth	15
2.6 Módulo Chan-Mobile.....	17
IMPLEMENTACIÓN DEL SISTEMA DE VIDEO VIGILANCIA	20
3.1 Requerimientos para la Implementación.....	21
3.1.1 Hardware	21
3.1.1.1 Servidor.....	21
3.1.1.2 Cámara Web.....	22
3.1.1.3 Bluetooth.....	22
3.1.1.4 Gateway	22
3.1.2 Software.....	23
3.1.2.1 Servidor.....	23
3.1.2.2 Motion	23
3.1.2.3 Bluetooth.....	24

3.1.2.4	Softphone X-Lite	24
3.2	Instalación.....	24
3.2.1	Instalación de Librerías Base.....	25
3.2.2	Instalación de Librerías para Bluetooth.....	28
3.2.3	Instalación de Asterisk.....	28
3.2.4	Instalación de Asterisk-Addons.....	30
3.2.5	Instalación de Chesse.....	31
3.2.6	Instalación de Motion.....	31
3.2.7	Instalación de X-Lite	31
3.3	Configuración	32
3.3.1	Configuración de Motion	32
3.3.1.1	Descripción del Archivo MOTION.CONF	33
3.3.2	Configuración del Softphone X-Lite.....	37
3.3.3	Configuración de ASTERISK.....	40
3.3.3.1	Configuración del Archivo SIP.CONF.....	40
3.3.3.1.1	Descripción del Archivo SIP.CONF	41
3.3.3.2	Configuración del Archivo EXTENSIONS.CONF.....	42
3.3.3.2.1	Descripción del Archivo EXTENSIONS.CONF	44
3.3.3.3	Configuración del Archivo MOBILE.CONF.....	47
3.3.3.3.1	Descripción del Archivo MOBILE.CONF	47
3.3.4	Creación de SCRIPTS	51
3.3.4.1	Script de Control de Motion.....	52
3.3.4.1.1	Descripción del Archivo CONTROL_MOTION.SH	52
3.3.4.2	Script de Ejecución del Programa MOTION.....	53
3.3.4.3	Script de Gestión de Eventos.....	54
3.3.4.3.1	Descripción del Archivo CAM_EVENT.SH.....	55
3.3.4.4	Script de Ejecución de Alarma	56
3.3.4.5	Script de Empezar el Sonido de Alarma.....	57
3.3.4.6	Script de Detener el Sonido de Alarma	58
	FUNCIONAMIENTO Y PRUEBAS DEL PROYECTO	59
4.1	Inicialización de Asterisk.....	60
4.2	Activación del Softphone.....	60
4.3	Conexión de Asterisk con el Gateway	62
4.4	Activación del Sistema de Vigilancia	62
4.5	Funcionamiento del Sistema de Vigilancia.....	64
4.6	Ejecución de Llamada.....	65
	CONCLUSIONES Y RECOMENDACIONES	67
	BIBLIOGRAFÍA	70

INDICE DE FIGURAS

Figura 1.1: Esquema del Sistema de Vigilancia	5
Figura 2.1: Sistemas de Vigilancia	10
Figura 2.2: Esquema de Funcionalidad de Asterisk	11
Figura 2.3: Características de Asterisk.....	13
Figura 2.4: Operación de las Cámaras Web	14
Figura 2.6: Dispositivos con Bluetooth	17
Figura 3.1: Celular Nokia 6300	22
Figura 3.2: Indicadores de Usuarios	25
Figura 3.3: Versión del Kernel	26
Figura 3.4: Archivo motion.conf.....	32
Figura 3.5: Softphone X-Lite	37
Figura 3.6: Menú Principal de X-Lite	38
Figura 3.7: Ingreso a la Configuración del SIP Proxy	38
Figura 3.8: Configuración de X-Lite	39
Figura 3.9: Dirección IP del PC	39
Figura 3.10: Archivo sip.conf.....	41
Figura 3.11: Archivo extensions.conf.....	43
Figura 3.12: Archivo mobile.conf	47
Figura 3.13: Dirección MAC del Dispositivo Bluetooth	48
Figura 3.14: Escaneo de Dispositivos Bluetooth	49
Figura 3.15: Inicio del Servicio de Asterisk.....	49
Figura 3.16: Ingreso a la Consola Remota de Asterisk	50
Figura 3.17: Obtención del Puerto del Teléfono Móvil.....	50
Figura 3.18: Terminación del Servicio de Asterisk	51
Figura 3.19: Archivo control_motion.sh	52
Figura 3.20: Archivo exec_motion.sh.....	53
Figura 3.21: Archivo cam_event.sh	54
Figura 3.22: Descripción del cam_event.sh	56
Figura 3.23: Archivo start_sound.sh.....	57
Figura 3.24: Archivo play_sound.sh	57
Figura 3.25: Archivo stop_sound.sh	58
Figura 4.1: Inicio de Asterisk.....	60
Figura 4.2: Ingreso a la Consola Remota de Asterisk	60

Figura 4.3: Registro de X-Lite en Asterisk	61
Figura 4.4: Usuarios SIP	62
Figura 4.5: Conexión de Asterisk con el Gateway	62
Figura 4.6: Activación del Sistema de Vigilancia.....	63
Figura 4.7: Activación de Motion	63
Figura 4.8: Ejecución del Script cam_event.sh.....	64
Figura 4.9: Ejecución de la Llamada en Consola.....	65
Figura 4.9: Realización de la Llamada.....	66
Figura 4.10: Fotografías Capturadas.....	66

INDICE DE TABLAS

Tabla 1 – Características del Servidor	21
Tabla 2 – Componentes del Servidor	23

INTRODUCCIÓN

El continuo crecimiento y evolución de la tecnología de Voz sobre IP y la acogida que actualmente presenta la utilización de Centrales Telefónicas No Privativas en el Ecuador, junto a la necesidad de utilizar sistemas de vigilancia en oficinas o en hogares ha guiado a la realización del presente trabajo, el cual trata de la Implementación de un Sistema de Video Vigilancia bajo una plataforma de Linux, aprovechando las funciones y aplicaciones del programa de código abierto ASTERISK vinculado con el modulo CHAN-MOBILE y trabajando conjuntamente con el software MOTION, adicionalmente se utiliza una cámara web y un teléfono celular con Bluetooth para establecer comunicación con la telefonía móvil celular.

La cámara web será la encargada de monitorear los movimientos que ocurren dentro de un espacio escogido en el hogar u oficina; con la ayuda de MOTION se tomarán fotografías y se realizará un video cuando se detecte algún movimiento, posteriormente con la ejecución de un SCRIPT se activa una ALARMA audible y se realiza una llamada desde ASTERISK, esta llamada será dirigida hacia la red GSM a través de un enlace Bluetooth entre el servidor y el celular Gateway, todo esto gracias al modulo CHAN-MOBILE. Al establecerse la llamada entre el servidor y el celular destino, en este caso el número celular del propietario del sistema, el sistema informa de la activación de la alarma y brinda la opción de apagar dicha alarma a través del celular.

CAPITULO 1

ANTECEDENTES Y JUSTIFICACIÓN

1.1 Antecedentes

La seguridad es un objetivo que el hombre anhela constantemente como una necesidad primaria, es por esta razón que los sistemas que brindan seguridad son tan importantes e indispensables en las empresas y hogares, puesto que ayudan en gran parte a que los usuarios se sientan protegidos y a su vez mantengan vigilados sus bienes materiales. Con la ayuda de estos sistemas electrónicos de vigilancia se obtiene un hogar u oficina más segura y confortable.

En el mercado existe gran variedad de productos que ofrecen este servicio y así mismo hay muchas compañías que se dedican a dar seguridad a empresas y hogares. Algunos equipos son más complejos en funcionamiento, pudiendo así cubrir más áreas y servicios simultáneos, pero con mayor dificultad en manejo desde la perspectiva del usuario y además con costos muy elevados para su adquisición, instalación, soporte técnico y mantenimiento.

Adicionalmente con el crecimiento en la utilización de las tecnologías inalámbricas, enfocándose principalmente en el estándar *Bluetooth*, se tiene la posibilidad de adaptar el uso de la misma, ya que viene incorporada en los distintos dispositivos de comunicación comúnmente utilizados hoy en día, como por ejemplo PDA, teléfonos móviles, computadoras portátiles, impresoras, cámaras digitales, auriculares, etc.

A través del presente trabajo se ofrece una opción de vigilancia, utilizando programas de libre distribución, al implementar un sistema de detección de intrusos que avise por medio de una llamada celular cualquier anomalía que ocurra en el lugar de instalación del mismo.

1.2 Descripción del Proyecto

El presente trabajo consiste en la explicación detallada de la Implementación de un Sistema de Video Vigilancia instalado en una plataforma de Linux y con la utilización de programas de código abierto.

Además se hizo uso de la tecnología inalámbrica con su estándar de Bluetooth para el enlace entre el servidor Asterisk con Bluetooth y un teléfono celular con Bluetooth utilizado como puerta de salida hacia la red de telefonía móvil.

La cámara web será la encargada de detectar los movimientos que acontecen dentro del rango de visión de la misma. Con la ayuda de MOTION, la cámara tomará fotografías y realizará un video cuando localice algún movimiento, almacenando dichas fotografías y video en un directorio que posteriormente se podrá consultar.

Adicionalmente al detectarse un movimiento se ejecuta un SCRIPT que reproduce un sonido de alarma por los altavoces del computador portátil, y

que además efectúa la conexión con ASTERISK para realizar una llamada a un número de teléfono celular predefinido.

La realización de la llamada es posible gracias a la vinculación del modulo CHAN-MOBILE con Asterisk, mismo que permite utilizar celulares como puertas de salida hacia la red GSM a través de un enlace Bluetooth.

Al establecerse la llamada, Asterisk informa de la activación de la ALARMA por detección de movimiento y ofrece la opción de apagar el sonido de alarma a través del celular durante la llamada en curso.

En la Figura 1.1 se observa un esquema de cómo funciona el Sistema de Vigilancia aquí propuesto.

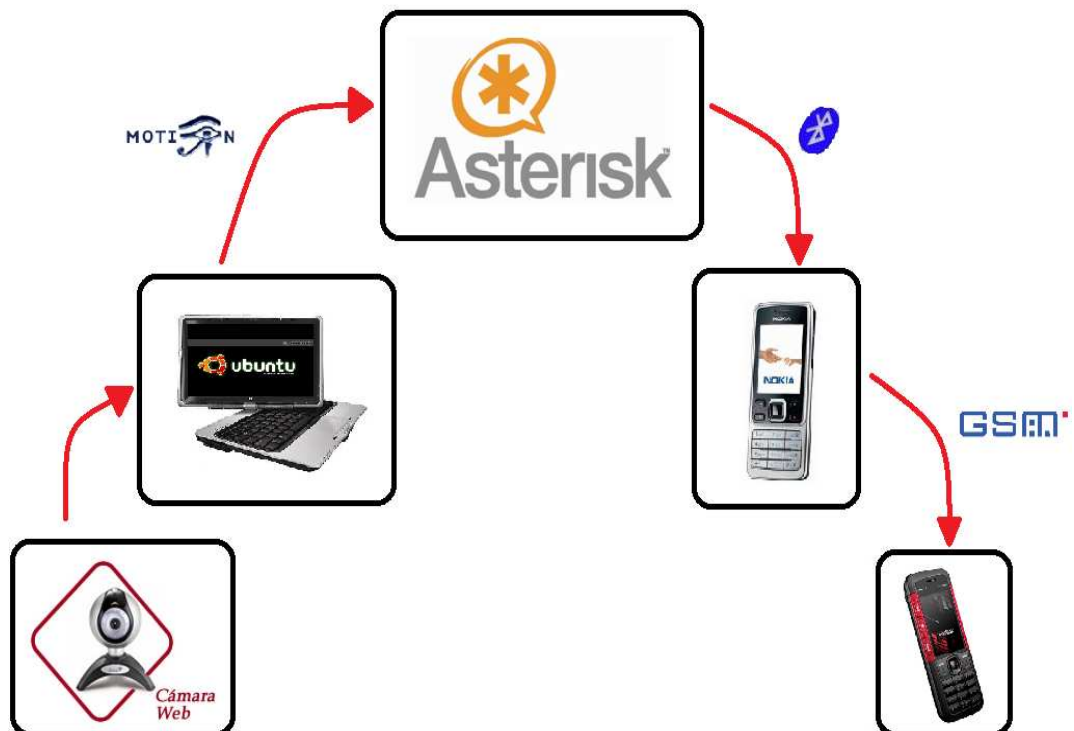


Figura 1.1: Esquema del Sistema de Vigilancia

1.2.1 Objetivo General

Implementar un Sistema de Video Vigilancia utilizando programas que sean de libre distribución, que proporcione la información necesaria de lo ocurrido y que además esa información llegue inmediatamente al teléfono móvil del usuario.

1.2.2 Objetivos Específicos

Para cumplir con el objetivo principal del proyecto se deben cumplir los siguientes objetivos:

- ❖ Identificar claramente cada uno de los componentes en Hardware y Software que intervienen en el Sistema de Video Vigilancia.
- ❖ Analizar las herramientas que facilitan la implementación del sistema, determinando así los procedimientos que deben ser llevados a cabo para la óptima funcionalidad del mismo.
- ❖ Entender el funcionamiento de los programas a utilizar, para obtener el máximo provecho de los mismos.
- ❖ Realizar una cuidadosa instalación de los recursos necesarios del sistema para el correcto funcionamiento del mismo.
- ❖ Identificar correctamente los archivos a configurar para lograr el funcionamiento exitoso, de manera conjunta, entre los distintos programas y dispositivos a utilizar.

- ❖ Verificar la compatibilidad entre los componentes utilizados y su funcionamiento coordinado.
- ❖ Proporcionar toda la información posible al usuario del Sistema para la correcta implementación y utilización del mismo.

1.3 Justificación

Debido al aumento de inseguridad en el Ecuador y junto a la gran demanda de recursos económicos para la adquisición de alguna solución para esto, surge la idea de buscar una alternativa más económica y con mayor facilidad de implementación que se ajuste a las necesidades del usuario. Una de estas alternativas es la utilización de Software de Libre Distribución y de Código Abierto para la Implementación de un Sistema de Vigilancia sencillo y eficaz.

1.4 Metodología

Para el correcto funcionamiento del Sistema de Vigilancia se utilizó una laptop como servidor con sistema operativo Ubuntu 10.04 LTS y se le añadió las dependencias necesarias para el uso de la cámara web y el módulo Bluetooth incorporados en el servidor, también se instaló el programa de control de la cámara web, MOTION, y el software de PBX,

ASTERISK, incluyendo el modulo CHAN-MOBILE para poder establecer la comunicación con la red de telefonía celular.

Posteriormente se configuró correctamente los archivos necesarios para el funcionamiento coordinado del sistema. Luego se crearon y describieron los SCRIPTS a utilizar.

Finalmente se hizo la demostración de cómo opera el Sistema de Video Vigilancia.

1.5 Perfil de la Tesis

En el capítulo 2, se explica sobre las diferentes generalidades de los componentes utilizados en el proyecto, entre los que tenemos: Sistemas de Vigilancia, Asterisk, Cámaras Web, Motion, Bluetooth y Chan-Mobile.

En el capítulo 3, se describe las especificaciones técnicas, también se explica el diseño y se realiza la implementación de la solución propuesta.

En el capítulo 4, se realizan las pruebas de cada uno de los elementos que intervienen, y finalmente se muestra la operación y el funcionamiento del Sistema de Video Vigilancia.

CAPITULO 2

FUNDAMENTOS TEÓRICOS

2.1 Sistemas de Vigilancia

Las compañías que ofrecen seguridad y ofertan sistemas de vigilancia han llegado a ser muy populares en estos últimos años, entre quienes necesitan del servicio en sus empresas y aquellos que lo solicitan en sus hogares, tanto así que algunas constructoras de viviendas ofrecen la inclusión de los requisitos apropiados para instalar y accionar la mayoría de los sistemas caseros de vigilancia. Sin embargo esto no es indispensable, puesto que debido a lo fácil que resulta su instalación, esta se la puede realizar en edificaciones previamente construidas.

Los sistemas de vigilancia, como los mostrados en la Figura 2.1, no son solo alarmas sino que también incorporan cámaras de vigilancia en los lugares internos y externos, y además pueden trabajar con el sistema existente de prevención de incidentes, tales como, detectores de humo, sensores de rotura de cristal, etc. Algunos de los sistemas de vigilancia existentes en el mercado también permiten que el usuario controle el sistema y observe lo que ocurre remotamente a través de Internet.



Figura 2.1: Sistemas de Vigilancia

2.2 Asterisk

Asterisk es un sistema de comunicaciones de software libre (bajo licencia GPL) que proporciona funcionalidades de una central telefónica, tal y como lo muestra la Figura 2.2, con una extensa variedad de aplicaciones, y está basado totalmente en software. Como cualquier PBX, se puede obtener a partir de su empleo una central telefónica conectada directamente a la red pública de teléfono por medio de líneas troncales para gestionar, además de las llamadas internas, las entrantes y/o salientes con autonomía sobre cualquier otra central telefónica.

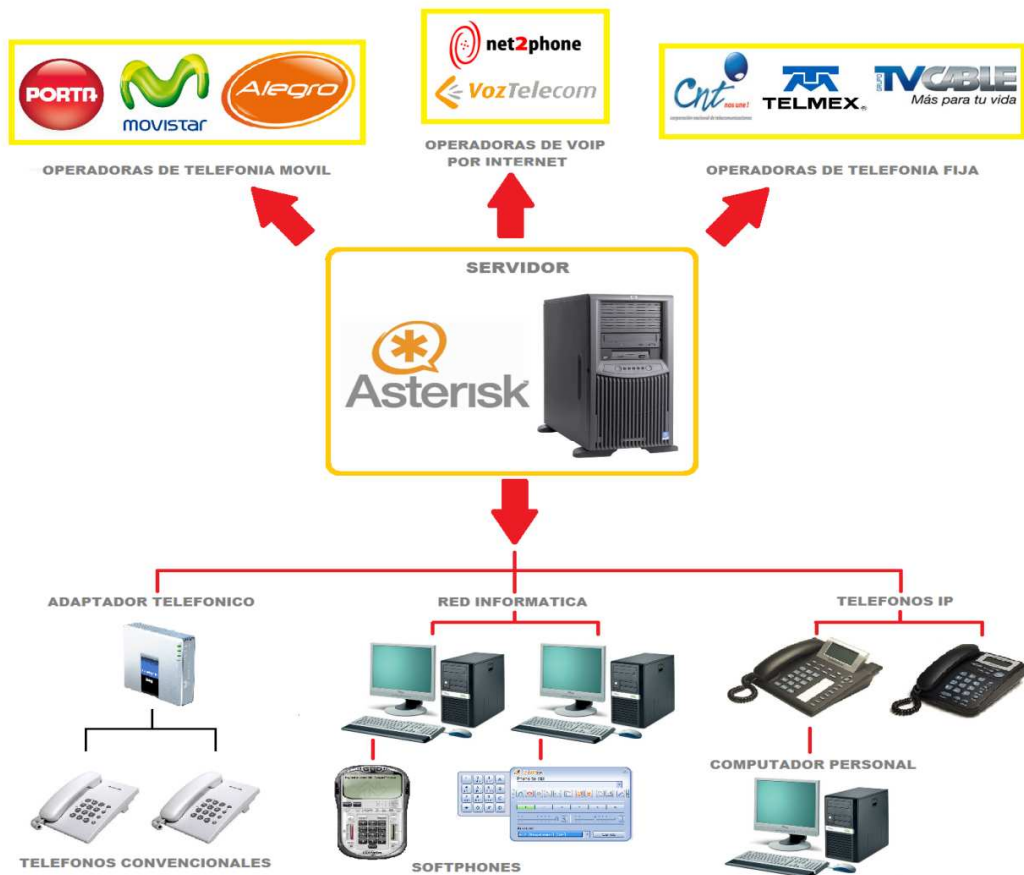


Figura 2.2: Esquema de Funcionalidad de Asterisk

El empleo de sistemas PBX evita conectar todos los teléfonos de una empresa de manera separada a la red de telefonía local pública (RTC), ya que funciona como un switch de red, con esto se evita gastos al establecer comunicación interna. Si se posee una oficina pequeña y se desea emplear software PBX tradicional, los costos serían altos, pero con Asterisk se reducen notablemente, ya que emula este tipo de centrales telefónicas.

Gracias a que Asterisk es de *Código Abierto*, permite la utilización de hardware estándar, y aprovecha lo mejor que encuentra en otros PBX. Además su desarrollo es modular, dinámico, flexible y adaptable, y realizado por la comunidad, misma que también se encarga del mantenimiento. Con todo esto se entiende que su crecimiento es ilimitado, con lo cual posibilita la escalabilidad. Otra facilidad de ser de código abierto, es que permite la personalización por parte del usuario, con esto el cliente decide lo que quiere, y elimina lo que no le interesa.

Asterisk incluye muchas características anteriormente sólo disponibles en costosos sistemas propietarios PBX como buzón de voz, conferencias, IVR, distribución automática, transferencia y desvío de llamadas, servicios de Call-Center como sistemas de colas, grabación de llamadas, también permite la posibilidad de integrarlo con bases de datos y otras muchas más opciones ilustradas en la Figura 2.3.

Adicionalmente los usuarios pueden crear nuevas funcionalidades escribiendo un *plan de marcado* en scripts o añadiendo módulos escritos en cualquier lenguaje de programación soportado por Linux.



Figura 2.3: Características de Asterisk

2.3 Cámaras Web

Una *cámara web* o *webcam* es una pequeña cámara digital conectada al computador, la cual puede capturar imágenes y transmitir las a través de internet, ya sea a una página web u otros computadores de forma privada.

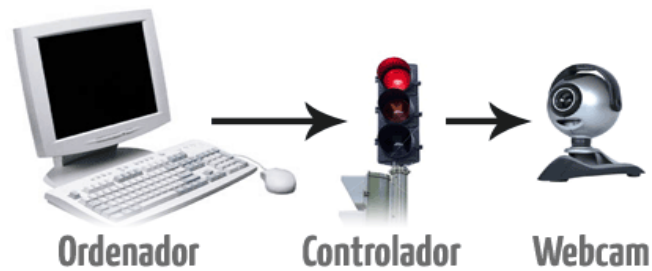


Figura 2.4: Operación de las Cámaras Web

Las cámaras web necesitan de un computador para transmitir las imágenes, tal y como lo muestra la Figura 2.4. Sin embargo, existen otras cámaras autónomas que tan sólo necesitan un punto de acceso a la red informática, bien sea Ethernet o inalámbrico, y se las denomina cámaras de red.

Son muy utilizadas en mensajería instantánea y chat. Este dispositivo se clasifica como de entrada, ya que por medio de él se puede transmitir imágenes hacia el computador.

2.4 Motion

Motion es el software que se encarga de monitorizar la cámara web en busca de actividad, utiliza el sistema V4L (Video 4 Linux), que no es más que el conjunto de funciones y procedimientos empleados para la captura de video en Linux. Funciona como la mayoría de programas de detección de movimiento basados en cámaras, simplemente comprueba la diferencia de píxeles entre fotogramas consecutivos capturados y si esta diferencia

es superior a un umbral predefinido, asume que hay movimiento. Este umbral debe ser bien estudiado para que un simple movimiento de cortina o el reflejo del sol no salten como un falso positivo. Una de las características que lo diferencian de otros programas similares es que permite monitorizar varias cámaras.

Motion sólo captura fotos cuando detecta movimiento, y las almacena en un directorio que luego se puede consultar, también se puede crear video con las fotografías tomadas. Además permite crear un pequeño servidor web de audio y video, con el que es posible visualizar la cámara web desde cualquier lugar del mundo y permite ejecutar acciones al suceder algún evento. Motion no dispone de entorno gráfico, pero no es nada complicado hacerlo funcionar, y el hecho de que solo sea un pequeño demonio que corre en segundo plano le da mayor flexibilidad. Motion está en la mayoría de los repositorios de las distintas distribuciones, pero se recomienda compilarlo e instalarlo manualmente.

2.5 Bluetooth

Fundamentalmente, el Bluetooth vendría a ser el nombre común de la especificación industrial IEEE 802.15.1, que define un estándar global de comunicación inalámbrica que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia

segura en la banda ISM de los 2,4 [GHz], globalmente y sin licencia de corto rango.

Los principales objetivos que se pretenden conseguir con esta norma son:

- Facilitar las comunicaciones entre equipos móviles y fijos.
- Eliminar cables y conectores entre éstos.
- Ofrecer la posibilidad de crear pequeñas redes inalámbricas y facilitar la sincronización de datos entre equipos personales.

Los dispositivos que con mayor frecuencia utilizan esta tecnología pertenecen a sectores de las telecomunicaciones y la informática personal, como los mostrados en la Figura 2.6, en la que se encuentra a los PDA, teléfonos móviles, computadoras portátiles, ordenadores personales, impresoras o cámaras digitales.

Para utilizar Bluetooth, un dispositivo debe implementar alguno de los perfiles Bluetooth. Estos definen el uso del canal Bluetooth, así como ayudan a canalizar al dispositivo que se quiere vincular.



Figura 2.6: Dispositivos con Bluetooth

2.6 Módulo Chan-Mobile

Es un driver de canal de Asterisk que permite la utilización de teléfonos móviles con bluetooth como dispositivos FXO, y auriculares bluetooth como dispositivos FXS.

En las primeras versiones se utilizaba el paquete *chan_bluetooth* de Theo Zourzouvillys el cual estaba en cierto sentido desactualizado, luego se comenzó con el desarrollo del software *chan_cellphone*, el cual derivó en la versión que actualmente se conoce como *chan_mobile* y que está disponible como un add-on vía SVN y en la versión 1.6.x de Asterisk.

Entre las características de este módulo se encuentran las siguientes:

- Gran variedad de modelos de teléfonos celulares son compatibles con este módulo.
- Múltiples adaptadores bluetooth pueden ser conectados.
- Asterisk se conecta automáticamente a cada teléfono celular cuando el mismo se encuentra dentro del alcance.
- Tiene comando para detectar otros dispositivos bluetooth, útil para la configuración. Necesita de un adaptador bluetooth desocupado.
- Las llamadas entrantes a los teléfonos celulares son manejadas por Asterisk, al igual que las llamadas entrantes en un canal DAHDI.
- Marcación a un número de teléfono celular usando *Dial (celular / dispositivo / nnnnnn)* en el plan de marcado.
- Puede usar un auricular bluetooth como extensión con el comando *Dial (celular / dispositivo)* en el plan de marcado.
- La aplicación *CellStatus* puede ser utilizada en el plan de marcado para ver si está conectado un teléfono celular.
- Existe la aplicación *MobileSMS* para enviar SMS a través de un teléfono móvil conectado.
- Se necesita usar un dongle bluetooth por celular al momento de la conexión con el servidor Asterisk.

La correcta utilización de este modulo implica el conocimiento de la lista de compatibilidad del mismo con dispositivos dongle bluetooth y teléfonos celulares. Esta lista se la encuentra en la siguiente página web:

http://www.voip-info.org/wiki/view/chan_mobile

CAPITULO 3

IMPLEMENTACIÓN DEL SISTEMA DE VIDEO VIGILANCIA

3.1 Requerimientos para la Implementación

3.1.1 Hardware

Las características para seleccionar un servidor se puede considerar una tarea sencilla porque puede ser cualquier equipo basado en plataforma x86, y a la vez complicada, porque el rendimiento de su sistema dependerá de la atención que le ponga a la plataforma de implementación. Cuando seleccione el hardware, usted debe considerar cuidadosamente el diseño general de su sistema y las funcionalidades que requerirá, esto le ayudará a determinar la marca y modelo del CPU, tarjeta madre, y fuente de energía.

3.1.1.1 Servidor

El computador que se encarga de realizar las tareas de servidor para el sistema de vigilancia es un Computador Portátil HP Pavilion dv5-1244la Entertainment PC, el cual tiene las siguientes características:

Tabla 1 – Características del Servidor

Procesador	Intel Core 2 Duo T6400 de 2 GHZ
RAM	3 GB DDR2 800 MHz
Disco Duro	250 GB
Dispositivo Bluetooth	HP Integrated Bluetooth module
Cámara Web Integrada	HP Webcam

3.1.1.2 Cámara Web

La captura de imágenes y el monitoreo de movimiento se realizará a través de la cámara web incorporada en el equipo servidor, cuyo modelo es el siguiente: *HP Webcam*.

3.1.1.3 Bluetooth

La conexión del servidor Asterisk con el teléfono celular gateway se realizó a través de bluetooth, es por esta razón que se utilizó el modulo bluetooth incorporado en el computador portátil servidor.

3.1.1.4 Gateway

Para realizar la comunicación con la Red GSM se utilizó un dispositivo móvil con tecnología bluetooth incorporada, en este caso se ha hecho uso del teléfono celular NOKIA 6300 mostrado en la Figura 3.1.



Figura 3.1: Celular Nokia 6300

3.1.2 Software

3.1.2.1 Servidor

El servidor que fue utilizado como sistema de vigilancia tiene los siguientes componentes instalados:

Tabla 2 – Componentes del Servidor

Sistema Operativo	Linux Ubuntu 10.04 LTS
Kernel	Versión 2.6.32-23-generic
Software PBX	Asterisk versión 1.6.0.28
Asterisk Add-on	Asterisk Addons 1.6.0.3

A continuación se muestra las librerías necesarias para que Asterisk funcione correctamente como PBX:

```
kernel      kernel-devel  bison      bison-devel  ncurses
ncurses-devel  zlib        zlib-devel  openssl      openssl-devel
gnutls-devel  gcc         gcc-c++
```

3.1.2.2 Motion

Es el software encargado de controlar la cámara web para que detecte movimiento. Se recomienda adicionalmente la instalación de un software para probar el funcionamiento de la cámara web, en este proyecto se utilizó el programa *Chesse*.

3.1.2.3 Bluetooth

La pila de protocolos **Bluez** proporciona soporte para el núcleo Bluetooth y los protocolos de capas en Linux. Es flexible, eficiente y utiliza una aplicación modular.

Librerías necesarias: bluez-utils, bluez-libs y bluez-libs-devel

3.1.2.4 Softphone X-Lite

Los softphones son simuladores de extensiones de una central telefónica, en este proyecto se ha utilizado X-Lite para que simule una extensión, misma que permitirá controlar la activación y desactivación del sistema de vigilancia. Este softphone utiliza el protocolo de voz sobre IP *SIP* para comunicarse con Asterisk.

3.2 Instalación

A continuación se detalla todos los paquetes instalados para el funcionamiento del sistema de vigilancia, para esto se ha tomado como punto de partida la instalación previa del Sistema Operativo Linux en su distribución de Ubuntu 10.04 LTS.

Adicionalmente se recomienda realizar todos los pasos de la instalación como usuario “*root*”, para evitar problemas con los permisos de los directorios y archivos a modificar posteriormente.

Para los siguientes pasos de la instalación se necesita proveer de una conexión a Internet al computador.

3.2.1 Instalación de Librerías Base

Se procede a la instalación de las dependencias necesarias para que Asterisk funcione en la distribución de Linux.

Se abre un Terminal (**Aplicaciones -> Accesorios -> Terminal**), y se ejecuta los siguientes comandos:

Se ingresa como “*root*”, solo en el caso en que se haya iniciado sesión con otra cuenta:

```
$ sudo -s
```

La Figura 3.2 muestra la diferencia del indicador en el Terminal al cambiar de usuario.

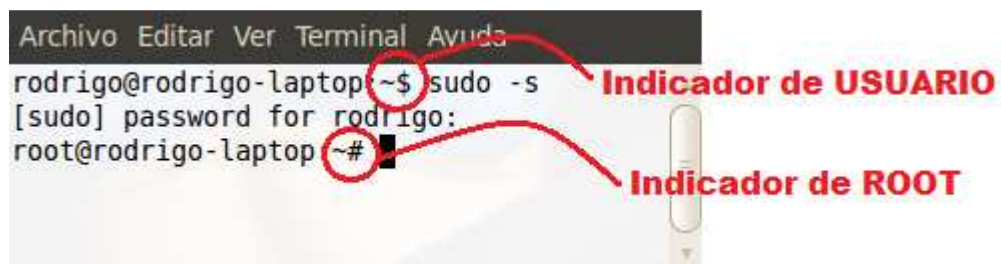


Figura 3.2: Indicadores de Usuarios

Se edita el archivo *sources.list* ubicado en el directorio */etc/apt/*

```
# gedit /etc/apt/sources.list
```

Se agrega las siguientes líneas al archivo *sources.list*:

```
deb http://us.archive.ubuntu.com/ubuntu/ dapper universe
```

```
deb-src http://us.archive.ubuntu.com/ubuntu/ dapper universe
```

Luego se guarda y cierra el archivo *sources.list*

Se actualiza los repositorios ejecutando el siguiente comando:

```
# apt-get update
```

Se instala las librerías para acceso vía SSH:

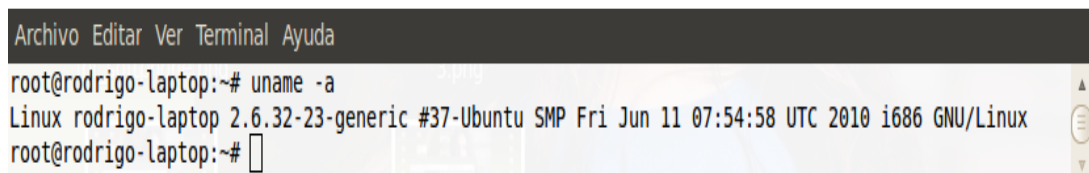
```
# apt-get install openssh libssl-dev ssh
```

Se instala las librerías de compilación de software:

```
# apt-get install gcc make g++
```

Se verifica que versión del *kernel* se tiene instalada en el servidor, el resultado debe ser algo como lo mostrado en la Figura 3.3:

```
# uname -a
```



```
Archivo Editar Ver Terminal Ayuda
root@rodrigo-laptop:~# uname -a
Linux rodrigo-laptop 2.6.32-23-generic #37-Ubuntu SMP Fri Jun 11 07:54:58 UTC 2010 i686 GNU/Linux
root@rodrigo-laptop:~#
```

Figura 3.3: Versión del Kernel

Luego se busca las fuentes del *kernel* con la version instalada en el servidor

```
# apt-cache search 2.6.32-23
```

Se procede a instalar los encabezados para la versión del *kernel* (*linux-headers*, *linux-source*, *linux-image*):

```
# apt-get install linux-headers-2.6.32-23-server linux-image-2.6.32-23-server linux-source-2.6.32-23
```

Se instalan los paquetes para el manejo de voz:

```
# apt-get install festival festival-dev
```

Para poder usar la consola de Asterisk se instalaron las siguientes librerías:

```
# apt-get install ncurses-base ncurses-bin ncurses-term libncurses5 libncursesw5 libncurses5-dev libncursesw5-dev
```

Se instalaron las librerías de compresión:

```
# apt-get install zlib1g zlib1g-dev
```

Se instalaron las librerías para Base de Datos:

```
# apt-get install mysql-client-5.0 mysql-server-5.0 libmysqlclient15-dev
```

Se descargó e instaló la librería *Lame*, misma que brinda soporte para la reproducción de MP3's en Linux:

```
# cd /usr/src
# wget http://ufpr.dl.sourceforge.net/sourceforge/lame/lame-398-2.tar.gz
# tar -xzvf lame-398-2.tar.gz
# cd lame-398-2
# ./configure --prefix=/usr
# make
# make install
# ldconfig -v
```

3.2.2 Instalación de Librerías para Bluetooth

Se ejecuta los siguientes comandos:

```
# apt-get install bluez-utils
# apt-get install bluez-libs
# apt-get install bluez-libs-devel
```

3.2.3 Instalación de Asterisk

Se creó un directorio para la instalación de los paquetes de Asterisk:

```
# mkdir /usr/src/asterisk
```

Se ingresa al directorio:

```
# cd /usr/src/asterisk
```

Se procede a la descarga de la versión actual del Asterisk 1.6.0:

```
# wget http://downloads.digium.com/pub/telephony/asterisk/asterisk-1.6.0-current.tar.gz
```

A continuación se desempaqueta el archivo descargado:

```
# tar -xzf asterisk-1.6.0-current.tar.gz
```

Luego se ingresa al directorio que se creó al ejecutar el paso anterior:

```
# cd asterisk-1.6.0.28/
```

De haberse realizado una instalación de Asterisk anteriormente, se ejecuta el siguiente comando:

```
# make clean
```

Se abre el menú de Asterisk, recomendado para usuarios con experiencia en la utilización de este software:

```
# make menuselect
```

En el menú que aparece se puede escoger los módulos a instalar/activar (siempre y cuando previamente se haya instalado las dependencias necesarias).

Finalmente se procede con la compilación y creación de los archivos de ejemplo de Asterisk al ejecutar los siguientes comandos:

```
# make
```



```
# make install  
  
# make samples  
  
# make config
```

3.2.4 Instalación de Asterisk-Addons

Ahora se procede a descargar los "addons" de Asterisk:

```
# cd /usr/src/asterisk  
  
# wget http://downloads.digium.com/pub/telephony/asterisk/asterisk-  
addons-1.6.0-current.tar.gz
```

A continuación se procede a desempaquetar el archivo descargado:

```
# tar -xvzf asterisk-addons-1.6.0-current.tar.gz
```

Luego se ingresa al directorio que se creó al ejecutar el paso anterior:

```
# cd asterisk-addons-1.6.0.3/
```

Finalmente se procede a realizar la instalación de los "Addons" con el modulo de soporte Bluetooth "*chan_mobile*":

```
# ./configure --with-bluetooth  
  
# make  
  
# make install  
  
# make samples
```

3.2.5 Instalación de Chesse

Con la instalación de este software se cargan las dependencias necesarias para la utilización de la cámara web.

```
# apt-get install cheese
```

3.2.6 Instalación de Motion

La instalación del programa en Ubuntu es muy sencilla, ya que *Motion* se encuentra dentro de los repositorios:

```
# apt-get install Motion
```

3.2.7 Instalación de X-Lite

Se descarga el instalador para Linux del softphone (*X-Lite_Install.tar.gz*) desde el siguiente link:

```
http://xlite.counterpath.com/download/linux
```

Se ubica el archivo descargado en el siguiente directorio: **/usr/src**

Se procede a desempaquetar el archivo descargado, y listo:

```
# cd /usr/src
```

```
# tar -zxvf X-Lite_Install.tar.gz
```

3.3 Configuración

A continuación se procede a indicar los archivos y programas a configurar para que el sistema de vigilancia funcione perfectamente. Adicionalmente se realiza una explicación detallada del contenido de cada archivo configurado.

3.3.1 Configuración de Motion

Primero se realiza una copia del archivo de configuración existente:

```
# cp /etc/motion/motion.conf /etc/motion/motion.conf.backup
```

Luego se edita el archivo de configuración de Motion, *motion.conf*

```
# gedit /etc/motion/motion.conf
```



```
motion.conf ✕
daemon on
videodevice /dev/video0
width 640
height 480
framerate 30
auto_brightness on
brightness 0
contrast 0
saturation 0
threshold 20000
noise_tune on
minimum_motion_frames 9
output_normal on
quality 85
ffmpeg_cap_new on
ffmpeg_video_codec mpeg4
locate on
text_right %Y-%m-%d\n%T
text_left SISTEMA DE VIGILANCIA
target_dir /tmp/motion
webcam_port 8081
webcam_quality 30
webcam_motion on
webcam_maxrate 1
webcam_localhost off
quiet on
on_event_start /usr/local/bin/cam_event.sh
```

Figura 3.4: Archivo motion.conf

3.3.1.1 Descripción del Archivo MOTION.CONF

A continuación se realiza la explicación de los parámetros de configuración utilizados para cumplir el propósito propuesto. Cabe recalcar que el usuario tiene la posibilidad de ajustar los parámetros a conveniencia.

daemon. – Si se quiere que inicie en modo demonio (en segundo plano), para las primeras pruebas es recomendable configurarlo “*off*”, para ver la ejecución del programa en el terminal.

➤ Opciones del dispositivo de Captura:

videodevice. – Dispositivo de video utilizado para la captura de imágenes.

width. – La anchura de la imagen (píxeles), esto dependerá de lo que soporte la cámara web.

height. – La altura de la imagen (píxeles), esto dependerá de lo que soporte la cámara web.

framerate. – Número máximo de imágenes por segundo que se capturarán. Eso dependerá de la calidad que se le quiera dar a la imagen de video.

auto_brightness. – Si se quiere que Motion regule el brillo de la cámara. Sólo recomendable “*on*” para cámaras sin auto brillo.

brightness. – El brillo. Si se tiene “*auto_brightness*” activado, este será el valor medio en el que se base Motion para ajustar el brillo.

contrast. – Establece el contraste de la cámara web. Si es 0, utiliza el propio de la cámara.

saturation. – Establece la saturación de la cámara web. Si es 0, utiliza el propio de la cámara.

➤ Opciones de la Detección de Movimiento:

threshold. – Es la sensibilidad, indica el número de píxeles cambiados en una imagen para que se active la grabación del movimiento (Por defecto: 1500). Con valores más bajos será más sensible al movimiento y viceversa.

noise_tune. – Activa el ajuste automático del nivel de ruido. Esta opción es muy potente, con ella prácticamente se elimina los falsos movimientos.

lightswitch. – Ignorar cambios repentinos de intensidad lumínica en un área porcentual de la imagen. Rango válido: 0 – 100.

minimum_motion_frames. – Tiempo mínimo en segundos entre los marcos de captura de la cámara. Por defecto: 0 = desactivado - la tasa de captura está dada por la tasa de fotogramas de la cámara.

- Opciones de guardado de imagen:

quality. – La calidad de las imágenes JPEG en porcentaje.

output_normal. – Manera en que se guarda las imágenes mientras detecta movimiento. Valores validos: on, off, first, best. Con 'on' se guardan todas las imágenes. Cuando se indica 'first', sólo la primera imagen de cada evento se guarda. Se captura la imagen con más movimiento cuando se establece en 'best'. Con 'off' no se guarda ninguna imagen.

- Opciones relacionadas con FFMPEG, para el archivo de video:

ffmpeg_cap_new. – Para utilizar ffmpeg para codificar video mpeg en tiempo real.

ffmpeg_video_codec. – Códec a utilizar por ffmpeg para la compresión de video. Los formatos soportados son: mpeg1, mpeg4, msmpeg4, swf.

- Opciones del Texto de la Pantalla:

locate. – Permite localizar y dibujar un cuadro alrededor del objeto en movimiento, sólo válido para imágenes y no para videos.

text_right. – Texto definido por el usuario en cada imagen. Superpuesto en la esquina inferior derecha.

text_left. – Texto definido por el usuario en cada imagen. Superpuesto en la esquina inferior izquierda.

➤ Opciones de Directorio Destino de archivos y videos:

target_dir. – El directorio donde Motion guardará los videos y las imágenes.

➤ Opciones de Servidor de la Cámara Web en Directo:

webcam_port. – El puerto donde se podrá observar la cámara web en vivo a través de un mini servidor HTTP creado por Motion.

webcam_quality. – La calidad en que se mostrarán las imágenes.

webcam_maxrate. – Limita la tasa de fotogramas de la cámara.

webcam_localhost. – Si se quiere restringir el uso del servidor solo localmente.

webcam_motion. – Si se establece a *'on'* Motion envía el flujo de webcam a 1 imagen por segundo cuando no se detecta ningún movimiento. Cuando se detecta movimiento el flujo se define por *webcam_maxrate*. Cuando esta *'off'* el flujo de la cámara web siempre se ejecuta según lo definido por *webcam_maxrate*.

➤ Opciones de Comandos Externos, Alertas y Registro:

quiet. – Emite un pitido de salida cuando detecta movimiento.

on_event_start. – Comando para ser ejecutado cuando un evento empieza. Un evento comienza al primer movimiento detectado después de un periodo de no detección de movimiento definido por *gap*.

3.3.2 Configuración del Softphone X-Lite

Como primer paso se ubica en el directorio donde se instaló el X-Lite y luego se procede a ejecutar el programa:

```
# cd /usr/src/xten-xlite/
```

```
# ./xtensoftphone
```

Enseguida aparece la interfaz gráfica del X-Lite, como lo muestra la Figura 3.5.



Figura 3.5: Softphone X-Lite

Luego se da un clic en el ícono de *MENU*, y aparece la ventana del menú principal del X-Lite, mostrada en la Figura 3.6.



Figura 3.6: Menú Principal de X-Lite

Se ingresa a la Configuración del Sistema (*System Settings*) y posteriormente a la configuración del *SIP Proxy*, tal como se indica en la Figura 3.7.



Figura 3.7: Ingreso a la Configuración del SIP Proxy

Se ingresa a la configuración por defecto (*Default*) y se realiza la configuración mostrada en la Figura 3.8.

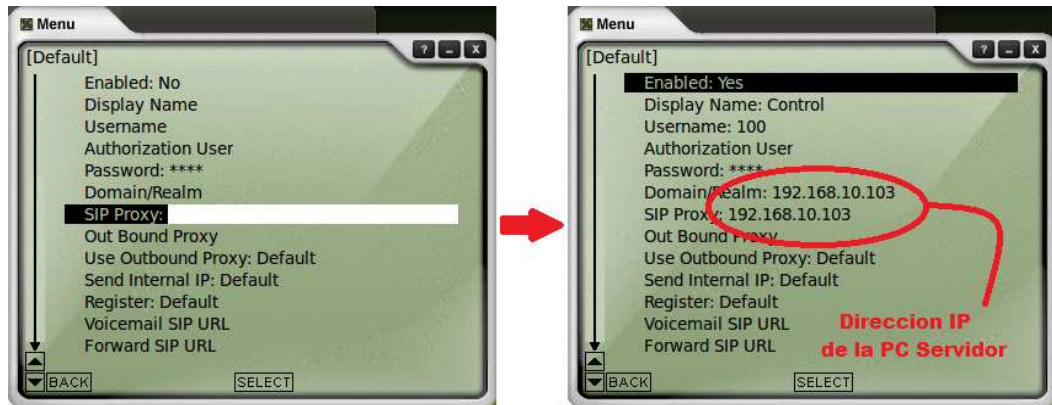


Figura 3.8: Configuración de X-Lite

En el campo de *Password* se digita: **100**

En *Domain/Realm* se escribe la dirección IP del Host.

En *SIP Proxy* se escribe la dirección IP del servidor SIP, en este caso es el servidor Asterisk.

Como se está configurando una extensión en el servidor Asterisk, los campos son iguales.

Para obtener la dirección IP del PC, se ejecuta en un Terminal:

ifconfig eth0

```
Archivo Editar Ver Terminal Ayuda
rodrigo@rodrigo-laptop:/$ ifconfig eth0
eth0      Link encap:Ethernet direcciónHW 00:23:8b:c5:bd:86
          Direc. inet:192.168.10.103 Difus.:192.168.10.255 Másc:255.255.255.0
          Dirección inet6: fe80::223:8bff:fec5:bd86/64 Alcance:Enlace
          ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST MTU:1500 Métrica:1
          Paquetes RX:175 errores:0 perdidos:0 overruns:0 frame:0
          Paquetes TX:182 errores:0 perdidos:0 overruns:0 carrier:0
          colisiones:0 long.colaTX:1000
          Bytes RX:17168 (17.1 KB) TX bytes:15078 (15.0 KB)
          Interrupción:30

rodrigo@rodrigo-laptop:/$
```

Figura 3.9: Dirección IP del PC

3.3.3 Configuración de ASTERISK

Para que Asterisk controle el funcionamiento del sistema de vigilancia se tiene que configurar 3 archivos:

SIP.CONF – Aquí se declara la única extensión que se necesita para controlar la activación y desactivación del sistema de vigilancia.

EXTENSIONS.CONF – Aquí se configura el plan de marcado que seguirá Asterisk. Este archivo controla el comportamiento de las llamadas.

MOBILE.CONF – Aquí se configura el dispositivo Bluetooth que utiliza Asterisk para la conexión con el celular gateway, mismo que también fue definido aquí.

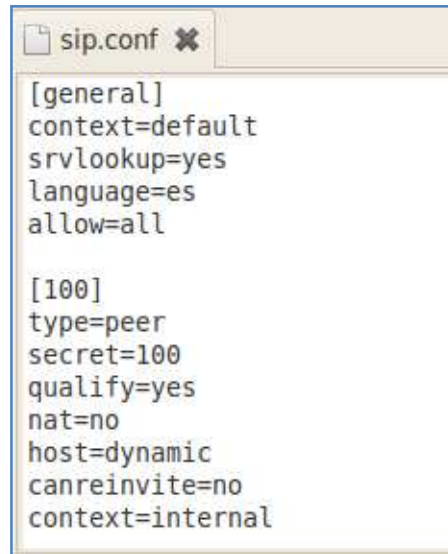
3.3.3.1 Configuración del Archivo SIP.CONF

Primero se realiza una copia del archivo de configuración existente:

```
# cp /etc/asterisk/sip.conf /etc/asterisk/sip.conf.backup
```

Luego se edita el archivo de configuración de extensiones SIP “*sip.conf*” y se lo configura tal y como se muestra en la Figura 3.10.

```
# gedit /etc/asterisk/sip.conf
```



```
[general]
context=default
srvlookup=yes
language=es
allow=all

[100]
type=peer
secret=100
qualify=yes
nat=no
host=dynamic
canreinvite=no
context=internal
```

Figura 3.10: Archivo sip.conf

3.3.3.1.1 Descripción del Archivo SIP.CONF

[general] ; Etiqueta que introduce la parte general de la configuración.

context=default ; Define el contexto predeterminado para las extensiones SIP, en caso de no configurarlo dentro de la misma.

srvlookup=yes ; Permite hacer búsquedas de registros DNS y SRV para llamadas SIP salientes basadas en los nombres de dominio.

language=es ; Si se ha instalado locuciones en más de un idioma, aquí se puede definir el idioma a usar en la extensión. En este caso se instalaron las locuciones en español, y se lo especifica con el prefijo 'es'.

allow=all ; Configura los códec a usar, en este caso se habilita todos.

[100] ; Número de la extensión.

type=peer ; Tipo de extensión. Puede ser *“friend”*, *“user”* o *“peer”*. Con *“friend”* puede hacer y recibir llamadas, *“user”* solo recibir y *“peer”* solo puede realizar llamadas.

secret=100 ; Define la contraseña de la extensión.

qualify=yes ; Determina el tiempo de respuesta de una extensión e indica si está alcanzable o no.

nat=no ; Si la extensión se conecta al servidor Asterisk detrás de un firewall hay que poner *“yes”*, caso contrario *“no”*.

host=dynamic ; Si la extensión se conecta remotamente cambiando continuamente su dirección IP se pone este parámetro. Recomendable.

canreinvite=no ; Si se quiere que la extensión intente conectarse directamente con la extensión llamada, ponemos *“yes”*. Si se requiere que Asterisk haga de puente entre las dos extensiones, entonces *“no”*.

context=internal ; El contexto que usa la extensión.

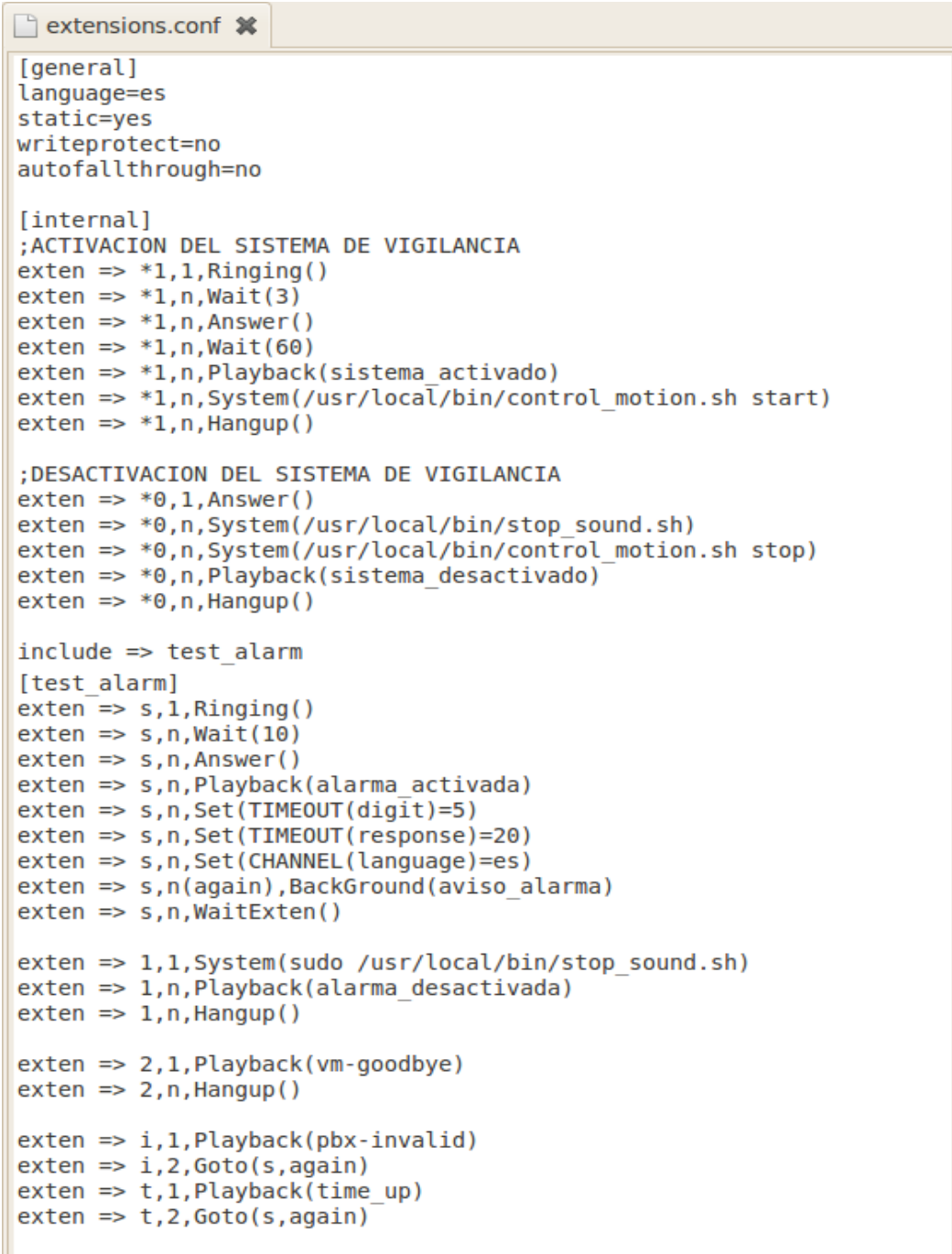
3.3.3.2 Configuración del Archivo *EXTENSIONS.CONF*

Primero se realiza una copia del archivo de configuración existente:

```
#cp /etc/asterisk/extensions.conf /etc/asterisk/extensions.conf.backup
```

Luego se edita el archivo de configuración del plan de marcado, *extensions.conf*

gedit /etc/asterisk/extensions.conf



```
[general]
language=es
static=yes
writeprotect=no
autofallthrough=no

[internal]
;ACTIVACION DEL SISTEMA DE VIGILANCIA
exten => *1,1,Ringing()
exten => *1,n,Wait(3)
exten => *1,n,Answer()
exten => *1,n,Wait(60)
exten => *1,n,Playback(sistema_activado)
exten => *1,n,System(/usr/local/bin/control_motion.sh start)
exten => *1,n,Hangup()

;DESACTIVACION DEL SISTEMA DE VIGILANCIA
exten => *0,1,Answer()
exten => *0,n,System(/usr/local/bin/stop_sound.sh)
exten => *0,n,System(/usr/local/bin/control_motion.sh stop)
exten => *0,n,Playback(sistema_desactivado)
exten => *0,n,Hangup()

include => test_alarm
[test_alarm]
exten => s,1,Ringing()
exten => s,n,Wait(10)
exten => s,n,Answer()
exten => s,n,Playback(alarma_activada)
exten => s,n,Set(TIMEOUT(digit)=5)
exten => s,n,Set(TIMEOUT(response)=20)
exten => s,n,Set(CHANNEL(language)=es)
exten => s,n(again),Background(aviso_alarma)
exten => s,n,WaitExten()

exten => 1,1,System(sudo /usr/local/bin/stop_sound.sh)
exten => 1,n,Playback(alarma_desactivada)
exten => 1,n,Hangup()

exten => 2,1,Playback(vm-goodbye)
exten => 2,n,Hangup()

exten => i,1,Playback(pbx-invalid)
exten => i,2,Goto(s,again)
exten => t,1,Playback(time_up)
exten => t,2,Goto(s,again)
```

Figura 3.11: Archivo *extensions.conf*

3.3.3.2.1 Descripción del Archivo *EXTENSIONS.CONF*

[general] ; Etiqueta de inicio de la configuración general.

language=es ; Ajusta el lenguaje a utilizarse.

static=yes ; Permite guardar el plan de marcado con el comando “*save dialplan*” desde la consola de Asterisk.

writeprotect=no ; Protección frente a escritura. Con ‘no’, se permite modificar los archivos de configuración desde la consola de Asterisk.

autofallthrough=yes ; Si esta opción es activada, cuando una extensión termine de ejecutar sus prioridades, le permite terminar la llamada inmediatamente con *BUSY*, *CONGESTION*, o *HANGUP*.

[internal] ; Etiqueta de encabezado del contexto *INTERNAL*.

Dentro de este contexto se definió únicamente 2 extensiones, las cuales se encargan de Activar y Desactivar el Sistema de Vigilancia.

La extensión (*1) será la que active el sistema de la siguiente manera: Primeramente ejecuta un sonido de timbrado durante 3 segundos, luego Asterisk toma la llamada y espera 60 segundos, este tiempo es el que se tiene para salir de la habitación u oficina a fin de no activar la alarma. Después de transcurrido dicho tiempo se ejecuta un archivo de audio pregrabado, el cual anuncia la activación del sistema.

Posteriormente se inicia el software de detección de movimientos, a través de la ejecución del script de control del mismo. Por último se cuelga la llamada.

La extensión (*0) será la que desactive el sistema. Primero se hace que Asterisk acoja la llamada, después se detiene la existencia de un posible sonido de alarma a través de la ejecución del script “*stop_sound.sh*”, luego se finaliza el programa Motion con ayuda del script de control del mismo. Finalmente se reproduce un archivo de audio indicando que el sistema ha sido desactivado y se termina la llamada.

include => test_alarm ; se incluye el contexto *TEST_ALARM*.

[test_alarm] ; etiqueta de encabezado del contexto *TEST_ALARM*.

Este contexto es el que controla la llamada realizada hacia el teléfono móvil cuando se detecta algún intruso en la habitación u oficina.

Al inicio se ejecuta un tono de llamada durante 10 segundos, luego Asterisk responde y se reproduce un archivo de audio para indicar que la alarma ha sido activada. Posteriormente se configuran algunos parámetros de la llamada en curso:

Set(TIMEOUT(digit)=5) ; Configura el tiempo válido entre la pulsación de uno y otro dígito. En este caso 5 segundos.

Set(TIMEOUT(response)=20) ; Dispone que el tiempo de espera para marcar una extensión sea de 20 segundos.

Set(CHANNEL(language)=es) ; Ajusta el lenguaje a utilizar por este canal durante la llamada. En este caso el español.

Luego se reproduce un archivo de audio, el cual da 2 opciones: la primera opción es la de apagar el sonido de alarma marcando “1”, y la segunda es la de marcar “2” para terminar la llamada sin detener la alarma sonora activada.

WaitExten() ; espera a que una extensión sea marcada.

Al escoger la opción “1”, se detiene el sonido de alarma con la ejecución del script “*stop_sound.sh*”, luego se reproduce un archivo de audio indicando que la alarma sonora ha sido apagada y se termina la llamada.

En la opción “2”, simplemente se reproduce un archivo de audio y se termina la llamada.

Adicionalmente se definieron dos extensiones para control del contestador automático de llamada descrito en los párrafos anteriores.

La extensión “*i*”, la cual toma la llamada en caso de que se marque una extensión no definida en este contexto y la envía a la extensión “*s*” en la instrucción etiquetada como “*again*”.

Y la extensión “t”, misma que acoge la llamada cuando se cumple el tiempo de espera para que se marque alguna extensión y la ubica en la extensión “s” con prioridad etiquetada como “again”.

3.3.3.3 Configuración del Archivo MOBILE.CONF

Primero se realiza una copia del archivo de configuración existente:

```
# cp /etc/asterisk/mobile.conf /etc/asterisk/mobile.conf.backup
```

Luego se edita el archivo de configuración de medios Bluetooth, “mobile.conf” y se lo deja como el mostrado en la Figura 3.12.

```
# gedit /etc/asterisk/mobile.conf
```



```
mobile.conf x
[general]
interval=45

[adapter]
id=blue
address=00:24:7E:61:06:13

[Nokia6300]
address=00:1A:DC:8E:B9:94
port=13
context=test_alm
adapter=blue
nocallsetup=yes
```

Figura 3.12: Archivo mobile.conf

3.3.3.3.1 Descripción del Archivo MOBILE.CONF

A continuación se explica cada uno de los parámetros configurados en este archivo, y adicionalmente se detallan los pasos a seguir para obtener algunos datos importantes. No olvidar modificar los campos de acuerdo a su necesidad.

[general] ; Etiqueta de inicio de la configuración global.

Interval =45 ; Indica el número de segundos en tratar de conectarse a un dispositivo.

[adapter] ; Etiqueta que indica el adaptador bluetooth a configurar.

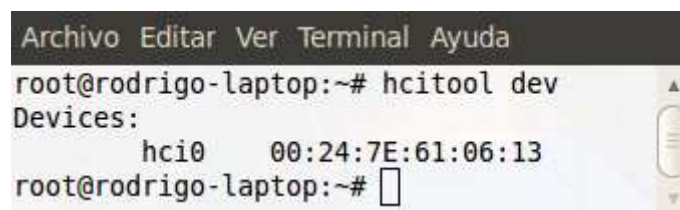
Id=blue ; Identificador único del adaptador.

address=00:24:7E:61:06:13 ; Indica la Mac del dispositivo Bluetooth del computador.

Para conocer la dirección Mac del dispositivo bluetooth en el computador, se abre un Terminal y se ejecuta el siguiente comando:

hcitool dev

Aparece algo como lo mostrado en la Figura 3.13.



```
Archivo Editar Ver Terminal Ayuda
root@rodrigo-laptop:~# hcitool dev
Devices:
    hci0    00:24:7E:61:06:13
root@rodrigo-laptop:~#
```

Figura 3.13: Dirección MAC del Dispositivo Bluetooth

[Nokia_6300] ; Nombre del teléfono móvil utilizado como gateway.

address=00:1A:DC:8E:B9:94 ; Indica la Mac del teléfono móvil.

Para obtener el nombre y la dirección Mac del teléfono móvil que va servir de gateway, se debe activar el bluetooth del teléfono y ponerlo como visible. Y luego ejecutar el comando (**# hcitool scan**) en un Terminal, y luego aparece algo como lo mostrado en la Figura 3.14.



```
Archivo Editar Ver Terminal Ayuda
root@rodrigo-laptop:~# hcitool scan
Scanning ...
    00:1A:DC:8E:B9:94      Nokia_6300
root@rodrigo-laptop:~#
```

Figura 3.14: Escaneo de Dispositivos Bluetooth

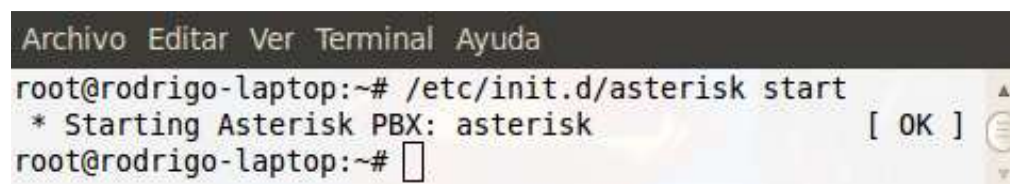
port=13 ; Este es el puerto del teléfono móvil al cual se comunica el servidor Asterisk.

Para saber que puerto del teléfono móvil se utiliza, se debe activar el bluetooth en el celular y en la PC, y seguir los siguientes pasos:

Se abre un Terminal y se inicia Asterisk ejecutando el siguiente comando:

/etc/init.d/asterisk start

Debe observarse en el Terminal algo como lo mostrado en la Figura 3.15.

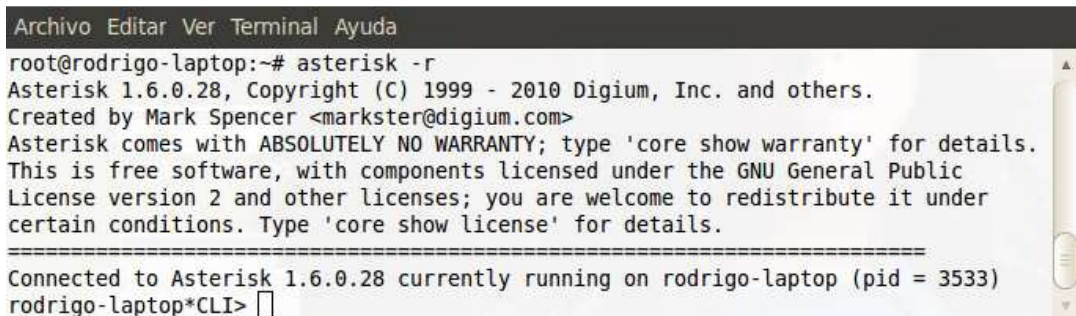


```
Archivo Editar Ver Terminal Ayuda
root@rodrigo-laptop:~# /etc/init.d/asterisk start
* Starting Asterisk PBX: asterisk      [ OK ]
root@rodrigo-laptop:~#
```

Figura 3.15: Inicio del Servicio de Asterisk

Luego se ingresa a la consola remota de Asterisk, como lo ilustra la Figura 3.16, al ejecutar el siguiente comando:

asterisk -r



```
Archivo Editar Ver Terminal Ayuda
root@rodrigo-laptop:~# asterisk -r
Asterisk 1.6.0.28, Copyright (C) 1999 - 2010 Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
=====
Connected to Asterisk 1.6.0.28 currently running on rodrigo-laptop (pid = 3533)
rodrigo-laptop*CLI> 
```

Figura 3.16: Ingreso a la Consola Remota de Asterisk

Una vez dentro del CLI de Asterisk, se ejecuta el siguiente comando:

CLI > mobile search

Luego de unos segundos aparece algo como lo mostrado en la Figura 3.17, y se procede a anotar el valor del puerto, que en este caso es 13.



```
Archivo Editar Ver Terminal Ayuda
rodrigo-laptop*CLI> mobile search
Address          Name          Usable Type  Port
00:1A:DC:8E:B9:94 Nokia_6300    Yes   Phone  13
rodrigo-laptop*CLI> 
```

Figura 3.17: Obtención del Puerto del Teléfono Móvil

Luego se procede a salir de la consola remota:

CLI > exit

Y finalmente se termina con el proceso Asterisk, tal y como se muestra en la Figura 3.18, al ejecutar el siguiente comando:

/etc/init.d/asterisk stop

```
Archivo Editar Ver Terminal Ayuda
rodrigo-laptop*CLI> exit
root@rodrigo-laptop:~# /etc/init.d/asterisk stop
* Stopping Asterisk PBX: asterisk [ OK ]
root@rodrigo-laptop:~# █
```

Figura 3.18: Terminación del Servicio de Asterisk

Continuando con la explicación del archivo *“mobile.conf”*:

context=test_alarm ; Contexto del plan de marcado asociado al teléfono móvil durante su utilización.

adapter=blue ; Identificador del adaptador asociado al dispositivo móvil.

nocallsetup=yes ; activar sólo si el teléfono informa que admite la notificación de progreso de llamada, pero no se quiere que lo haga.

3.3.4 Creación de **SCRIPTS**

A continuación se crearán los scripts utilizados en el Sistema de Video Vigilancia, y se realiza una breve descripción de cada uno de los siguientes:

- ❖ Script de Control de Motion.
- ❖ Script de Gestión de Eventos.
- ❖ Script de Ejecución de Alarma.
- ❖ Script de Empezar el Sonido de Alarma.
- ❖ Script de Detener el Sonido de Alarma.

3.3.4.1 Script de Control de Motion

Con este script se controla la inicialización y terminación del software de Detección de Movimiento “*MOTION*”.

Primero se ubica en el directorio `/usr/local/bin`:

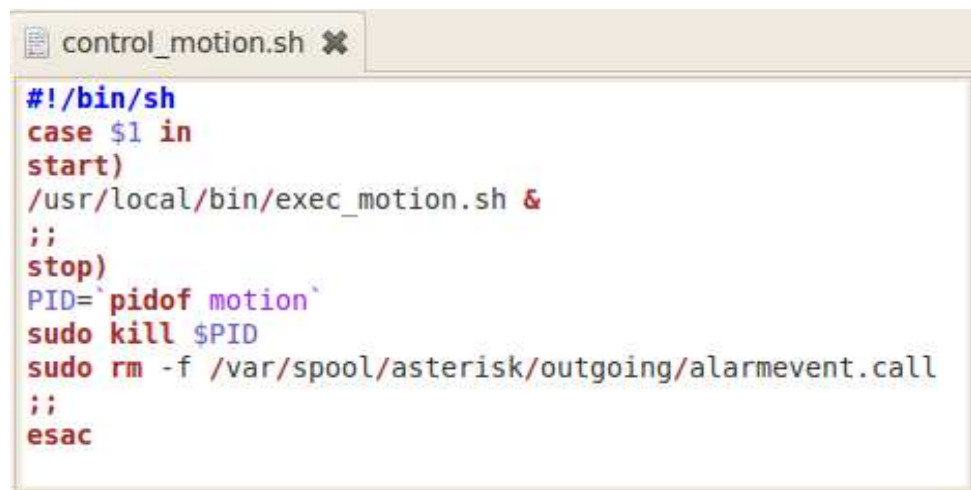
```
# cd /usr/local/bin
```

Ahora se crea el archivo “`control_motion.sh`”:

```
# touch control_motion.sh
```

Luego se procede a editar el script de control, tal y como se muestra en la Figura 3.19, ejecutando el siguiente comando:

```
# gedit control_motion.sh
```

A screenshot of a terminal window titled 'control_motion.sh'. The window contains the following shell script code:

```
#!/bin/sh
case $1 in
start)
/usr/local/bin/exec_motion.sh &
;;
stop)
PID=`pidof motion`
sudo kill $PID
sudo rm -f /var/spool/asterisk/outgoing/alarmevent.call
;;
esac
```

Figura 3.19: Archivo `control_motion.sh`

3.3.4.1.1 Descripción del Archivo `CONTROL_MOTION.SH`

Este script utiliza la estructura “`case`” y consta de dos opciones posibles para su ejecución.

La primera opción es **START**, en la cual simplemente se ejecuta en segundo plano un script llamado “*exec_motion.sh*”.

La segunda opción es **STOP**, aquí se obtiene el identificador del proceso de nombre *MOTION*. Luego termina ese proceso, y finalmente se elimina el archivo de llamada creado en Asterisk durante la detección de movimiento.

3.3.4.2 Script de Ejecución del Programa *MOTION*

Este script simplemente ejecuta el programa *MOTION* como súper usuario, es decir como usuario *root*.

Recordar que este script se ejecutará en segundo plano, puesto que con esto se evita que al realizar la llamada de activación del *MOTION* halla conflictos entre Asterisk y Motion.

Se procede a crear y editar el archivo “*exec_motion.sh*”:

```
# touch /usr/local/bin/exec_motion.sh
```

```
# gedit /usr/local/bin/exec_motion.sh
```



```
#!/bin/sh
sudo /usr/bin/motion
```

Figura 3.20: Archivo *exec_motion.sh*

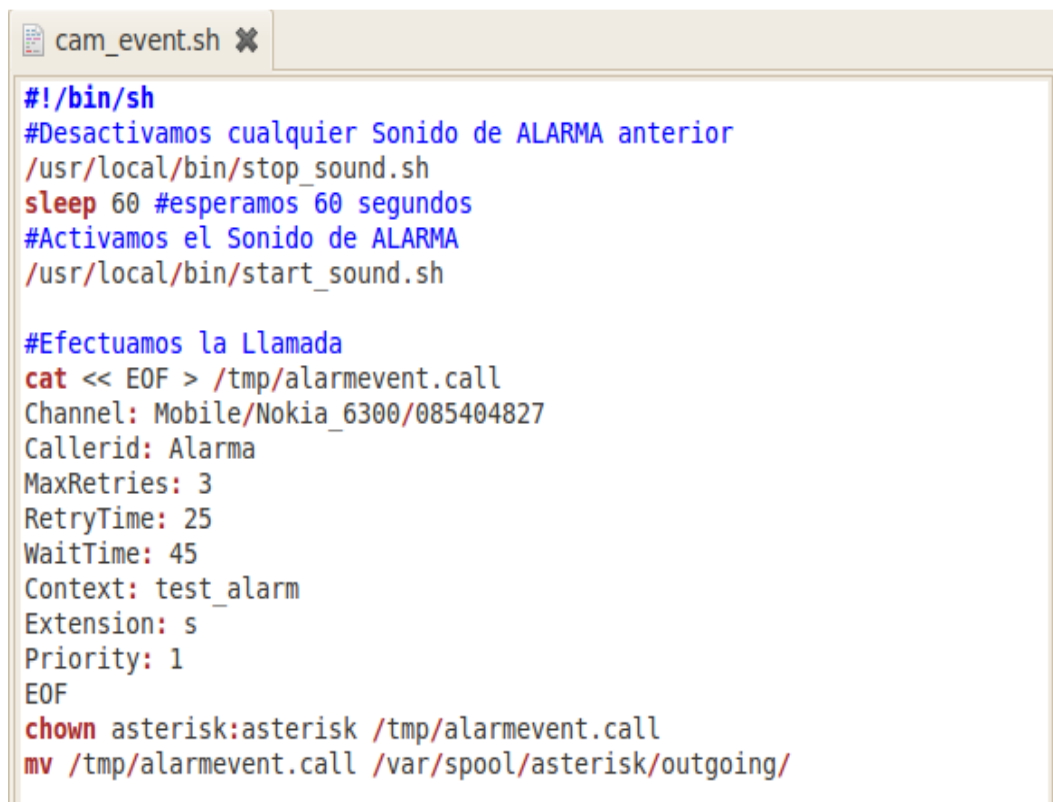
3.3.4.3 Script de Gestión de Eventos

Este script se ejecuta al iniciar un evento, es decir cuando MOTION detecta movimiento.

Se procede a crear y editar el archivo “*cam_event.sh*”, tal y como se muestra en la Figura 3.21:

```
# touch /usr/local/bin/cam_event.sh
```

```
# gedit /usr/local/bin/cam_event.sh
```

The image shows a terminal window titled 'cam_event.sh'. The script content is as follows:

```
#!/bin/sh
#Desactivamos cualquier Sonido de ALARMA anterior
/usr/local/bin/stop_sound.sh
sleep 60 #esperamos 60 segundos
#Activamos el Sonido de ALARMA
/usr/local/bin/start_sound.sh

#Efectuamos la Llamada
cat << EOF > /tmp/alarmevent.call
Channel: Mobile/Nokia_6300/085404827
Callerid: Alarma
MaxRetries: 3
RetryTime: 25
WaitTime: 45
Context: test_alarm
Extension: s
Priority: 1
EOF
chown asterisk:asterisk /tmp/alarmevent.call
mv /tmp/alarmevent.call /var/spool/asterisk/outgoing/
```

Figura 3.21: Archivo *cam_event.sh*

3.3.4.3.1 Descripción del Archivo CAM_EVENT.SH

Primero se ejecuta el script *“stop_sound.sh”*, para detener un posible sonido de alarma activado anteriormente, esto para evitar que se solapen dichos sonidos y simplemente se ejecute un único sonido.

Luego se realiza una espera de 60 segundos con el objetivo de que si se activa la alarma, se pueda detener la ejecución de la llamada y se apague oportunamente el sistema.

Posteriormente se ejecuta el script *“start_sound.sh”* para activar la alarma sonora.

Después se crea un archivo de llamada (.call), con la información necesaria para que Asterisk lo interprete correctamente. Recordar cambiar los datos del nombre del bluetooth del teléfono móvil gateway, y el número de teléfono celular a llamar, tal y como lo ilustra la Figura 3.22.



Figura 3.22: Descripción del cam_event.sh

Finalmente se mueve el archivo creado al siguiente directorio: /var/spool/asterisk/outgoing/, para que Asterisk al encontrar algún archivo nuevo realice la llamada con los datos dentro del archivo.

3.3.4.4 Script de Ejecución de Alarma

Este script es ejecutado por el script de gestión de eventos explicado en el punto anterior. Y simplemente realiza la ejecución de un script adicional en segundo plano, esto para no interrumpir los procesos activos tales como Asterisk y Motion.

Se ubica en el directorio */usr/local/bin*, y se crea el archivo “start_sound.sh”, al ejecutar los siguientes comandos:

```
# cd /usr/local/bin
```

```
# touch start_sound.sh
```

Se edita el archivo “start_sound.sh”, como lo muestra la Figura 3.23.

```
# gedit start_sound.sh
```

A screenshot of a text editor window titled 'start_sound.sh'. The window contains two lines of code: a shebang line '#!/bin/sh' and a command line '/usr/local/bin/play_sound.sh &'. The text is color-coded: blue for the shebang, red for the command, and black for the ampersand.

```
#!/bin/sh
/usr/local/bin/play_sound.sh &
```

Figura 3.23: Archivo start_sound.sh

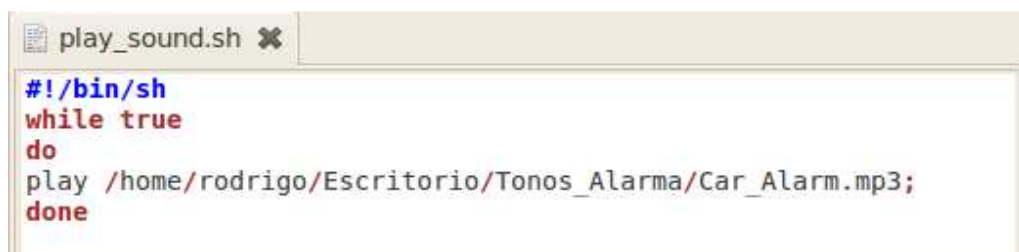
3.3.4.5 Script de Empezar el Sonido de Alarma

Este script realiza la reproducción repetitiva e indefinida de un archivo de audio por los altavoces del PC. Recordar cambiar el directorio y el fichero a reproducir según vuestras necesidades.

Se realiza lo siguiente, desde el Terminal de Linux:

```
# touch /usr/local/bin/play_sound.sh
```

```
# gedit /usr/local/bin/play_sound.sh
```

A screenshot of a text editor window titled 'play_sound.sh'. The window contains a shell script with a while loop. The code is color-coded: blue for the shebang, red for the loop keywords, and black for the command and the end of the loop.

```
#!/bin/sh
while true
do
play /home/rodrigo/Escritorio/Tonos_Alarma/Car_Alarm.mp3;
done
```

Figura 3.24: Archivo play_sound.sh

3.3.4.6 Script de Detener el Sonido de Alarma

Al ejecutar este script se termina con la reproducción del sonido de alarma. Primero se obtiene el PID del proceso ejecutado en segundo plano, es decir el *“play_sound.sh”*. Y luego se termina con ese proceso.


Se ubica en el directorio */usr/local/bin*:

```
# cd /usr/local/bin
```

Se crea y edita el archivo *“stop_sound.sh”*, al ejecutar los siguientes comandos, y se deja el archivo como lo muestra la Figura 3.25.

```
# touch stop_sound.sh
```

```
# gedit stop_sound.sh
```

A screenshot of a terminal window with a title bar that says "stop_sound.sh". The terminal content shows a shell prompt followed by three lines of code: a shebang line, a command to find the PID of the current process, and a command to kill it.

```
#!/bin/bash
PID=$(pgrep -u $LOGNAME -o -x play_sound.sh)
sudo kill -9 $PID
```

Figura 3.25: Archivo stop_sound.sh

CAPITULO 4

FUNCIONAMIENTO Y PRUEBAS DEL PROYECTO

4.1 Inicialización de Asterisk

Lo primero que se realiza es iniciar el servicio de Asterisk ejecutando el siguiente comando en un Terminal:

```
# /etc/init.d/asterisk start
```

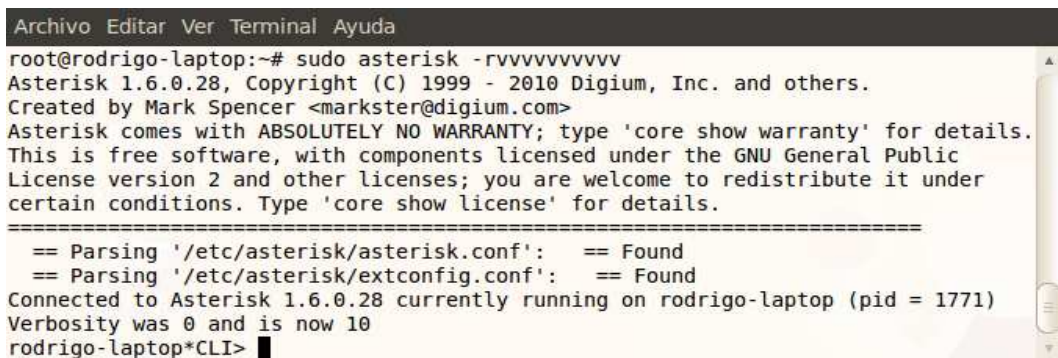
A screenshot of a terminal window with a dark title bar containing 'Archivo Editar Ver Terminal Ayuda'. The terminal text shows the command `root@rodrigo-laptop:~# /etc/init.d/asterisk start` being executed. The output is `* Starting Asterisk PBX: asterisk` followed by a cursor and a `[OK]` status indicator. The prompt returns to `root@rodrigo-laptop:~#`.

```
Archivo Editar Ver Terminal Ayuda
root@rodrigo-laptop:~# /etc/init.d/asterisk start
* Starting Asterisk PBX: asterisk [ OK ]
root@rodrigo-laptop:~#
```

Figura 4.1: Inicio de Asterisk

Luego se procede a ingresar a la consola remota de Asterisk, al ejecutar lo siguiente:

```
# asterisk -rvvvvvvvvvv
```

A screenshot of a terminal window with a dark title bar containing 'Archivo Editar Ver Terminal Ayuda'. The terminal text shows the command `root@rodrigo-laptop:~# sudo asterisk -rvvvvvvvvvv` being executed. The output displays the Asterisk version (1.6.0.28), copyright information, and a license notice. It then shows configuration files being parsed and the connection to the Asterisk service on the local machine (pid = 1771). The prompt changes to `rodrigo-laptop*CLI>`.

```
Archivo Editar Ver Terminal Ayuda
root@rodrigo-laptop:~# sudo asterisk -rvvvvvvvvvv
Asterisk 1.6.0.28, Copyright (C) 1999 - 2010 Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
=====
== Parsing '/etc/asterisk/asterisk.conf': == Found
== Parsing '/etc/asterisk/extconfig.conf': == Found
Connected to Asterisk 1.6.0.28 currently running on rodrigo-laptop (pid = 1771)
Verbosity was 0 and is now 10
rodrigo-laptop*CLI>
```

Figura 4.2: Ingreso a la Consola Remota de Asterisk

4.2 Activación del Softphone

Se procede a iniciar el softphone X-Lite al ejecutar los siguientes comandos en un Terminal diferente al paso anterior:

```
# cd /usr/src/xten-xlite/
```

```
# ./xtensoftphone
```

Luego se espera unos minutos para que se registre el softphone en el servidor Asterisk y se observará en la consola de Asterisk y en la pantalla del softphone algo como lo mostrado en la Figura 4.3

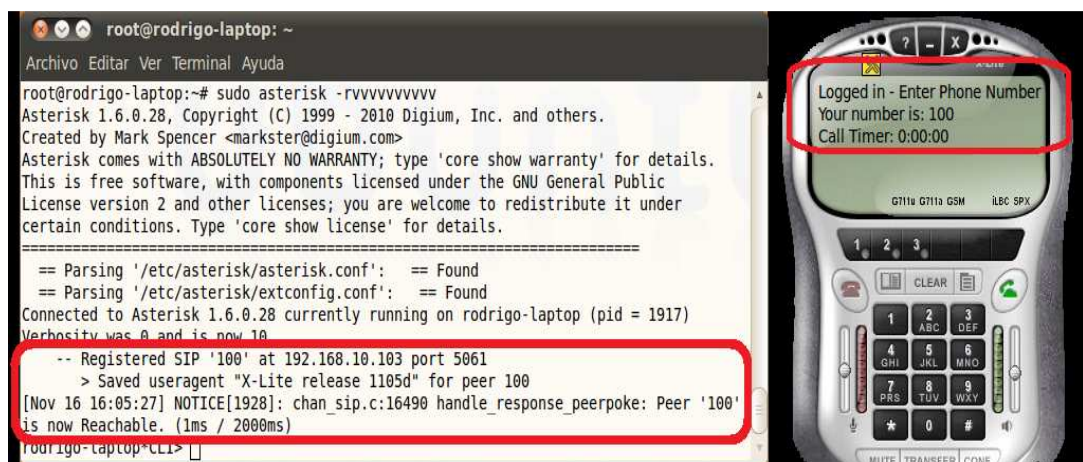


Figura 4.3: Registro de X-Lite en Asterisk

Luego se procede a verificar el correcto registro del usuario SIP configurado en Asterisk, a través de la ejecución del siguiente comando en la consola de Asterisk:

```
CLI> sip show peers
```

Y se observa la única extensión configurada, tal y como lo muestra la Figura 4.4


```

rodrigo-laptop*CLI> sip show peers
Name/username      Host                Dyn Nat ACL Port      Status
100/100            192.168.10.103     D                5061      OK (2 ms)
1 sip peers [Monitored: 1 online, 0 offline Unmonitored: 0 online, 0 offline]
rodrigo-laptop*CLI> █

```

Figura 4.4: Usuarios SIP

4.3 Conexión de Asterisk con el Gateway

Se procede a encender el bluetooth del teléfono móvil que va a estar conectado con el servidor de Asterisk.

Y luego de unos minutos se confirma la conexión del dispositivo al visualizar en la consola de Asterisk mensajes de confirmación tal y como lo ilustra la Figura 4.5

```

rodrigo-laptop*CLI>
-- Bluetooth Device Nokia_6300 has connected.
-- Bluetooth Device Nokia_6300 initialised and ready.
rodrigo-laptop*CLI> █

```

Figura 4.5: Conexión de Asterisk con el Gateway

4.4 Activación del Sistema de Vigilancia

Ahora que Asterisk se encuentra operando se procede a la activación del Sistema de Vigilancia marcando en el softphone la extensión [*1] y se observa en la consola de Asterisk algo como lo mostrado en la Figura 4.6



```

Archivo Editar Ver Terminal Ayuda
rodrigo-laptop*CLI>
== Using SIP RTP CoS mark 5
-- Executing [*1@internal:1] Ringing("SIP/100-00000008", "") in new stack
-- Executing [*1@internal:2] Wait("SIP/100-00000008", "3") in new stack
-- Executing [*1@internal:3] Answer("SIP/100-00000008", "") in new stack
-- Executing [*1@internal:4] Wait("SIP/100-00000008", "60") in new stack
-- Executing [*1@internal:5] Playback("SIP/100-00000008", "sistema activado") in new stack
-- <SIP/100-00000008> Playing 'sistema activado.gsm' (language 'es')
-- Executing [*1@internal:6] System("SIP/100-00000008", "/usr/local/bin/control_motion.sh start") in new stack
rodrigo-laptop*CLI>

```

Figura 4.6: Activación del Sistema de Vigilancia

A continuación se observa la ejecución en consola del archivo “*extensions.conf*”, a través del cual se ejecuta el script “*control_motion.sh*” para activar el software MOTION, tal y como se describe en la Figura 4.7.

```

Archivo Editar Ver Terminal Ayuda
rodrigo-laptop*CLI>
== Using SIP RTP CoS mark 5
-- Executing [*1@internal:1] Ringing("SIP/100-00000008", "") in new stack
-- Executing [*1@internal:2] Wait("SIP/100-00000008", "3") in new stack
-- Executing [*1@internal:3] Answer("SIP/100-00000008", "") in new stack
-- Executing [*1@internal:4] Wait("SIP/100-00000008", "60") in new stack
-- Executing [*1@internal:5] Playback("SIP/100-00000008", "sistema activado") in new stack
-- Executing [*1@internal:6] System("SIP/100-00000008", "/usr/local/bin/control_motion.sh start") in new stack
rodrigo-laptop*CLI>

```



```

control_motion.sh ✕
#!/bin/sh
case $1 in
start)
/usr/local/bin/exec_motion.sh &
;;
stop)
PID=`pidof motion`
sudo kill $PID
sudo rm -f /var/spool/asterisk/outgoing/alarmevent.call
;;
esac

```

➔

```

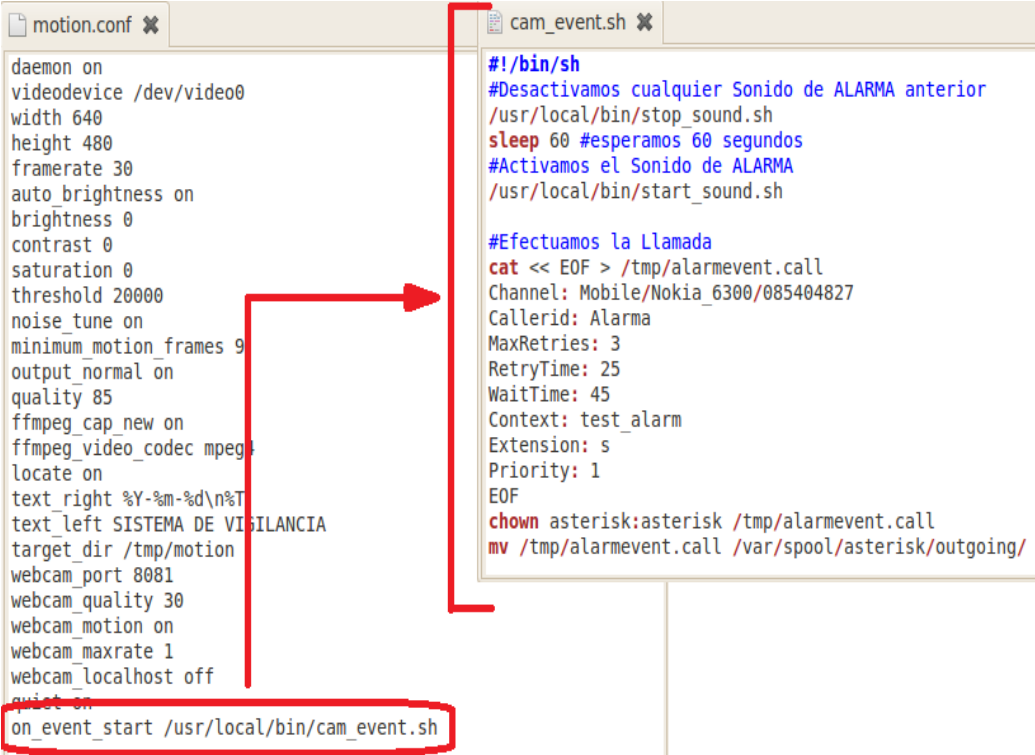
exec_motion.sh ✕
#!/bin/sh
sudo /usr/bin/motion

```

Figura 4.7: Activación de Motion

4.5 Funcionamiento del Sistema de Vigilancia

El sistema se encuentra encendido en espera de algún tipo de movimiento, cuando exista un movimiento extraño comenzará a tomar fotos, con los parámetros definidos en el archivo motion.conf, y al leer la línea de comando “on_event_start” se procede a ejecutar el script “cam_event.sh”, como se indica en la Figura 4.8.



```
motion.conf x cam_event.sh x
daemon on
videodevice /dev/video0
width 640
height 480
framerate 30
auto_brightness on
brightness 0
contrast 0
saturation 0
threshold 20000
noise_tune on
minimum_motion_frames 9
output_normal on
quality 85
ffmpeg_cap_new on
ffmpeg_video_codec mpeg4
locate on
text_right %Y-%m-%d\n%T
text_left SISTEMA DE VIGILANCIA
target_dir /tmp/motion
webcam_port 8081
webcam_quality 30
webcam_motion on
webcam_maxrate 1
webcam_localhost off
quiet on
on_event_start /usr/local/bin/cam_event.sh

#!/bin/sh
#Desactivamos cualquier Sonido de ALARMA anterior
/usr/local/bin/stop_sound.sh
sleep 60 #esperamos 60 segundos
#Activamos el Sonido de ALARMA
/usr/local/bin/start_sound.sh

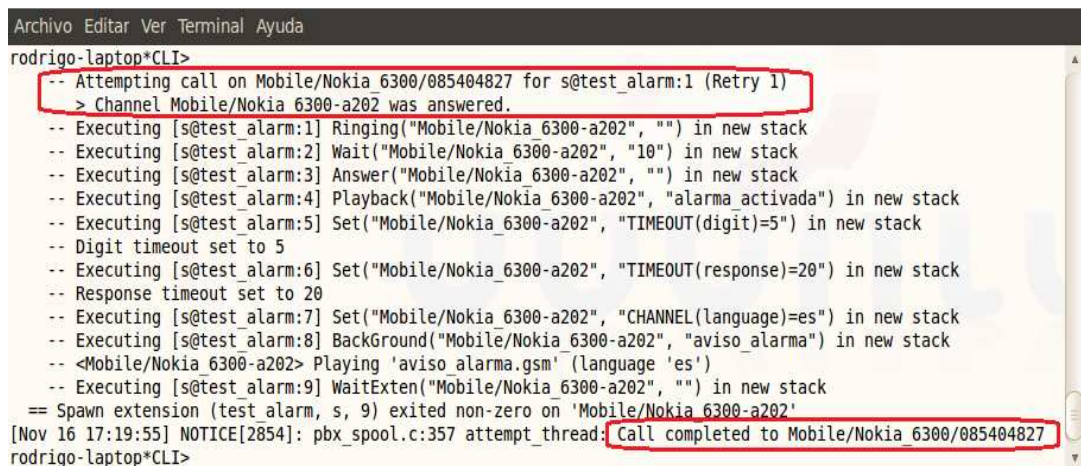
#Efectuamos la Llamada
cat << EOF > /tmp/alarmevent.call
Channel: Mobile/Nokia_6300/085404827
Callerid: Alarma
MaxRetries: 3
RetryTime: 25
WaitTime: 45
Context: test_alarm
Extension: s
Priority: 1
EOF
chown asterisk:asterisk /tmp/alarmevent.call
mv /tmp/alarmevent.call /var/spool/asterisk/outgoing/
```

Figura 4.8: Ejecución del Script cam_event.sh

4.6 Ejecución de Llamada

Se procede a ejecutar el script “*cam_event.sh*”, mismo que espera 60 segundos para que se pueda desactivar la alarma sin que se realice la llamada e inicie el sonido de alarma por la detección de movimiento.

Si No se desactivó la alarma, se reproduce un sonido para tratar de ahuyentar al intruso, posteriormente se crea y envía hacia Asterisk un archivo de llamada (.call) para que se realice la llamada de aviso, esto se puede verificar en la consola de Asterisk, tal y como se muestra en la Figura 4.9.



```
Archivo Editar Ver Terminal Ayuda
rodrigo-laptop*CLI>
-- Attempting call on Mobile/Nokia 6300/085404827 for s@test_alarm:1 (Retry 1)
> Channel Mobile/Nokia 6300-a202 was answered.
-- Executing [s@test_alarm:1] Ringing("Mobile/Nokia 6300-a202", "") in new stack
-- Executing [s@test_alarm:2] Wait("Mobile/Nokia 6300-a202", "10") in new stack
-- Executing [s@test_alarm:3] Answer("Mobile/Nokia 6300-a202", "") in new stack
-- Executing [s@test_alarm:4] Playback("Mobile/Nokia 6300-a202", "alarma activada") in new stack
-- Executing [s@test_alarm:5] Set("Mobile/Nokia_6300-a202", "TIMEOUT(digit)=5") in new stack
-- Digit timeout set to 5
-- Executing [s@test_alarm:6] Set("Mobile/Nokia_6300-a202", "TIMEOUT(response)=20") in new stack
-- Response timeout set to 20
-- Executing [s@test_alarm:7] Set("Mobile/Nokia 6300-a202", "CHANNEL(language)=es") in new stack
-- Executing [s@test_alarm:8] Background("Mobile/Nokia 6300-a202", "aviso_alarma") in new stack
-- <Mobile/Nokia 6300-a202> Playing 'aviso_alarma.gsm' (language 'es')
-- Executing [s@test_alarm:9] WaitExten("Mobile/Nokia 6300-a202", "") in new stack
== Spawn extension (test_alarm, s, 9) exited non-zero on 'Mobile/Nokia 6300-a202'
[Nov 16 17:19:55] NOTICE[2854]: pbx_spool.c:357 attempt_thread: Call completed to Mobile/Nokia 6300/085404827
rodrigo-laptop*CLI>
```

Figura 4.9: Ejecución de la Llamada en Consola

A continuación en la Figura 4.10 se muestra la realización de la llamada desde el celular gateway hacia el usuario propietario del Sistema de Vigilancia.



Figura 4.9: Realización de la Llamada

Mientras se realiza esto, el programa Motion hace capturas (fotografías) del intruso y las almacena en el directorio definido en la configuración, para que posteriormente se pueda verificar quién ha sido el intruso.

A continuación en la Figura 4.10, se muestran algunas de las fotografías que el sistema de vigilancia logró capturar.



Figura 4.10: Fotografías Capturadas

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

1. La utilización de este tipo de aplicaciones en el Ecuador ayuda en gran medida a las empresas, debido a su bajo costo, sin contar con lo sencillo que resulta la implementación del mismo, ya que se puede lograr una seguridad más sofisticada con solo agregar más cámaras al sistema de video vigilancia.
2. La principal ventaja que se obtiene con esta aplicación es de que no existe límites de información y cuenta con una gran variedad de alternativas, con la ayuda de Asterisk como PBX, se permiten futuras configuraciones para lograr una mejor efectividad y alcance, todo esto acorde a las necesidades del entorno donde se lo aplique y los requerimientos del cliente.
3. La instalación de un sistema de vigilancia en una oficina o casa, basado en este proyecto, se presenta como accesible para cualquier persona que desee implementarlo gracias a la ventaja de utilizar software de libre distribución y de código abierto, y por los bajos costos incurridos en la implementación.
4. La facilidad de implementación y configuración resulta una gran ventaja a considerar, para tomar este sistema como base para el control y monitoreo de zonas, permitiendo alertar en tiempo real y evitar situaciones de riesgo.
5. El uso del módulo bluetooth para la conexión con la red GSM es más económico que adquirir una tarjeta FXO/FXS para conectar una base celular.

RECOMENDACIONES

1. Tener conocimiento intermedio en LINUX, útil para la instalación de librerías y dependencias necesarias para el funcionamiento de la cámara web y el módulo bluetooth que se utilicen.
2. Configurar la sensibilidad de la cámara y el contraste a un nivel estable para que cuando haya cambios bruscos de luz no se active la alarma de manera innecesaria.
3. Configurar al sistema para que tome solo unas cuantas fotos al detectar movimiento para que no se llene el disco duro del servidor.
4. Tener precaución de que los dispositivos móviles que se comunican al servidor y los dispositivos dongle bluetooth, se encuentren dentro de la lista de compatibilidad de equipos que funcionan correctamente, para evitar malos resultados en la calidad del servicio.
5. Se podría hacer que el Sistema envíe un correo electrónico o un mensaje de texto (si el proveedor lo permite) en vez de una llamada.
6. Otra opción a tomar en cuenta es la de enviar el video o las fotografías grabadas a un servidor FTP o a un servidor de correo por internet.
7. También se podría configurar al Sistema para que realice una video llamada a un teléfono IP con soporte para tal efecto, o realizarla hacia un teléfono móvil, para así visualizar en tiempo real lo que ha causado la alarma.

BIBLIOGRAFÍA

- [1] **DABAX.NET**, Sistema de Vigilancia mediante una WebCam y Motion, <<http://www.dabax.net/sistemavigilancia>>, Septiembre 2009
- [2] **MOTION – WEB HOME**, Motion página oficial, <<http://www.lavrsen.dk/twiki/bin/view/Motion/WebHome>>, Septiembre 2009
- [3] **SERVITUX.ORG**, Sistema de Alarma con Asterisk y WebCam, <<http://www.servitux.org/view.php/page/alarma>>, Septiembre 2009
- [4] **TELEFONÍA VOIP GRANDSTREAM**, Telefonía IP, <<http://miqueridopinwino.blogspot.com/2007/07/motion-protgete-con-una-cmara-web-y.html>>, Septiembre 2009
- [5] **BLOGPOST.COM**, Tecnología para un proceso Sostenible, http://en.wikipedia.org/wiki/Inverted_index, Septiembre 2009
- [6] **Umanzor P**, Conectar Asterisk a la red GSM - chan mobile , http://info.linuxmall.cl/wiki/index.php/Conectar_Asterisk_a_la_red_GSM_-_chan_mobile, 2009
- [7] **Voip IP Telephony**, Configuring, using and debugging chan_mobile , <http://snapvoip.blogspot.com/2007/10/configuring-using-and-debugging.html>, 2007
- [8] **Voip-Info.org**, Chan mobile, http://www.voip-info.org/wiki/view/chan_mobile, 2008