

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DEL PERÚ

DISEÑO DE UN SISTEMA DE CONTROL DOMÓTICO Y VIDEO VIGILANCIA SUPERVISADO POR UN TELÉFONO MÓVIL

Tesis para optar el Título de **Ingeniero de las Telecomunicaciones**, que presenta el
bachiller:

Felipe Guerra Ruiz

ASESOR: M.Sc.Ing. Gumercindo Bartra Gardini

Lima, noviembre de 2013

RESUMEN

En los últimos años, la percepción de inseguridad ha aumentado significativamente, gran parte de los peruanos perciben la falta de seguridad, y la delincuencia como el problema más importante que afronta nuestro país, y debido a los altos costos de los sistemas de vigilancia automatizados, éstos no son una herramienta muy utilizada en el segmento residencial, sin embargo son una solución que presenta muchas ventajas en éste ámbito.

Es por ello que en la presente tesis se realiza el diseño de un sistema de video vigilancia pensado para el hogar y que permita al usuario acceder a él de manera rápida y sencilla utilizando un dispositivo móvil, ya sea dando las órdenes mediante una llamada telefónica o visualizando las capturas de las cámaras mediante una interfaz web, que además incluya un segmento orientado al control domótico de manera que se pueda centralizar el control de luces o aparatos eléctricos mediante llamadas.

Se quiere que en la tesis se diseñe un sistema orientado a reducir los costos en comparación con otras soluciones disponibles en el mercado actual, y para ello este diseño se centra en el micro controlador y micro computador Raspberry Pi, que es un computador del tamaño de una tarjeta de crédito y costo mínimo, pero con gran funcionalidad.

En el capítulo 1 se realiza un análisis de la problemática a la que se busca tentar una solución con este diseño, la inseguridad ciudadana.

En el capítulo 2 da una breve reseña sobre las tecnologías utilizadas en el diseño de este sistema.

En el capítulo 3 se formulan los objetivos del diseño y dan los planteamientos iniciales del sistema que se va a diseñar.

El capítulo 4 describe cada componente y cada paso del diseño del sistema de video vigilancia y control domótico, dividido en segmentos.

Finalmente en el capítulo 5 se evalúan los costos del sistema para justificar si se trata de un sistema que tiene un lugar como solución al problema de la delincuencia en nuestro país.

ÍNDICE

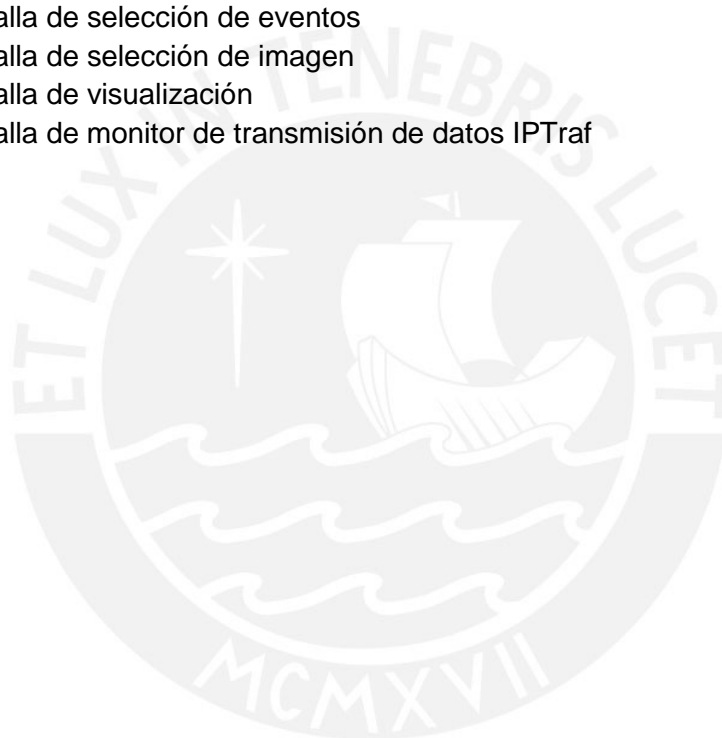
Introducción	8
Capítulo 1: Marco Problemático	9
1.1 Problemática de la seguridad ciudadana	9
1.2 Problemática de la domótica	11
1.3 Problemática de acceso	12
Capítulo 2: Marco Teórico	13
2.1 Domótica.	13
2.2 Video Vigilancia	14
2.2.1 Sistemas Analógicos	14
2.2.2 Sistemas de vigilancia IP	15
2.3 Telefonía IP	16
2.3.1 Central IP PBX	16
2.3.2 Características de un sistema PBX	16
2.3.3 Servidor Asterisk	17
2.3.4 Sistema interactivo de respuesta por voz	18
2.4 Micro controlador	19
2.4.1 Micro controlador ARM	19
2.4.2 Pines GPIO	20
2.4.3 Raspberry Pi	20
2.5 Relés	21
2.6 Sensores	22
2.6.1 Sensores de movimiento	22
2.6.2 Sensor de movimiento PIR	22
2.7 Lenguaje de programación	23
2.7.1 Java	23
2.8 Aplicación Web	24
2.8.1 HTML	24
2.8.2 Java Server Page	25
2.8.3 Servidor Web	25
2.8.4 Apache Tomcat	25
2.8.5 JQuery Mobile	26
Capítulo 3: Planteamientos iniciales del sistema	27
3.1 Objetivos	28
3.1.1 Objetivo general	28
3.1.2 Objetivos específicos	28
3.2 Planteamientos iniciales	29
Capítulo 4: Diseño del sistema de control domótico y video vigilancia supervisado desde un teléfono móvil	30
4.1 Arquitectura del sistema	30
4.2 Componentes del sistema	31
4.2.1 Selección del micro controlador	31
4.2.2 Selección de las cámaras de vigilancia	34

4.2.3 Dispositivos móviles	34
4.2.4 Softphone Zoiper	35
4.2.5 Selección de los relés	37
4.2.7 Sistema operativo Raspbian	39
4.2.8 Lenguaje Java	39
4.2.9 Librería Asterisk-Java	39
4.2.10 Librería PI4J	39
4.2.11 Servidor web Apache Tomcat	40
4.3 Diseño de la red	40
4.4 Segmento de IVR	41
4.4.1 Archivo de configuración Sip	41
4.4.2 Archivo de configuración Extensions	41
4.4.3 Archivo de configuración Manager	43
4.4.4 Comunicación con el programa de Java	44
4.4.5 Salida de la red local	45
4.5 Segmento de control domótico	46
4.5.1 Diagrama del segmento de control domótico	46
4.5.2 Conexiones del segmento de control domótico	46
4.5.3 Flujo del segmento de control domótico	48
4.5.4 Comunicación con el programa de Java	50
4.6 Segmento de vigilancia	50
4.6.1 Ubicación de los componentes	50
4.6.2 Conexiones del segmento de vigilancia	51
4.6.3 Funcionamiento de las cámaras	53
4.6.4 Flujo del segmento de vigilancia	55
4.7 Segmento de visualización	57
4.7.1 Diagrama de casos de uso de la aplicación Web	58
4.7.2 Autenticación de usuario	58
4.7.3 Selección de evento	59
4.7.4 Selección de imagen para visualizar	60
4.8 Pruebas de funcionamiento	61
Capítulo 5: Evaluación de costos del sistema	64
Conclusiones	66
Recomendaciones	67
Fuentes	68
Anexos	71
Código de programa Controlador (Java)	
Archivo de servidor Asterisk: extensions.conf (dial plan)	
Quick Start Guide: The Raspberry Pi – Single Board Computer	
Datasheet: Delcomp – Relay X4	

ÍNDICE DE FIGURAS

Capítulo 1	
Figura 1.1 Cantidad de faltas registradas en Lima 2005-2012	10
Figura 1.2 Cantidad de faltas registradas 2012 por tipo de falta	11
Capítulo 2	
Figura 2.1 Diseño de un sistema de video vigilancia analógico	15
Figura 2.2 Diseño de un sistema de video vigilancia analógico	15
Figura 2.3 Diseño de un sistema IP PBX basado en Asterisk	17
Figura 2.4 Ejemplo de un menú con mensajes pregrabados	18
Figura 2.5 Componentes esenciales de un micro controlador	19
Figura 2.6 Modelo del Raspberry Pi modelo B	21
Figura 2.7 Ejemplo de un relé para controlar un circuito eléctrico	21
Figura 2.8 Funcionamiento de un sensor PIR	23
Figura 2.9 Ejemplo de aplicación Java	24
Figura 2.10 Ejemplo de código HTML	25
Capítulo 4	
Figura 4.1 Arquitectura del sistema	31
Figura 4.2 Pines GPIO y salida Ethernet del Raspberry Pi	33
Figura 4.3 Softphone Zoiper multi-plataforma	35
Figura 4.4 Tarjeta de tarjeta Delcomp RelayX4	36
Figura 4.5 terminales de los relés tarjeta relayx4	36
Figura 4.6 Pines de control de la tarjeta relayx4	37
Figura 4.7 Sensor de movimiento PIR	38
Figura 4.8 Sensor de movimiento PIR HC-SR501	38
Figura 4.9 Agregar una cuenta de usuario en el archivo sip.conf	41
Figura 4.10 Menú de control del sistema	42
Figura 4.11 Ejemplo de configuración del archivo extensions.conf para un IVR	43
Figura 4.12 Archivo extensions.conf – Envío de una alerta con un mensaje pregrabado	43
Figura 4.13 Archivo manager.conf – Cuenta de administrador del servidor para acceso remoto	44
Figura 4.14 Declaración y construcción de la conexión con la central PBX	44
Figura 4.15 Listener para eventos del servidor Asterisk	45
Figura 4.16 Originar acción – Envío de alerta	45
Figura 4.17 Diagrama de segmento de control domótico	46
Figura 4.18 Conexiones necesarias para controlar un circuito con un relé y el micro controlador	47
Figura 4.19 Conexiones en el raspberry pi para la comunicación con la tarjeta de relés	48
Figura 4.20 Diagrama de flujo del segmento de control domótico	49
Figura 4.21 Declaración de los objetos de los pines GPIO_07 y GPIO_01	50
Figura 4.22 Declaración de los objetos de los pines GPIO_07 y GPIO_01	50
Figura 4.23 Ubicación de componentes de vigilancia	51

Figura 4.24 Diagrama para la conexión de los sensores y el micro controlador	52
Figura 4.25 Raspberry Pi con las conexiones correspondientes a los dos sensores de movimiento	52
Figura 4.26 Conexión del sensor PIR	53
Figura 4.27 Captura de una cámara D-Link DCS-2130 a 640x480	54
Figura 4.28 Cámara Dcs-2130 conectada mediante un cable Ethernet	54
Figura 4.29 Salida del terminal ante una descarga de imagen de la cámara IP	55
Figura 4.30 Diagrama de flujo del segmento de capturas con las cámaras de vigilancia	56
Figura 4.31 Captura de imágenes desde la cámara de vigilancia	57
Figura 4.32 Diagrama de caso de uso – Aplicación web	58
Figura 4.33 Pantalla de ingreso del usuario y la contraseña	59
Figura 4.34 Pantalla de selección de eventos	59
Figura 4.35 Pantalla de selección de imagen	60
Figura 4.36 Pantalla de visualización	61
Figura 4.37 Pantalla de monitor de transmisión de datos IPTraf	62



Capítulo 4

Tabla 4.1 Tabla comparativa de micro controladores	32
Tabla 4.2 Tabla comparativa de cámaras de vigilancia	34
Tabla 4.3 Direccionamiento IP para la red local	40
Tabla 4.4 Conexiones entre la tarjeta de relés y el micro controlador	47
Tabla 4.5 Conexiones de los sensores de movimiento y el micro controlador	51
Tabla 4.6 Cálculo de la capacidad de almacenamiento	54
Tabla 4.7 Tabla de tráfico generado por acción realizada	62
Tabla 4.8 Tabla de ancho de banda por acción realizada	63

Capítulo 5

Tabla 5.1 Tabla de costos del sistema diseñado	64
Tabla 5.2 Costos del sistema de seguridad ofrecido por Prosegur	65



Uno de los principales problemas que azotan a nuestra sociedad, así como a muchas otras en todo el mundo, es la delincuencia, la cual en nuestra ciudad ha alcanzado cifras alarmantes en los últimos años, las cuales nos indican que vivimos en una de las ciudades con mayor cantidad de delitos cometidos cada año, a esto debemos agregarle el hecho de que la gran mayoría de personas no confían en que la policía estará en el momento en que se le necesite.

Las cámaras de video vigilancia han tenido una gran aceptación como elemento que ofrece seguridad a entidades tales como bancos u oficinas del gobierno, así como en hogares, a pesar de que en ellos esta técnica de seguridad no haya logrado penetrar en el mercado con la fuerza con que la tomaron las entidades grandes anteriormente mencionadas.

Además de ello, es importante mencionar que a un sistema de seguridad basado en una red con cámaras de vigilancia y sensores se le puede agregar un elemento importante como lo es la domótica, incluyendo en este sistema la posibilidad de controlar de manera remota los aparatos eléctricos y las luces en el hogar para brindar seguridad, así como comodidad al residente.

Es por ello, que esta tesis se centra en el diseño de un sistema de seguridad basado en la vigilancia por medio de cámaras de video y el uso de sensores de movimiento para que se pueda detectar cualquier evento en el hogar cuando el usuario se encuentre lejos, y se le pueda alertar inmediatamente de este evento por medio de una central telefónica que se encargue de enviar estas alertas, así como de recibir y ejecutar las acciones solicitadas por el usuario.

Por último, el usuario podrá acceder a las imágenes capturadas por las cámaras de vigilancia que se encontrarán en un servidor de archivos remoto, por medio de una interfaz web de fácil acceso desde cualquier dispositivo que cuente con internet.

CAPÍTULO 1

MARCO PROBLEMÁTICO

1.1. Problemática de la seguridad ciudadana

Actualmente en Lima, uno de los problemas que más preocupa a la mayoría de ciudadanos es la falta de seguridad, este problema es atribuido en muchas ocasiones al crecimiento de manera muy rápida de la ciudad, el incremento muy acelerado de la densidad poblacional en nuestra ciudad en los últimos años.

Lima es uno de los departamentos con mayores indicadores de criminalidad en el país, según las cifras del ministerio del interior, el 40.7% de faltas cometidas en el país en el año 2012 corresponden al departamento de Lima [2], debido a que en el año 2012 en Lima se registraron 96,832 faltas en este departamento, es por ello que los mecanismos de seguridad usados para contrarrestar la delincuencia en Lima deben ser un punto de énfasis en cualquier política de gobierno, así como también debe ser una responsabilidad de los ciudadanos el velar por su propia seguridad cumpliendo con ciertas consideraciones esenciales para mantener su seguridad.

En la figura 1.1 se puede apreciar como en los últimos años se ha dado un crecimiento en la cantidad de faltas cometidas en nuestra ciudad, lo cual nos hace pensar que en un futuro la cifra será mucho mayor que ahora, y el problema también será mucho más complicado de resolver de lo que es actualmente.

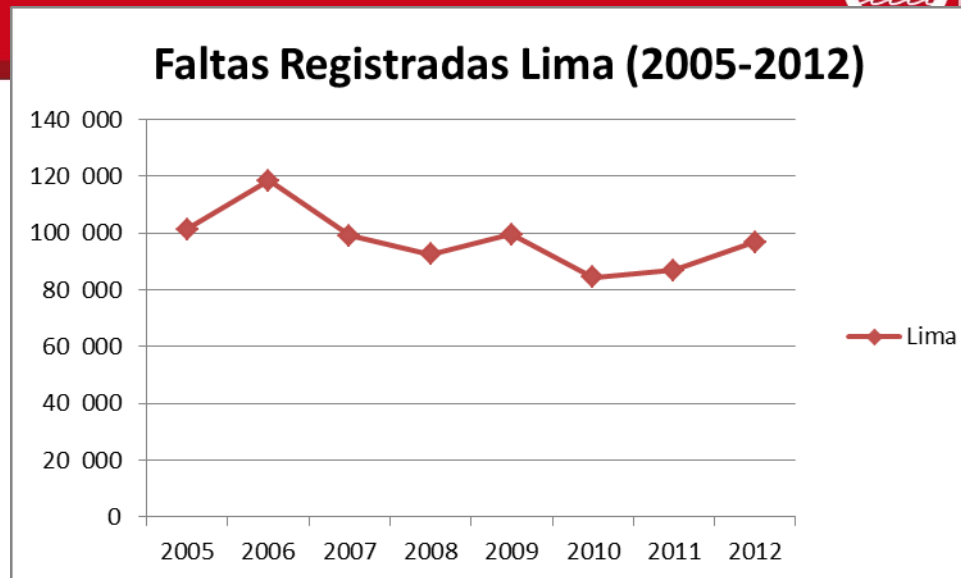


Figura 1.1 Cantidad de faltas registradas en Lima 2005-2012
 Fuente: Ministerio del Interior (MININTER)

Dentro de todas las faltas cometidas en nuestro país, el ministerio del interior las clasifica de acuerdo al tipo de falta, existiendo: Contra la familia y la persona; Contra el patrimonio; Contra las buenas costumbres; Contra la seguridad pública; Contra la tranquilidad pública; Otras faltas.

Siendo las faltas contra el patrimonio las más representativas dentro del total, pues representan el 28.9% del total de faltas cometidas durante el año 2012, a esto es importante agregar el hecho de que el aumento de estas ha sido del 7.8% del año 2011 al año 2012 [2], lo cual es una cifra alarmante considerando que este indicador nos dice que hay un aumento considerable en la cantidad de delitos cometidos cada año, en especial los delitos contra el patrimonio, que son sobre los cuales se hará énfasis en ésta tesis. En la figura 1.2 se puede apreciar el alto porcentaje de faltas registradas en el año de tipo “Contra el patrimonio”, como parte del total de faltas registradas en el año.

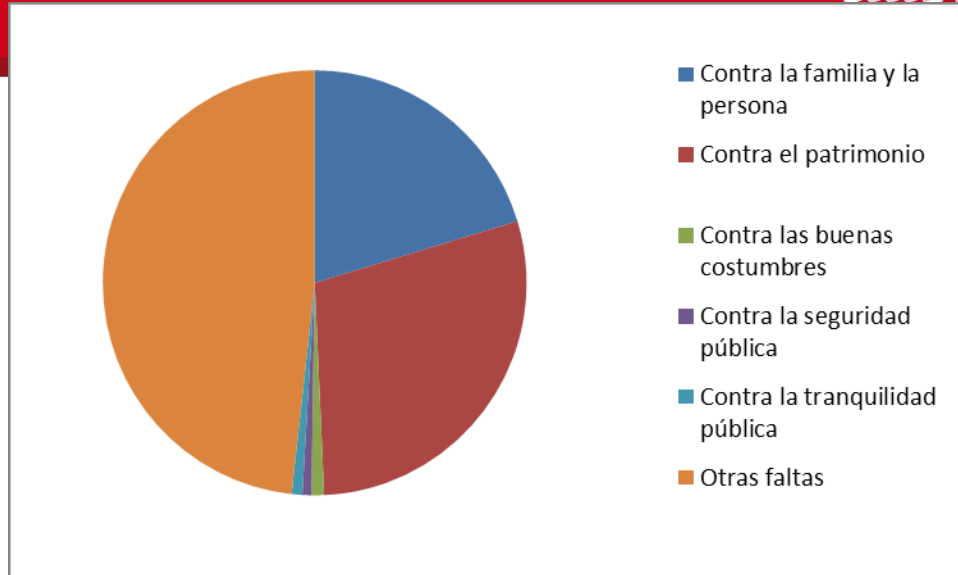


Figura 1.2 Cantidad de faltas registradas 2012 por tipo de falta
Fuente: Ministerio del Interior (MININTER)

La inseguridad ciudadana se define como una sensación psicológica que se encuentra vinculada a una posible percepción de vulnerabilidad a ser víctima de un delito. Es decir, que a pesar de no haber sido víctima de un delito necesariamente, por influencia colectiva, tiene esta sensación. Esta inseguridad se ve incrementada debido al hecho de que la policía nacional, no se abastece para la demanda de seguridad que existe en nuestra ciudad, si a esto se le agrega el hecho de que el despliegue policial va decreciendo cada año, esto hace que los indicadores de sensación de inseguridad se agraven.

1.2 Problemática de la domótica

La domótica se define como un conjunto de sistemas que automatizan las diferentes instalaciones de una vivienda [3], y es una tecnología que tiene por finalidad brindar una mayor comodidad y seguridad al usuario en el manejo de las instalaciones eléctricas en su hogar, es una tecnología que, en países del primer mundo, está aumentando su uso en el contexto residencial muy rápidamente debido, entre otras cosas, a sus diversas posibles aplicaciones y su gran utilidad como herramienta para brindar a los usuarios esta comodidad antes mencionada, así como la posibilidad de supervisar y controlar los elementos eléctricos en su casa de manera remota.

Sin embargo, es una tecnología que debido , en parte, a sus altos costos, tanto de adquisición como de operación no se encuentra todavía difundida en nuestro país con la misma fuerza que en América del norte, Europa o Asia; otra de las causas que

limitan la existencia de sistemas de domótica en nuestro país es la falta de iniciativas de desarrollo de tecnología, además de ello no existe una cultura tecnológica en los peruanos, lo cual significa que la población no cuenta con la información necesaria con respecto a la domótica, hecho que lo lleva a pensar en esta tecnología como un lujo que se encuentra fuera de su alcance económico.

1.3 Problemática de acceso

Un problema muy importante en los sistemas de vigilancia basados en teléfonos móviles presentes en la actualidad es la dificultad que representa el acceder a él por parte de ciertas personas, ya sea por problemas en la visión de parte de la persona, así como también puede deberse al hecho de que algunos posibles usuarios prefieren el uso de los sistemas low-tech para sus actividades cotidianas. Siendo los sistemas low-tech definidos como sistemas tecnológicos que tienen la característica de no ser muy sofisticados o complejos [4], y son más amigables a los usuarios de mayor edad, así como a los que sin importar su edad prefieren un sistema rápido y simple que cumpla con la función deseada.



CAPÍTULO 2

MARCO TEÓRICO

2.1 Domótica

La domótica se define como el conjunto de sistemas que automatizan las instalaciones de una vivienda [4]. Estos sistemas pueden ser controlados o supervisados de forma presencial, así como también remotamente, y este control puede darse de diferentes maneras, ya sea mediante una red interna con controladores centralizados, así como también a través de la internet, o usando la telefonía para ello.

El principal objetivo de cualquier sistema domótico es el poder centralizar el control de los diferentes dispositivos eléctricos de una vivienda, así como también las luces y puertas siempre y cuando cuenten con alimentación eléctrica, de esta manera se puede brindar una gran facilidad al usuario para controlar todas estas funciones en tiempo real [5].

Otro de los fines con los cuales contribuye la domótica es en la seguridad de la vivienda, debido a que se pueden controlar y programar los cierres de puertas, así como el encendido de alarmas y luces como reacción ante posibles amenazas que se puedan detectar mediante los sensores que se tengan instalados, tales como la detección de movimiento cuando haya una intrusión, detección de fuga de gas, entre otras [5].

Además de ello, la domótica contribuye a la eficiencia del uso de recursos eléctricos en la vivienda, y de esta forma contribuye tanto a la economía del usuario como a la preservación de recursos del medio ambiente [5].

2.2 Video Vigilancia

Se define la video vigilancia como la supervisión presencial o remota de una determinada localidad utilizando una o más cámaras de video, puede ser para fines de seguridad, o de supervisión de personal, así como para muchas otras funciones.

Los sistemas de video vigilancia actuales se clasifican en de acuerdo a la tecnología que utilizan en:

2.2.1 Sistemas Analógicos

Los sistemas de vigilancia analógica o, también conocidos como circuito cerrado de televisión tienen como característica principal el uso de cámaras analógicas para la captura de señal analógica, la cual es transmitido por medio de cables coaxiales a un Grabador de Video Digital o DVR por sus siglas en inglés. En el DVR se realiza la conversión de la señal analógica en digital, y se realiza la compresión del video, para posteriormente poder transmitirla al dispositivo en el cual se dará el almacenamiento. También permite transmitir en broadcast a todos los dispositivos de la red.

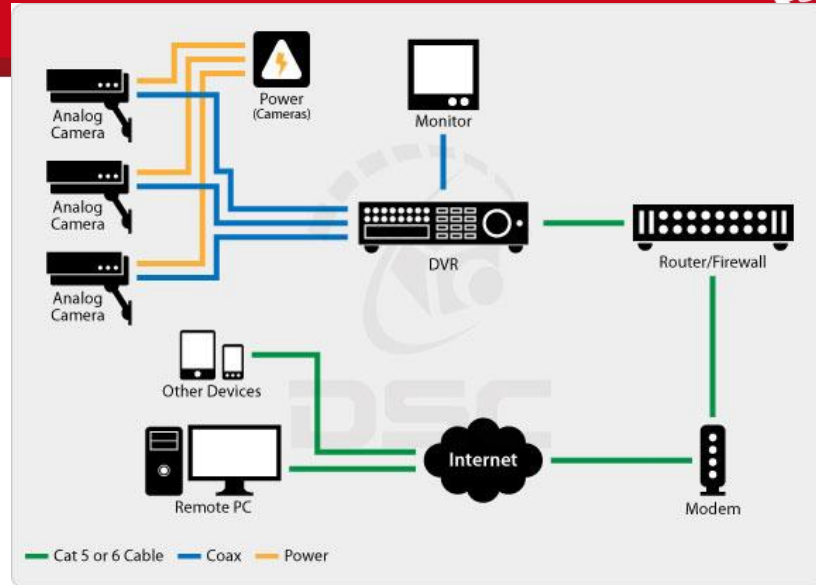
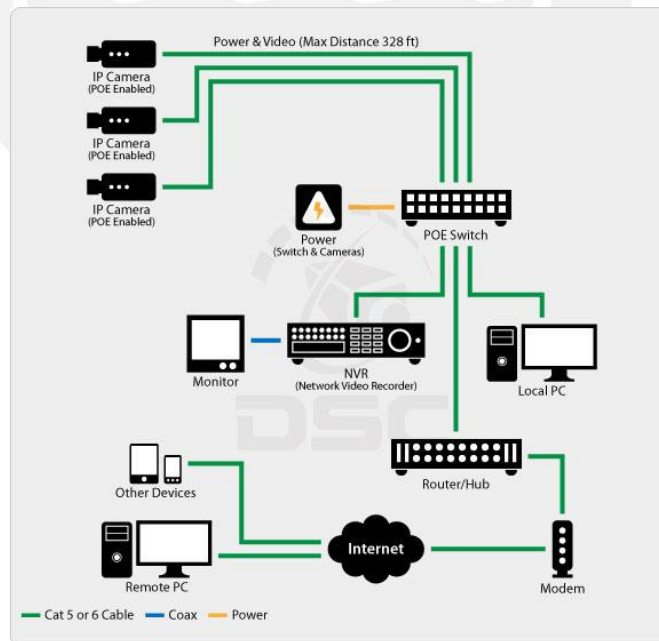


Figura 2.1 Diseño de un sistema de video vigilancia analógico
Fuente: [7]

2.2.2 Sistemas de vigilancia IP

En los sistemas de vigilancia basados en IP, las cámaras capturan el video analógico e inmediatamente la convierte en señal digital y la comprimen para su transmisión. Luego de ello se transmite la señal de video en broadcast a través de la red de área local.



2.2 Diseño de un sistema de video vigilancia analógico
Fuente: [7]

En la actualidad, los sistemas analógicos siguen representando una opción más económica que los sistemas basados en tecnología IP, sin embargo el costo de las cámaras IP va decreciendo rápidamente, lo cual hace esta tecnología como la mejor opción para la vigilancia basándose en una proyección a futuro, debido a que cuentan con importantes ventajas con respecto a los sistemas analógicos, tales como un ahorro de recursos de capacidad de transmisión al transmitir el video ya comprimido, en comparación con la transmisión de la tecnología analógica de video vigilancia; así como también la mayor resolución en la imagen que permite la transmisión.

2.3 Telefonía IP

La telefonía IP, también conocida como VoIP (Voice Over Internet Protocol) es la tecnología que permite la transmisión de voz en una red utilizando el protocolo de red IP.

2.3.1 Central IP PBX

También conocida como PBX, o Private Branch Exchange por sus siglas en inglés es una tecnología que debe su nombre a que su principal ámbito de uso son las comunicaciones dentro de una empresa, un PBX cumple las funciones de conmutación dentro de una red de telefonía en ella.

Actualmente, una red privada de telefonía VoIP cuenta con muchas más funcionalidades que simplemente llamadas, funcionalidades que incluyen atención automatizada mediante mensajes pregrabados, mensajes de voz, llamadas en espera y conferencias telefónicas que agregan valor a las redes de telefonía IP [8].

2.3.2 Características de un sistema PBX

Listo para la transmisión de voz sobre IP, Dado que la telefonía está apuntando hacia esta tecnología, permite la transmisión mediante el estándar SIP.

Mensajería de voz, tecnología actualmente disponible en cualquier sistema de telefonía basado en IP, que permite funcionalidades tales como el envío de estos mensajes a un correo electrónico.

Movilidad, debido a que permite la redirección de llamadas de una extensión a otra en caso no recibir respuesta.

Conferencias, que permiten a los usuarios comunicarse entre más de 2 extensiones.

Reportes de llamadas, una central IP PBX permite un fácil acceso al historial de llamadas para una mejor gestión de las comunicaciones.

2.3.3 Servidor Asterisk

Asterisk es un software de libre distribución creado para brindar una solución de IP PBX e incluye todos los componentes necesarios para la creación de un sistema de telefonía IP completo [8].

Asterisk puede ser conectado a la PSTN utilizando VoIP, o tecnologías de telefonía tradicionales, soporta el estándar de telefonía SIP y presenta una constante evolución para mantenerse a la par con la tecnología actual, cada año se realizan mejoras al software con visión a futuro.

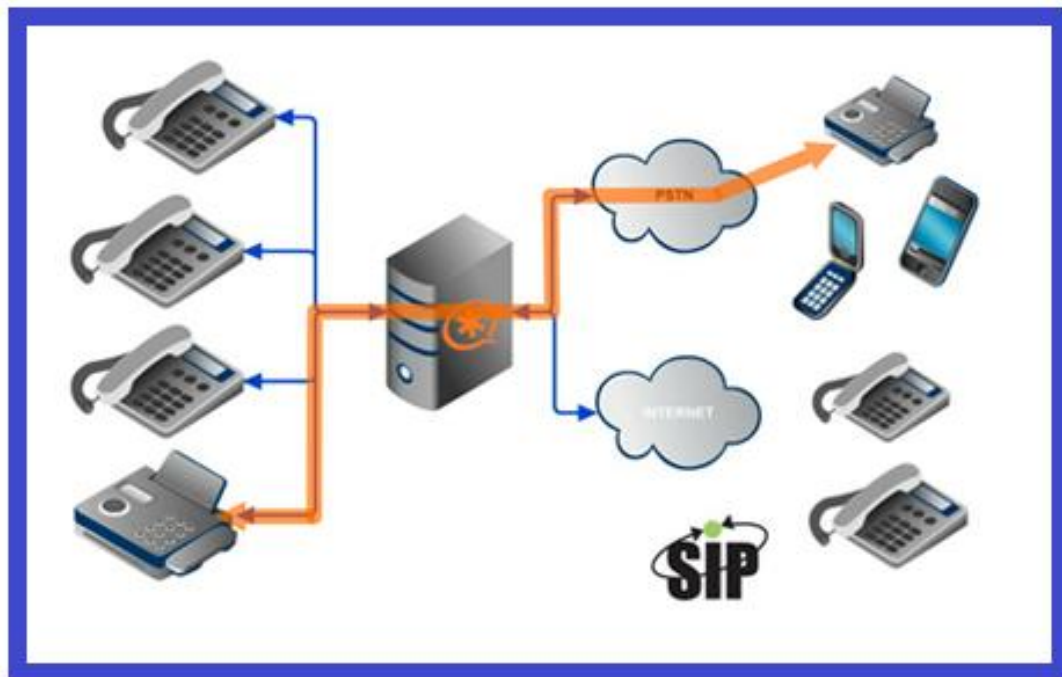


Figura 2.3 Diseño de un sistema IP PBX basado en Asterisk
Fuente: [8]

En la figura 2.3 se puede apreciar un sistema IP PBX que tiene conectadas tres extensiones telefónicas simples y un fax orientadas al envío y recepción de documentos, esta central de IP PBX utiliza software de libre distribución Asterisk y se comunica con la red PSTN (red de conmutación telefónica pública), así como también a otras extensiones utilizando la internet.

2.3.4 Sistema interactivo de respuesta por voz

Los sistemas interactivos de respuesta por voz, IVR por sus siglas en inglés, son sistemas que permiten que una determinada extensión en una red de telefonía sea atendida por un servidor con un menú automatizado con mensajes de voz previamente pregrabados, de esta manera se puede redireccionar la llamada, así como escuchar mensajes informativos, e incluso controlar componentes externos a la red de telefonía [8].

Por ejemplo, se puede configurar el servidor de voz sobre IP con un sistema IVR para que indique al usuario que efectúa las llamadas los posibles anexos a los cuales se puede redireccionar esta llamada, el usuario entonces marca la opción deseada y el sistema se encarga de realizar la tarea previamente programada para esa opción.

En esta tesis se usará un servidor de voz sobre IP con un sistema IVR basado en el Asterisk para poder realizar el manejo de los elementos del sistema que se quiere diseñar.

```
[menu]
exten => s,1,Answer()
exten => s,n,wait(0.5)
exten => s,n,Background(menu0)
exten => s,n,waitExten(5)
exten => 1,1,Goto(menu-a,s,1)
exten => 2,1,Goto(menu-b,s,1)
exten => 3,1,Goto(menu-c,s,1)
exten => 0,1,Dial(SIP/123)
exten => *,1,Goto(s,1)
exten => i,1,Playback(invalida)
exten => i,n,Goto(s,1)
include => menu-a
include => menu-b
```

Figura 2.4 Ejemplo de un menú con mensajes pregrabados.
Fuente: Elaboración propia

En la figura 2.4 podemos apreciar un ejemplo claro del uso de un menú simple en un servidor IVR, en el cual, al recibir la llamada, la extensión previamente configurada como el servidor con mensajes pregrabados se encarga de contestar las llamadas entrantes, primero se reproduce el menú inicial y saludo, mientras se espera por la selección del usuario de la actividad para ejecutar la acción escogida, en el caso del

ejemplo se tienen más menús anidados dentro de este menú principal, es decir cada opción llevar al usuario a una nueva extensión con un nuevo menú.

2.4 Micro controlador

Un microcontrolador es un circuito intergrado diseñado para cumplir un número de funciones previamente programadas en su memoria, se compone de al menos tres elementos principales, aunque podría contar con otros:

- Un procesador, que se encarga de realizar las operaciones lógicas e interpretar las instrucciones recibidas por el microcontrolador.
- Una memoria, que se encarga de almacenar de manera temporal o permanente los datos para el procesamiento del microcontrolador.
- Dispositivos de entrada/salida para recibir y enviar la información que procesa el microcontrolador.



Figura 2.5 Componentes esenciales de un microcontrolador
Fuente: Elaboración propia

2.4.1 Micro controlador ARM

Un micro controlador ARM o (Advanced Risc Machine) son también considerados como computadores con un número reducido de instrucciones como lo dice su nombre (RISC significa computador de instrucciones reducidas), y su tamaño reducido, bajo consumo y bajo consumo de potencia los convierte en una opción muy útil en soluciones de electrónica en hogares o en entornos móviles, y existen varios microcontroladores ARM que cuentan con sistemas operativos de libre distribución como lo es el Raspberry Pi, el cual cuenta con pines de entrada y salida para enviar y recibir señal de dispositivos externos [18].

2.4.2 Pines GPIO

Los pines de entrada y salida de propósito general, o GPIO por sus siglas en inglés, son pines presentes en varios micro controladores ARM y que permiten su conexión con dispositivos externos ya sean de entrada o salida, para que estos puedan ser controlados por el micro controlador, algunos dispositivos que se pueden conectar usando estos pines son sensores, servos, relés, interruptores, entre muchos otros, la señal recibida por estos pines o enviada por estos pines es procesada en el micro controlador e interpretada de acuerdo a la programación que se les haya dado previamente.

2.4.3 Raspberry pi

El raspberry pi es un microcontrolador de bajo costo y que cumple con muchas más funcionalidades que la mayoría de competidores, tiene la característica de contar con hardware y software de libre distribución, su sistema operativo es de la distribución Linux, y esta desarrollado bajo la licencia del mismo [9].

El raspberry pi cuenta con varias funcionalidades y prestaciones que lo hacen una opción muy atractiva como controlador de dispositivos:

- Cuenta con una unidad central de procesamiento que funciona a 700MHz, pero que puede ser sobrecorrida hasta 1GHz sin perder su garantía.
- Cuenta con una memoria RAM de 512 MB, que si tenemos en cuenta las aplicaciones en las cuales se utiliza el raspberry pi, es la apropiada para su correcto funcionamiento.
- Cuenta con dos puertos USB, es compatible con la mayoría de periféricos con los cuales es compatible una PC, pueden usarse para conectar todo tipo de periféricos de entrada y de salida.
- Tiene un puerto Ethernet para la conexión a internet, sin embargo también puede darse la conexión usando un adaptador USB para su uso en una red inalámbrica.
- Permite todas las ventajas y facilidades del almacenamiento externo pues cuenta con un puerto para tarjetas SD, SDHC y SDXC, es decir permite un almacenamiento de hasta 2TB.
- Permite una variedad de sistemas operativos de distribución Linux.
- Tiene un consumo energético bastante reducido en 3.5W.

Estos componentes se pueden observar en la figura 2.6 extraído de la página web del producto [9].

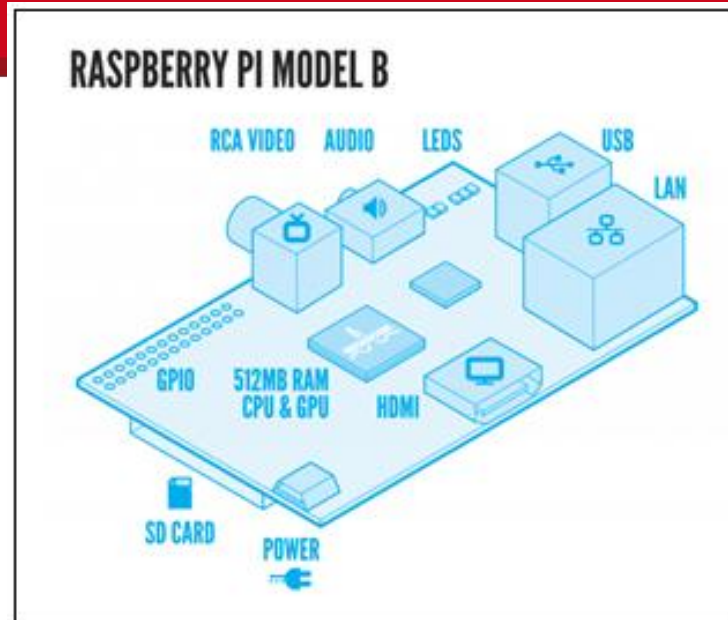


Figura 2.6 Modelo del Raspberry Pi modelo B
 Fuente: [9]

2.5 Relés

También conocidos como relays por su nombre en inglés, son dispositivos electromecánicos que cumplen la función de un interruptor controlado por un circuito eléctrico, su funcionamiento es magnético, y permite el accionar de contactos eléctricos con la finalidad de abrir ciertos circuitos eléctricos. Se pueden utilizar como interruptores para abrir y cerrar circuitos controlándolos mediante una señal eléctrica proveniente de un microcontrolador.



Figura 2.7 Ejemplo de un relé para controlar un circuito eléctrico.
 Fuente: [1]

En la figura 2.7 se puede observar un relé que será usado como interruptor para abrir y cerrar el circuito entre el cable rojo y el negro para controlar la alimentación de un dispositivo eléctrico.

2.6 Sensores

Los sensores son dispositivos electrónicos que se encargan de detectar variaciones en las condiciones del ambiente que se busca estudiar, y ante una de estas variaciones, son capaces de enviar una señal de alerta al micro controlador pues estas variaciones son interpretadas como eventos, de forma que se pueda realizar una acción previamente programada, por ejemplo se tienen los sensores de movimiento, de humo, así como de fugas de gas, entre muchos otros.

2.6.1 Sensores de movimiento

Los sensores de movimiento se clasifican según su principio de funcionamiento, los pueden haber infrarrojos, de ultra-sonido, entre otros; pero todos cumplen con detectar movimiento en un ambiente determinado, para traducirlos en una señal eléctrica a ser analizada y procesada posteriormente en otro dispositivo, su aplicación más importante es en la seguridad y en la domótica debido a que permite activar dispositivos externos al detectar una intrusión.

2.6.2 Sensor de movimiento PIR

Un sensor de movimiento pasivo infrarrojo o PIR basa su funcionamiento en la detección de la radiación infrarroja que emiten los cuerpos, al moverse los objetos, esta radiación infrarroja varía con ellos, un sensor PIR detecta la variación en los patrones de radiación en el ambiente, y envía un pulso a través de una salida que puede conectarse a un micro controlador.

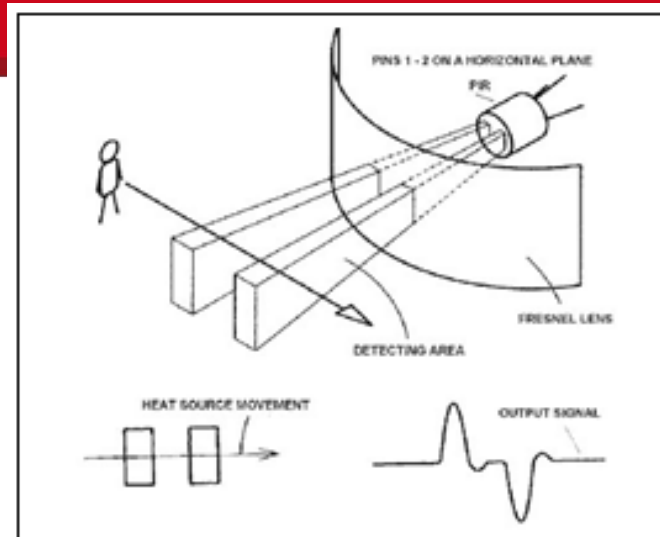


Figura 2.8 Funcionamiento de un sensor PIR
 Fuente: www.ladyada.net

2.7 Lenguaje de programación

Un lenguaje de programación es un lenguaje estandarizado que se utiliza para dar órdenes a las computadoras para la realización de tareas, consisten en ordenes organizadas y secuenciales que permiten que un computador procese la información recibida y brinde una respuesta, de acuerdo a lo que esté previamente programado en el sistema.

2.7.1 Java

Es un lenguaje de programación multiplataforma, orientado a objetos y la arquitectura de sus aplicaciones se basa en el uso de clases, y se encuentra bajo la licencia de GNU, es uno de los lenguajes de programación más utilizados para aplicaciones de cliente y servidor.

Cuenta con la posibilidad de agregar librerías para su integración con software externo, tal como Asterisk para llamadas por voip, o conexión con bases de datos relacionales, también para su comunicación con servidores FTP, entre muchas otras librerías de gran utilidad cuando se quiere utilizar este lenguaje de programación para integrar sistemas distintos para una misma finalidad.

```
2 package ejemplo;
3
4 import java.io.BufferedReader;
5 import java.io.IOException;
6 import java.io.InputStreamReader;
7
8
9
10 public class Ejemplo
11 {
12     public static void main (String[] args) throws IOException
13     {
14
15         InputStreamReader reader=new InputStreamReader(System.in);
16         BufferedReader Input=new BufferedReader (reader);
17
18         System.out.println ("Introduce tu nombre: ");
19
20         String name= Input.readLine();
21
22         System.out.println ("Hello " + name + "!");
23
24     }
25 }
26
27
```

Figura 2.9 Ejemplo de aplicación Java
Fuente:Elaboración propia

2.8 Aplicación web

Una aplicación web es un programa al cual pueden acceder los usuarios mediante la internet usando un navegador web, no es necesario instalar el aplicativo en todos los dispositivos que vayan a correr la aplicación, basta con que tengan un navegador compatible con el lenguaje que se utilizó para el desarrollo de esta aplicación [12].

Toda aplicación web centra su funcionamiento en un cliente y un servidor, y una de las más importantes ventajas que ofrece con respecto a aplicaciones tradicionales es la facilidad que implica el no tener que instalarlo en todos los clientes que vayan a utilizar la aplicación, la independencia del sistema operativo debido a que sin importar el sistema operativo del cual se acceda, si se tiene un navegador apropiado se tendrá un funcionamiento óptimo del aplicativo [12].

2.8.1 HTML

El lenguaje HTML o Hyper Text Markup Language es el lenguaje que se utiliza para el desarrollo de páginas web. Es un lenguaje basado en etiquetas que se abren y cierran mediante palabras reservadas envueltas en corchetes que dan jerarquía a ciertos elementos en la página, permite el uso de elementos externos mediante el uso de referencias a sus ubicaciones en el servidor web correspondiente [12].

```
8 <!DOCTYPE html>
9 <html>
10
11 <head>
12 <title>Ejemplo HTML</title>
13 </head>
14
15 <body>
16 <p>Este es un ejemplo de HTML en el que el cuerpo está
17     constituido de un párrafo</p>
18 </body>
19
20 </html>
21
```

Figura 2.10 Ejemplo de código HTML
Fuente: Elaboración propia

2.8.2 Java Server Page

Un JSP o Java Server Page por sus siglas en inglés es un código utilizado para el desarrollo de páginas web usando el lenguaje de programación Java para agregar funcionalidades a la página web y hacerla interactiva con el usuario, permite la interacción con componentes de Java tal como applets de java para realizar una acción determinada dentro de la página. Para el funcionamiento de una aplicación web basada en JSP es necesario el uso de un servidor web, por ejemplo el servidor Apache Tomcat.

2.8.3 Servidor Web

Un servidor web es un servidor que brinda acceso a un número de usuarios a una aplicación web, en él se realiza todo el procesamiento correspondientes a el aplicativo que corre en la página web a la cual accede el usuario a través de su navegador, así como servidor web se tiene un cliente web en cada una de las computadoras que se conectan a este servicio, y el uso de este modelo cliente-servidor permite reducir el procesamiento en los clientes, y hace que la aplicación web se vuelva portable y multiplataforma al no requerir un software específico para su funcionamiento.

2.8.4 Apache Tomcat

Apache Tomcat es un servidor web que soporta JSP y servlets de java, permite el acceso de los usuarios remotos a la aplicación web en este servidor además de agregar funcionalidades a la página web haciéndola más completa de cara al usuario.

2.8.5 JQuery Mobile

JQuery Mobile es un framework de red orientado a dispositivos móviles y con pantalla táctil, se diseña la aplicación web y se introduce la librería del jquery mobile para darle una forma amigable al usuario que accede a través de un dispositivo móvil, con botones grandes y demás características que optimizan el uso de las aplicaciones web en celulares, tablets, etc.

El jQuery Mobile tiene como una de sus más importantes ventajas la capacidad de acceder al aplicativo sin importar el sistema operativo subyacente, de esta forma se realiza una sola implementación del programa y se compila una sola vez lo cual contribuye con la portabilidad al sistema [13].



CAPÍTULO 3

PLANTEAMIENTOS INICIALES DEL SISTEMA

Dada la problemática descrita en el capítulo 1 surge la necesidad de una solución al problema de la falta de seguridad ciudadana en nuestro país y nuestra ciudad que utilice herramientas de la tecnología actual para la búsqueda de una solución a éste problema, por lo cual ésta tesis plantea diseñar un sistema que permita acceder a la gestión y visualización remota de cámaras de vigilancia instaladas en el hogar y que permitan tener un registro en imágenes de los eventos que se den cuando el residente se encuentre alejado de casa, que permita también alertar al usuario inmediatamente en caso de que un evento sea detectado por el sistema y que además permita al usuario controlar de manera remota el encendido y apagado de luces en el hogar, así como de cualquier otro aparato eléctrico que esté funcionando en el hogar; todo esto integrado en un solo sistema que integre estas funcionalidades y permita controlarlas por medio de la realización de llamadas y la selección de opciones de acuerdo a mensajes previamente grabados, aprovechando así la telefonía IP para brindar un fácil acceso por parte del usuario a esta solución.

3.1 Objetivos

A continuación se listan los objetivos establecidos para esta tesis de acuerdo a la problemática descrita en el capítulo 1.

3.1.1 Objetivo general

El objetivo general es el diseño de un sistema de control domótico y de seguridad que permita supervisar la actividad en el hogar mediante sensores de movimiento, videocámaras, que permita tener puertas controladas remotamente; y que además brinde la facilidad de controlarlo a través de un dispositivo móvil.

3.1.2. Objetivos Específicos

- Evaluación y selección de sensores y cámaras IP de video vigilancia.
- Evaluación y selección del micro controlador a usarse para controlar todos los dispositivos del sistema.
- Diseño de la red de datos para la comunicación entre los diferentes componentes del sistema.
- Programación de red de los diferentes componentes del sistema.
- Configuración de un servidor VoIP que permita al usuario acceder al sistema domótico.
- Desarrollo del programa que integre el servidor de llamadas VoIP con todos los demás componentes de la red.
- Diseño de una base de datos que permita la gestión de cuentas de usuario del sistema.
- Diseño de la aplicación web para supervisión de las cámaras por el usuario
- Análisis económico de la propuesta de solución.

3.2 Planteamientos iniciales

A continuación se explicarán los principales planteamientos que se desean cumplir en el diseño del sistema propuesto.

- El control del sistema deberá estar centralizado en un servidor que permita la gestión de la red interna de telefonía IP, así como el resto de componentes, tales como los sensores y cámaras de video.
- El sistema debe permitir un acceso rápido y sencillo de manera que cualquier usuario sin mayor experiencia o instrucción en temas de computación pueda acceder a él sin problemas.
- Se debe incluir en el sistema el envío de parte del controlador hacia los teléfonos registrados en la red.
- Se plantea como diseño básico el sistema que incluye en su diseño 2 cámaras de vigilancia y dos sensores de movimiento instalados en lugares críticos dentro de la vivienda, así como el control centralizado de las luces en 2 habitaciones.
- El sistema deberá tener un bajo costo, de forma que no implique un gasto muy elevado para la persona que desee implementarlo en su hogar.
- La vivienda para la que este sistema está diseñado cuenta con una conexión de internet de al menos 1Mbps con una IP fija y un Módem/Router inalámbrico brindado.

CAPÍTULO 4

DISEÑO DEL SISTEMA DE CONTROL DOMÓTICO SUPERVISADO DESDE UN TELÉFONO MÓVIL

4.1 Arquitectura del sistema

El diseño del sistema se realizó en base a una vivienda de una sola planta y con un área de 140 metros cuadrados, pero se puede implementar en viviendas de mayor o menor tamaño o con mayor número de plantas considerando el tamaño de la misma para el cableado o el número de sensores y cámaras, además sería pertinente centralizar las conexiones de los dispositivos que forman parte del sistema para cada planta.

A continuación se muestra un esquema de la arquitectura que se plantea para los componentes que formarán parte del sistema:

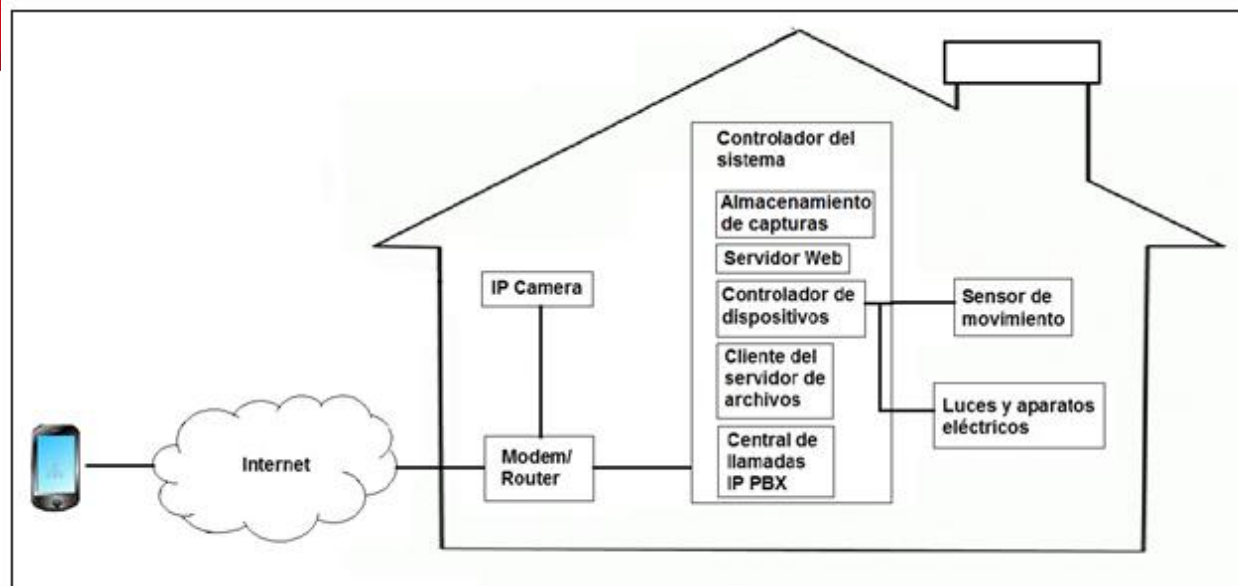


Figura 4.1 Arquitectura del sistema
Fuente: Elaboración propia

En la figura 4.1 podemos observar la arquitectura del sistema que se va a diseñar, una representación de cada uno de los componentes del sistema y como se interconectan entre sí. Posteriormente analizaremos detalladamente cada etapa del diseño de este sistema.

Esta tesis está orientada a su instalación en un hogar con un servicio de internet con salida a través de un módem/router con tecnología inalámbrica proporcionado por el proveedor de servicios de internet.

4.2 Componentes del sistema

Una parte importante del diseño del sistema es seleccionar los dispositivos óptimos para el correcto funcionamiento del sistema, a continuación se describen los dispositivos seleccionados a adquirir para poder implementar el sistema.

4.2.1 Selección del micro controlador

Se tomaron en cuenta un conjunto de micro controladores disponibles actualmente en el mercado, los cuales son:

- Raspberry Pi (Modelo B)
- Arduino Due
- CubieBoard

Teniendo en cuenta que para el sistema es indispensable contar con un microprocesador y con una capacidad de memoria de al menos 512MB para poder procesar el video y la voz necesaria en el diseño del sistema, se realizó una comparativa entre ellos tomando en cuenta factores importantes para nuestro diseño y se desarrolló la siguiente tabla en base a tres micro controladores disponibles actualmente en nuestro país:

	Arduino Due	Raspberry Pi Modelo B	CubieBoard
Velocidad de procesador	85MHz	700MHz	1GHz
RAM	96KB	512MB	1GB
Máximo almacenamiento	512KB	32GB (SD)	32GB (SD)
GPIO(entrada y salida)	54	26	96
Puerto Ethernet	No	Sí	Sí
WIFI	No	No	No
Procesador gráfico	Ninguno	VideoCore IV	ARM Mali-400
Precio	\$ 50.00	\$ 35.00	\$ 50.00

Tabla 4.1 Tabla comparativa de micro controladores
Elaboración propia

Debido a que el Arduino Due no cuenta con el puerto Ethernet ni procesador gráfico que son importantes para el uso que se le va a dar como controlador de las cámaras de video se descarta.

Luego, vemos que el controlador CubieBoard, a pesar de contar con más procesamiento, memoria RAM y número de pines; se ve que ambos tienen las características necesarias para un uso en una casa, en que no hará falta el uso de más de 26 pines para el control doméstico.

Finalmente, debido a que el precio del Raspberry Pi es considerablemente menor que el del CubieBoard, entonces, se escoge el Raspberry Pi como controlador para el sistema.

Dado que el Raspberry Pi ha sido escogido como el micro controlador a usarse en el sistema, se pasará a explicar como se utilizará cada uno de sus componentes para este sistema:

Puerto Ethernet para conector RJ45: Este puerto se usa para la conexión del micro controlador hacia el módem/router; y a través de éste, con los demás elementos del sistema.

Pines GPIO: Los pines de entrada y salida del Raspberry Pi permitirán la conexión del micro controlador con los componentes que se controlarán.

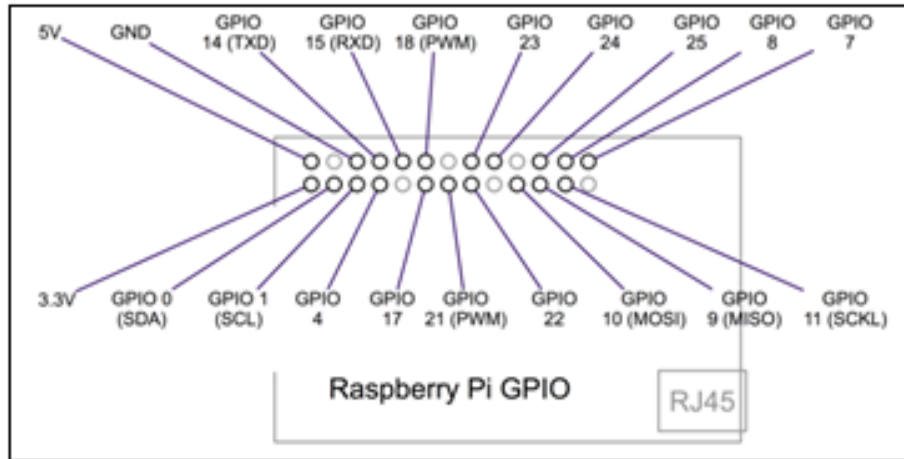


Figura 4.2 Pines GPIO y salida Ethernet del Raspberry Pi
Fuente: [22]

Como se observa en la imagen 4.2, el Raspberry Pi cuenta con 26 pines GPIO, de los cuales algunos son usados para alimentación de 5V o de 3.3V, o para el conector a tierra GND, el resto de pines GPIO pueden ser usados para conectar dispositivos periféricos, sin embargo para este diseño se tendrán en cuenta los GPIOs 4,17,22,23,24,25,7,8; los cuales soportan niveles de entrada/salida de 3.3V y serán usados para conectar el micro controlador a los relés y sensores.



Cámara	TRENDnet TV-IP751WC	TRENDnet TV-IP751WIC	D-Link DCS-930L	D-Link DCS-931L
Tipo	Interior	Interior	Interior	Interior
Máxima resolución	640x480	640x480	640x480	640x480
Compresion	JPEG/MJPEG	JPEG/MJPEG	JPEG/MJPEG	JPEG/MJPEG/H.264
Wireless	Sí	Sí	Sí	Sí
Zoom	4x digital	4x digital	4x digital	4x digital
Vision nocturna	No	Sí	No	Sí
Precio	\$ 105.00	\$ 130.00	\$ 69.00	\$ 109.00

Tabla 4.2 Tabla comparativa de cámaras de vigilancia
Fuente: [19]

Las cuatro cámaras son muy similares, y todas cumplen con los requisitos de tener tecnología IP, compresión JPEG, las cuatro permiten conexión inalámbrica usando Wifi, sin embargo hay una cámara significativamente más económica que las otras, no cuenta con visión nocturna pero el diseño del sistema no lo necesita. Entonces se elige la cámara D-Link DCS-930L.

4.2.3 Dispositivos móviles

Debido a que se trata de un sistema basado en una central IP PBX Asterisk, cualquier dispositivo móvil con un softphone para telefonía IP y acceso a internet bastará para el funcionamiento del sistema.

4.2.4 Softphone Zoiper

Zoiper es un softphone multi-plataforma, disponible en android, iOS(iphone), Windows, Mac, Linux, y permite que el dispositivo en el cual está instalado, sea una computadora, una tablet o un smartphone puedan registrarse como extensiones de una central IP PBX para su uso en una red de telefonía de voz sobre IP.



Figura 4.3 Softphone Zoiper multi-plataforma
Fuente: [28]

4.2.5 Selección de los relés

Para el control de las luces y aparatos eléctricos en el hogar se necesitará una tarjeta de relés que soporte circuitos de 220V de voltaje, al ser éste el voltaje de las conexiones eléctricas en las viviendas en nuestro país, pues se desea controlar más de un circuito de forma individual.

Se usará una tarjeta de relés fabricada en el Perú por la marca Delcomp, pues al ser un producto nacional, facilita el mantenimiento por ser menor el tiempo y la dificultad que implicaría contactarse con el fabricante para la adquisición y mantenimiento del producto.



Figura 4.4 Tarjeta de tarjeta Delcomp RelayX4
Fuente: [21]

Ésta tarjeta funciona con una alimentación de 12 Vdc, para el funcionamiento de los relés, cuenta con cuatro de ellos, los cuales son controlados por cuatro pines que recibirán la órden de encendido y apagado del micro controlador, y su funcionamiento es el siguiente [21]:

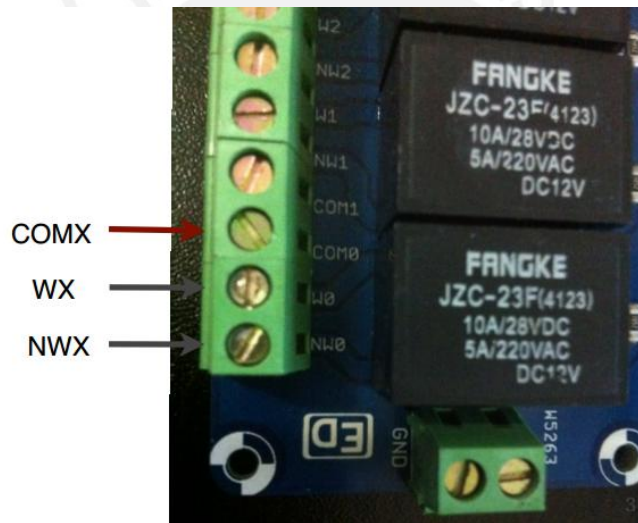


Figura 4.5 terminales de los relés tarjeta relayx4
Fuente: [21]

El primer relé tiene tres terminales, tal como se observa en la figura anterior, cuando el relé está en el modo de apagado, se tiene un circuito que une el terminal COMX con el NWX, y si el relé se encuentra en la posición de encendido, se cerrará el circuito entre COMX y WX.

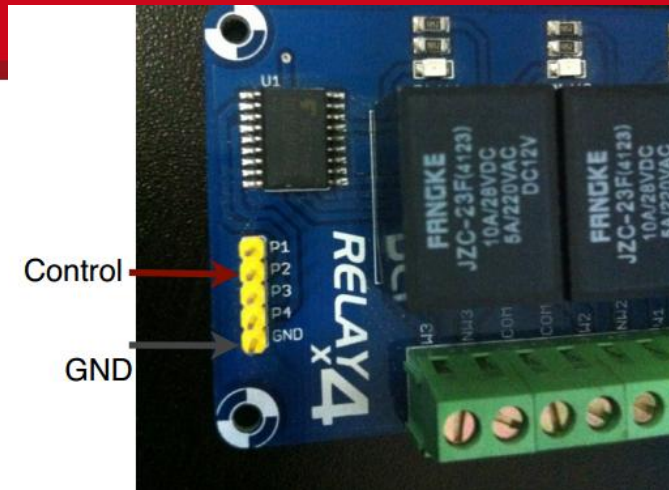


Figura 4.6 Pines de control de la tarjeta relayx4
 Fuente: [21]

El pin P1 es el pin de control del primer relé, si éste recibe una señal de alta por parte del micro controlador, este pondrá el relé en la posición de encendido, esto se cumple para todos los relés de manera similar.

4.2.6 Sensor de movimiento PIR HC-SR501

Luego de realizar una investigación para la elección del sensor de movimiento, se encontró una sola alternativa en el mercado nacional que pueda funcionar con la solución planteada debido a tener características de una alimentación de 5 Vdc y con una cobertura de 180°, el cual es el sensor infrarrojo PIR HC-501..

El HC-SR501 es un sensor de movimiento infrarrojo , el único que es óptimo para aplicaciones de domótica y se distribuye comercialmente en nuestro país y tiene las siguientes características:

- Alimentación: 3,3 a 5 Vdc
- Comunicación: un solo bit de alta / baja
- Dimensiones: 32,2 x 24,3 x 25,4 mm
- Rango de temperatura: 0 a +50 ° C
- Ángulo de detección: 180°
- Distancia de detección: Hasta 7 metros

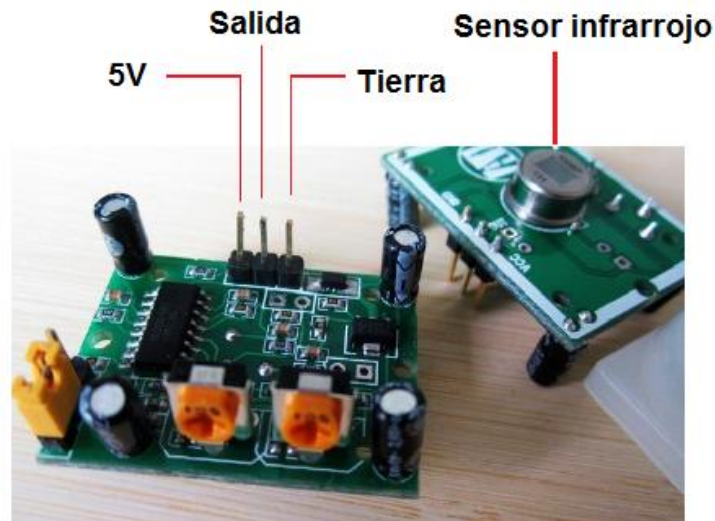


Figura 4.7 Sensor de movimiento PIR
 Fuente: Elaboración propia

Este sensor además incluye una cubierta plástica en forma de domo que actúa como un lente que permite que el sensor detecte con un ángulo de 180°



Figura 4.8 Sensor de movimiento PIR HC-SR501
 Fuente: [20]

Se eligió el sensor pues sus características como la distancia y ángulo de detección, así como su temperatura de operación y su tamaño reducido lo hace óptimo para éste diseño, además tiene un costo bajo en \$5.90.

4.2.7 Sistema operativo Raspbian

En el micro controlador se usará el sistema operativo Raspbian desarrollado para arquitectura ARM basado en Debian, este sistema operativo es una distribución de Linux, y como tal es software libre y permite la instalación del servidor IP PBX que se tiene en el diseño del sistema, también permite el uso del lenguaje de programación Java, de manera que lo hace óptimo para este diseño [23].

4.2.8 Lenguaje JAVA

Se escogió para el sistema el lenguaje de programación java debido a que es un sistema operativo basado en objetos con una sintaxis bastante simple y familiar a la mayoría de programadores, su máquina virtual es más veloz que sus competidores Ruby, Python, Perl y Squeak, y si bien cuenta con una gran cantidad de librerías, será necesario instalar algunas adicionales para la interacción con los otros componentes del sistema [24].

4.2.9 Librería Asterisk-Java

Esta librería consiste en un conjunto de clases de java que facilitan la comunicación entre un programa y un servidor PBX Asterisk, trabaja bajo la licencia Apache 2.0.

Requiere la instalación de Asterisk 1.0 o mayor, así como del Java Runtime Environment 1.6 o mayor, para su funcionamiento necesita que el servidor Asterisk también se encuentre en funcionamiento [25].

4.2.10 Librería PI4J

La librería PI4J tiene por finalidad integrar las librerías nativas del Raspberry Pi con Java para brindar a los programas escritos en Java completo acceso a las funcionalidades del Raspberry Pi:

Permite exportar pines GPIO, configurar pines como entrada o salida, permite fijar el estado de los pines para salida, así como leer el estado de los mismos para entrada, facilita la automatización basada en cambios de estado de los pines [26].

4.2.11 Servidor web Apache Tomcat

Se necesitará una interfaz que permita que el usuario tenga acceso a las imágenes captadas por las cámaras de video, para ello se ha considerado un servidor web, debido a las propiedades mencionadas en el capítulo 2, tales como la capacidad de acceder a él desde cualquier plataforma, sin importar el sistema operativo del cliente.

4.3 Diseño de la red

Para el diseño de este sistema se consideró una red local de una vivienda de una extensión de 140 metros cuadrados con una sola planta (la cual se describe en el punto 1 del capítulo 4 del documento) definida por el módem/router brindado por el operador de servicio de internet, el cual cumple función de servidor dhcp, es decir, puede brindar direcciones IP a los elementos que se conecten a la red.

La red de área local es la red 192.168.1.0 con máscara de subred 255.255.255.0, y para el sistema los elementos de la red serán los siguientes:

	Dirección de red	Máscara de subred	Puerta de enlace
Modem/Router	192.168.1.1	255.255.255.0	
Raspberry pi	192.168.1.100	255.255.255.0	192.168.1.1
Cámara de la sala	192.168.1.101	255.255.255.0	192.168.1.1
Cámara de la otra habitación	192.168.1.102	255.255.255.0	192.168.1.1
Teléfono móvil	192.168.1.103	255.255.255.0	192.168.1.1

Tabla 4.3 Direccionamiento IP para la red local
Fuente: Elaboración propia

El sistema que se diseña en esta tesis cuenta con cuatro segmentos claramente diferenciados, de acuerdo a la función que se realiza en cada uno de ellos y que se integran para cumplir todas las funciones del sistema:

- Segmento de IVR
- Segmento de control
- Segmento de vigilancia
- Segmentos de visualización

A continuación se explicará en detalle el diseño de cada uno de estos segmentos.

4.4 Segmento de IVR

Primero, será necesaria la instalación del servidor IP PBX en el micro controlador, como ya se definió se usará un servidor Asterisk, se instalará la versión 11.5.0 de Asterisk, la cual tiene buen desempeño en procesadores ARM, y para poder establecer el servidor será necesario configurar los archivos siguientes:

4.4.1 Archivo de configuración Sip

Éste es un archivo de configuración del servidor Asterisk que permite al administrador agregar cuentas de usuario que puedan registrarse en el servidor.

Para agregar la cuenta de usuario 1001 con contraseña 1001 y que tenga también ese número como callerid, que se encuentre en el contexto internos y soporte los códecs g729, ulaw, alaw y gsm para el audio se agrega las siguientes líneas al sip.conf.

```
GNU nano 2.2.6 File: /etc/asterisk/sip.conf
[1001]
type=friend
secret=1001
callerid="1001" <1001>
host=dynamic
context=internos
allow=g729
allow=ulaw
allow=alaw
allow=gsm
```

Figura 4.9 Agregar una cuenta de usuario en el archivo sip.conf
Elaboración propia

4.4.2 Archivo de configuración Extensions

El archivo extensions.conf es también conocido como plan de discado pues en él se configuran todas las acciones que se pueden realizar con el marcado de dígitos de los usuarios.

Es en este archivo donde se realizará la configuración necesaria para que el usuario reciba los mensajes pre-grabados al llamar a la central y pueda dar todas las órdenes que se han contemplado para el sistema, así como las llamadas realizadas por el servidor como alertas para eventos determinados, se configurarán las extensiones de manera que se siga la siguiente estructura de menús pregrabados:

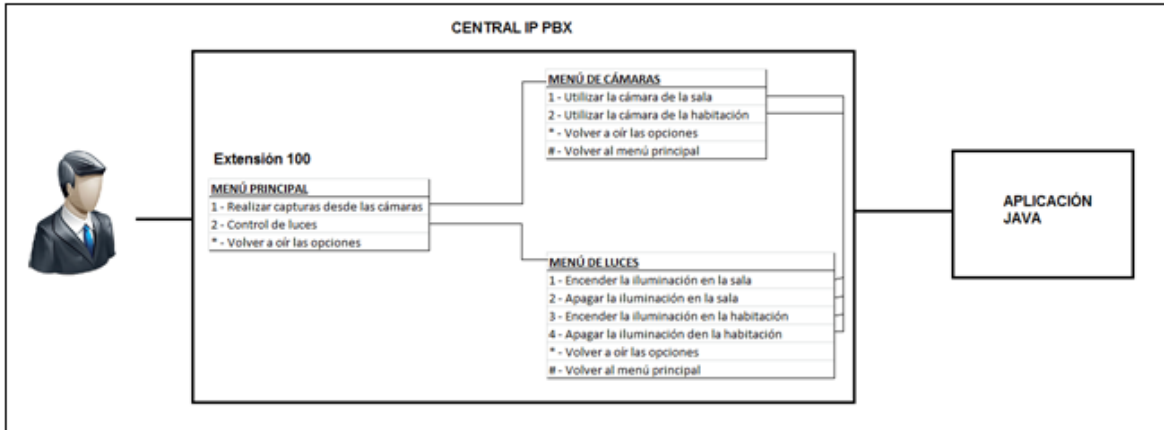


Figura 4.10 Menú de control del sistema
Elaboración propia

Como se observa en la figura 4.10, el usuario al marcar el número de la central IVR (extensión 100) ingresará a un menú, el cual reproducirá un mensaje pregrabado con las opciones de marcación, al ser derivado a uno de los menús siguientes, escuchará un mensaje de manera similar y podrá escoger una de las opciones, la cual será capturada por la aplicación Java gracias a la interacción entre el proceso Java y la central de llamadas usando la librería Asterisk-Java y se ejecutará la acción solicitada.

A continuación se muestra un ejemplo del uso del archivo extensions.conf para un menú con mensajes pre-grabados.

```

GNU nano 2.2.6      File: /etc/asterisk/extensions.conf      Modified
exten => 1001,1,Dial(SIP/1001)
exten => 1001,2,Hangup ()
exten => 100,1,Goto(menu,s,1)
exten => 101,1,Goto(deteccion,s,1)
include => menu
include => deteccion
[menu]
exten => s,1,Answer()
exten => s,n,Wait(0.5)
exten => s,n,Background(menu0)
exten => s,n,WaitExten(5)
exten => 1,1,Goto(menu-a,s,1)
exten => 2,1,Goto(menu-b,s,1)
exten => 3,1,Goto(menu-c,s,1)
exten => 0,1,Dial(SIP/123)
exten => *,1,Goto(s,1)
exten => i,1,Playback(invalida)
exten => i,n,Goto(s,1)

include => menu-a
include => menu-b
    
```

Figura 4.11 Ejemplo de configuración del archivo extensions.conf para un IVR
Elaboración propia

Se puede observar las funciones del usuario 1001, que puede marcar y colgar, y a la extensión 100, que envía al usuario al menú principal, se reproduce mediante la función Background el audio del menú principal menu0.gsm y se redirige a los otros menús dependiendo del dígito marcado.

Este archivo también se usa para el envío de las alertas que tendrá que enviar el sistema al usuario en caso de un evento en el hogar, de manera similar al ejemplo se usan tres acciones:

```

GNU nano 2.2.6      File: /etc/asterisk/extensions.conf      Modified
include => menu

[deteccion]
exten => s,1,Answer()
exten => s,n,Background(movimiento)
exten => s,n,Hangup()
    
```

Figura 4.12 Archivo extensions.conf – Envío de una alerta con un mensaje pregrabado
Elaboración propia

4.4.3 Archivo de configuración Manager

Finalmente, se configurará el archivo manager.conf de manera que se agregue un administrador del servidor para dar acceso remoto a la gestión del servidor, en el sistema se usará para permitir la conexión con la aplicación Java mediante la librería Asterisk-Java, la configuración consiste en agregar la cuenta manager al archivo:

```

GNU nano 2.2.6      File: /etc/asterisk/manager.conf      Modified
[manager]
secret=manager
deny=0.0.0.0/0.0.0.0
permit =192.168.1.100/255.255.255.0
permit=127.0.0.1/255.255.255.0
read=all
write=all
  
```

Figura 4.13 Archivo manager.conf – Cuenta de administrador del servidor para acceso remoto
Elaboración propia

4.4.4 Comunicación con el programa de Java

Como ya se mencionó, el servidor de llamadas se comunica con el programa de Java utilizando la librería de java Asterisk-Java-0.3.jar.

De forma específica se usa el Manager API, el cual es un conjunto de objetos y métodos utilizados para establecer una sesión de gestión remota del servidor de llamadas utilizando una cuenta declarada en el archivo manager.conf.

Para ello se usa un objeto ManagerConnection que es una instancia de la conexión con la central PBX:

```

private ManagerConnection managerConnection;

public Control(String machine, String user, String password) throws IOException
{
    ManagerConnectionFactory factory = new ManagerConnectionFactory(machine, user, password);
    this.managerConnection = factory.createManagerConnection();
}
  
```

Figura 4.14 Declaración y construcción de la conexión con la central PBX
Elaboración propia

En la imagen anterior se observa parte de un código que corresponde a la declaración y construcción del objeto ManagerConnection, como vemos su constructor requiere los parámetros Machine (Dirección del servidor de llamadas), User (usuario configurado en manager.conf) y Password (Contraseña de ese usuario).

Este objeto permite el uso de Listeners para detectar llamadas, dígitos marcados, o nuevos canales de comunicación abiertos.

```
public void run() throws Exception
{
    // Eventos de asterisk
    managerConnection.addEventListener(this);
}
```

Figura 4.15 Listener para eventos del servidor Asterisk
Elaboración propia

En la imagen anterior observamos como se agrega un listener que detecte una nueva acción, debido al objeto ManagerConnection se podrá acceder a toda la información de este evento, tal como callerid del que inició la llamada, canal de comunicación usado, dígitos marcados, entre muchos otros.

Para realizar el envío de alertas se utiliza el objeto originateAction del API:

```
public void MandarAlerta(ManagerConnection managerConnection) throws IOException, IllegalStateException,
OriginateAction originateAction;
ManagerResponse originateResponse;
System.out.println(managerConnection.getState().name());
originateAction = new OriginateAction();
originateAction.setChannel("SIP/1002");
originateAction.setContext("internos");
originateAction.setExten("101");
originateAction.setPriority(new Integer(1));
originateAction.setTimeout(new Integer(30000));

originateResponse = managerConnection.sendAction(originateAction, 30000);
System.out.println(originateResponse.getResponse());
}
```

Figura 4.16 Originar acción – Envío de alerta
Elaboración propia

El método mostrado en la figura 4.16 origina una acción que consiste en una llamada a partir de la información brindada por el objeto ManagerConnection, Se origina una acción, la cual tiene como destino el "SIP-1002" y contexto "Internos", que es el contexto que contiene a esta extensión. Se lo comunica con la extensión 101, la cual está programada en el IVR para dar una alerta al usuario sobre un evento en su hogar.

4.4.5 Salida de la red local

Para poder registrar las cuentas en los teléfonos que servirán de interfaz para el usuario, se adquirirá una dirección privada del proveedor de servicios, una vez que se registra el teléfono al servidor de llamadas puede acceder a éste siempre que tenga acceso a internet sin importar su ubicación.

4.5 Segmento de control domótico

4.5.1 Diagrama del segmento de control domótico

Este segmento comprende los componentes del sistema relacionados al control domótico, desde la orden hasta la ejecución de la acción, a continuación se muestra el diagrama de los componentes del segmento:

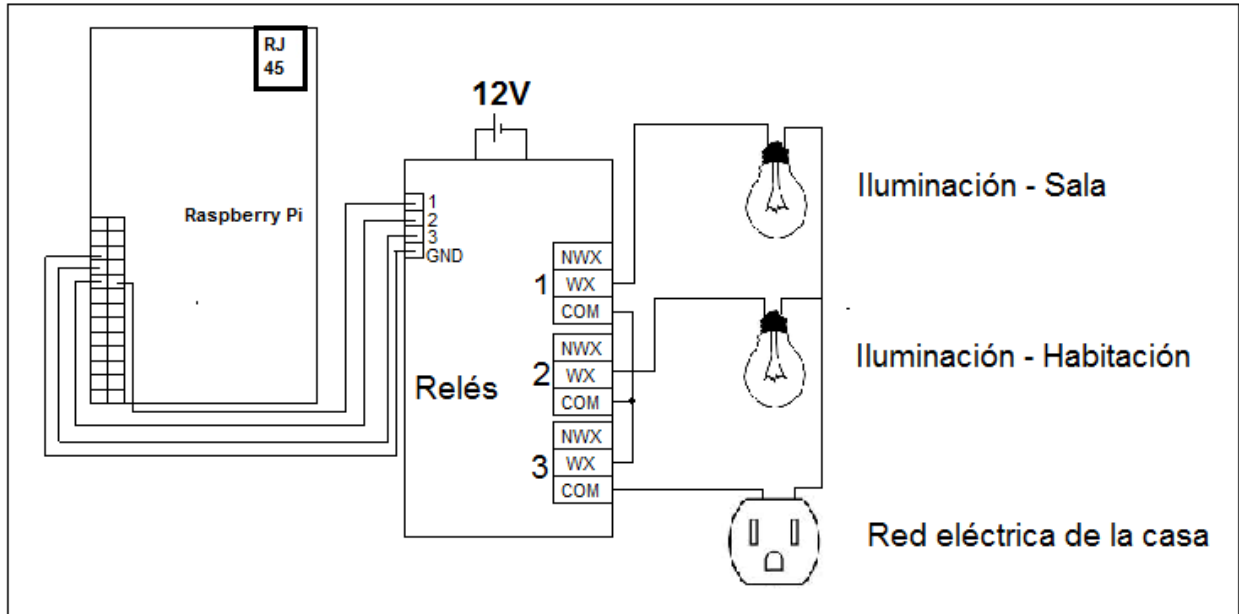


Figura 4.17 Diagrama de segmento de control domótico
Fuente: Elaboración propia

Los sistemas de iluminación de la sala y la habitación se conectan a la red eléctrica a través de la tarjeta de relés, de tal forma cuando los relés correspondientes a cada uno se activen, se completará el circuito respectivo y se encenderán las luces, además de ello estos relés son controlados a través de los pines de control de la tarjeta, los cuales están conectados a los pines GPIO del Raspberry Pi.

4.5.2 Conexiones del segmento de control domótico

En el segmento de control domótico se comunican las luces y aparatos eléctricos a ser controlados por medio de la tarjeta de relés, y los pines a utilizar tanto en la tarjeta RelayX4 como en el Raspberry Pi son los siguientes:

Red eléctrica		Luces (Sala)		Luces (Habitación)	
Relay X4	Raspberry Pi	Relay X4	Raspberry Pi	Relay X4	Raspberry Pi
Pin 1	GPIO_15	Pin 2	GPIO_16	Pin 3	GPIO_18
GND(Tierra)	GPIO_20(Tierra)	GND(Tierra)	GPIO_20(Tierra)	GND(Tierra)	GPIO_20(Tierra)

Tabla 4.4 Conexiones entre la tarjeta de relés y el micro controlador
Elaboración propia

De esta forma se controla de forma separada la alimentación del segmento de control domótico desde la red eléctrica de la vivienda, así como las luces de la sala, y las luces de la habitación, este sistema será controlado por el sensor de movimiento, de tal forma que se enciendan las luces al detectar movimiento y puedan realizar las capturas desde las cámaras con las luces encendidas, también será posible controlar las luces por orden del usuario para su encendido y apagado mediante los menús de llamadas en caso el usuario así lo necesite.

En la figura 4.18 se puede observar la forma en que se realizan las conexiones necesarias para cada controlar un circuito conectado a la red eléctrica en el segmento de control domótico.



Figura 4.18 Conexiones necesarias para controlar un circuito con un relé y el micro controlador
Elaboración propia

Por otro lado, las conexiones a realizar en el micro controlador, usando los pines definidos en la tabla de conexiones, son las siguientes.



Figura 4.19 Conexiones en el raspberry pi para la comunicación con la tarjeta de relés
Fuente: Elaboración propia

4.5.3 Flujo del segmento de control doméstico

Las órdenes de este segmento son recibidas por el servidor IP PBX en la forma de dígitos marcados en el menú de luces de la central IVR, se interpreta en el programa de java la extensión marcada como un evento, y se ejecuta la acción, como se observa en el siguiente diagrama de flujo.

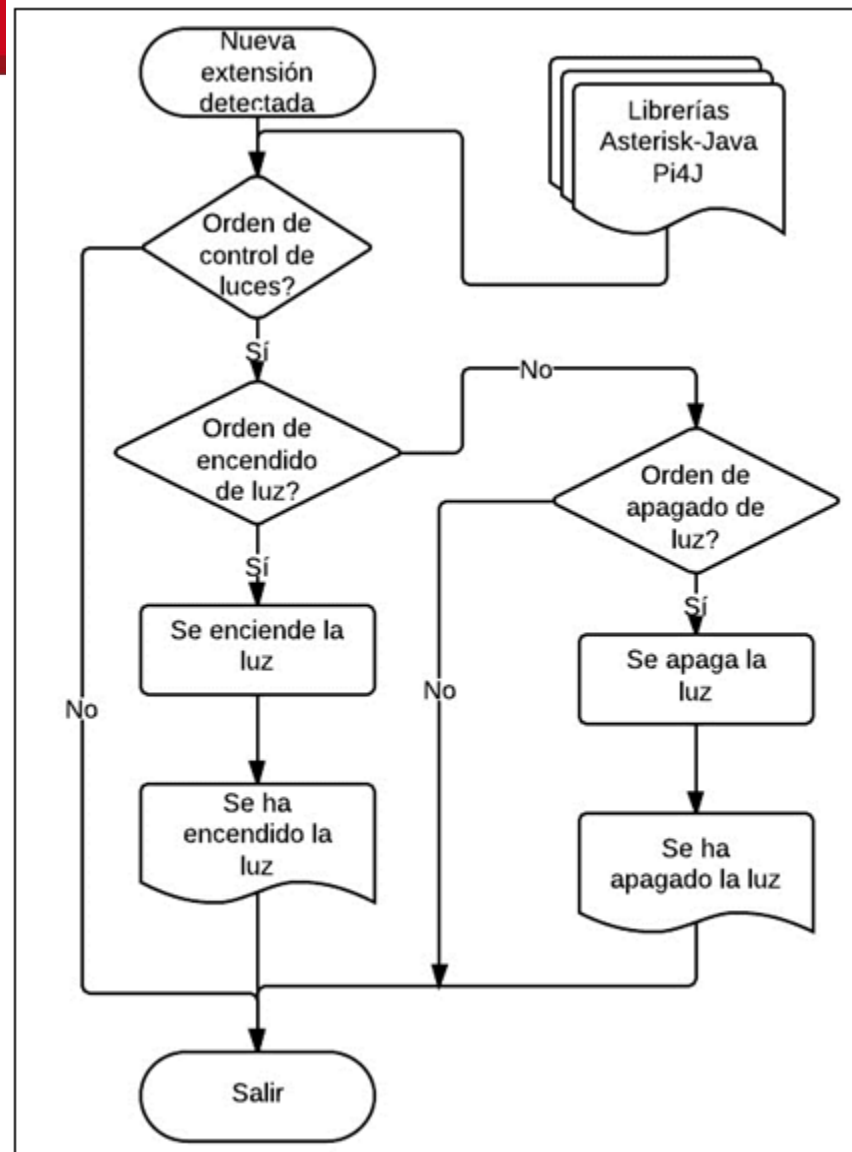


Figura 4.20 Diagrama de flujo del segmento de control doméstico
Elaboración propia

En el diagrama de flujo se observa que la entrada para el flujo del segmento de control doméstico es la extensión marcada por el usuario en el menú de la central IVR, se importan las librerías Asterisk-Java para la comunicación entre el programa y la central de llamadas, y PI4J para el control de los pines GPIO del raspberry pi desde el programa, en si la orden era de encendido de las luces, se ejecuta la acción y se reproduce un mensaje de éxito de la acción al usuario, de manera similar si la orden era de apagado de las luces, se ejecuta la acción y se envía el mensaje pre grabado para expresar el éxito de la acción.

4.5.4 Comunicación con el programa de Java

Como se mencionó anteriormente, para poder controlar los pines del raspberry Pi desde el programa de Java se usa la librería Pi4J, cuyo funcionamiento se basa en la creación de un objeto para controlar los pines o GpioController y un objeto para cada uno de los pines que se desee manejar desde el programa, ya sea de entrada (GpioPinDigitalInput) o de salida (GpioPinDigitalOutput).

```
final GpioController gpio = GpioFactory.getInstance();
final GpioPinDigitalInput sensor7 = gpio.provisionDigitalInputPin(RaspiPin.GPIO_07, PinPullResistance.PULL_DOWN)
final GpioPinDigitalOutput luz1 = gpio.provisionDigitalOutputPin(RaspiPin.GPIO_01, "pin18", PinState.LOW);
```

Figura 4.21 Declaración de los objetos de los pines GPIO_07 y GPIO_01
Elaboración propia

En la figura 4.21 se muestra parte de un código que corresponde a las declaraciones de tres objetos, el primero gpio es un GpioController, el cual establece y gestiona la comunicación con los pines del raspberry pi, los dos siguientes son pines específicos, uno de entrada y el otro de salida, además se inicializan con estado inicial Down.

```
sensor7.addListener(new GpioPinListenerDigital() {
public void handleGpioPinDigitalStateChangeEvent(GpioPinDigitalStateChangeEvent event) {
if(sensor7.isHigh()){
System.out.println("\n --> GPIO PIN STATE CHANGE: " + event.getPin().getName() + " = " + event.getState().getName()
luz1.high();
```

Figura 4.22 Declaración de los objetos de los pines GPIO_07 y GPIO_01
Elaboración propia

En la figura 4.22 se observa parte de un código que corresponde a un listener que se establece en el gpio "sensor7", de forma que al detectarse un cambio de estado de low hacia high, se pone el gpio "luz1" a posición high, y de esta forma se puede prender una luz conectada a este PIN, en este caso a través de la tarjeta de relés.

4.6 Segmento de vigilancia

4.6.1 Ubicación de los componentes

Este segmento consiste en el proceso desde la detección de movimiento, la captura de imágenes en las cámaras de vigilancia hasta el almacenamiento de estas imágenes en el servidor

De acuerdo a los planteamientos iniciales del sistema, se usarán dos cámaras de vigilancia, una de ellas será instalada apuntando hacia la entrada principal de la

vivienda, acompañada por un sensor de movimiento que detecte el movimiento y active la captura de imágenes.

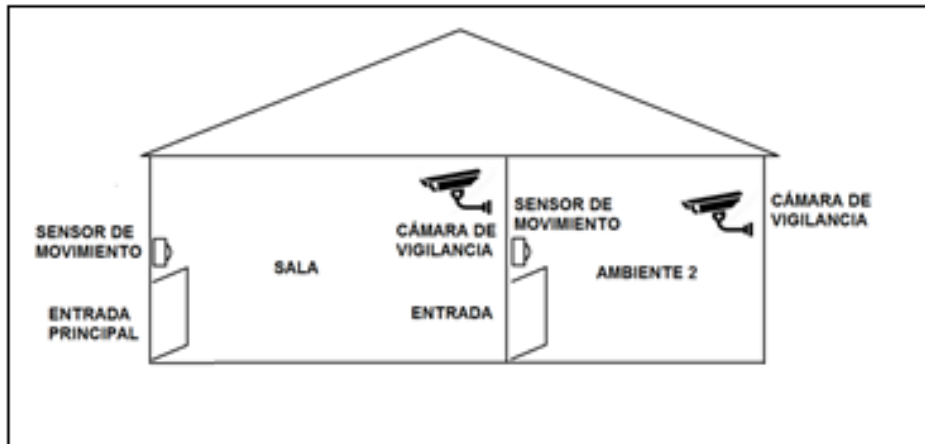


Figura 4.23 Ubicación de componentes de vigilancia
Fuente: Elaboración propia

En la figura 4.23 se puede observar la propuesta de ubicación de los componentes del segmento de vigilancia dentro de la vivienda, como ya se mencionó, se tienen dos puntos de vigilancia: El primero es la entrada principal de la vivienda y el otro es un ambiente que sea determinado como un punto crítico donde se necesite registrar los eventos, por ejemplo el cuarto donde el residente guarde sus objetos de valor.

4.6.2 Conexiones del segmento de vigilancia

Considerando que la cámara de vigilancia escogida para el sistema soporta el estándar 802.11 WiFi, es decir no es necesario realizar la conexión alámbrica con la puerta de enlace de la red, con alimentar la cámara bastará.

Por otro lado la conexión de los sensores de movimiento PIR se conectarán al micro controlador de la siguiente manera:

Sensor (Sala)		Sensor (Habitación)	
PIR HC-SR501	Raspberry Pi	PIR HC-SR501	Raspberry Pi
5V	GPIO_2 (5V)	5V	GPIO_4(5V)
OUT	GPIO_7	OUT	GPIO_11
GND(Tierra)	GPIO_6 (Tierra)	GND(Tierra)	GPIO_9(Tierra)

Tabla 4.5 Conexiones de los sensores de movimiento y el micro controlador
Elaboración propia

En la tabla se observa la selección de pines del raspberry pi que serán usados para las conexiones con los dos sensores infrarrojos de movimiento, relacionadas con los pines de los dos sensores respectivamente.

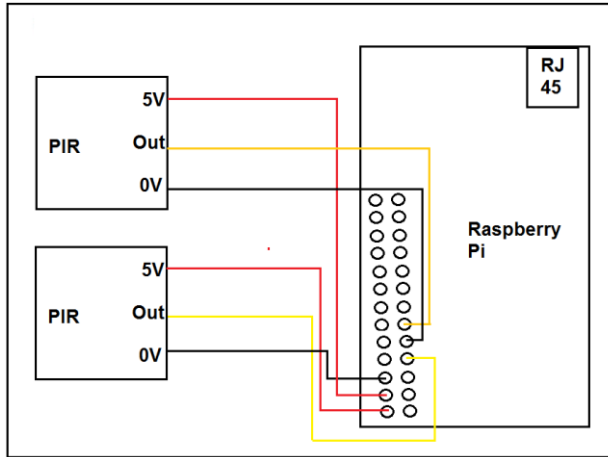


Figura 4.24 Diagrama para la conexión de los sensores y el micro controlador
Elaboración propia

En la figura 4.24 se muestra gráficamente los pines escogidos y el diagrama de como se conectan los sensores y el micro controlador.

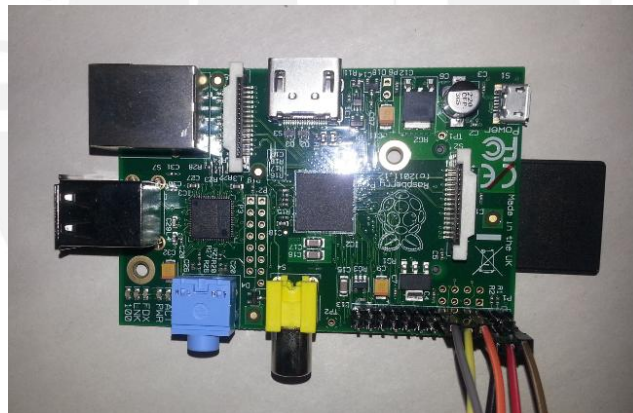


Figura 4.25 Raspberry Pi con las conexiones correspondientes a los dos sensores de movimiento
Elaboración propia

En la figura 4.25 se muestra un micro controlador raspberry pi con las conexiones necesarias para su comunicación con los sensores de movimiento PIR HC-SR501.

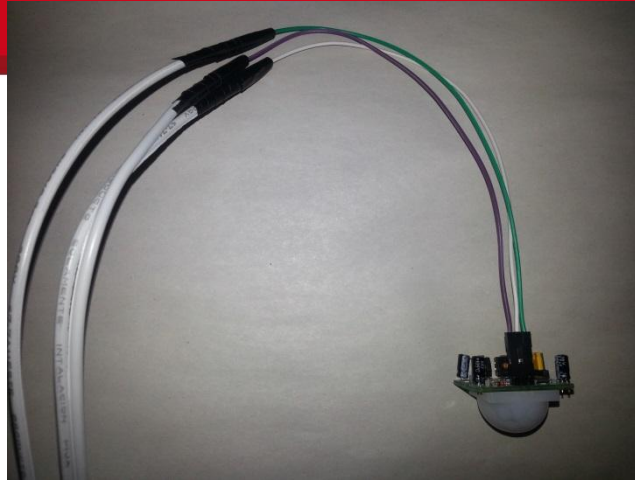


Figura 4.26 Conexión del sensor PIR
Elaboración propia

En la figura 4.26 se puede apreciar una fotografía de un sensor PIR conectado mediante un cable de 15 metros con el cual se realizaron las pruebas de funcionamiento del sensor a los 3 pines del raspberry pi.

4.6.3 Funcionamiento de las cámaras

Las cámaras de vigilancia IP, como ya se mencionó, serán las D-link DCS-930L, las cuales utilizan compresión Motion JPEG, en el cual el video capturado consiste de una secuencia de imágenes comprimidas utilizando el formato JPEG.

Las cámaras D-link DCS-930L permiten una resolución de video de 640x480 pixels y una frecuencia de imagen de hasta 20fps en esa resolución. Sin embargo para el diseño se ha optado por la captura individual de imágenes JPEG cada dos segundos, debido a que en aplicaciones de video vigilancia no es necesario tener acceso a 20 fps, esto considerando el movimiento que realizaría el intruso en la vivienda, además de ello un video de M-JPEG de un minuto en esa resolución podría llegar a ocupar hasta 60 Mega Bytes, mientras que cada captura de imagen en JPEG de esta cámara puede ocupar hasta 30 Kilo Bytes.

Por último el uso de imágenes individuales facilita la navegación del usuario para visualizar cada instante del evento.

Si consideramos una tarjeta de almacenamiento de 4 GB en la que se ha instalado el sistema operativo del microcontrolador el cual ocupa 1.8 GB, se considera 1.2 GB para el almacenamiento de las capturas, tenemos la siguiente tabla muestra el cálculo del espacio ocupado en la memoria por las capturas de las cámaras de vigilancia.

Captura JPEG (640x480)	Número de capturas por evento	Evento almacenado en la memoria	Capacidad de la memoria	Eventos que se pueden almacenar
30 KB	10	0.3MB	1,200 MB	4000 Eventos

Tabla 4.6 Cálculo de la capacidad de almacenamiento
Elaboración propia



Figura 4.27 Captura de una cámara D-Link DCS-2130 a 640x480
Elaboración propia

La figura 4.27 muestra una imagen en formato jpeg capturada por la cámara con la resolución de 640x480, como se observa está a 2 metros de la puerta que aparece en la imagen y se puede apreciar bien. Como se quiere tener la cámara en la misma sala a no más de 5 metros, no se tendrá problemas con la resolución de la cámara.



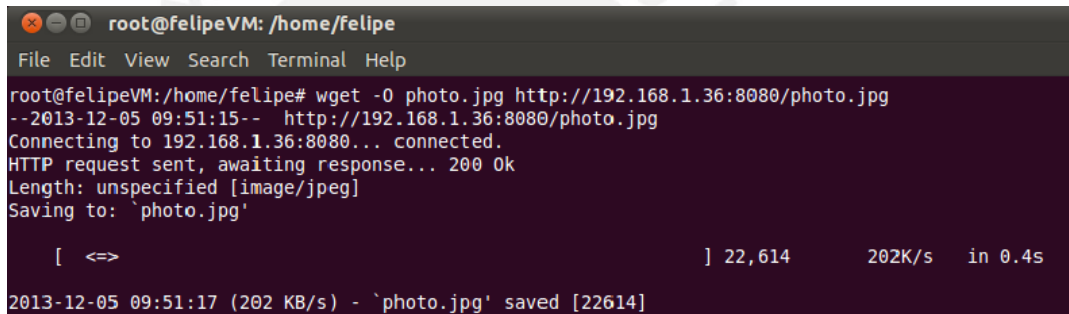
Figura 4.28 Cámara Dcs-2130 conectada mediante un cable Ethernet
Elaboración propia

Como se observa en la figura 4.28 se usó una cámara DCS-2130, que a pesar de ser más costosa que la cámara que se tiene en el diseño soporta también el formato de compresión de imágenes JPEG a 640x480 para la captura y descarga de la imagen a través desde la red hacia el dispositivo raspberry pi a través del terminal de Linux se usa el siguiente comando:

```
➤ wget -O /home/photo.jpg --user=admin --password=admin http://192.168.1.33:80/dms.jpg
```

El comando `wget` permite la descarga de archivos desde la red y la opción `-O /home/photo.jpg` permite el almacenamiento de la imagen en ese archivo, debido a que las cámaras d-link usadas permiten verificación de usuario y contraseña se usan los comandos `--user=admin` `--password=admin` y finalmente el socket de red de la cámara que consiste en la dirección 192.168.1.33 y el puerto 80, el comando `dms.jpg` permite la captura de imágenes individuales en las cámaras d-link.

En la siguiente imagen se muestra el output del terminal de Linux ante una descarga de una imagen desde una cámara IP.



```
root@felipeVM: /home/felipe
File Edit View Search Terminal Help
root@felipeVM:/home/felipe# wget -O photo.jpg http://192.168.1.36:8080/photo.jpg
--2013-12-05 09:51:15-- http://192.168.1.36:8080/photo.jpg
Connecting to 192.168.1.36:8080... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [image/jpeg]
Saving to: `photo.jpg'

 [ <=> ] 22,614 202K/s in 0.4s
2013-12-05 09:51:17 (202 KB/s) - `photo.jpg' saved [22614]
```

Figura 4.29 Salida del terminal ante una descarga de imagen de la cámara IP
Elaboración propia

Nótese que en este caso se usó una cámara sin usuario ni contraseña y con dirección IP 192.168.1.36 y puerto 8080 para la captura de la imagen.

4.6.4 Flujo del segmento de vigilancia

Debido a que el segmento de vigilancia implica el encendido de las luces para poder realizar las capturas desde las cámaras de vigilancia, esto tiene que verse reflejado en el diagrama de flujo correspondiente a este segmento del sistema, el cual se observa a continuación:

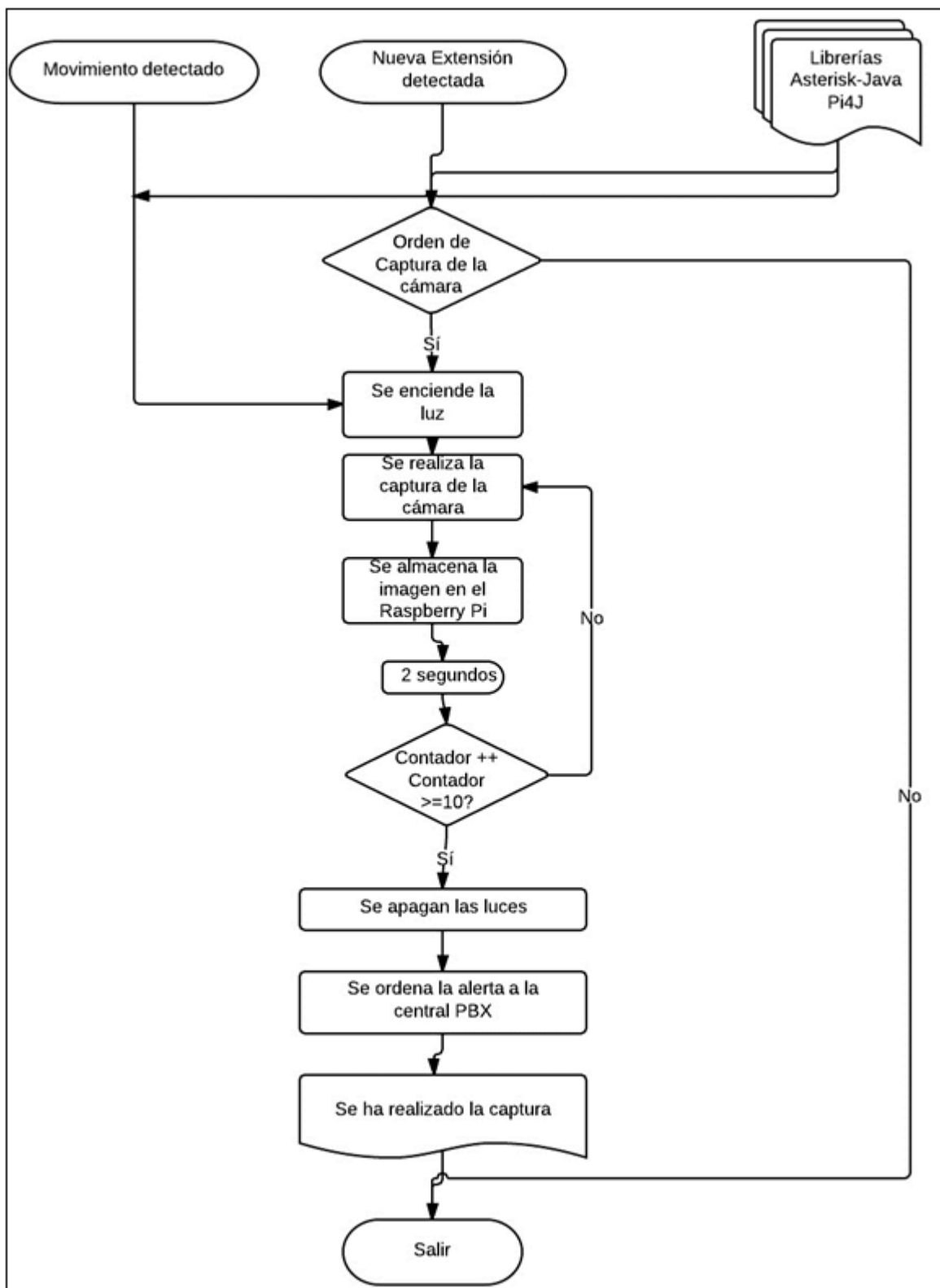


Figura 4.30 Diagrama de flujo del segmento de capturas con las cámaras de vigilancia
Elaboración propia

En el diagrama se puede observar que existen dos posibles eventos que accionan este segmento, el primero es una llamada desde el servidor de llamadas, y el segundo es mediante la detección de movimiento en uno de los sensores, luego de ello se enciende la luz para poder capturar la imagen con más calidad, y se toman las capturas de imagen cada dos segundos durante veinte segundos, y finalmente se envía una alerta al usuario también mediante la central de llamadas.

```

sensor7.addListener(new GpioPinListenerDigital() {
public void handleGpioPinDigitalStateChangeEvent(GpioPinDigitalStateChangeEvent event) {
if(sensor7.isHigh()){
System.out.println("\n --> GPIO PIN STATE CHANGE: " + event.getPin().getName() + " = " + event.getState().getName());
luz1.high();
for(int x=1;x<=10;x=x+1){
SimpleDateFormat df = new SimpleDateFormat("ddmmyyHHmmss");
Date rn = Calendar.getInstance().getTime();
String reportdate = df.format(rn);
String rutadescarga = reportdate + ".jpg";
String comandodescarga="wget -O "+ rutadescarga + " http://" + ip_cam1 + "/photo.jpg";
TomarFoto(ip_cam1, comandodescarga);
try {
Thread.sleep(2000);
} catch (InterruptedException ex) {}
try {MandarAlerta(managerConnection);} catch (IOException ex1) {} catch (IllegalStateException ex) {}
} catch (AuthenticationFailedException ex) {} catch (TimeoutException ex) {}
luz1.low();
}
}
}

```

Figura 4.31 Captura de imágenes desde la cámara de vigilancia
Elaboración propia

En la figura 4.31 se observa el código correspondiente a un listener que se activa cuando se da un cambio de estado del objeto sensor7, entonces se activa el pin luz1 para el encendido de una luz, se entra a un bucle que captura 10 fotos en formato de compresión jpeg soportado por la cámara de vigilancia desde la dirección ip_cam1, esperando 2000 milisegundos entre capturas y las descarga guardándolas con la fecha en que se capturaron como nombre del archivo, y finalmente se envía una alerta utilizando la conexión con el servidor PBX y se apaga el pin luz1, apagando así la luz conectada a él mediante la tarjeta de relés.

4.7 Segmento de visualización

Para brindar al usuario acceso a las capturas realizadas por las cámaras de vigilancia se opta por instalar un servidor web Apache en el Raspberry Pi, en el cual el usuario, luego de identificarse con una cuenta de usuario y una contraseña, puede acceder a las capturas, las cuales están archivadas en el servidor y ordenadas por fecha y hora del evento que las originó.

Gracias a que se cuenta con una dirección IP pública se podrá acceder a este servidor Web desde cualquier ubicación siempre que se tenga acceso a internet.

4.7.1 Diagrama de casos de uso de la aplicación Web

En la siguiente imagen se muestra el diagrama de caso de uso de la aplicación web, en la cual el usuario empieza por autenticarse usando un usuario y contraseña, selecciona de una lista de eventos, el evento cuyas capturas desea visualizar y accede a éstas:

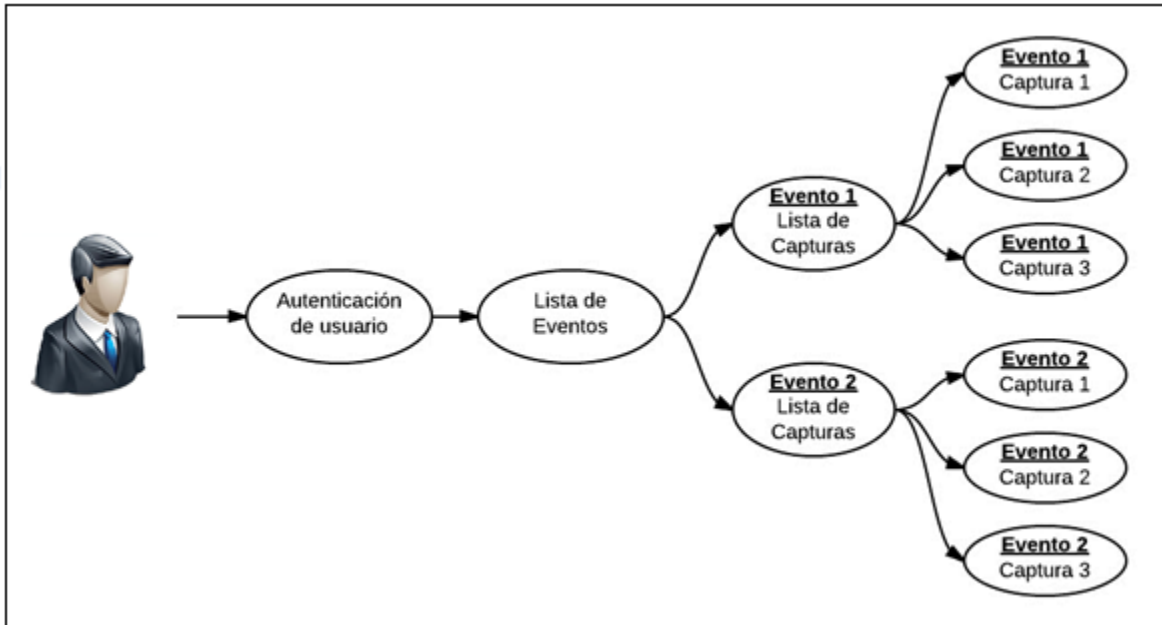


Figura 4.32 Diagrama de caso de uso – Aplicación web
Elaboración propia

Para la aplicación web se utilizó jQuery mobile para optimizar su diseño en un dispositivo móvil, como se explicó en marco teórico, a continuación se describen cada una de las pantallas de la aplicación web.

4.7.2. Autenticación de usuario

En la pantalla de autenticación de usuario se permite al usuario ingresar su cuenta de usuario y contraseña para acceder a la vigilancia de su casa:

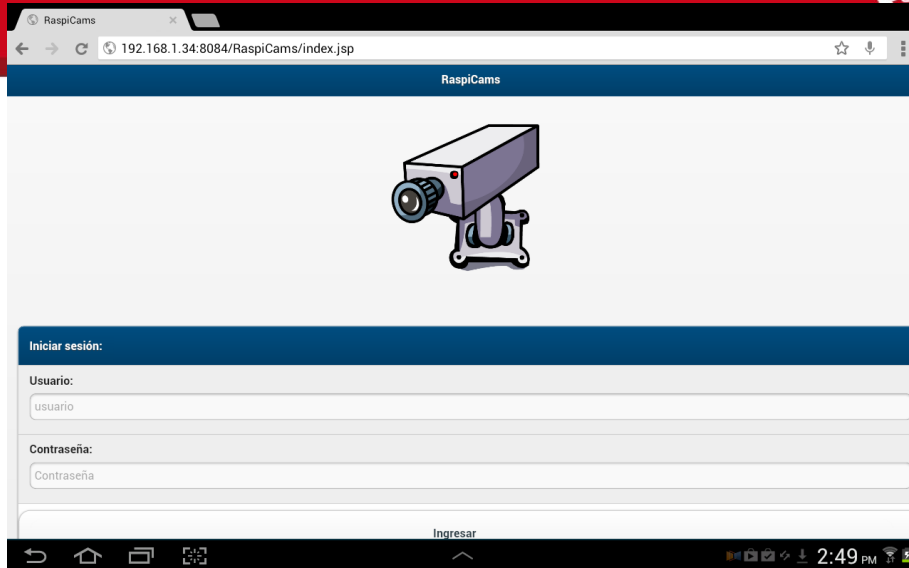


Figura 4.33 Pantalla de ingreso del usuario y la contraseña
Elaboración propia

4.7.3 Selección de evento

En la figura 4.34 se observa la pantalla de selección de eventos, en la cual el usuario observa los eventos que ha registrado el sistema con sus fechas y horas respectivas, y puede seleccionar uno de esos eventos para visualizar las capturas realizadas por la cámara.

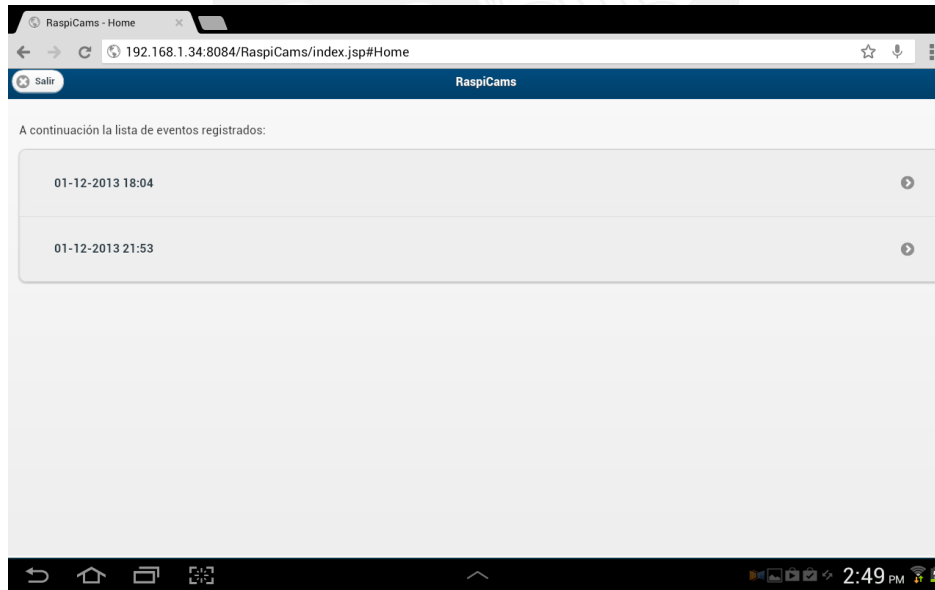


Figura 4.34 Pantalla de selección de eventos
Elaboración propia

4.7.4 Selección de imagen para visualizar

En la figura 4.35 se observa la pantalla correspondiente a la selección de imagen que se desea visualizar en toda la pantalla, a partir de los íconos de cada una.

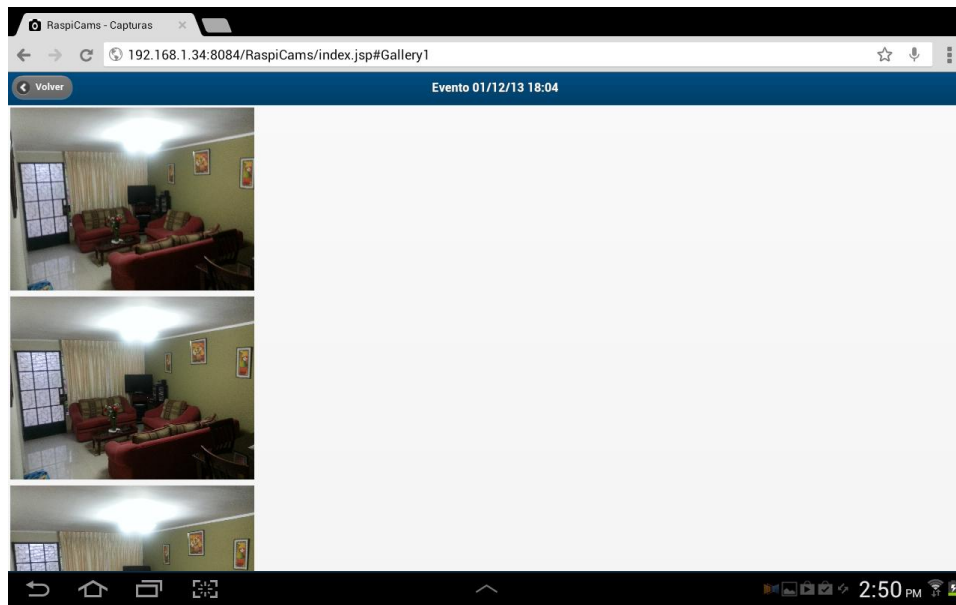


Figura 4.35 Pantalla de selección de imagen
Elaboración propia

Finalmente, luego de elegir la captura que se quiere visualizar, el usuario la puede ver en la pantalla de visualización de imagen, tal como se muestra en la figura 4.36.

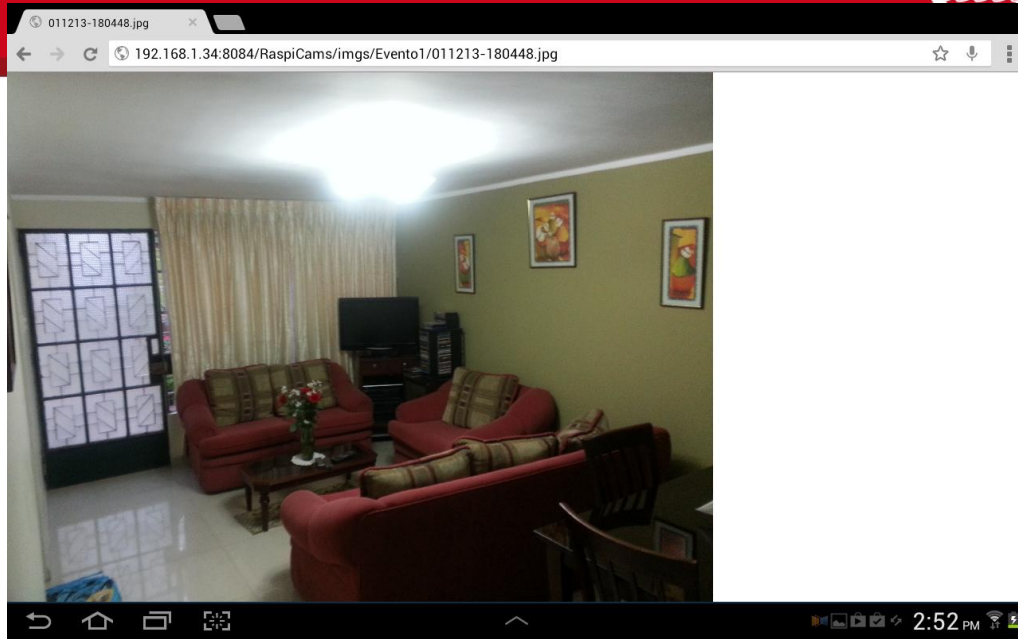


Figura 4.36 Pantalla de visualización
Elaboración propia

4.8 Pruebas de funcionamiento

Al tratarse de un sistema que incluye servicios de red como la descarga de capturas desde el micro controlador y la central de llamadas, es necesario analizar su consumo de ancho de banda para determinar su factibilidad en una red local.

Para ello, se usó el software iptraf, el cual permite monitorear el envío y recepción de paquetes por una determinada interfaz tal como se muestra en la figura 4.38.

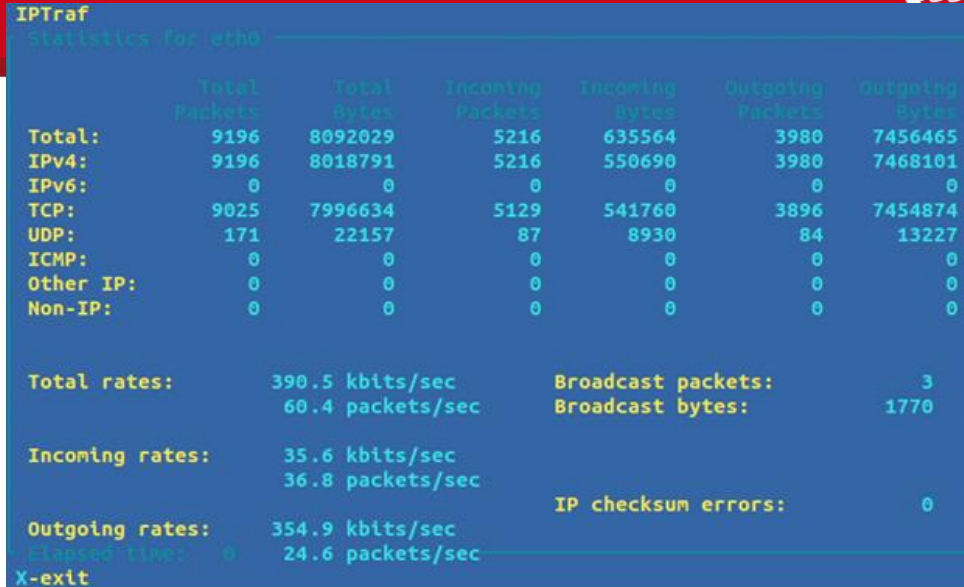


Figura 4.37 Pantalla de monitor de transmisión de datos IPTráf
Elaboración propia

En la figura anterior se puede apreciar que la información brindada por la aplicación se separa en Bytes y paquetes enviados y recibidos, y tasas de bits de entrada y salida de la interfaz seleccionada.

A partir de esta información se midieron cuatro veces consecutivas el tráfico generado por las acciones siguientes en el raspberry pi:

- Llamada a la central IVR para encendido y apagado de iluminación
- Llamada a la central IVR para realizar capturas con la cámara IP
- Captura de fotos y envío de alerta generados por detección de movimiento

Y el resultado fue el siguiente:

KiloBytes transmitidos (KB)	Medición 1	Medición 2	Medición 3	Medición 4	Promedio
Llamada	661.0	727.0	759.0	675.0	705.5
Llamada con fotos	943.0	1,034.0	1,070.0	1,036.0	1,020.8
deteccion de movimiento	453.0	379.0	401.0	462.0	423.8

Tabla 4.7 Tabla de tráfico generado por acción realizada
Elaboración propia

Ancho de banda (Kbps)	Medición 1	Medición 2	Medición 3	Medición 4	Promedio
Llamada	204.0	196.0	224.0	221.0	211.3
Llamada con fotos	226.0	245.0	209.0	230.0	227.5
deteccion de movimiento	160.0	144.0	199.0	152.0	163.8
Sistema en espera	46.1	46.7	45.7	46.3	46.2

Tabla 4.8 Tabla de ancho de banda por acción realizada
Elaboración propia

Este ancho de banda es posible en una red de área local, y permite la escalabilidad de la red pues se pueden usar más de dos circuitos de iluminación, dos cámaras IP y dos sensores de movimiento sin alterar el diseño ni generar una ocupación de ancho de banda que represente un cuello de botella para el sistema, y dado que la descarga de fotos se realiza internamente una conexión promedio a internet de 1 Mbps permite el tráfico que implicaría la salida de la llamada a la internet para el acceso al IVR y para el envío de alertas.



CAPÍTULO 5

EVALUACIÓN DE COSTOS DEL SISTEMA

Considerando los requerimientos iniciales del sistema de contar con una conexión a internet de 1 Gbps de velocidad, con una IP fija y que se cuenta con una puerta de enlace proporcionada por el operador (módem/router inalámbrico), entonces la inversión necesaria para implementar el sistema será la mostrada en la tabla 5.1.

	Cantidad	Precio Unitario	Precio Total
Raspberry Pi - Modelo B	1	\$ 61.40	\$ 61.40
Adaptador para alimentación Raspberry Pi	1	\$ 9.00	\$ 9.00
Tarjeta Delcomp Relay X4	1	\$ 14.50	\$ 14.50
Fuente de 12V para tarjeta Relay X4	1	\$ 9.00	\$ 9.00
Cámaras IP Dlink DCS-930L	2	\$ 59.00	\$ 118.00
Sensores PIR HC-SR501	2	\$ 5.90	\$ 11.80
Instalación y Cableado(3 Horas)	1	\$ 100.00	\$ 100.00
Costo de Desarrollo	1	\$ 100.00	\$ 100.00
Total			\$ 423.70
Pago mensual (Pago por el sistema de seguridad)			\$ 35.93

Tabla 5.1 Tabla de costos del sistema diseñado
Elaboración propia

Cuando se compara este sistema con la solución equivalente brindada por la marca Prosegur, la cual comprende la instalación de 3 sensores de movimiento y alarmas, 2 cámaras de vigilancia y acceso en tiempo real a la captura de las cámaras mediante una aplicación Web, se obtienen los siguientes datos:

PROSEGUR

Instalación 3 detectores de movimiento y alarmas	s/. 1200	\$ 433.21
Instalación 2 cámaras de vigilancia inalámbricas	s/. 780	\$ 281.59
Total – Instalación	s/. 1980	\$ 714.80
Pago mensual	s/. 130	\$ 46.93

Tabla 5.2 Costos del sistema de seguridad ofrecido por Prosegur
Fuente: Centro de atención al cliente - Prosegur

Cabe resaltar que la solución brindada por PROSEGUR incluye un sistema de baterías para que el sistema mantenga su funcionamiento en caso se de un corte de energía, a diferencia del diseño planteado en esta tesis, el cual no cuenta con ese respaldo.

Y se concluye que la diferencia en costo de instalación del sistema diseñado es significativa, pues es de 323 Dólares comparados con los 714 Dólares de la solución de Prosegur, además la solución brindada por Prosegur requiere un pago de 50 Dólares mensuales por concepto de operación del servicio, mientras la solución diseñada contempla 35 dólares por este mismo concepto.

A partir de la investigación realizada y el diseño siguiente se puede concluir lo siguiente:

- El diseño del sistema en que consiste esta tesis es factible en el ámbito de una red local, debido a que sus requerimientos de ancho de banda son muy bajos, además no requiere un servidor con grandes capacidades de procesamiento para su funcionamiento ni la instalación de muchos componentes complejos.
- El sistema, puede ser incrementado en tamaño y funcionalidades con relativa facilidad, debido a que ocupa una parte mínima de la capacidad de la red local, así como de la capacidad en la salida a internet de la vivienda, este incremento lo puede realizar la empresa que provee el servicio.
- Debido a las características de los componentes utilizados, el sistema no sería apto para su instalación en exteriores, sino únicamente dentro de la vivienda.
- Debido al reducido tamaño del micro controlador de dispositivos, que funciona también como servidor web y central telefónica, la instalación del sistema será sencilla y puede darse en cualquier lugar de la casa.
- Debido al bajo costo del sistema y a utilizarse hardware y software de libre distribución, se presenta como una muy buena propuesta de negocio.
- Gracias a las alertas instantáneas que genera el sistema puede ayudar al usuario a prevenir que se complete un robo.
- El control de los luces, así como la posibilidad de incluir otros aparatos eléctricos al sistema controlado de manera remota, y así brindar una facilidad a las personas con poca movilidad.
- Todo el procesamiento y almacenamiento ocurre dentro de la casa, lo cual brinda facilidad de acceso para el usuario, pero puede ser una vulnerabilidad.

RECOMENDACIONES

- Complementar el uso del sistema en la vivienda de parte del usuario con el cumplimiento de sugerencias básicas para asegurar su patrimonio como el cerciorarse de cerrar las puertas y ventanas al salir de la casa, o no abrir la puerta a extraños, si bien se puede usar herramientas tecnológicas, el ser precavido y seguir este tipo de pautas puede ser incluso más efectivo para proteger la vivienda.
- Ubicar el Raspberry Pi, que es el componente que centraliza las órdenes y acciones, en un lugar seguro de la casa y de difícil acceso para los intrusos de manera que se pueda garantizar su funcionamiento en caso de intrusiones.
- Analizar de manera cuidadosa la ubicación donde se instalarán los sensores y segmentos, si bien en este diseño se plantean puntos críticos de la seguridad de la vivienda, los puntos críticos para la vigilancia puede variar de acuerdo a la utilidad que se le quiera dar al sistema y la vivienda en que se instale.
- Trabajar en colaboración con las autoridades para mejorar el diseño del sistema utilizando alertas que se envíen directamente a ellas para así optimizar la respuesta ante una intrusión.

- [1] Nelly Pillhuamán Caña , Julio Ramos Ramírez and Guillermo Vallenas
2010 Percepción sobre inseguridad ciudadana
Investigaciones sociales. 14.24 (Ene. 2010): p351.
- [2] Ministerio del Interior
2005-2012 Estadísticas sobre seguridad ciudadana.
Consulta: 07 de Agosto de 2013.
<http://www.inei.gob.pe/estadisticas/indice-tematico/seguridad-ciudadana/>
- [3] Real Academia Española
Consulta: 10 de Agosto de 2013.
<http://rae.es/recursos/diccionarios/drae>
- [4] Merriam-Webster Dictionary
Consulta: 10 de Agosto de 2013.
<http://www.merriam-webster.com/dictionary>
- [5] Domótica Viva
2003 Bricolaje X-10 Curso de Domótica a través de la red eléctrica
Consulta: 11 de Agosto de 2013
<http://www.domoticaviva.com/X-10.htm>
- [6] Wilson, Mark S.
2011 A checklist for jumping from analog to IP surveillance.(Security)
Cabling Installation & Maintenance, Nov, 2011, Vol.19(11), p.11(3)
- [7] Analog vs. IP technologies
Consulta: 15 de Agosto de 2013.
<http://www.discount-security-cameras.net/analog-vs-ip-technology.aspx>
- [8] IP PBX
Consulta: 15 de Agosto de 2013.
<http://www.asterisk.org/get-started/applications/pbx>
- [9] Raspberry Pi
Consulta: 16 de Agosto de 2013.
<http://www.raspberry-pi.org>

- [10] Dispositivos electrónicos de potencia: Relés
Consulta: 14 de Agosto de 2013.
http://www.ugr.es/amroldan/enlaces/dispo_potencia/reles
- [11] Dispositivos electrónicos de potencia: Relés
Consulta: 14 de Agosto de 2013.
<http://www.cyberhades.com/wp-content/uploads/2008/12/rele.jpg>
- [12] Web-Based Application Development
GROVE, Ralph F
Jones & Bartlett Publishers 2009
- [13] jQuery mobile
Consulta: 20 de Agosto de 2013
<http://jquerymobile.com/>
- [14] Databases Organizing Information
ROZA, Greg.
Rosen Central 2010
- [15] Diggita.com
Consulta: 20 de Agosto de 2013
http://diggita.com/2009_12_01_archive.html
- [16] Cubieboard: A series of ARM miniPCs
Consulta: 27 de Agosto de 2013
<http://cubieboard.org>
- [17] Arduino Due
Consulta: 27 de Agosto de 2013
<http://arduino.cc/en/Main/arduinoBoardDue>
- [18] ARM architecture: Documentation
Consulta: 27 de Agosto de 2013
<http://infocenter.arm.com/help/index.jsp>
- [19] Telware Perú
Consulta: 27 de Agosto de 2013
<http://www.telwareperu.com/tienda.php>

- [20] Micropicplus - Sensor PIR HC-SR501
Consulta: 27 de Agosto de 2013
<http://www.micropicplus.com/shop/sensores/72-pir-sensor-de-movimiento.html>
- [21] Manual de tarjeta RelayX4
DELCOMP
- [22] Adefruit Learning System - The GPIO connector
Consulta: 27 de Agosto de 2013
<http://learn.adafruit.com/adafruits-raspberry-pi-lesson-4-gpio-setup/the-gpio-connector>
- [23] Order a slice of Pi.(Product/service evaluation)
Halfacree, Gareth
Computer Act!ve, Sept 27, 2012
- [24] **Why Java** is hot: Sun's radically new programming tool could shift the balance of power in computers.(Sun Microsystems)
Elmer - Dewitt, Philip
Time, Jan 22, 1996, Vol.147(4), p.58(3)
- [25] Asterisk-Java
Consulta: 27 de Agosto de 2013
<http://www.asterisk-java.org/development/>
- [26] Asterisk-Java
Consulta: 27 de Agosto de 2013
<http://www.pi4j.com/>
- [27] Apache Commons Net Library
Consulta: 27 de Agosto de 2013
<http://commons.apache.org/proper/commons-net/>
- [28] Zoiper cross-platform softphone
Consulta: 27 de Agosto de 2013
<http://www.zoiper.com/en>