Trabajo Fin de Grado Grado en Ingeniería de las Tecnologías de Telecomunicación

Aplicaciones orientadas a la domótica con Raspberry Pi

Autor: Claudio González Domínguez Tutor: Bernardo Palomo Vázquez

> Departamento de Ingeniería Electrónica Escuela Técnica Superior de Ingeniería Universidad de Sevilla

> > Sevilla, 2015



Trabajo Fin de Grado Grado en Ingeniería de las Tecnologías de Telecomunicación

Aplicaciones orientadas a la domótica con Raspberry Pi

Autor: Claudio González Domínguez

Tutor: Bernardo Palomo Vázquez

Departamento de Ingeniería Electrónica Escuela Técnica Superior de Ingeniería Universidad de Sevilla Sevilla, 2015

Trabajo Fin de Grado: Aplicaciones orientadas a la domótica con Raspberry Pi

Autor: Claudio González Domínguez

Tutor: Bernardo Palomo Vázquez

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2015

El Secretario del Tribunal

A mi familia A mis maestros

Agradecimientos

En primer lugar, quiero agradecer a mi tutor Bernardo Palomo Vázquez, por darme la oportunidad de realizar este trabajo fin de grado y por la ayuda y dedicación durante el desarrollo del mismo.

También, dar las gracias a todos mis profesores y compañeros durante todos los años de estudio por ayudarme a adquirir los conocimientos para poder realizar este proyecto.

Finalmente, agradecer a mi familia y amigos por apoyarme y animarme durante estos meses de dedicación continua a este trabajo.

Claudio González Domínguez Sevilla, 2015

Este trabajo final de grado consiste en diseñar y construir una aplicación orientada a la domótica con la placa Raspberry Pi. En concreto un sistema de seguridad, el cual es posible administrar remotamente y que funcionará tanto manual como automáticamente. Se explicará en que consiste la Raspberry Pi y cual es su estructura. Se verán los sistemas operativos más usados.

En cuanto a la aplicación, se explicarán cuales han sido los componentes elegidos para desarrollar el trabajo y los motivos de esa elección, el montaje y configuración de estos y el funcionamiento del sistema completo.

Se mostrará la interfaz de usuario implementada para la aplicación y se explicará como usarla así como los pasos necesarios para programarla.

En definitiva se verá cómo se puede construir con una pequeña placa un sistema de seguridad que detecte la presencia de intrusos y cómo poder vigilar una habitación desde cualquier lugar con conexión a internet.

Además se estudiarán varias posibilidades para mejorar la aplicación en general.

This final degree project consists of designing and building an application oriented to home automation with Raspberry Pi board. In particular a security system, which can remotely manage and operate both manually and automatically. It will explain that the Raspberry Pi consists and what is its structure. The most used operating systems will be seen.

As to the application, this document will explain which components have been selected to develop the work and the reasons for that choice, installation and configuration of these and operation of the entire system.

It will be shown the user interface for the application implemented and shall be explained how to use it and the steps required to program.

Ultimately it will be seen how to build a security system that detects intruders with a small board and how to monitor a room from anywhere with internet access.

In addition, several possibilities will be explored to improve the application in general.

-Translation by google-

Agradecimientos	іх
Resumen	xi
Abstract	xiii
Índice	xv
Índice de Tablas	xvii
Índice de Figuras	xix
Notación	ххі
1. Introducción	1
1.1. Objetivos Del Provecto	2
1.2. Motivaciones	3
2. Raspberry Pi	5
2.1. Hardware	6
2.1.1. Puertos USB	8
2.1.2. Pines GPIO	9
2.1.3. Conector DSI	9
2.1.4. Conector CSI	9
2.1.5. Almacenamiento	10
2.1.6. Puerto Ethernet	10
2.1.7. Alimentación	10
2.1.8. Salidas de audio	10
2.1.9. Conector HDMI	10
2.1.10. Broadcom BCM2835	10
2.2. Software	11
2.2.1. Instalación del Sistema Operativo	12
3. Diseño e implementación de la aplicación	13
3.1. Componentes	13
3.1.1. Raspberry PI. Modelo B+	13
3.1.2. Camara	14
$\begin{array}{cccc} 3.1.5. & \text{IVICIO SelvO SOBO} \\ 2.1.4 & \text{Sensor DIP} \end{array}$	15
3.1.4. Sensor Fin	10
3.1.6 Adaptador LISB Wifi	18
3.2. Montaie	19
3.2.1. Diagrama de conexiones	21
3.3. Entorno Web	22
3.4. Configuración	25
3.4.1. Raspberry Pi Camera	25
3.4.2. Adaptador Wifi	27
3.4.3. Servidor PHP	27
3.4.4. Servicio NO-IP	32

3.4.5.	Motion	33
3.4.6.	Yowsup	33
3.5. Fu	ncionamiento	34
3.5.1.	Servidor PHP + Base de datos	34
3.5.2.	PHP + Python	36
4. Conclus	siones	45
4. Conclus 4.1. Po	siones sibles mejoras futuras	45
4. Conclus 4.1. Po. Referencias	siones sibles mejoras futuras	45 45 49

ÍNDICE DE TABLAS

Tabla 2–1 Modelos Raspberry Pi	7
Tabla 2–2 Elementos Raspberry Pi. Modelo B+	8
Tabla 3–1 Características Raspberry Pi. Modelo B+	13
Tabla 3–2 Características Raspberry Pi Camera	14
Tabla 3–3 Cracterísticas Micro Servo SG90	15
Tabla 3–4 Características sensor PIR KC7783	16
Tabla 3–5 Especificaciones adaptador Wifi TL-WN725N	18

ÍNDICE DE FIGURAS

Figura 1-1 Diagrama domótica	2
Figura 2-1 Logo Raspberry Pi	5
Figura 2-2 Primer prototipo Raspberry Pi	6
Figura 2-3 Elementos Raspberry Pi. Modelo B+	7
Figura 2-4 Diagrama LAN9514	8
Figura 2-5 Pines GPIO	9
Figura 2-6 Captura programa Win32 Disk Imager	12
Figura 3-1 Raspberry Pi. Modelo B+	14
Figura 3-2 Raspberry Pi Camera	15
Figura 3-3 Micro Servo SG90	16
Figura 3-4 Sensor PIR KC 7783	17
Figura 3-5 Módulo 4 relés	17
Figura 3-6 Adaptador USB Wifi TL-WN725N	18
Figura 3-7 Caja de registro	19
Figura 3-8 Orificios sujeción caja de registro	19
Figura 3-9 Vista anterior sistema final	20
Figura 3-10 Vista superior sistema final	20
Figura 3-11 Diagrama de conexiones	21
Figura 3-12 Conexión Raspberry Pi Camera	22
Figura 3-13 Página de inicio	22
Figura 3-14 Modo de operación	23
Figura 3-15 Página aviso activación	23
Figura 3-16 Página de streaming	24
Figura 3-17 Página de control	24
Figura 3-18 Conexión cámara a Raspberry Pi	25
Figura 3-19 Menú de configuración Raspberry Pi	26
Figura 3-20 Submenú de configuración Raspberry Pi Camera	26
Figura 3-21 Fichero /etc/php5/apache2/php.ini	28
Figura 3-22 Interfaz phpMyAdmin	29
Figura 3-23 Interfaz phpMyAdmin (Crear base de datos)	30
Figura 3-24 Interfaz phpMyAdmin (Crear tabla)	30
Figura 3-25 Interfaz phpMyAdmin (Campos tabla)	31
Figura 3-26 Interfaz phpMyAdmin (Insertar usuario)	31
Figura 3-27 Fichero /etc/init.d/noip2	32

Figura 3-28 Funcionamiento servidor web3	5
Figura 4-1 Batería recargable4	6
Figura 4-2 Conexiones dispositivos a módulo de relés4	6
Figura 4-3 Zumbador 12V 4	7
Figura 4-4 Raspberry Pi NoIR Camera4	7
Figura 4-5 Imagen tomada Pi NoIR Camera4	8
Figura 4-6 Imagen tomada Raspberry Pi Camera4	8

Notación

TFG	Trabajo Fin de Grado	
HTML	HyperText Markup Language (Lenguaje de marcas de hipertexto)	
PHP	PHP Hypertext Pre-processor	
GPIO	General Purpose Input/Output (Entrada/Salida de Propósito General)	
UART	Universal Asynchronous Receiver-Transmitter (Transmisor-Receptor Asíncrono Universal)	
I2C	Bus de comunicaciones en serie. Inter-Integrated Circuit (Inter-Circuitos Integrados)	
SPI	Serial Peripheral Interface (Bus de Interfaz de Periféricos Serie)	
WIFI	WI-FI. Nombre comercial comunicación inalámbrica.	
USB	Universal Serial Bus	
PIR	Passive Infrared (Sensor Infrarrojo Pasivo)	
HDMI	High-Definition Multimedia Interface (Interfaz multimeda de alta definición)	
NoIR	No Infrared (No emplea filtro infrarrojo)	
SBC	Single Board Computer (Ordenador de placa reducida)	

E ste trabajo fin de grado trata como su propio nombre indica, "Aplicaciones orientadas a la domótica con Raspberry Pi", del diseño y programación de una aplicación orientada a la domótica usando para ello un mini ordenador.

En este proyecto, por un lado, se tiene la Raspberry Pi que es una pequeña placa de tamaño similar al de una tarjeta de crédito a la que se puede conectar ratón, teclado y pantalla y tener multitud de usos como ordenador convencional dado que dispone hasta de conexión a internet. Además de esto es capaz de realizar muchas otras funciones programándolas.

El dispositivo es un proyecto orientado a niños para que estos se inicien en la programación y comprendan cómo funcionan los ordenadores, aunque personas de todas las edades podrían adentrarse en el mundo de la programación y aprender a programar en lenguajes como Python o Scratch.

Por otro lado, en el título del TFG aparece la palabra domótica, que se puede definir como el conjunto de las tecnologías aplicadas al control y la automatización inteligente de la vivienda, que permite una gestión eficiente del uso de la energía, que aporta seguridad y confort, además de comunicación entre el usuario y el sistema.

Un sistema que hace uso de la domótica es aquel que toma información a través de unos sensores, la procesa y es capaz de tomar decisiones por sí solo.

Uno de los usos más expandidos es el de usar la Raspberry Pi para realizar programas orientados al ocio como crear juegos o centros multimedia. También es posible programar un servidor web o tener un sistema "Ambilight", que consiste en un sistema de retroiluminación para los televisores que permite una mejor experiencia visual.

Hay que tener en cuenta que todo esto se puede conseguir con un dispositivo tanto de bajo coste, alrededor de 30 euros, como de reducido tamaño. Todo esto junto con su gran potencial y versatilidad a la hora de programar hacen que la Raspberry Pi tenga gran popularidad.

Otro de los usos y en el que se centra este trabajo es en el uso como sistema domótico, existen proyectos como control de puertas de garajes, calderas u otras aparatos electrónicos de la casa, implementación de estaciones meteorológicas y sistemas de seguridad.

En estos usos orientados a la domótica la Raspberry Pi hace la función de procesar la información obtenida a través de unos sensores que les suministran una serie de señales. Es decir, funciona como si fuera el cerebro del sistema tomando las decisiones oportunas.



Figura 1-1 Diagrama domótica

Este último es el uso que se le va a dar a la Raspberry Pi en este TFG, conjuntamente con el de servidor web. Consiste de un sistema de vigilancia inalámbrico que permite la detección de intrusos así como la comunicación entre usuario y sistema mediante la interfaz web alojada en el servidor, desde cualquier lugar donde exista una conexión a internet. Se opta por la opción de la interfaz web para facilitar el uso al usuario y tener acceso desde cualquier dispositivo con navegador web.

1.1. Objetivos Del Proyecto

El propósito de este proyecto consiste en diseñar e implementar un sistema de vigilancia inalámbrico mediante una Raspberry Pi modelo B+, aprender a configurar el dispositivo y su funcionamiento. Además de ver el potencial del que dispone este dispositivo, así como la cantidad de posibles usos que se le pueden dar.

Otro de los objetivos de este trabajo es el de aprender un nuevo lenguaje de programación como es Python y hacer uso de otros ya aprendidos como Script.

El resultado esperado es el de tener un sistema de seguridad autónomo capaz de detectar cualquier anomalía y avisar a los usuarios, así como tener una interfaz manejable para cualquier tipo de usuario con la que ver el estado de una habitación o poder controlar tanto la cámara como otros dispositivos electrónicos existentes.

Como se ha dicho anteriormente hay dos partes que se pueden separar, una de ellas es la parte interfaz/usuario en la que se pretende hacer una página web aplicando los conocimientos en programación web HTML y PHP adquiridos en la titulación Grado en Ingeniería de las Tecnologías de Telecomunicación.

Por otro lado está la parte informática/electrónica en la que se habrá de conectar los distintos sensores y

actuadores y donde se incluye la programación en Python para la lectura, procesado y escritura de los datos de los sensores.

1.2. Motivaciones

Se escogió la realización de este trabajo fin de grado por diferentes motivos.

El primero de ellos es el trabajar con la Raspberry Pi, ya que es un dispositivo que últimamente se está usando mucho por la mayoría de personas que les gustan la tecnología y la programación para realizar multitud de diferentes proyectos.

Otro de los motivos es que no solo hay que aplicar los conocimientos adquiridos durante la carrera sino que también te da la oportunidad de seguir adquiriendo nuevos conocimientos como es aprender un nuevo lenguaje de programación.

Por último, este trabajo te da la oportunidad de iniciarte en la domótica que es una tecnología que en un futuro muy próximo es posible que todo el mundo tenga presente en sus viviendas ya que permite un gran ahorro energético y dan mucha comodidad a los usuarios.

2. RASPBERRY PI

R aspberry Pi es un ordenador de placa reducida (SBC) de bajo coste, que se podría considerar como un ordenador de muy pequeño tamaño, comparable con el de una tarjeta de crédito, desarrollado en Reino Unido por la fundación Raspberry Pi, con el objetivo principal de incitar tanto a niños en sus colegios como a adultos a que aprendan sobre ordenadores y todo lo relacionado con ellos.



Figura 2-1 Logo Raspberry Pi

La idea es tener una placa a la que poder conectar monitor, ratón y teclado y ayudar a personas de todas las edades a adentrarse en el mundo de la computación y la programación.

La idea de desarrollar algo así surgió en 2006 cuando Eben Upton, Rob Mullins, Jack Lang and Alan Mycroft del laboratorio de informática de la Universdad de Cambridge empezaron a ver como había cambiado los conocimientos de los niños sobre la informática. En la década de 1990 la mayoría de los niños tenían mucha experiencia como programadores aficionados, en cambio en la década del 2000 solo eran capaces de realizar diseño web.

Los primeros diseños de Raspberry Pi se basaban en el microcontrolador Atmel ATmega644. En la siguiente figura se puede ver el prototipo basado en ese microcontrolador.



Figura 2-2 Primer prototipo Raspberry Pi

En mayo de 2009, la Fundación Raspberry Pi fue fundada en Caldecote, South Cambridgeshire, Reino Unido como una asociación caritativa que es regulada por la Comisión de Caridad de Inglaterra y Gales.

La fundación Raspberry Pi surge con un objetivo en mente: Desarrollar el uso y entendimiento de los ordenadores en los niños. La idea es conseguir ordenadores portables y muy baratos que permitan a los niños usarlos sin miedo, abriendo su mentalidad y educándolos en la ética del "ábrelo y mira cómo funciona".

En agosto de 2011, se fabricaron cincuenta placas Alpha, que tenían las mismas características que el modelo B, pero eran un poco más grandes para integrar bien unas interfaces para depuración. Las primeras ventas del modelo B tuvieron lugar el 29 de febrero de 2012.

El 4 de febrero de 2013, se lanzó el modelo A, que venía con solo 256Mb de RAM y sin puerto ethernet a un precio más asequible que el modelo B.

2.1. Hardware

Hoy en día la Raspberry Pi presenta tres modelos, los modelos A+ y B+, que están basados en sus predecesores A y B, y la nueva Raspberry Pi 2 modelo B. Las principales diferencias entre los modelos antiguos A y B y sus versiones más recientes A+ y B+ respectivamente están en el almacenamiento, pasando de ser SD a microSD y en el caso del modelo A una reducción de tamaño y por tanto de peso. Otra diferencia del modelo B es el aumento de 2 a 4 puertos USB. En la siguiente tabla se puede observar la diferencia entre las tres versiones que se comercializan actualmente.

Características	Modelo A+	Modelo B+	RPI 2 Modelo B
SoC	Broadcom BCM2835	Broadcom BCM2835	Broadcom BCM2836
CPU	ARM11 ARMv6	ARM11 ARMv6	ARM11 ARMv7 ARM
	700MHz	700MHz	Cortex-A7 4 núcleos (a) 900MHz
GPU	Broadcom VideoCore IV 250MHz OpenGL ES 2.0		
RAM	256 MB LPDDR	512 MB LPDDR	1 GB LPDDR SDRAM 450
	SDRAM 400 MHZ	SDRAM 400 MHZ	MHZ
USB 2.0	1	4	4
Salidas de vídeo	HDMI 1.4 @ 1920x1200 píxeles		
Almacenamiento		microSD	
Ethernet	No	Sí, 10/100 Mbps	Sí, 10/100 Mbps
Tamaño	85,60x56,5 mm	85,60x56,5 mm	85,60x56,5 mm
Peso	23g	45g	45g

Tabla 2-1 Modelos Raspberry Pi

Dado que las demás características son similares y el modelo usado en este trabajo ha sido el modelo B+, a continuación se va a pasar a detallar los componentes existentes en esta versión.



Figura 2-3 Elementos Raspberry Pi. Modelo B+

N°	Identificador
1	Salida puertos USB 2.0
2	Pines GPIO
3	Conector DSI Display Module
4	Conector CSI Camera Module
5	Ranura para la tarjeta microSD
6	Puerto Ethernet (RJ45)
7	Alimentación micro USB
8	Leds indicadores
9	Salida de audio
10	Salida HDMI

Tabla 2-2 Elementos Raspberry Pi. Modelo B+

2.1.1. Puertos USB

El modelo B+ posee cuatro puertos USB del estándar 2.0 gesstionados por el microchip LAN9514. Este chip esta especialmente diseñado para casos como el de Raspberry Pi en el que se tiene que integrar puertos en una pequeña placa. El hecho de que en este modelo se haya aumentado la cantidad de puertos permite conectar a la vez ratón, teclado y conexión WIFI USB.



Figura 2-4 Diagrama LAN9514

2.1.2. Pines GPIO

Pin#	NAME		NAME	Pin#
01	3.3v DC Power		DC Power 5v	02
03	GPIO02 (SDA1, I2C)	\odot	DC Power 5v	04
05	GPIO03 (SCL1, I2C)	\odot	Ground	06
07	GPIO04 (GPIO_GCLK)		(TXD0) GPIO14	08
09	Ground	00	(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)	$\mathbf{O}\mathbf{O}$	(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)	\mathbf{O}	Ground	14
15	GPIO22 (GPIO_GEN3)	$\mathbf{O}\mathbf{O}$	(GPIO_GEN4) GPIO23	16
17	3.3v DC Power		(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)	\odot	Ground	20
21	GPIO09 (SPI_MISO)	\odot	(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)	\odot	(SPI_CE0_N) GPIO08	24
25	Ground		(SPI_CE1_N) GPIO07	26
27	ID_SD (I2C ID EEPROM)	\odot	(I2C ID EEPROM) ID_SC	28
29	GPIO05	00	Ground	30
31	GPIO06	$\mathbf{O}\mathbf{O}$	GPIO12	32
33	GPIO13	00	Ground	34
35	GPIO19	00	GPIO16	36
37	GPIO26	00	GPIO20	38
39	Ground	00	GPIO21	40

Raspberry Pi B+ J8 Header

Figura 2-5 Pines GPIO

Como se puede observar en la Figura 2-5 la Raspberry Pi B+ cuenta con dos filas de 20 pines cada una. Estos pines sirven de interfaz entre la Raspberry Pi y el exterior. GPIO (General Purpose Input/Output) es un sistema de Entrada/Salida de propósito general que te permite dar un uso más amplio y realizar multitud de proyectos.

Con estos pines se pueden controlar multitud de dispositivos como luces leds, motores o relés.

De los 40 pines que contiene la Raspberry Pi, 12 de ellos son de alimentación (5V o 3.3V) o tierra y después hay tanto pines GPIO normales, que puedes programar para tus proyectos, como pines GPIO especiales para usar como puerto UART, I2C o SPI.

2.1.3. Conector DSI

Puerto DSI (Display Serial Interface), que permite la conexión de pequeñas pantallas LCD directamente a la GPU del dispositivo.

2.1.4. Conector CSI

Consiste en un conector tipo bus de 15 pines utilizado para añadir un dispositivo compatible con la interfaz CSI-2 (o Camera Serial Interface versión 2). Mediante este conector se puede conectar la cámara de la Raspberry Pi.

2.1.5. Almacenamiento

La Raspberry Pi B+ no dispone de un disco duro, para ello trae un lector/ranura para memorias microSD, un sistema de almacenamiento en estado sólido. El arranque del sistema se hará desde la propia tarjeta microSD, con lo que debido a que tiene que albergar todo el sistema operativo, es necesario que la tarjeta sea de al menos 2 GB de capacidad para almacenar todos los archivos requeridos.

2.1.6. Puerto Ethernet

Se dispone de un conector RJ-45 conectado al integrado LAN9514 de SMSC, nombrado en el apartado 2.1.1, que proporciona conectividad a 10/100 Mbps.

Es posible conectar la raspberry directamente a un PC sin pasar por un router conectando ambos equipos de manera directa con un cable RJ45.

Los modelos actuales de la Raspberry Pi no cuentan con con un componente para poder acceder a redes inalámbricas, pero es posible añadir soporte Wi-Fi a la Raspberry utilizando un adaptador USB para red inalámbrica.

2.1.7. Alimentación

La placa carece de botones de encendido y apagado. Para su alimentación dispone de un conector micro USB tipo B que proporciona 5 V de tensión. La mayoría de los cargadores para smartphones (que suministren más de 700 mA) son compatibles para dar tensión a la Raspberry Pi.

2.1.8. Salidas de audio

Para la salida de audio posee un conector de audio Jack de 3,5mm, además del propio HDMI. Si se está usando el puerto HDMI de la Raspberry Pi, obtener el audio es sencillo: cuando está configurado apropiadamente, el puerto HDMI transporta ambas señales, la de video y la de audio. Esto significa que conectando un único cable a la pantalla es suficiente para sacar video y audio. Si el display no tiene entrada HDMI se tendría que utilizar la salida de audio Jack.

2.1.9. Conector HDMI

Permite la conexión de un dispositivo compatible con la interfaz HDMI 1.3 y 1.4 para la extracción de vídeo y audio.

2.1.10. Broadcom BCM2835

El diseño incluye un System-on-a-chip Broadcom BCM2835, que contiene un procesador central (CPU) ARM1176JZF-S a 700 MHz, un procesador gráfico (GPU) VideoCore IV y 512 MB de memoria RAM todo en un mismo integrado.

2.2. Software

La Raspberry Pi está diseñada para ejecutar el sistema operativo GNU/Linux de código abierto. Varias versiones de Linux (conocidas como distribuciones) que soportan la Raspberry Pi son:

- Raspbian OS es la distribución por excelencia para la Raspberry Pi. Es la más completa y optimizada de las existentes, por eso cuenta con apoyo oficial. Raspbian OS se basa en la potente distro Debian Wheezy (Debian 7.0) optimizando el código de ésta para la Raspberry Pi. La distribución permite moverse ágilmente en el hardware de la Raspberry Pi, con un entrono de escritorio LXDE y Midori como navegador web predeterminado. Además incluye herramientas de desarrollo muy interesantes, como IDLE para Python, Scratch para programar videojuegos.
- **RISC OS** es uno de los pocos sistemas operativos no basados en Linux que existen para la Raspberry Pi. De hecho, RISC OS es un sistema operativo británico desarrollado por Acorn Computers (los creadores de ARM) y que se distribuye bajo licencia Open-Source. Aunque su soporte y catálogo de aplicaciones disponibles no sea tan amplio como el de otras distribuciones, también está considerado como una de los sistemas operativos oficiales de la Raspberry Pi y es especialmente interesante en cuanto a que se ha creado en torno a la plataforma ARM desde cero.
- Arch Linux es otro de los grandes nombres en cuanto a distribuciones Linux. Se caracteriza por su simplicidad, elegancia, coherencia del código y minimalismo. Pero la simplicidad no quiere decir facilidad de uso, ya que Arch Linux es bastante conocida por ser poco amigable y recomendable solo para gente con conocimientos más elevados. Ahora Arch Linux soporta ARM y por tanto también puede instalarse en la Raspberry Pi.
- **Pidora** es básicamente una distribución Linux Fedora especialmente optimizada para funcionar en ARM. Por el resto de características es similar a Fedora, la hermana pequeña de Red Hat, y mantenida por los mismos desarrolladores de esta comunidad libre.
- OpenELEC es otro de los sistemas operativos oficiales de la Raspberry Pi y por tanto se incluye en NOOBS. Se trata de una distribución Linux especialmente pensada para crear un centro multimedia barato con la Raspberry Pi. Con él se puede disponer de todo el contenido multimedia y acceso a Internet para transformar una TV en una smartTV.

Para ello, OpenELEC incluye paquetes de codecs de audio y vídeo, drivers, y se basa en el famoso Kodi (anteriormente conocido como XBMC, siglas de Xbox Media Center). Kodi es un centro multimedia que fue creado en un inicio para la videoconsola Xbox, pero el desarrollo hizo que se portara a otras plataformas. Se completa con reproductores de audio, vídeo, presentación de diapositivas, visores de imágenes, reportes de clima, y otras funciones implementadas mediante plug-ins.

2.2.1. Instalación del Sistema Operativo

Dado que se va a trabajar con Python el sistema operativo que se va a instalar en este caso es Raspbian.

Para poder instalar la distribución elegida es necesario usar un ordenador con lector de tarjetas SD. Usando un ordenador con Windows, el primer paso que hay que realizar es obtener el sistema operativo desde el siguiente enlace:

https://www.raspberrypi.org/downloads/

Posteriormente, se necesita descargar el software gratuito "Win32 Disk Imager". En Image File se escoge el fichero que contenga la imagen del Sistema Operativo, en Device hay que asegurarse que se elige la unidad correcta asociada a la tarjeta SD y mediante el botón Write se pasa el sistema a la tarjeta.

😼 Win32 Disk Imager	
Image File	Device
MD5 Hash:	
Progress	
Cancel Read	Write Exit
Waiting for a task.	

Figura 2-6 Captura programa Win32 Disk Imager

Una vez finalizado este proceso, ya se puede insertar la tarjeta en Raspberry Pi y comenzar a trabajar.

3. DISEÑO E IMPLEMENTACIÓN DE LA APLICACIÓN

Se opta por desarrollar una aplicación llamada "Security Pi" centrada en la seguridad y control de una vivienda. Como punto de partida a la hora de diseñar la aplicación de seguridad con la Raspberry Pi se empezó por la elección de los componentes que satisficieran las necesidades del TFG. Se ha intentado ahorrar lo máximo reutilizando componentes que existían en el Departamento de Ingeniería Electrónica y adquiriendo algunos de bajo coste. Posteriormente se llevó a cabo la configuración de la Raspberry Pi así como el desarrollo de los códigos para el control de todos los dispositivos conectados a ésta. Por último para facilitar al usuario la interacción y configuración del sistema se implementó un entorno web.

3.1. Componentes

Los siguientes componentes son los elegidos para el montaje del sistema así como para la construcción del soporte para todos los dispositivos.

3.1.1. Raspberry Pi. Modelo B+

Características	Modelo B+
SoC	Broadcom BCM2835
CPU	ARM11 ARMv6 700MHz
GPU	Broadcom VideoCore IV 250MHz OpenGL ES 2.0
RAM	512 MB LPDDR SDRAM 400 MHz
USB 2.0	4
Salidas de vídeo	HDMI 1.4 @ 1920x1200 píxeles
Almacenamiento	microSD
Ethernet	Sí, 10/100 Mbps
Tamaño	85,60x56,5 mm
Peso	45g.

Tabla 3-1 Características Raspberry Pi. Modelo B+



Figura 3-1 Raspberry Pi. Modelo B+

Se elige la Raspberry Pi para desarrollar el sistema porque era uno de los requisitos de este TFG. A la hora de elegir modelo se escoge el modelo B+ ya que era el último existente en el mercado en ese momento y del que disponía el Departamento de Ingeniería Electrónica.

3.1.2. Cámara

La cámara usada para este TFG es la Raspberry Pi Camera, que aunque no da las mismas prestaciones que algunas cámaras que podemos conectar vía USB, al ser la oficial de Raspberry Pi existen librerías para su control y no es necesario descargar nuevos programas para su instalación. También permite realizar multitud de cambios como color, calidad o brillo. En la Tabla 3-2 podemos ver las principales características.

Características	Valor
Dimensiones	25 x 20 x 9 mm
Peso	3g
Resolución	5 Megapíxeles
Modos vídeo	1080p30, 720p60, VGA90
Resolución imagen	2592 x 1944 pixel

Tabla 3–2 Características Raspberry Pi Camera


Figura 3-2 Raspberry Pi Camera

3.1.3. Micro Servo SG90

La opción de usar un servo en el sistema se plantea cuando es necesario dar movilidad a la cámara, vista en el apartado 1.1.2, para poder tener un ángulo de visión mucho mayor. Dado que la cámara va a estar instalada en una pared solo necesitamos un movimiento de 180 grados para tomar imágenes panorámicas. Por ese motivo se escoge el micro servo de SG90 de Tower Pro que da la funcionalidad requerida.

Características	Valor
Dimensiones	22 x 11.5 x 27 mm
Peso	9g
Torque	1.2 kg/cm (4.8V)
Voltaje de operación	4.0 a 7.2 V
Velocidad de giro	0.12 seg / 60°
Ángulo de giro	180°

Tabla 3–3 Cracterísticas Micro Servo SG90



Figura 3-3 Micro Servo SG90

3.1.4. Sensor PIR

Para solucionar el problema de detectar cualquier intruso se opta por incorporar al sistema un sensor infrarrojo pasivo (sensor PIR) que mide la luz infrarroja radiada por los objetos situados en su campo de visión, es decir, detecta cualquier objeto que emita calor. El sensor que se usa es el KC7783 facilitado por el departamento.

Características	Valor
Dimensiones	25 x 35 mm
Ancho pulso salida	0.5 seg min
Corriente de operación	400µA (5V)
Voltaje de operación	4.0 a 12 V
Ganancia entrada	68dB

Tabla 3-4 Características sensor PIR KC7783



Figura 3-4 Sensor PIR KC 7783

3.1.5. Relés

La tensión de salida de la Raspberry Pi es de 5 voltios, pero podemos conectar luces y otros elementos de mayor voltaje usando un relé para hacer de intermediario entre ellos. Se opta por un módulo de 4 relés de Arduino compatible con la Raspberry Pi dado que funciona con alimentación de 5V. Este dispositivo lo que hace es cerrar o abrir el circuito recibiendo una señal desde el controlador.



Figura 3-5 Módulo 4 relés

3.1.6. Adaptador USB Wifi

Para que el sistema funcione es necesario una conexión a internet y en el caso de la Raspberry Pi se puede conectar a través de la clavija RJ45 con un cable Ethernet pero para darle mayor movilidad al sistema y ahorrar conectar más cables se ha decidido invertir en un dispositivo USB para tener conexión inalámbrica a la red. Mirando en la lista de dispositivos compatibles se opta por el adaptador TL-WN725N de TP-LINK.

Especificaciones	Valor
Estándar	IEEE 802.11 b/g/n
Frecuencia	2.4~2.4835 GHz
Velocidad	150 Mbps
Dimensiones	18.6 x 15 x 7.1 mm
Seguridad	WEP, WPA/WPA2, WPA- PSK/WPA2-PSK

Tabla 3-5 Especificaciones adaptador Wifi TL-WN725N



Figura 3-6 Adaptador USB Wifi TL-WN725N

3.2. Montaje

En este apartado se va a describir paso a paso como ha sido el montaje del sistema así como el conexionado de cada componente con la Raspberry Pi.

En primer lugar para hacer el sistema más compacto y poder dar sujeción a diversos componentes se decide encapsular la Raspberry Pi y la mayoría de los componentes en una caja de registro como la mostrada en la Figura 3-7.



Figura 3-7 Caja de registro

En segundo lugar se hacen los orificios en la tapa de la caja para la instalación del servo y para la conexión de la cámara a la Raspberry Pi como se puede ver en la Figura 3-8.



Figura 3-8 Orificios sujeción caja de registro

En la Figura 3-9 y 3-10 se puede ver una vista anterior y superior del sistema final con todos sus componentes conectados así como el adaptador fabricado para sujetar la cámara al servo. Se puede observar como se ha usado una placa de pruebas para conectar los distintos dispositivos ya que no había suficientes pines de alimentación a 5V en la Raspberry Pi.



Figura 3-9 Vista anterior sistema final



Figura 3-10 Vista superior sistema final

Como aparece en la Figura 3-10 se ha realizado una segunda caja para encapsular las conexiones de la bombilla, a 12V, y futuros elementos con el módulo de relés. De este modo se tienen separadas distintas alimentaciones (12V o 5V) en cajas independientes para hacer más fácil futuras implementaciones.

3.2.1. Diagrama de conexiones



Figura 3-11 Diagrama de conexiones

En la Figura 3-11 se puede ver un esquema casi completo, a falta de conectar la Raspberry Pi Camera en su puerto correspondiente, de las conexiones necesarias en el sistema. La Raspberry Pi únicamente tiene dos pines de alimentación a 5V y se necesita conectar tres elementos por lo tanto es necesario usar una placa de prueba.

El servo está conectado al pin GPIO18 de la Raspberry Pi en el que posteriormente se configurará un PWM para controlar dicho servo.

La salida del sensor PIR se encuentra conectada al pin GPIO23 para su posterior tratamiento de señal y poder así detectar la presencia de algún individuo en la habitación donde se instale el sistema.

Aunque en el diagrama únicamente conectamos la entrada (IN1) del módulo de relés al pin GPIO24 es posible conectar las otras entradas restantes a cualquiera de los pines GPIO libres y tratar los diodos indicadores de los relés como luces o avisos de activación de cualquier dispositivo.

En la siguiente figura se puede ver cómo se conecta la cámara a la Raspberry Pi.



Figura 3-12 Conexión Raspberry Pi Camera

3.3. Entorno Web

Para que el usuario pudiera interactuar con el sistema desde cualquier lugar se opta por diseñar una web que se alojará en la Raspberry Pi. Anteriormente se ha debido configurar un servidor PHP en ésta como se puede ver en el apartado 3.4.3.1.

El primer punto a tener en cuenta ha sido la seguridad dado que la finalidad de esta web es dar tanto acceso al sistema de seguridad de una habitación, como poder manipularlo y controlarlo remotamente. En la figura 3-13 se muestra la página de inicio de la web cuya finalidad es restringir el acceso a usuarios que previamente el desarrollador ha tenido que dar de alta. La configuración de usuarios se realiza mediante "phpMyAdmin" como se muestra en el apartado 3.4.3.2.



En segundo lugar, una vez ingresado como usuario, se dan las opciones de operar en modo automático o en modo manual como se ve en la siguiente figura:



Figura 3-14 Modo de operación

Dentro del modo de vigilancia automática el sistema se encargar de detectar la presencia de algún individuo en la habitación así como de avisar mediante "whatsapp". Si el modo se ha activado correctamente la web nos avisa como se observa en la figura 3-15.



Vigilancia activada

Figura 3-15 Página aviso activación

En el caso del modo de vigilancia manual la interfaz web mediante un menú permite dos opciones, una de ellas es ver el estado de la habitación en tiempo real mediante streaming como se puede ver en la figura 3-16.



Figura 3-16 Página de streaming

La otra opción permite tomar imágenes del estado de la sala mostrándolas por la web y controlar la activación de las luces como se muestra en la figura 3-17.



Figura 3-17 Página de control

3.4. Configuración

Una de las partes claves de este proyecto es la correcta configuración de los elementos que lo conforman así como instalación de las librerías necesarias para que el sistema funcione. En los siguientes apartados se va a describir dicha configuración necesaria.

3.4.1. Raspberry Pi Camera

La cámara es sensible a descargas electrostáticas por lo que se debe manipular con especial cuidado.

Con la Raspberry Pi apagada, conectamos un monitor por HDMI, teclado y ratón por USB, para posteriormente poder acceder a la configuración de la cámara, y para conectar la cámara correctamente levantamos el conector y conectamos la cinta con la banda azul como se muestra en la figura 3-18 y después bajamos el conector.



Figura 3-18 Conexión cámara a Raspberry Pi

Una vez conectado todo se enciende la Raspberry Pi y se ingresa por la línea de comandos lo siguiente:

pi@raspberrypi ~ \$ sudo apt-get update

pi@raspberrypi ~ \$ sudo apt-get upgrade

Ahora se accede al menú de configuración con el siguiente comando:

pi@raspberrypi ~ \$ sudo raspi-config

Aparece el menú mostrado en la figura 3-19 y se selecciona la opción 5 y posteriormente en el submenú de la figura 3-20 se pone la cámara en "Enable" y se reinicia la Raspberry Pi para poder y dar uso a la cámara.



Figura 3-19 Menú de configuración Raspberry Pi

😣 🖻 🗊 🏾 pi@	Draspberrypi: ~	
	Enable support for Raspberry Pi camera?	
	<mark><disable></disable></mark> <enable></enable>	

Figura 3-20 Submenú de configuración Raspberry Pi Camera

3.4.2. Adaptador Wifi

En el caso del adaptador escogido es necesario descargar el software para instalarlo y poder poner en funcionamiento el adaptador. En este caso depende de la versión de Raspbian que se esté usando. Para ello el primer paso es comprobar esa versión con el siguiente comando:

```
uname -a
```

Y como resultado del comando se obtiene:

```
Linux raspberrypi 3.18.13+ #784 PREEMPT Sat May 9 15:49:42 BST 2015 armv61 GNU/Linux
```

La parte importante es lo mascado en negrita y subrayado (3.18.13+ #784) que es lo necesario a la hora de buscar el driver correspondiente en el siguiente link:

https://www.raspberrypi.org/forums/viewtopic.php?f=28&t=62371&sid=a880b14059c80371f70f0a7c961a3cf

Una vez comprobado y seleccionado el driver correcto se ejecuta los siguientes comandos donde se sustituye lo marcado en negrita por el correspondiente a la versión en uso:

```
wget https://dl.dropboxusercontent.com/u/80256631/<u>8188eu-20150509.tar.gz</u>
tar xzf 8188eu-2015yyzz.tar.gz
./install.sh
```

Para seguir con la configuración del wifi de una forma más simple se accede al modo monitor escribiendo lo siguiente en la línea de comandos:

startx

El último paso es conectarse a la red preferida introduciendo los datos necesarios.

3.4.3. Servidor PHP

La finalidad de instalar un servidor en la Raspberry Pi es alojar en el dispositivo una web que sirva de interfaz de usuario y a su vez, mediante PHP, pueda ejecutar los códigos escritos en Python que están almacenados en la memoria.

3.4.3.1. Instalación de Apache y PHP

El primer paso será crear el grupo que usa apache por defecto y darle permisos mediante los siguientes comandos:

sudo addgroup www-data

sudo usermod -a -G www-data www-data

Posteriormente se actualizan los repositorios y se procede a la instalación de Apache y PHP:

sudo apt-get update

sudo apt-get install apache2 php5 libapache2-mod-php5

Por último se reinicia el servidor apache con el siguiente comando y ya está disponible la carpeta *var/www* para alojar las páginas web del proyecto, a las que se podrá acceder desde cualquier lugar.

sudo /etc/init.d/apache2 restart

3.4.3.2. Instalación y configuración de MySQL y phpMyAdmin

MySQL es una base de datos Open Source que sirve para almacenar datos y phpMyAdmin es un software que permite administrar bases de datos MySQL a través de una interfaz web que en el caso de este sistema de seguridad se usara para almacenar los usuarios registrados en la aplicación.

Para la instalación de la base de datos se escribe por línea de comandos lo siguiente:

```
sudo apt-get install mysql-server mysql-client php5-mysql
```

Durante la instalación se tendrá que administrar una contraseña para poder acceder posteriormente a la base de datos con el usuario root.

Para la instalación de la instalación de phpMyAdmin se escribe lo siguiente:

```
sudo apt-get install libapache2-mod-auth-mysql php5-mysql phpmyadmin
```

Durante la instalación habrá que decir que servidor web se usa, en este caso apache, que si se quiere configurar una base de datos y facilitar una nueva contraseña.

Al finalizar habrá que modificar el siguiente archivo con el editor nano:

```
sudo nano /etc/php5/apache2/php.ini
```

Y escribimos extension=mysql.so en la sección "Dynamic Extensions" como se ve en la figura 3-21.



Figura 3-21 Fichero /etc/php5/apache2/php.ini

Después habrá que reiniciar la Raspberry Pi con el comando sudo reboot y tras el reinicio lo único que queda es ejecutar los siguientes comandos:

```
sudo ln-s /etc/phpmyadmin/apache.conf /etc/apache2/conf.d/phpmyadmin.conf
sudo/etc/init.d/apache2reload
```

Para acceder a configurar la base de datos hay que escribir en el navegador la siguiente dirección: <u>http://ipdispositivo/phpmyadmin</u> y saldrá la siguiente interfaz:

phpMyAdmin
Bienvenido a phpMyAdmin
Idioma - Language
Español - Spanish
Iniciar sesión 🕡
Usuario:
root
Contraseña:
Continuar

Figura 3-22 Interfaz phpMyAdmin

Al acceder a phpMyAdmin se puede crear una base de datos para guardar los usuarios que tendrán acceso al sistema de seguridad. En primer lugar se crea una nueva base de datos llamada Seguridad y dentro de ésta una tabla con los usuarios como se observa en las siguientes figuras.

En la figura 3-23 se observa los pasos necesarios para crear una nueva base de datos. Pinchando en la pestaña base de datos, aparece lo mostrado en la imagen. Luego se introduce un nombre y se hace clic en crear y así se dispone de una nueva base de datos.

Para crear una tabla en la base de datos correspondiente, primero se accede clicando sobre ella y posteriormente como se muestra en la figura 3-24, directamente la propia interfaz da la opción de crear la tabla introduciendo nombre y número de columnas. A la hora de crear la tabla se necesita especificar el número de campos que se van a introducir, que en el caso de este sistema son: id, usuario y contraseña. Este paso se puede observar en la figura 3-25.

Por último si se accede a la tabla es posible insertar nuevos usuarios clicando en la pestaña insertar y rellenando los campos correspondientes como se muestra en la figura 3-26.

php MuAdmin										
	Bases de datos	📕 SQL	🌗 Estado actual	Procesos	Privilegios	🐻 Exportar	📑 Importar	Variables	Juegos de caracteres	▼ Más
🏡 🗾 😹 🥹 🗊 😋	\checkmark									
information_schema impsql	Bases de da	tos								
performance_schema phpmyadmin seguridad	Crear nueva base	de datos	Cotejamiento	۲	Crear					
	Base de datos 🔔				<u> </u>	•				
	information_schema	🛤 Comp	obar los privilegios							
	mysql el Comprobar los privilegios									
	📄 performance_schema 📧 Comprobar los privilegios									
	phpmyadmin	a: Comp	obar los privilegios							
	seguridad	🛤 Comp	obar los privilegios							
	Total: 5									
	Marcar todos / Desmarcar todos Para los elementos que están marcados: 📪 Eliminar									
	III Activar las estadísticas									
	A Nota: Activar aquí las estadisticas de la base de datos podría causar tráfico pesado entre el servidor web y el servidor MySQL.									

Figura 3-23 Interfaz phpMyAdmin (Crear base de datos)

phpMyAdmin	🤃 localhost y 🗴 seguridad	
	🖗 Estructura 📙 SQL 🔍 Buscar 🕢 Generar una consulta 🚔 Exportar 🚔 Importar 🤌 Operaciones 💌 Privilegios 💿 Seguimiento 🔻 Más	
<u> </u>	Tabla Acción Filas Tipo Cotejamiento Tamaño Residuo a depurar usuarios Examinar Hestar Structura Buscar Structura Vaciar Structura Eliminar 1 InnoDB Iatin 1_swedish_ci 16.8 18	
seguridad	1 tabla Número de filas 1 InnoDB latin1_swedish_ci 16.0 KB 0 8	
usuanos	Marcar todos / Desmarcar todos Para los elementos que están marcados: •	
Crear tabla	🚔 Vista de impresión 👼 Diccionario de datos	
	Tear nueva tabla en la base de datos seguridad	
	Nombre: Número de columnas:	
	Continuar	
		-
	Nombre. Número de columnas:	ar

Figura 3-24 Interfaz phpMyAdmin (Crear tabla)

ohoMuAdmin	🕼 localhost 🖡 🐻 seguridad							
P P	Nombre de la tabla:							
2 🧃 🔒 😣 🗊 🤤								
		Estr	uctura 😠					
isuarios	Columna	id user pass						
reas table	Tipo 😦	INT •	VARCHAR •	VARCHAR •				
	Longitud/Valores1		30	30				
	Predeterminado ²	Ninguno	Ninguno 🔻	Ninguno				
	Cotejamiento	•	•					
	Atributos	•	T	•				
	Nulo							
	Índice	*	- •	*				
	AUTO_INCREMENT							
	Comentarios							
	MIME-type	•	•	•				
	Transformación del navegador		•	•				
	Opciones de transformación ³							
	Comentarios de la tabla:	Mo	otor de almacenamiento: 💩	Cotejamiento:				
		I	nnoDB 🔹		•			
	definición de la PARTICIÓN: 🤢							
		<i>i</i>						

Figura 3-25 Interfaz phpMyAdmin (Campos tabla)

php MyAdmin	🤃 locathost y 👩 seguridad y 🐻 usuarios			
	🗉 Examinar 🦻 Estructura 📳 SQL 🔍 Buscar 🐉 Insertar 🖶 Exportar 🚔 Importar 🧬 Operaciones 💿 Seguimiento			
<u>∆</u> <u>≣</u> ≽ ⊌ î` ¢	Columna Tipo Función Nulo Valor			
seguridad 🔻	id int(11)			
📰 usuarios	user varchar(30)			
O Crear tabla	pass varchar(30)			
	Continuar			
	Ø Ignorar			
	Columna Tipo Función Nulo Valor			
	id int(11)			
	user varchar(30)			
	pass varchar(30)			
	Continuar			
	insertar como una nueva ma voiver voiver Continuar Reiniciar			
	Continuar inserción con 2 🔹 filas			

Figura 3-26 Interfaz phpMyAdmin (Insertar usuario)

3.4.4. Servicio NO-IP

Dado que se va a usar la Raspberry Pi como servidor y en este caso se quiere acceder remotamente desde cualquier red es más cómodo y seguro crear un subdominio y así no es necesario acceder al dispositivo usando su ip. Además como el posible que la ip del router sea dinámica, es decir, puede variar cada vez que éste se reinicie, con este servicio de hostname virtual se soluciona el problema de que cambie la ip.

En primer lugar, se debe de crear una cuenta gratuita en la página web <u>http://www.noip.com</u> y configurar un hostname. Una vez que ya se obtiene un usuario y contraseña se pasa a instalar el paquete NO-IP en la Raspberry Pi.

Se procede a descargar y descomprimir el paquete con los siguientes comandos:

```
wget http://www.no-ip.com/client/linux/noip-duc-linux.tar.gz
tar -zxvf noip-duc-linux.tar.gz
```

A continuación accedemos a la carpeta que se ha descomprimido e instalamos el programa con las siguientes órdenes:

```
cd noip-2.1.9-1/
make
sudo make install
```

Durante el proceso de instalación se pedirá que se introduzca los datos de la cuenta creada anteriormente.

El siguiente paso es crear un fichero con el contenido de la figura 3-27:

```
sudo nano /etc/init.d/noip2
```



Figura 3-27 Fichero /etc/init.d/noip2

Después se guarda el fichero, se le da permiso de ejecución y se coloca en la cola de arranque con las siguientes sentencias:

```
sudo chmod +x /etc/init.d/noip2
sudo update-rc.d noip2 defaults
```

Para finalizar se necesita abrir el puerto 80 del router, ya que se usa el protocolo http, al que esté conectada la Raspberry Pi para así poder utilizar el servicio NO-IP.

3.4.5. Motion

El uso que se le da a motion en este proyecto es el de crear un flujo de transmisión de video en tiempo real de lo que ocurre en la habitación. Se ha usado este software porque permite configurar multitud de características fácilmente mediante un fichero de configuración y porque resultaba más sencillo incluir ese stream en la interfaz de usuario.

En primer lugar, se realiza una actualización de la Raspberry Pi con los comandos y un reinicio:

```
sudo apt-get update
sudo apt-get upgrade
sudo reboot
```

Posteriormente se instalan las siguientes librerías necesarias para que funcione el programa:

```
sudo apt-get install motion
sudo apt-get install libjpeg62
```

Seguidamente se crea una carpeta en la que se descargará los ficheros del software:

```
cd ~/
mkdir mmal
cd mmal
wget https://www.dropbox.com/s/xdfcxm5hu71s97d/motion-mmal.tar.gz
tar -zxvf motion-mmal.tr.gz
```

Ahora lo único que queda por hacer es modificar el fichero de configuración con lo que cada usuario requiera. Por ejemplo modificando los siguientes parámetros se está especificando la resolución del video, la carpeta donde se almacenará los videos y la ubicación del fichero.log.

width 640 height 480 target_dir /home/pi/m-video logfile /home/pi/mmal/motion.log

Para ejecutar el programa se usa la sentencia:

./motion -n -c motion-mmalcam.conf

3.4.6. Yowsup

Yowsup es una librería escrita en Python que permite interactuar con el servidor de WhatsApp, pudiendo así recibir y enviar mensajes desde la Raspberry Pi. La funcionalidad que se quiere conseguir usando esta librería es la de avisar mediante un mensaje a los usuarios del sistema de la intrusión de algún individuo en la habitación puesta en vigilancia.

Como siempre, en primer lugar es necesario actualizar el sistema a la versión más reciente y en este caso también son necesarias ciertas librerías para la instalación.

sudo apt-get update

```
sudo apt-get upgrade
sudo rpi-update
sudo apt-get install python-dateutil
sudo apt-get install argparse
sudo apt-get install python-setuptools
```

A continuación se copia el repositorio de yowsup en la Raspberry Pi.

git clone git://github.com/tgalal/yowsup.git

Ahora, se instala mediante el siguiente comando, para ello hay que estar en el directorio /yowsup que se ha copiado con anterioridad.

sudo python sepup.py install

En esta misma carpeta se crea un fichero con el siguiente contenido:

cc=34 #Código de país

phone=6123456789 #Número completo sin '+' o '00'

id=FF:FF:FF:FF:FF #MAC Raspberry Pi

password=PASSWORD #Completar después del registro

Finalmente, solo queda registrar el número a usar. Para ello se usa el primer comando con el que se enviará un sms con el código de registro XXX-XXX, que sustituyéndolo en el segundo debe de generar un mensaje con el password que hay que completar en el archivo de configuración creado anteriormente.

yowsup-cli registration -c config -r sms yowsup-cli registration -c config -R XXX-XXX

Para enviar mensajes desde la línea de comandos se usa la siguiente instrucción:

yowsup-cli demos --config ficheroconfig --send numero "mensaje"

3.5. Funcionamiento

En este apartado se va a describir tanto el funcionamiento del servidor y su consulta a la base de datos para comprobar el usuario y clave, como la interactuación entre PHP y los códigos escritos en Python.

3.5.1. Servidor PHP + Base de datos

La función de convertir la Raspbery Pi en un servidor que soporte PHP, es la de crear un panel de control de cara al usuario.

El único uso que se da a la base de datos en este sistema es la comprobación de los usuarios que son válidos a la hora de ingresar en la aplicación. Aunque se podría haber creado una página de registro, no es este el caso, ya que lo que se quiere es restringir el acceso. Por lo tanto estos usuarios han debido de ser registrados por el administrador con anterioridad.

En la siguiente figura se puede ver un diagrama del funcionamiento de un servidor web con la única diferencia de que en el caso de la Raspberry Pi el servidor web y la base de datos se encuentran en el mismo dispositivo.



Figura 3-28 Funcionamiento servidor web

Al introducir los datos e iniciar sesión se redirecciona a la página login.php, que incluye a funciones.php, cuyos códigos se muestran a continuación:

Código 3-1 login.php

```
<?php
require once('funciones.php');
conectar('localhost', 'root', 'Klaugd07.', 'seguridad');
//Recibir
$user = strip_tags($_POST['user']);
$pass = strip tags($ POST['pass']);
$query
                @mysql query('SELECT
                                               FROM
                                                        usuarios
                                                                     WHERE
          =
                                         *
user="'.mysql real escape string($user).'"
                                                                        AND
pass="'.mysql real escape string($pass).'"');
if($existe = @mysql fetch object($query))
{
                                                                      echo
'<script>window.location="opcion.html"</script>';
}else{
                                                                      echo
'El usuario y/o password son incorrectos.';
}
?>
```

Código 3-2. funciones.php

En el código se puede ver como mediante el método POST se obtienen los argumentos introducidos en el formulario al iniciar sesión.

Haciendo uso de las funciones de mysql se hace la consulta a la base de datos y con la sentencia if del final se redirecciona a la siguiente página, que permite elegir el modo de operación, o muestra por pantalla que los datos no son correctos.

3.5.2. PHP + Python

Cuando aparece el modo de selección hay dos opciones, que el sistema funcione automáticamente o que el usuario pueda controlar de forma manual la vigilancia de la sala.

En el caso que se seleccione el modo automático, mediante php se ejecuta por la línea de comandos un script en el que se llama al programa presencia.py.

Código 3-3. opcion.php

```
<?php
//Funcion PHP
$opcion = $_POST[opcion];
if($opcion == "auto"){
echo '<script>window.location="fin.html"</script>';
$a- exec("sh startauto.sh > salida.txt");
echo $a;
}
else
echo '<script>window.location="portada.php"</script>';
?>
```

En el código 3-3 se puede ver como se llama al script startauto.sh redirigiendo la salida a un archivo de texto para que se muestre la siguiente página web y no se quede esperando la a que acabe el programa.

Código 3-4. presencia.py

```
#!/usr/bin/env python
import RPi.GPIO as GPIO
import time
import picamera
import os
GPIO.setmode (GPIO.BCM)
GPIO.setup(23,GPIO.IN)
GPIO.setup(24,GPIO.OUT,initial=1)
camera = picamera.PiCamera()
Previous State = 0
Current State = 0
try:
while True:
Current State = GPIO.input(23)
if Current State==1 and Previous State==0:
# PIR is triggered
print " Movimiento en la entrada!"
GPIO.output(24,False)
time.sleep(1)
camera.capture('/var/www/intruso.jpg')
os.system("yowsup-cli demos
                               --config
                                         /home/pi/yowsup/config
                                                                   --send
34645198559 'Intruso!!! http://klaugd07rpi.ddns.net/intruso.jpg'")
time.sleep(4)
# Record previous state
Previous State = 1
elif Current State==0 and Previous State==1:
# PIR has returned to ready state
GPIO.output(24, True)
Previous State=0
print " No hay movimiento"
```

```
Wait for 10 milliseconds
time.sleep(0.01)
time.sleep(1)
```

```
except KeyboardInterrupt:
print("Exit")
GPIO.cleanup()
```

El código 3-4 es el que se ejecuta al iniciar el modo de operación automático. El código lo que hace es comprobar el pin de la Raspberry Pi donde está conectado el sensor PIR y en el momento que detecta un movimiento captura una imagen y mediante la librería "yowsup" manda un mensaje al móvil del usuario avisando de que existe un intruso y mandándole un enlace de la foto tomada.

Por otro lado, en el caso que se seleccione el modo de operación manual la web nos manda a los paneles de control que se muestran en las figuras 3-16 y 3-17.

En el código 3-5 se muestra parte del código de la web portada.php. Esta web consiste en una serie de formularios para controlar tanto los relés como para tomar una imagen panorámica del estado de la habitación.

Mediante PHP se comprueba que acción se ha querido realizar el usuario y se llama al programa escrito en Python que corresponda por línea de comandos.

Código 3-5. portada.php

. . .

```
. . .
<div align="center">CONTROL LAMPARA 1</div>
     <div align="center">
           <form action="" method="post">
                <input type="submit" name="encender24"
value="Encender">
                <input type="submit" name="apagar24" value="Apagar">
                <input type="submit" name="parpadear24"
value="Parpadear">
           </form>
     </div>
    <br>
<div align="center">ESTADO HABITACION</div>
     <div align="center">
           <form action="" method="post">
                <input type="submit" name="panoramica" value="Tomar
Imagenes">
           </form>
```

```
</div>
     <div align="center">
     <img src="image1.jpg" width="300px">
           <img src="image2.jpg" width="300px">
           <img src="image3.jpg" width="300px">
    </div>
     <hr>
</body>
</html>
<?php
//Funcion PHP fotos
     if($ POST[panoramica]) {
           $a- exec("sudo python /var/www/fotos/panoramica.py");
           echo $a;
     }
//Funciones PHP del pin GPIO 24
     if($ POST[encender24]){
           $a- exec("sudo python /var/www/leds/led24on.py");
           echo $a;
     }
     if($ POST[apagar24]) {
           $a- exec("sudo python /var/www/leds/led24off.py");
           echo $a;
     }
     if($ POST[parpadear24]){
           $a- exec("sudo python /var/www/leds/led24parpadea.py");
           echo $a;
     }
. . .
. . .
?>
```

Los códigos 3-6, 3-7 y 3-8 muestran el control del relé para hacer encender, apagar o parpadear una lámpara respectivamente. En este caso se ha tenido que tener en cuenta a la hora de programar que los relés son activos a nivel bajo.

Código 3-6. led24on.py

```
#!/usr/bin/env python
```

import RPi.GPIO as GPIO

#Configuramos los pines
GPIO.setmode(GPIO.BCM)
#Pin 24 como salida
GPIO.setup(24,GPIO.OUT)
#Pin ON
GPIO.output(24,False)

Código 3-7. led24off.py

#!/usr/bin/env python

import RPi.GPIO as GPIO

#Configuramos los pines
GPIO.setmode(GPIO.BCM)
#Pin 24 como salida
GPIO.setup(24,GPIO.OUT)
#Pin ON
GPIO.output(24,True)
#Liberamos pines
GPIO.cleanup()

Código 3-8. led24parpadea.py

```
#!/usr/bin/env python
```

import RPi.GPIO as GPIO
import time

#Configuramos los pines
GPIO.setmode(GPIO.BCM)
#Pin 24 como salida
GPIO.setup(24,GPIO.OUT)

for i in range(0,5):
#Pin ON
GPIO.output(24,False)

```
time.sleep(1)
#Pin OFF
GPIO.output(24,True)
time.sleep(1)
#Liberamos los pines
GPIO.cleanup()
```

Si se cliquea en la opción tomar imágenes, se toma una imagen panorámica de la habitación mediante el código 3-9.

Código 3-9. panoramica.py

```
import picamera
import time
                            #Importamos para usar time.sleep()
import os
os.system("sudo sh /var/www/stopmotion.sh")
time.sleep(1)
camera=picamera.PiCamera()
camera.start preview()
os.system("sudo python /var/www/servo/left.py")
camera.capture('image1.jpg')
time.sleep(0.5)
os.system("sudo python /var/www/servo/center.py")
camera.capture('image2.jpg')
time.sleep(0.5)
os.system("sudo python /var/www/servo/right.py")
camera.capture('image3.jpg')
time.sleep(0.5)
os.system("sudo python /var/www/servo/center.py")
camera.stop preview()
camera.close()
```

```
#os.system("sudo sh /var/www/startauto.sh")
```

En este caso hay que tener en cuenta que es posible que se pulse esta opción estando activa la cámara por lo tanto lo primero que se hace es parar el modo "streaming" y después se toman tres imágenes en tres ángulos distintos. Para mover el servo en el que se encuentra montada la cámara se usan los códigos siguientes, llamándolos usando la librería "os", que permite ejecutar por línea de comando desde Python.

Código 3-10. left.py

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setup(18,GPIO.OUT)
pwm = GPIO.PWM(18,50)
pwm.start(7.5)
pwm.ChangeDutyCycle(10.5)
time.sleep(0.5)
pwm.stop()
GPIO.cleanup()
```

Código 3-11. center.py

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setup(18,GPIO.OUT)
pwm = GPIO.PWM(18,50)
pwm.start(7.5)
pwm.ChangeDutyCycle(7.5)
time.sleep(0.5)
pwm.stop()
GPIO.cleanup()
```

Código 3-12. right.py

import RPi.GPIO as GPIO
import time

```
GPIO.setmode(GPIO.BCM)
GPIO.setup(18,GPIO.OUT)
pwm = GPIO.PWM(18,50)
pwm.start(7.5)
pwm.ChangeDutyCycle(4.5)
time.sleep(0.5)
pwm.stop()
GPIO.cleanup()
```

Otra opción que permite el modo de operación manual es la de ver que está ocurriendo en tiempo real en la habitación desde la pestaña "Streaming". Desde esta interfaz se puede tanto iniciar como parar el stream y mover la cámara para tener más ángulo de visión controlando el servo con los códigos visto anteriormente.

E ste trabajo fin de grado tenía como principal objetivo diseñar y construir un sistema de vigilancia inalámbrico, el cual se ha cumplido con ciertas dificultades y algunos inconvenientes que se nombrarán a continuación.

El principal incoveniente se tiene a la hora de implementar la detección de intrusos, ya que la primera idea era usar una de las opciones que da el archivo de configuración del programa motion. Este programa mediante la opción on_motion_detected permite la ejecución de comandos al detectar la cámara un cambio de píxeles, pero en este caso y tras muchos intentos fallidos de poder configurarla conrrectamente se decide programar la detección de intrusos haciendo uso de un sensor infrarrojo pasivo (PIR), consiguiendo de este modo no tener en continuo funcionamiento la cámara.

Una de las partes que en principio podía entrañar más dificultad a la hora de comenzar a desarrollar este TFG era el aprendizaje de un nuevo lenguaje de programación, aunque todo lo contrario ya que Python usa una sintaxis simple, clara y sencilla que permite que el desarrollo de una aplicación en este lenguaje sea sencillo y rápido.

Lo más difícil de este trabajo, en cuanto a horas de estudio y pruebas es la programación de la librería "Yowsup" ya que la mayoría de la documentación existente es sobre versiones antiguas de este programa, con un funcionamiento y configuración totalmente distinto.

Otro punto en el que se ha empleado un gran número de horas es en la programación de la interfaz web, pero al contrario de la parte de configuración, no entraña una gran dificultad dado que la programación web es algo que hoy en dia se encuentra bastante extendida y existe documentación suficiente para que alguien que tenga conocimientos básicos sobre HTML y PHP pueda realizar bastantes ideas de las que se plantee, siempre empleando un tiempo considerable.

En cuanto a la calidad de las imágenes de la Raspberry Pi Camera se ha de comentar que en condiciones de luz presenta una calidad aceptable y en cuanto al video en sreaming también decir que se obtiene un retraso de unos 2-3 segundos. Todo esto se podría mejorar usando una cámara USB de mayor calidad. La captura de imágenes con baja luminosidad se soluciona activando la iluminación de la habitación cuando se detecta la presencia de algún intruso.

4.1. Posibles mejoras futuras

Una vez finalizado este Trabajo Fin de Grado se han tenido en cuenta una serie de aspectos que se podrían mejorar o implementar algunos nuevos para darle mayor funcionalidad al sistema.

Uno de los aspectos que se deben de mejorar es la alimentación del sistema ya que actualmente se usa una fuente de alimentación de 5V para la Raspberry Pi y 12V para la iluminación. Todo esto se podría sustituir por una batería de litio que además le daría total movilidad al sistema dentro del rango de alcance del Wifi.

En la siguiente imagen se observa un ejemplo de batería con el que podría tener total funcionamiento el proyecto ya que inlcuyo ambos voltajes necesarios.





Otra mejora que se le puede aplicar al sistema es la ampliación con nuevas funcionalidades aprovechando el modulo de relés, en el cual solo está en uso uno de ellos con la conexión a una lámpara. En los relés vacios se pueden conectar tanto nuevas lámparas para iluminar otras partes de la casa tanto motores para el control de persianas o la inclusión de un zumbador para advertir de la presencia del intruso. En la siguiente imagen podemos ver tanto la conexión de la bombilla como la del zumbador al módulo de relés.



Figura 4-2 Conexiones dispositivos a módulo de relés



Figura 4-3 Zumbador 12V

En la Figura 4-3 se puede observar un ejemplo de zumbador que se podría conectar al sistema.

Una de las mejoras principales debería de ser la sustitución de la cámara por la Raspberry Pi NoIR Camera que permitiría la visión de imágenes en condiciones de baja luminosidad.



Figura 4-4 Raspberry Pi NoIR Camera

En las siguientes figuras se pueden ver las diferencias entre imágenes tomadas por la Pi Camera y por la Pi NoIR Camera.



Figura 4-5 Imagen tomada Pi NoIR Camera



Figura 4-6 Imagen tomada Raspberry Pi Camera

- [1] Alejandro Esquiva Rodríguez. Tutorial Raspberry Pi 15. Instalación de Apache + MySQL + PHP, 16 Agosto 2013, [consulta: 12 Abril 2015]. Disponible en: https://geekytheory.com/tutorial-raspberry-pi-15instalacion-de-apache-mysql-php/
- [2] FjRamirez. Instalación del módulo cámara de Raspberry Pi, 15 Julio 2013, 22:48, [consulta: 15 Julio 2015]. Disponible en: http://www.tuelectronica.es/tutoriales/raspberry-pi/instalacion-del-modulo-camara-de-raspberry-pi.html#
- [3] Héctor Herrero Hermida. Raspberry Pi Uso de Relé, X10 y Detector de Movimiento, 26 Enero 2015, [consulta: 15 Julio 2015]. Disponible en: http://www.bujarra.com/raspberry-pi-uso-de-rele-y-detector-demovimiento/
- [4] Tarek (tgalal). Yowsup (The Python WhatsApp Library), 23 Junio 2015, [cosulta: 15 Julio 2015]. Disponible en: https://github.com/tgalal/yowsup
- [5] MrEngman. (UPDATE) Drivers for TL-WN725N V2 3.6.11+ -> 4.0.xx+, 16 Julio 2015, [consulta: 15 Julio 2015]. Disponible en: https://www.raspberrypi.org/forums/viewtopic.php?f=28&t=62371
- [6] Raspberry Pi Foundation. The making of Pi. Disponible en: https://www.raspberrypi.org/about/
- [7] Peatonet. Raspberry Pi y los pines GPIO: Controlando el LED desde una interfaz web, 5 Octubre 2014, [consulta: 15 Julio 2015]. Disponible en: http://www.peatonet.com/raspberry-pi-y-los-pines-gpiocontrolando-el-led-desde-una-interfaz-web/
- [8] Using the Raspberry Pi. Disponible en: https://www.raspberrypi.org/forums/viewforum.php?f=12
- [9] Programming. Disponible en: https://www.raspberrypi.org/forums/viewforum.php?f=14
- [10] Lady Ada. Introducing the Raspberry Pi Model B+, 4 Mayo 2015, [consulta: 15 Julio 2015]. Disponible en: https://learn.adafruit.com/introducing-the-raspberry-pi-model-b-plus-plus-differences-vs-modelb/overview
- [11] Guido van Rossum. The Python Tutorial, Septiembre 2014, [consulta: Julio 2015] Disponible en: https://docs.python.org/2/tutorial/index.html#the-python-tutorial
- [12] Scavix. Raspberry Pi as low-cost HD surveillance camera, Step 7: Installing the motion detection software, [consulta: Julio 2015]. Disponible en: http://www.instructables.com/id/Raspberry-Pi-as-low-cost-HD-surveillance-camera/step7/Installing-the-motion-detection-software/
- [13] Huguidugui. Login y registro: Validar con PHP (servidor) y enviar a la BD, 10 Octube 2013, [consulta: 15 Julio 2015]. Disponible en: https://huguidugui.wordpress.com/2013/10/10/login-y-registro-validarcon-php-servidor-y-enviar-a-la-bd/
- [14] HTML(5) Tutorial, [consulta: 15 Julio 2015]. Disponible en: http://www.w3schools.com/html/default.asp
ANEXO A: CÓDIGO INTERFAZ DE USUARIO

En este anexo se muestra el código de la página web al completo creada para servir de interfaz de usuario para un control mas simplificado de cara a cualquier tipo de usuario.

Código A-1 index.html

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>SecurityPi</title>
<style type="text/css">
body,td,th {
     color: #000000;
     font-family: Gotham, "Helvetica Neue", Helvetica, Arial, sans-
serif;
}
body {
    background-image: url();
    background-repeat: repeat-x;
    background-color: #FFFFFF;
}
</style>
<div align="center"><a href="index.html"><img src="Cabecera.jpg"</pre>
></a></div>
</head>
<body>
<hr>
<center>
<td width="847" height="66" style="text-align: center; font-
family: Gotham, 'Helvetica Neue', Helvetica, Arial, sans-serif; font-
style: normal;"><h2><strong>Iniciar Sesion</strong></h2><//r>
   <form id="form1" name="form1" method="post"
action="login.php">
       <label for="textfield"></label>
         Usuario:
         <input type="text" name="user">
       Contraseña:
         <input type="password" name="pass" >
       <div align="center">
         <input type="submit" name="submit" value="Acceder" />
       </div>
     </form>
```

</center>
</body>
</html>

Código A-2 login.php

```
<?php
require once('funciones.php');
conectar('localhost', 'root', 'Klaugd07.', 'seguridad');
//Recibir
$user = strip tags($ POST['user']);
$pass = strip tags($ POST['pass']);
$query = @mysql query('SELECT * FROM usuarios WHERE
user="'.mysql_real_escape_string($user).'" AND
pass="'.mysql real escape string($pass).'"');
if($existe = @mysql fetch object($query))
{
     echo '<script>window.location="opcion.html"</script>';
}else{
     echo 'El usuario y/o password son incorrectos.';
}
?>
```

Código A-3 funciones.php

```
<?php
function conectar($servidor, $user, $pass, $name)
{
     $con = @mysql_connect($servidor, $user, $pass);
     @mysql_select_db($name, $con);
}
?>
```

Código A-4 opcion.php

```
<?php
    //Funcion PHP
    $opcion = $_POST[opcion];
    if($opcion == "auto"){
        echo '<script>window.location="fin.html"</script>';
        $a- exec("sh startauto.sh > salida.txt");
        echo $a;
    }
    else
        echo '<script>window.location="portada.php"</script>';
```

Código A-5 fin.html

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>SecurityPi</title>
<div align="center"><a href="index.html"><img src="Cabecera.jpg"</pre>
></a></div>
<style type="text/css">
body,td,th {
     color: #000000;
     font-family: Gotham, "Helvetica Neue", Helvetica, Arial, sans-
serif;
}
</style>
</head>
<body>
     <hr/>
     <div align="center">
                 <h1>Vigilancia activada</h1>
     </div>
    </body>
</html>
```

Código A-6 portada.php

```
<!doctype html>
<html>
<head>
<META HTTP-EQUIV="REFRESH" CONTENT="10;URL=portada.php">
<meta charset="utf-8">
<title>SecurityPi</title>
<div align="center"><a href="index.html"><img src="Cabecera.jpg"</pre>
></a></div>
<style type="text/css">
body,td,th {
     color: #000000;
     font-family: Gotham, "Helvetica Neue", Helvetica, Arial, sans-
serif;
}
#nav, #nav ul {
    background-color: #8899AA;
    list-style: none outside none;
    margin: 0;
    padding: 0;
}
#nav {
    display: block;
    padding: 5px;
    position: relative;
    width: 112px;
```

```
-moz-perspective: 200px;
    -ms-perspective: 200px;
    -webkit-perspective: 200px;
    -o-perspective: 200px;
   perspective: 200px;
}
#nav ul {
   left: -9999px;
   opacity:0;
    overflow: hidden;
   padding: 5px;
   position: absolute;
    top: -9999px;
    -moz-transform: rotateY(70deg);
    -ms-transform: rotateY(70deg);
    -o-transform: rotateY(70deg);
    -webkit-transform: rotateY(70deg);
    transform: rotateY(70deg);
    -moz-transform-origin: 0 0;
    -ms-transform-origin: 0 0;
    -o-transform-origin: 0 0;
    -webkit-transform-origin: 0 0;
    transform-origin: 0 0;
    -moz-transition: -moz-transform 0.3s linear, opacity 0.3s linear;
    -ms-transition: -ms-transform 0.3s linear, opacity 0.3s linear;
    -o-transition: -o-transform 0.3s linear, opacity 0.3s linear;
    -webkit-transition: -webkit-transform 0.3s linear, opacity 0.3s
linear;
    transition: transform 0.3s linear, opacity 0.3s linear;
}
#nav li {
   background-color: #FFFFF;
   position: relative;
}
#nav > li {
   -moz-transform-style: preserve-3d;
    -ms-transform-style: preserve-3d;
    -o-transform-style: preserve-3d;
    -webkit-transform-style: preserve-3d;
   transform-style: preserve-3d;
#nav li a {
   background-color: #AABBCC;
   border-color: #DDDDDD #555555 #555555 #DDDDDDD;
   border-style: solid;
   border-width: 1px;
    color: #000000;
   display: block;
    font-size: 15px;
   padding: 8px 10px 8px 5px;
    text-decoration: none;
   width:95px;
```

```
-moz-transition: all 0.2s linear;
    -ms-transition: all 0.2s linear;
    -o-transition: all 0.2s linear;
    -webkit-transition: all 0.2s linear;
    transition: all 0.2s linear;
}
#nav li:hover > a {
   background-color: #8899AA;
   border-color: #8899AA;
    color: #FFFFFF;
#nav ul li {
   width: 100%;
}
</style>
</head>
<body>
<!--GPI018-->
     <hr>
    <div class="container">
     <a href="portada.php">Control</a>
          <a href="stream.php">Streaming</a>
     </div>
     <div align="center">CONTROL LAMPARA 1</div>
     <div align="center">
           <form action="" method="post">
                <input type="submit" name="encender24"
value="Encender">
                <input type="submit" name="apagar24" value="Apagar">
                <input type="submit" name="parpadear24"
value="Parpadear">
          </form>
     </div>
    <br>
    <div align="center">CONTROL LAMPARA 2</div>
     <div align="center">
          <form action="" method="post">
                <input type="submit" name="encender17"
value="Encender">
                <input type="submit" name="apagar17" value="Apagar">
                <input type="submit" name="parpadear17"
value="Parpadear">
          </form>
     </div>
    <br>
    <div align="center">CONTROL LAMPARA 3</div>
     <div align="center">
          <form action="" method="post">
                <input type="submit" name="encender27"
value="Encender">
                <input type="submit" name="apagar27" value="Apagar">
                <input type="submit" name="parpadear27"
value="Parpadear">
          </form>
```

```
</div>
    <br>
    <div align="center">CONTROL LAMPARA 4</div>
     <div align="center">
           <form action="" method="post">
                <input type="submit" name="encender22"
value="Encender">
                <input type="submit" name="apagar22" value="Apagar">
                <input type="submit" name="parpadear22"
value="Parpadear">
           </form>
     </div>
     <br>
     <div align="center">ESTADO HABITACION</div>
     <div align="center">
           <form action="" method="post">
                <input type="submit" name="panoramica" value="Tomar
Imagenes">
           </form>
     </div>
     <div align="center">
     <img src="image1.jpg" width="300px">
           <img src="image2.jpg" width="300px">
           <img src="image3.jpg" width="300px">
    </div>
     <hr>
</body>
</html>
<?php
//Funcion PHP fotos
     if($ POST[panoramica]) {
           $a- exec("sudo python /var/www/fotos/panoramica.py");
           echo $a;
     }
//Funciones PHP del pin GPIO 24
     if($ POST[encender24]) {
           $a- exec("sudo python /var/www/leds/led24on.py");
           echo $a;
     }
     if($ POST[apagar24]) {
           $a- exec("sudo python /var/www/leds/led24off.py");
           echo $a;
     }
     if($ POST[parpadear24]){
           $a- exec("sudo python /var/www/leds/led24parpadea.py");
           echo $a;
     }
```

```
//Funciones PHP del pin GPIO 17
     if($ POST[encender17]) {
           $a- exec("sudo python /var/www/leds/led17on.py");
           echo $a;
     }
     if($ POST[apagar17]) {
           $a- exec("sudo python /var/www/leds/led17off.py");
           echo $a;
     }
     if($ POST[parpadear17]) {
           $a- exec("sudo python /var/www/leds/led17parpadea.py");
           echo $a;
     }
     //Funciones PHP del pin GPIO 27
     if($_POST[encender27]){
           $a- exec("sudo python /var/www/leds/led27on.py");
           echo $a;
     }
     if($ POST[apagar27]) {
           $a- exec("sudo python /var/www/leds/led27off.py");
           echo $a;
     }
     if($ POST[parpadear27]){
           $a- exec("sudo python /var/www/leds/led27parpadea.py");
           echo $a;
     }
     //Funciones PHP del pin GPIO 22
     if($ POST[encender22]){
           $a- exec("sudo python /var/www/leds/led22on.py");
           echo $a;
     }
     if($ POST[apagar22]) {
           $a- exec("sudo python /var/www/leds/led22off.py");
           echo $a;
     }
     if($_POST[parpadear22]){
           $a- exec("sudo python /var/www/leds/led22parpadea.py");
           echo $a;
     }
?>
```

Código A-7 stream.php

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>SecurityPi</title>
<div align="center"><a href="index.html"><img src="Cabecera.jpg"</pre>
></a></div>
<style type="text/css">
body,td,th {
     color: #000000;
     font-family: Gotham, "Helvetica Neue", Helvetica, Arial, sans-
serif;
}
#nav, #nav ul {
    background-color: #8899AA;
    list-style: none outside none;
    margin: 0;
    padding: 0;
}
#nav {
    display: block;
    padding: 5px;
    position: relative;
    width: 112px;
    -moz-perspective: 200px;
    -ms-perspective: 200px;
    -webkit-perspective: 200px;
    -o-perspective: 200px;
    perspective: 200px;
}
#nav ul {
    left: -9999px;
    opacity:0;
    overflow: hidden;
    padding: 5px;
    position: absolute;
    top: -9999px;
    -moz-transform: rotateY(70deg);
    -ms-transform: rotateY(70deg);
    -o-transform: rotateY(70deg);
    -webkit-transform: rotateY(70deg);
    transform: rotateY(70deg);
    -moz-transform-origin: 0 0;
    -ms-transform-origin: 0 0;
    -o-transform-origin: 0 0;
    -webkit-transform-origin: 0 0;
    transform-origin: 0 0;
    -moz-transition: -moz-transform 0.3s linear, opacity 0.3s linear;
    -ms-transition: -ms-transform 0.3s linear, opacity 0.3s linear;
```

```
-o-transition: -o-transform 0.3s linear, opacity 0.3s linear;
    -webkit-transition: -webkit-transform 0.3s linear, opacity 0.3s
linear;
    transition: transform 0.3s linear, opacity 0.3s linear;
}
#nav li {
   background-color: #FFFFFF;
   position: relative;
}
#nav > li {
    -moz-transform-style: preserve-3d;
    -ms-transform-style: preserve-3d;
    -o-transform-style: preserve-3d;
    -webkit-transform-style: preserve-3d;
    transform-style: preserve-3d;
}
#nav li a {
   background-color: #AABBCC;
   border-color: #DDDDDD #555555 #555555 #DDDDDDD;
   border-style: solid;
   border-width: 1px;
    color: #000000;
   display: block;
    font-size: 15px;
   padding: 8px 10px 8px 5px;
   text-decoration: none;
   width:95px;
    -moz-transition: all 0.2s linear;
    -ms-transition: all 0.2s linear;
    -o-transition: all 0.2s linear;
    -webkit-transition: all 0.2s linear;
    transition: all 0.2s linear;
#nav li:hover > a {
   background-color: #8899AA;
   border-color: #8899AA;
    color: #FFFFFF;
#nav ul li {
   width: 100%;
}
</style>
</head>
<body>
     <hr>
    <div class="container">
     <a href="portada.php">Control</a>
          <a href="stream.php">Streaming</a>
     </div>
     <div align="center">STREAMING</div>
    <div align="center">
          <form action="" method="post">
                <input type="submit" name="start" value="Iniciar">
```

```
<input type="submit" name="stop" value="Parar">
           </form>
     </div>
    <div align="center">
           <form action="" method="post">
                <input type="submit" name="girarizquierda"
value="Izquierda">
                <input type="submit" name="centrar" value="Centro">
                <input type="submit" name="girarderecha"
value="Derecha">
           </form>
     </div>
    <div align="center">
           <img src="http://192.168.0.102:8081/?action=stream"</pre>
width="640" height="480"/>
     </div>
     <hr>
</body>
</html>
<?php
//Funciones PHP del pin SERVO
     if($ POST[girarizquierda]){
           $a- exec("sudo python /var/www/servo/left.py");
           echo $a;
     }
     if($ POST[centrar]) {
           $a- exec("sudo python /var/www/servo/center.py");
           echo $a;
     }
     if($ POST[girarderecha]){
           $a- exec("sudo python /var/www/servo/right.py");
           echo $a;
     }
     if($ POST[start]) {
           $a- exec("sudo python /var/www/start.py");
           echo $a;
     }
     if($ POST[stop]) {
           $a- exec("sudo python /var/www/stop.py");
           echo $a;
     }
```

?>