UNIVERSIDAD TECNOLÓGICA ISRAEL

TRABAJO DE TITULACIÓN EN OPCIÓN AL GRADO DE:

INGENIERO EN ELECTRÓNICA DIGITAL Y TELECOMUNICACIONES

TEMA: SISTEMA DE SEGURIDAD ANTIRROBO VEHICULAR, UTILIZANDO RASPBERRY PI CON TECNOLOGÍA 3G Y UNA APLICACIÓN PARA SMARTPHONE

AUTOR: SANTIAGO FERNANDO QUILACHAMÍN SIMBAÑA

TUTOR: Mg. RENE ERNESTO CORTIJO LEYVA

AÑO: 2017

DECLARACIÓN

Yo, Santiago Fernando QuilachamÍn Simbaña, declaro bajo juramento que el trabajo descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he considerado las referencias bibliográficas que se incluyen en este documento.

_____

Santiago Fernando QuilachamÍn Simbaña

CERTIFICACION DEL TUTOR

UNIVERSIDAD TECNOLÓGICA ISRAEL

APROBACIÓN DEL TUTOR

En mi calidad de tutor del trabajo de titulación certifico:

Que el trabajo de titulación **"SISTEMA DE SEGURIDAD ANTIRROBO VEHICULAR, UTILIZANDO RASPBERRY PI CON TECNOLOGÍA 3G Y UNA APLICACIÓN PARA SMARTPHONE**", presentado por la Sr. Santiago Fernando QuilachamÍn Simbaña, estudiante de la carrera de Electrónica Digital y Telecomunicaciones, reúne los requisitos y méritos suficientes para ser sometido a la evaluación del Tribunal de Grado, que se designe, para su correspondiente estudio y calificación.

Quito D.M. 5 de Octubre del 2017

TUTOR

……..

Ing. Rene Ernesto Cortijo Leyva, Mg

AGRADECIMIENTOS

A DIOS, a mis padres a quienes, a lo largo de mi vida, me han apoyado y motivado mi formación académica, dándome toda su confianza en mí en todo momento y no dudar de mis habilidades. A mis queridos profesores a quienes les debo la mayor parte de mis conocimientos, muy agradecido por su paciencia y enseñanza, y finalmente un eterno agradecimiento a esta respetuosa y prestigiosa universidad la cual me abrió sus puertas, preparándome para un futuro competitivo y formándome como un buen profesional.

DEDICATORIA

El presente proyecto va dedicado a todas esas personas que supieron encaminarme y contribuyeron de una manera u otra para su realización. A todas esas personas que mediante sus ideas e incentivos permitieron se materialice este proyecto, del cual me siento totalmente satisfecho y muy orgulloso en presentar. Para todos ellos hago esta dedicatoria con mucho cariño.

TABLA DE CONTENIDO

LISTA DE FIGURAS

LISTA DE TABLAS

RESUMEN

El presente proyecto, está basado en el diseño de un sistema de seguridad antirrobo de video vigilancia vehicular, mediante el control en una aplicación Smartphone, el sistema se activa mediante la comunicación bluetooth para dar aviso del estado de las puertas si están abiertas, adicionalmente captar el ruido por medio de un sensor de sonido el cual activa una cámara que envía una imagen por medio de un modem USB 3G hacia el correo electrónico, de esta forma permite visualizar los eventos por medio de una aplicación Android en un Smartphone.

El análisis en desarrollo, implica un estudio de los componentes necesarios tanto en hardware como en software que más se acoplen al sistema de video vigilancia vehicular, es decir se analiza la tarjeta Raspberry Pi 3B, la cual posee la capacidad de soportar la transmisión de secuencia de imágenes mediante tecnología WIFI y se conecte a un módulo USB 3G para que trabaje con un sensor de ruido, la propuesta detalla la necesidad y los materiales utilizados en el proyecto con que se llevó a cabo, se analiza por separado cada parte del software y hardware con el fin que sea todo comprensible y dinámico, los resultados y análisis de las pruebas ejecutadas permite validar el comportamiento y el correcto funcionamiento del sistema de video vigilancia. Estos resultados muestran el potencial impacto del uso de estas tecnologías y abren nuevas preguntas con respecto a la investigación acerca de los procesos y aplicaciones en la sociedad. Para finalizar se establecen las conclusiones y recomendación respectivas al diseño del sistema.

PALABRAS CLAVES: Raspberry Pi 3B, tecnología WIFI, video vigilancia, sistema de seguridad, Smartphone

ABSTRACT

The present project, based on the design of a security system of vehicle video surveillance, through the control in a Smartphone application, the system is activated by the bluetooth communication to give notice of the state of the doors if they are open, additionally capture the noise by means of a sound sensor which activates a camera that sends an image by means of a USB 3G modem towards the electronic mail, of this form it allows to visualize the events through means of an Android application in a Smartphone.

The analysis in development implies a study of the components necessary in hardware and software that most are coupled to the vehicular video surveillance system, ie the Raspberry Pi 3B card is analyzed, which has the capacity to support the transmission of sequence of images using WIFI technology and connected to a USB 3G module to work with a noise sensor, the proposal details the need and the materials used in the project with which it was carried out, it analyzes separately each part of the software and hardware in order to be all understandable and dynamic, the results and analysis of the tests carried out validate the behavior and the correct operation of the video surveillance system. These results show the potential impact of the use of these technologies and open new questions regarding research about processes and applications in society. Finally, the respective conclusions and recommendations for the design of the system are established.

KEYWORDS: Raspberry Pi 3B, WIFI technology, video surveillance, security system, Smartphone

INTRODUCCIÓN

1.1. Antecedentes de la situación objeto de estudio

Sistema antirrobo fabricado en España y directamente conectado al móvil. Unos sensores de movimiento instalados en el coche permiten a través de esta APP, acceder a la posición del coche e incluso bloquearlo desde el móvil en caso de que se te lo hayan robado. Sin embargo, otras informaciones más como el bloqueo central entre otros. (Proenium, 2016)

La llamada de emergencia automática realiza una marcación al número 112 sin la intervención del conductor, cuando detecta que se ha producido un accidente grave (por ejemplo, si dispara un airbag). Este Inteligente sistema también envía las coordenadas GPS del coche, fecha, hora y sentido de la circulación. (Bureau, 2016)

La alarma vehicular es un dispositivo que tiene por objetivo el bloqueo mediante la señal de una sirena y parpadeo de las luces en caso de un intento de acceso no autorizado al interior del vehículo, generalmente esta clase de sistemas cuentan con una serie de sensores que pueden producir que la alarma se accione, como por ejemplo: pulsadores de puertas, pulsador del capot, sensor de impacto, sensor de movimiento, etc., donde dichos dispositivos aumentarán o reducirán su número según las funcionalidades del sistema de alarma vehicular. (Pablo, 2012)

1.2. Planteamiento del problema

En los últimos años el robo de vehículos en el Ecuador se ha incrementa considerablemente. Entre los meses de enero del año 2015 a julio del 2017 se robaron 13271 vehículos a nivel nacional, esto representa un crecimiento de 17% contra el mismo periodo del año anterior. Principalmente en las ciudades más importantes como, Quito, Guayaquil y Cuenca (TELÉGRAFO, 2017). Esto debido a la gran cantidad de vehículos existentes en el mercado las casas comerciales promocionan vehículos que tiene todo tipo de accesorios radios con pantalla táctil, asientos de

cuero, accesorios lujosos entre otros. Estos considerados como de gama media y gama alta.

El sector en la ciudad de Quito con mayor incide de robos de vehículos se diferencia de acuerdo con el sector (COMERCIO, 2016):

- Eugenio Espejo
- Eloy Alfaro
- La Delicia
- Quitumbe
- Manuela Sáenz
- Los Chillos
- Calderón
- Tumbaco

La cantidad de vehículos robados es mayor en los siguientes horarios (Mushoq, 2017),

- Mañana 6:00 AM a 12:00 PM número de vehículos 145 - 200
- Tarde 12:00 PM a 4:00 PM número de vehículos 210 - 275
- Noche 4:00 PM a 12:00 AM número de vehículos 305 - 414
- Madrugada 12:00 AM a 6 AM número de vehículos 80 - 120
- Otro número de vehículos 0 - 40

Muchos de estos vehículos son llevados a otros países cercanos como Colombia y Perú. Donde mediante la suplantación de datos de número de serie o suplantación los comercializan sin problema alguno.

Debido a esto al momento se tiene en el mercado empresas de seguridad vehicular que brindan servicios de rastreo satelital como Sherlock de Hyundai, Chevystar de Chevrolet, Hunter entre otras. (CHEVROLET, s.f.) Dichas empresas tienen servicios según la necesidad del usuario, y sus costos son variantes. Un servicio básico sobrepasa los 670 dólares, una renovación anual con un costo mayor de 320 dólares y atados asistir a un mantenimiento obligatorio. (ECUADORINMEDIATO, 2017)

Así también existen sistemas de alarmas como Némesis, Electrón, etc., No muy sofisticados que permiten tener seguridad en el vehículo para inmovilizarlo, por un costo módico inicial de (87 dólares) según sus prestaciones. (Ecuador, 2017)

Como otro medio alternativo de seguridad, han optado por las láminas de seguridades americanas y nacionales. Según sus características con un costo medio de (315 dólares) en adelante, las mismas que se instalan en los vidrios del vehículo para así de alguna forma incrementar la seguridad del vehículo. (Ecuador, 2017)

1.3. Formulación del problema

Este proyecto abarca el sistema de vigilancia para automóviles, el diseño con lleva la instalación de una cámara, un sensor de ruido de impacto en el interior del vehículo, un módulo de ubicación GPRS, un control de apertura de seguros, sensores en las puertas, un bloqueo central de bloque y una aplicación para un Smartphone que permita la visualización de imágenes en tiempo real del interior del vehículo, la emisión de alertas de robo y la conexión con él usuario.

La investigación que se realiza en este proyecto será del tipo experimental porque consiste en la manipulación de una (o más) variables no comprobadas en condiciones rigurosamente controladas, con el fin de describir la causa por la cual se produce una situación o acontecimiento particular.

1.4. Justificación

Al implementar el sistema de seguridad antirrobo vehicular con el uso de la tarjeta Raspberry Pi 3B con tecnología 3G, se pretende mejorar el nivel de seguridad del vehículo, conocer la ubicación del mismo, tener control sobre los seguros de las puertas y detectar si se produce ruptura de los vidrios mediante la visualización de los eventos en una aplicación para Smartphone.

Mediante la instalación del sistema de seguridad antirrobo vehicular, se disminuye el costo de instalación de 630 dólares a 250 dólares y de un

mantenimiento con un costo de 320 dólares a 75 dólares de los dispositivos que lo conforman.

El usuario puede tener gestión del sistema de seguridad antirrobo vehicular, mediante la aplicación Android la cual permite el acceso a los eventos y al dispositivo como tal.

Al implementar el sistema prototipo de seguridad vehicular se integrarán todos los dispositivos y se obtiene como resultado un sistema completo de multitareas que será muy eficiente y seguro.

La instalación del presente proyecto permite que el dispositivo sea independiente al sistema propio del vehículo, para evitar conflictos en los procesos y acciones como, control de seguros, estado de puertas, localización, bloqueo central y alimentación.

De esa manera se disminuirá de forma considerable en un índice de 4% el robo de vehículos, más aún el lote de vehículos que son transportados a otros países para su venta. (Aráuz, 2017)

Finalmente, su costo es menor debido a que los materiales son de adquisición nacional de alta resistente y de costo menor al 25% del valor total frente al mercado.

1.5. Objetivos

1.5.1 Objetivo general:

Implementar un sistema de seguridad antirrobo para vehículos, mediante Raspberry Pi 3B con tecnología 3G con un control autónomo y visualización en una aplicación para Smartphone.

1.5.2 Objetivos específicos:

- Analizar el incremento de la delincuencia y al alto índice de robos a vehículos en el Ecuador, su incidencia y la factibilidad técnica, económica para el desarrollo de la propuesta.
- Diseñar un sistema antirrobo vehicular con Raspberry PI que permita, mediante radio frecuencia tener el control de apertura y

bloqueo de los seguros de las puertas de forma manual o automática, con tecnología Bluetooth y visualizar el estado de las puertas en una interfaz desarrollada para Android, que admite el acceso mediante una clave.

- Implementar un sistema electrónico que permita la comunicación entre módulo de relés, módulo GPRS, módulo Raspberry Pi 3B, módulo sensor de ruido y módulo de control de bloque vehicular, para así con este tipo de elementos, permitir la transmisión bidireccional en él envió de datos. Mediante tecnología bluetooth, tecnología 3G y radio frecuencia permitir el envío y recepción de información.
- Realizar pruebas y sus respectivas acciones correctivas; que permitan comprobar el correcto funcionamiento del sistema antirrobo vehicular y la interfaz desarrollada para Android.

1.6. Descripción de los capítulos

A continuación, en el documento se describe el desarrollo del proyecto que se divide en cuatro capítulos diferentes.

En el capítulo 1, se describe el hardware, así como también el software, la tarjeta utilizada tiene sus periféricos, pines de entrada y de salida, tipo de procesador, capacidad de almacenamiento. Adicionalmente se describirá las características de los componentes como las cámaras, sensores de ruido de impacto, modem USB 3G y modem USB WIFI, módulo GPS, control remoto con tecnología RFID, entre otras.

En el capítulo 2, se elabora la propuesta, una descripción de la arquitectura del sistema y el diseño del mismo, es decir de todos los subsistemas de adquisición datos, envío de los mismos hacia un servidor remoto y finalmente con el desarrollo de la interfaz gráfica que permite visualizar los resultados.

En el capítulo 3, se desarrolla la implementación del sistema, se presenta todas las pruebas realizadas para asegurar el correcto funcionamiento del sistema prototipo así asegurando que el sistema no presente errores significativos y se indica los análisis de resultados.

Finalmente se presentan las conclusiones obtenidas durante el desarrollo del proyecto, así mismo se incluye las recomendaciones que podrán aportar a futuros proyectos relacionados.

CAPÍTULO I

FUNDAMENTACIÓN TEÓRICA

2.1 Antecedentes investigativos

Como antecedentes investigativos, se toma en cuenta trabajos que fueron desarrollados con enfoque principalmente a los sistemas de video vigilancia o de monitoreo, ha evolucionado notablemente en los últimos años. Desde los sistemas análogos que tuvieron su máximo representante en los CCTV (circuito cerrado de televisión), hasta los modernos sistemas digitales con transmisión inalámbrica.

El objetivo de todos estos sistemas es la supervisión, vigilancia, el control y eventualmente el registro de las actividades dentro de un espacio o ambiente en general. Los sistemas de vigilancia a través de video existen desde hace más de tres décadas, pero estos últimos años estos sistemas de video vigilancia avanzaron mucho debido principalmente a tres factores: el desarrollo tecnológico, la demanda de mayores niveles de seguridad y la evolución de técnicas de análisis de video. (Mata & Javier, 2010)

El esquema de un sistema CCTV clásico está descrito en la Figura 1, a partir de estos sistemas básicos de video vigilancia o de monitoreo por imágenes, estos sistemas evolucionan para adaptarse a las nuevas.

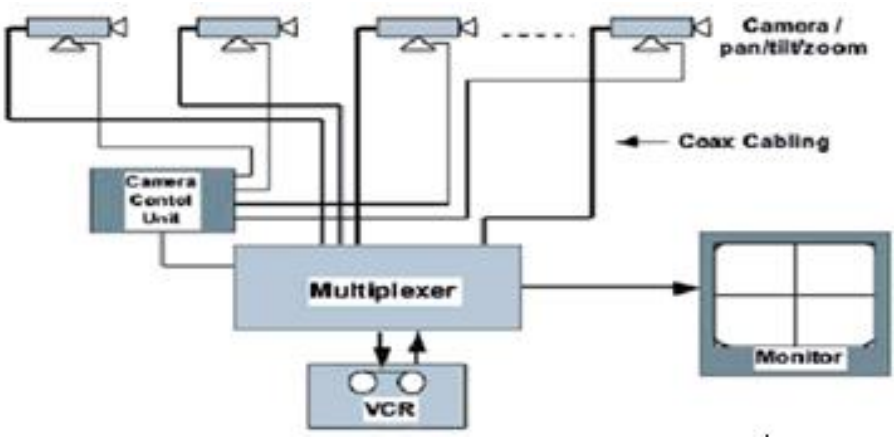


Figura 1. Esquema de un sistema CCTV
Fuente: (Siemon, 2016)

En la Figura 2, se observa el diagrama de video digital sobre IP (protocolo de internet), estos equipos electrónicos que manejan actualmente tráfico lP, son parte integral de los sistemas de vigilancia, puesto que los videos se almacenan en formato digital y pueden ser vistos en cualquier lugar de la red, lo que obliga el desempeño de nuevas capacidades de seguridad para los archivos administrativos como parte de las políticas de seguridad de red, un sistema así no solo es fácil de implementar, sino también es extremadamente versátil y las redes no son sobrecargadas con otro protocolo. (Cachiguango, 2010)

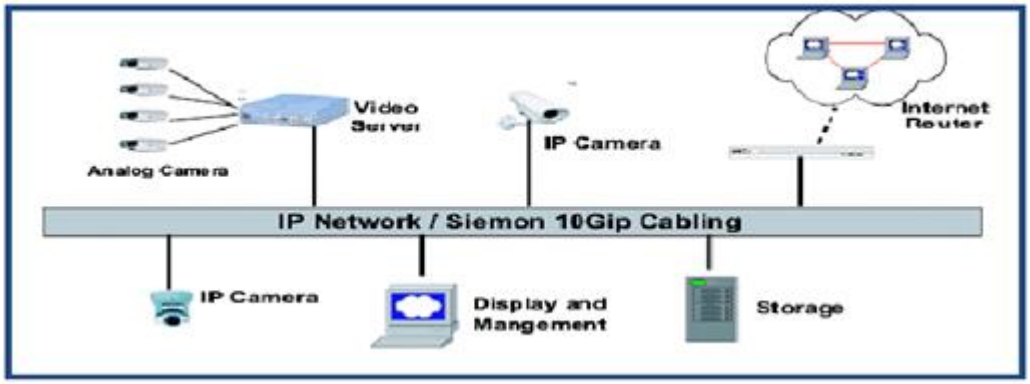


Figura 2. Diagrama de video digital sobre IP

Fuente: (Siemon, 2016)

Con los antecedentes descritos anteriormente para este proyecto se utiliza la comunicación de los componentes que están constituidos por bloques de transmisión y recepción como se observa en la Figura 3, adicionalmente se estudia y diseña el acople de los elementos físicos que son compatibles a la infraestructura de un software y hardware libre como por ejemplo Linux y Raspberry Pi 3B respectivamente.

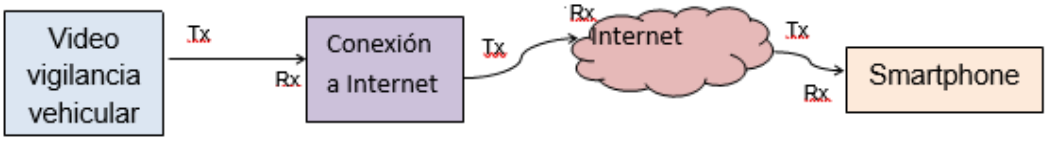


Figura 3. Diagrama de bloques transmisión y recepción

Fuente: (Elaborado por el Autor)

A continuación, se realiza el estudio y requerimientos de los diferentes componentes del sistema de video vigilancia.

2.2. Especificaciones Técnicas del Sistema

En esta sección se realiza el estudio de los diferentes componentes tanto del software como del hardware para el sistema de seguridad de video vigilancia.

2.3. Estudio del Hardware

Para analizar los diferentes componentes que constituyen la implementación de sistema, se describe a continuación las tecnologías, las funcionalidades del CPU o tarjeta madre, como el módulo SIM 908, el módulo GPS, control remoto RFID, cámara y sensor de ruido que ayudará al acople del sistema.

2.3.1. Plataforma Raspberry Pi

Existen varios modelos de tarjetas Raspberry Pi, entre las cuáles las versiones más conocidas son la versión 2 B y la versión 3 B, la plataforma consta de contactos de entrada/salida de uso general (GPIO), conectividad mejorada (4 puertos USB en lugar de 1 puerto USB que tiene el modelo A. En la Tabla 1 se indica un resumen de los modelos y las características generales de las tarjetas Raspberry. (Faican, 2016)

Tabla1: **Modelos y especificaciones técnicas, tarjetas Raspberry Pi**

Modelos de Raspberry Pi	Raspberry Pi A	Raspberry Pi 2 B	Raspberry Pi 3B
SoC (Chip)	Broadcom BCM 2835	Broadcom BCM 2835	Broadcom BCM 2836
CPU	ARM1176JZF-S a 700 MHz	ARM1176JZF-S a 700 MHz	ARM Cortex-A7 cuatro núcleos a 900 Mhz
GPU	Video Core IV a 250 Mhz	Video Core IV a 250 Mhz	Video Core IV a 250 Mhz
Memoria RAM	256 MB a 400 Mhz	512 MB a 400 Mhz	1 GB a 450 Mhz

Entradas de vídeo	Cámara CSI	Cámara CSI	Cámara CSI
Salidas de vídeo	HDMI 1.4, conector RCA	HDMI 1.4, conector TRRS	HDMI 1.4, conector TRRS
Salidas de audio (HDMI)	Jack de 3.5 mm (auriculares)	Jack de 3.5 mm (auriculares)	Jack de 3.5 mm (auriculares)
Conectores USB 2.0	1	4	4
Tarjetas de almacenamiento	SD	microSD	microSD
Conexión a red	No	Ethernet 10/100 Mbit/sg	Ethernet 10/100 Mbit/sg
Interfaz periféricos (GPIO)	20	40	40
Tamaño	85.6 x 56.5 mm	85.6 x 56.5 mm	85.6 x 56.5 mm
Peso	45 gramos	45 gramos	45 gramos
Consumo	1.5W/5V	3W/5V	5V

Nota. Fuente: (Premkumar, 2015)

2.3.2. Raspberry Pi 3B Modelo B

Tanto la tarjeta 2 B como la tarjeta 3 B tienen una idéntica distribución de sus componentes de manera que cualquier contenedor o aditamento es compatible para las dos tarjetas de 40 pines GPIO, las dos tarjetas tienen salida de video a 1080p, lector de tarjetas micro SD, puerto de red, interfaz de cámara CSI y la interfaz de pantalla DSI (Jon Holton, 2016).

Para el sistema de video vigilancia del proyecto se van a utilizar la tarjeta Raspberry Pi 3B modelo B, la Figura 4 muestra un esquema general de la plataforma, así como los periféricos y los pines de entrada salida conocidos como GPIO.

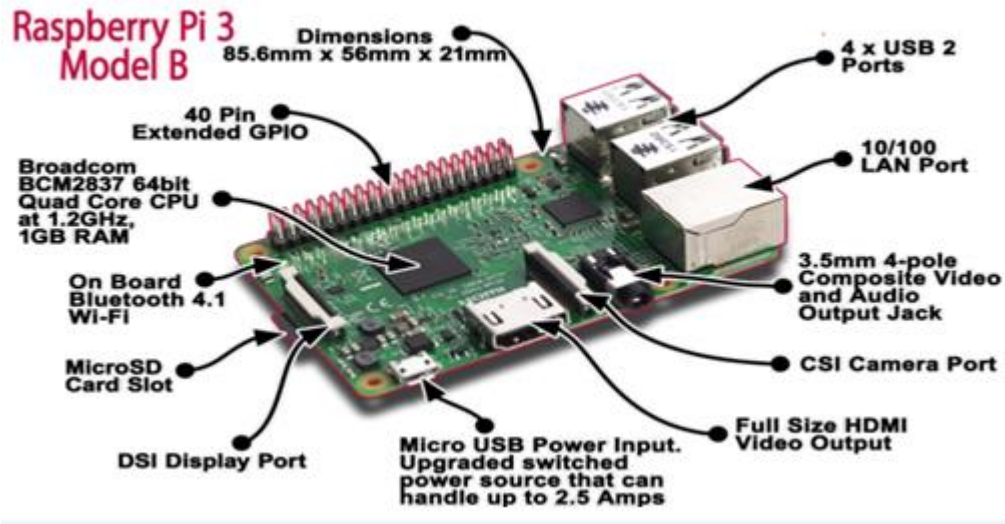


Figura 4. Tarjeta Raspberry Pi 3B modelo B

Fuente: (Princy, 2015)

Una característica importante de la tarjeta Raspberry Pi modelo 3B es la fila GPIO de pines que está ubicada a un costado, dicha fila tiene pines los cuales permiten tener una interfaz física entre la tarjeta pi y el mundo exterior. En su forma más simplificada se puede resumir como un grupo de interruptores que se pueden encender o apagar al introducir información a la tarjeta (input) o viceversa, así también se pueden encender o apagar de esta forma la mostrar información a la tarjeta (output), de esta manera se pueden programar estos pines para interactuar en formas asombrosas con el mundo exterior. (Krit Janard, 2016)

En la Figura 5, se muestra los pines de entrada y salida conocido como GPIO de propósito general de la tarjeta Raspberry Pi 3B. (Brown, 2016)

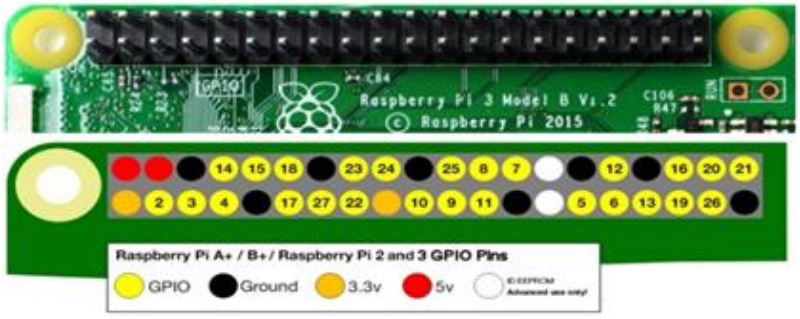


Figura 5. Esquema de pines de la Raspberry Pi 2 B

Fuente: (Brown, 2016).

2.3.3. Arduino mega 2560

Arduino es hardware libre basado en una placa con un microcontrolador de la familia Atmel AVR y un entorno de desarrollo que permite la creación de programas. Mediante el uso de Arduino se pueden realizar proyectos electrónicos, ya que el dispositivo tiene la capacidad de leer información de temperatura, presión, humedad, entre otros a través de sus pines de entrada y así controlar elementos del entorno que le rodean (luces, motores y otro tipo de actuadores).

El microcontrolador controla el procesamiento del Arduino se programa mediante el lenguaje de programación Arduino (basado en *Wiring*) y el entorno de desarrollo Arduino (basado en Processing). Finalmente se puede mencionar que el software es gratuito y se encuentra disponible en cualquier página web de Arduino. (Arduino, 2014)

En el mercado existe variedad de placas Arduino, cuya elección depende de la necesidad del proyecto que se quiere realizar. La Figura 6 muestra al Arduino Mega 2560, cuyas principales características son las siguientes: (Simbaña, 2015)

- Posee 54 pines digitales que funcionan como entradas/salidas.
- Su cristal oscilador es de 16 MHz.
- Se puede conectar a la computadora a través de una conexión USB (Universal Serial Bus).
- Posee un botón de reset y una entrada para la alimentación de la placa.

Figura 6. Arduino Mega 2560

Fuente: (Store, 2017).

2.3.3.1 Pines especiales

En la Tabla 2, se describen los que tiene el módulo Arduino mega 2560.

Tabla 2: **Pines especiales del Módulo Arduino.**

Serie: 0 (RX) y 1 (TX), Serie 1: 19 (RX) y 18 (TX); Serie 2: 17 (RX) y 16 (TX); Serie 3: 15 (RX) y 14 (TX).	**Rx (recepción) y Tx (transmisión).**
Interrupciones Externas: 2 (interrupción 0), 3 (interrupción 1), 18 (interrupción 5), 19	Los pines se pueden configurar para lanzar una interrupción en un valor LOW (0V), en flancos de subida o bajada (cambio de LOW a HIGH (5V) o viceversa), o en cambios de
PWM	Pines del 0-13 salida (Modulación por ancho de pulso) valores entre 0 a 255.
SPI: 50 (SS), 51 (MOSI), 52 (MISO), 53 (SCK)	Pines proporcionan comunicación SPI, usa la librería SPI.
LED	Integrado al pin digital 13, valor HIGH (5V) ON y valor LOW(0V) OFF.
16 entradas analógicas	Cada una de ellas proporciona una resolución de 10bits (1024 valores).
I2C: 20 (SDA) y 21 (SCL)	Soporte para el protocolo de comunicaciones I2C (TWI) usa la librería
AREF	Voltaje de referencia para las entradas analógicas.

Nota. Fuente: (ARDUINO MEGA 2560, 2017)

2.3.3.2 Características técnicas

En la Tabla 3, se describen las características que tiene el módulo Arduino mega 2560.

Tabla 3: **Características Técnicas Módulo Arduino Mega 2560.**

Microcontrolador	**ATMega8U2**
Voltaje de operación	5V
Voltaje de entrada (Recomendado)	7V a 12V
Voltaje de entrada (Límite)	6V a 20V
Pines para entrada - salida digital	54
Pines de salidas	14 se usan como salida de PWM

Corriente continua para las entradas	40 mA
Salida de alimentación a 3.3V	50 mA
Memoria Flash	256 KB (el bootloades ocupa 8 KB)
SRAM	8 KB
EEPROM	1 KB
Velocidad de reloj	16 MHz

Nota. Fuente: (Tinajero, 2014)

2.3.4. Cámara de Video Raspberry Pi 100003

La cámara de marca Raspberry Pi modelo 100003, tiene una resolución de 5 megapíxeles de imagen quieta: 2592 x 1944, vídeo máxima: 1080p, frecuencia máxima de cuadro: 30fps y soporta Windows XP/Vista/7/8, Linux y Mac OS X. (Marko Viitanen, 2015)

Se utiliza la cámara de video Raspberry Pi 100003, la cual va conectada al puerto de la cámara CSI de la tarjeta Raspberry Pi 3B modelo B como se muestra en la Figura 7, con la finalidad de capturar las imágenes que serán guardadas en la tarjeta micro SD y posterior ser enviadas por secuencias a la web.

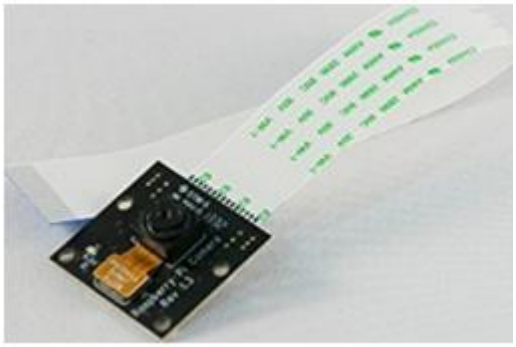


Figura 7. Cámara Raspberry Pi

Fuente: (PI-Supply, 2017)

2.3.5. Sensor de Ruido de Impacto

El sensor consta de tres pines los mismos que van conectados Pin 1 a VCC, Ping 3 a GND y el Pin 2 a la señal de canal individual, la misma que es digital de nivel bajo y tiene un led indicador cuando se activa la señal. Este módulo permite la conexión de un micrófono Electret hacia los pines

digitales de un microcontrolador. El nivel de detección sonoro puede ser modificado mediante un potenciómetro para censar la intensidad del sonido así validar su nivel se sensibilidad de recepción. (Teslabem, 2017)

Se analiza el sensor de ruido de impacto, el cual se encuentran conectado a los pines de propósito general (GPIO) de las tarjetas Raspberry Pi 3B modelo B como muestra la Figura 8, con la finalidad de detectar ruido y que pueda ser activada la cámara.

Figura 8. Sensor de sonido
Fuente: (Elcajondeardu, 2015)

2.3.6. Modem USB 3G

En la Figura 9, se muestra el modem USB 3G Huawei E3532, cuyas especificaciones son: interfaz USB 2.0; opera en las frecuencias GSM 850, 900, 1800, 1900 GHz y la frecuencia UMTS de 21000 GHz, cabe aclarar que estas son las frecuencias de 3G y 3.5G para Ecuador; y, tiene las siguientes velocidades máximas de transferencia de datos: HSDPA 7.2 Mbps, HSUPA 5,76 Mbit/s, EDGE 236,8 KBit/s y GPRS de 57,6 KBit/s. (Webantics, 2017)

Figura 9. Modem USB 3G Huawei
Fuente: (4gltemall, 2017)

2.3.7. Módulo Relés

Es una placa que consta de 4 relés, que funcionan con un voltaje de 5V, son capaces de manejar una carga de hasta 10 Amperes en 250 Voltios, tiene indicadores led por cada una de las entradas, convenientemente separadas mediante elementos optoacopladores de las entradas. (NOREN, 2017)

Figura 10. Modem USB 3G Huawei

Fuente: (NOREN, 2017)

En la Figura 10, se observa las entradas de alimentación (VCC), tierra (GND) y las entradas de la señal de control IN1, IN2, IN3 y IN4. Se debe tener en cuenta que para evitar un estado indeterminado de las entradas se debe conectar una resistencia de 10kOHM de forma independiente entre cada una de las señales y VCC.

2.3.8. Módulo GPRS

Si bien existen varios modelos de módems o módulos que permiten el envío de información a través de la red GPRS, los más fáciles de conseguir en el mercado local son: el módem Multitech GSM/GPRS MTCBA-GF2 y el SIM900 GSM/GPRS Shield.SIM900 GSM/GPRS Shield.

2.3.8.1 El SIM908 GSM/GPRS Shield

Es totalmente compatible con la mayoría de dispositivos Arduino, ya que se lo utiliza para desarrollar proyectos o aplicaciones como: control remoto, estaciones meteorológicas remotas, sistemas de rastreo de automóviles,

entre otros. La Figura 11 muestra el aspecto físico del SIM908 GSM/GPRS Shield. (Simbaña, 2015)

Figura 11. Aspecto físico del SIM908 GSM/GPRS Shield

Fuente: (Arduino, 2015)

Las especificaciones técnicas del módulo SIM908 GSM/GPRS Shield se presentan a continuación en la Tabla 4.

Tabla 4: **Especificaciones técnicas del SIM900 GSM/GPRS Shield**

ESPECIFICACIONES TÉCNICAS	
Voltaje de operación	5V - 20V
Comunicación	UART
Bandas de frecuencia	850/900/1800/1900MHz
Multi-GPRS	Slot clase 10/8
Estación móvil GPRS	Clase B
Clase 1	1W a 1800/1900MHz
Clase 4	2W a 850/1900MHz
Pila embebida TCP/UDP	Carga de datos a un servidor web
Bajo consumo de energía	1.5mA (modo sleep)
Temperatura de operación	-40ºC a +85ºC

Fuente: (Simbaña, 2015)

Después de analizar las especificaciones técnicas y características de los dos dispositivos mencionados anteriormente, se selecciona el módulo SIM908 GSM/GPRS Shield debido a los siguientes factores:

- Se puede tener una mejor presentación estética del sistema ya que se coloca directamente encima del dispositivo Arduino Mega 2560, mientras que con el módem Multitech GSM/GPRS MTCBA-GF2 es necesario utilizar cables para conectarse al dispositivo Arduino Mega 2560.
- Presenta menor costo con respecto al módem Multitech GSM/GPRS MTCBA-GF2.

2.3.8.2. Fuentes de Error GPS

El (Instituto de Automática Industrial, 2016) da a conocer las fuentes de error más importantes que actualmente afectan significativamente a las medidas que se realizan con el GPS, las cuáles se detallan a continuación:

- Perturbación ionosférica: La ionósfera es la parte de la atmósfera terrestre ionizada permanentemente por la radiación solar, lo cual repercute en modificar la velocidad de las señales de radio que viajan por este medio.

- Fenómenos meteorológicos: La tropósfera, que es en ella en donde se desarrollan todos los procesos meteorológicos y climáticos, la velocidad de las señales electromagnéticas disminuyen debido a la cantidad de vapor de agua que en este medio se genera.

2.3.8.3. Sistema global para las comunicaciones móviles (GSM)

Se define GSM a la Red del Sistema Global de Telefonía como aquel servicio portador constituido por todos los medios de transmisión y conmutación necesarios que permiten enlazar a voluntad dos equipos terminales móviles mediante un canal digital que se establece específicamente para la comunicación y que desaparece una vez completada la misma. (Rodríguez, 2016)

El Sistema Global para Comunicaciones Móviles originalmente fue desarrollado como estándar europeo para la telefonía móvil digital, se ha

convertido en el sistema móvil de uso más difundido en el mundo. Opera en las frecuencias de 900 a 1800 MHz en Europa, Asia y Australia. Así también en la frecuencia de 1900 MHz en Norteamérica y Latinoamérica. (Rodríguez, 2016)

2.3.8.4. Arquitectura de la Red GSM

La Arquitectura GSM está compuesta de los siguientes sistemas:

- Subsistema Radio (RSS, Radio SubSystem): Es el encargado de cubrir la comunicación que ocurre entre las estaciones móviles (MS) y las estaciones base (BTS).
- Subsistema de estaciones base (BSS): Este se encuentra contenido dentro de la parte del subsistema de radio, y está formado por los elementos que se muestran a continuación:
    - BTS (Base Transceiver Station): emisor, receptor y antena. Este se encarga de procesar los subsistemas de radio.
    - BSC (Base Station Controller): Handover, control de las BTS, mapeo de canales radio sobre los canales terrestres. (Velasco, 2005)

2.3.9. Control remoto RF module YK04

El radiocontrol emite la frecuencia codificada de los canales A, B, C y D, esto permite tener control sobre 4 variables. Se compone de (Emisor-receptor) utilizan los C.I PT2262/PT2272 que realizan un sistema de control remoto inalámbrico básico como se muestra en la Figura 12. Se utilizan normalmente en sistemas económicos, debido a que utilizan códigos fijos y no emiten la información encriptada. (Tecnologia-informatica, 2014)

Son comúnmente usados en controles de garajes (puertas), control de ventiladores, alarmas sencillas, juguetes entre otros.

Figura 12. Control RF YK04

Fuente: (Faranux, 2016)

2.3.10. Tarjeta Micro SD

En la Figura 13, se visualiza el logo de la tarjeta micro SD ScanDisk. Tiene un almacenamiento de 32 GB su velocidad de transmisión de 10 MB/s la cual permite almacenar gran cantidad de información. Se la utiliza para cargar el sistema operativo en la tarjeta Raspberry Pi 3B así permitiendo actualizar las librerías en la tarjeta y tener una conexión óptima de los dispositivos que controla. (Sandisk, 2012)

Figura 13. Scandisk

Fuente: (Corporation, 2017)

2.3.11. Regulador de Voltaje

Se ha implementado al sistema prototipo un regulador de voltaje como se muestra en la Figura 14, su función es regular el voltaje mediante un circuito electrónico, mediante un potenciómetro el cual permite variar los valores de voltaje, para alimentar a la tarjeta Raspberry Pi 3B con un voltaje fijo de DC, para evitar una sobre carga y dañar el dispositivo. (Cuadros, 2016)

Figura 14. Regulador de voltaje.

Fuente: (Princy, 2015)

2.3.12. Batería 12V a 7Ah

En la Figura 15 muestra una batería recargable que proporciona una fuente de alimentación alternativa al sistema, está se conecta a la batería principal del vehículo en paralelo con un diodo de 3 Amperios para protección, su duración es de 12 horas. Esta batería es la encargada de alimentar de energía eléctrica a todo el sistema, tiene un voltaje de 12 Voltios a una corriente continua con un amperaje de 7 Amperios.

Figura 15. Batería de respaldo.

Fuente: (Productosled, 2015)

2.4. Estudio del Software

A continuación, se describe la plataforma en la cual se va a desarrollar del proyecto, así como también los lenguajes de programación que utiliza esta tarjeta.

2.4.1. Sistema Operativo para la Raspberry

La tarjeta Raspberry Pi usa en uno de los sistemas operativos basados en el núcleo Linux Raspbian, una distribución optimizada para el hardware de

Raspberry Pi, el lanzamiento de este sistema fue durante julio de 2012 como se muestra en la Figura 16.

Slackware ARM (también llamada ARMedslack) versión 13.37 y posteriores arranca sin ninguna modificación. Los 128-496 MB de memoria RAM disponible en la Raspberry Pi, cubren los necesarios 64 MB de RAM para arrancar esta distribución en sistemas ARM y i386 (es un microprocesador con arquitectura de 32 bits) sin usar interfaz gráfica (el administrador de ventanas Fluxbox que funciona bajo X Window System requiere 48 MB de memoria RAM adicional). (Faican, 2016)

La GPU (Graphics Processor Unit), es la unidad de procesamiento gráfico, se asemeja a una CPU. Se puede acceder a través de una imagen del firmware de código cerrado, llamado blob binario, que se carga al arrancar desde la tarjeta SD. El blob binario está asociado a los drivers Linux que también son de código cerrado, esta aplicación hace llamadas a las librerías de tiempo de ejecución que son de código abierto en el kernel de Linux. La API (Application Programming Interface) del driver del kernel es específica para estas librerías (Sevilla, 2016).

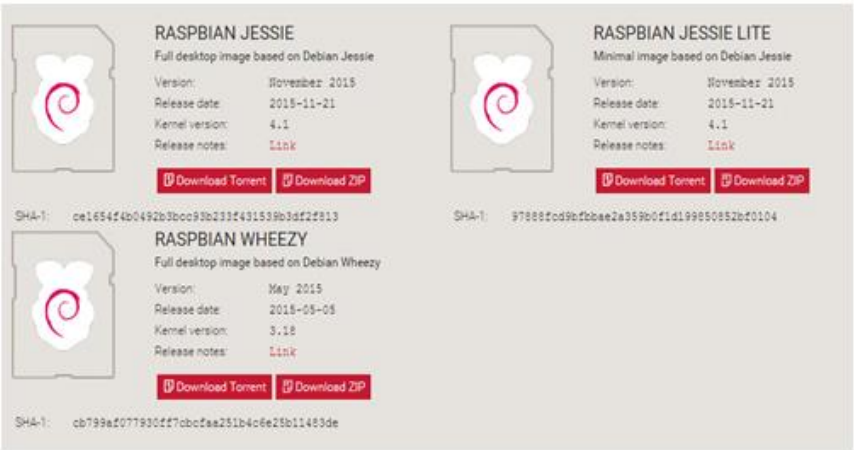


Figura 16. Opciones de descarga del Sistema Operativo
Fuente: (RASPBIAN, 2016).

2.4.2. Lenguajes de programación para la Raspberry Pi

A continuación, se muestran los principales lenguajes de programación utilizados para desarrollar proyectos en el dispositivo Raspberry Pi. Tras el

análisis de todos ellos, se tomará la decisión de cuál es el óptimo para el desarrollo del presente trabajo.

2.4.2.1. Ruby

El lenguaje Ruby como se muestra en la Figura 17, presentado en 1995 es un lenguaje de programación reflexivo y orientado a objetos. Combina una sintaxis inspirada en Python y Perl con características de programación orientada a objetos. Se puede considerar como un lenguaje de programación interpretado. Cabe destacar que su implementación oficial es distribuida bajo una licencia de software libre. (Gómez, 2011)

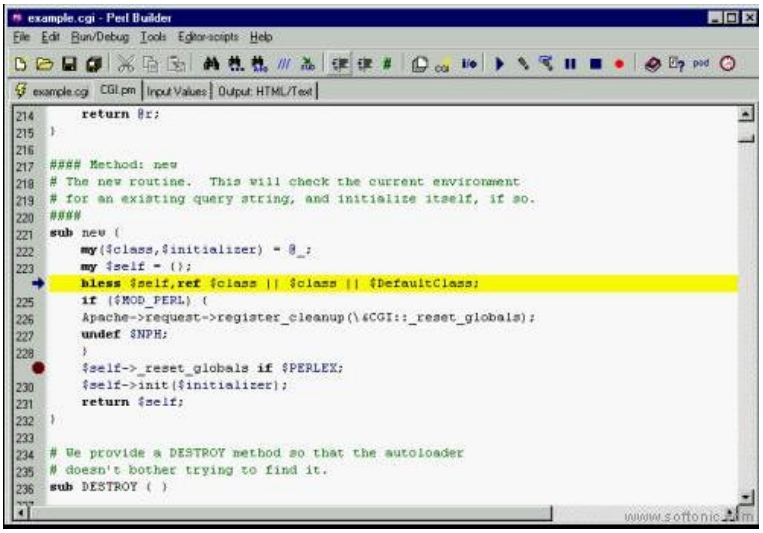


Figura 17. Entorno del lenguaje Ruby

Fuente: (Perez, 2011)

2.4.2.2. Lenguaje C

C es un lenguaje de programación orientado a la implementación de Sistemas Operativos, concretamente Unix. Es apreciado por la eficiencia del código que produce y es el lenguaje de programación más popular para crear software de sistemas, aunque también se utiliza para crear aplicaciones. El primer estándar del lenguaje C se produjo con ANSI X3.159-1989 como se muestra en la Figura 18. Posteriormente, en 1990, fue ratificado como estándar ISO (ISO/IEC 9899:1990. (Vele, 2016)

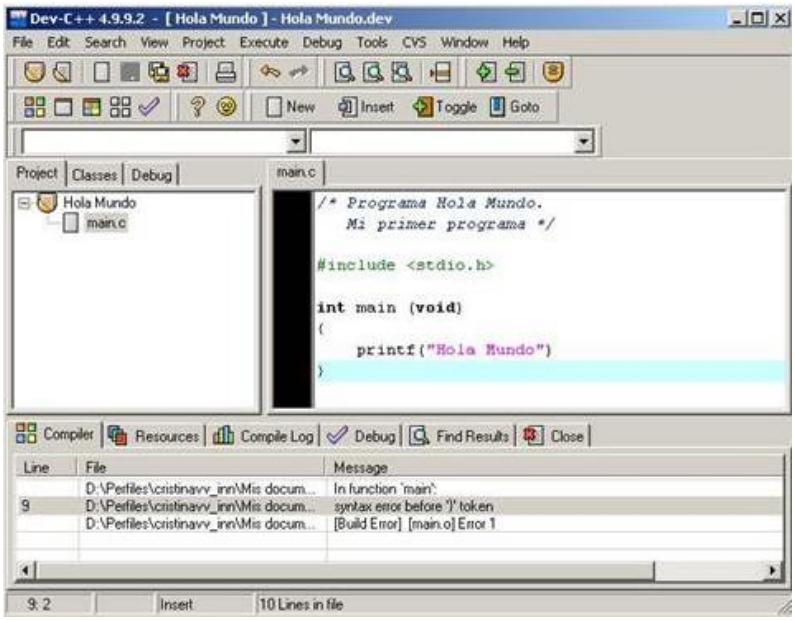


Figura 18. Entorno del lenguaje C

Fuente: (Torrelaguna, 2009)

2.4.2.3. Python

En la Figura 19, muestra el lenguaje de programación interpretado phyton cuya filosofía hace hincapié en una sintaxis que favorezca un código legible. Se trata de un lenguaje de programación multiparadigma ya que soporta orientación a objetos y programación estructurada. Es administrado por la Python Software Foundation y posee una licencia de código abierto denominada Python Software Foundation License1 la cual es compatible con la Licencia pública general de GNU a partir de la versión 2.1.1 e incompatible en ciertas versiones anteriores. (Orbegozo, 2015)

Tras el estudio de los lenguajes de programación anteriormente mencionados se escoge para el desarrollo del presente trabajo el denominado Python debido principalmente a:

- Dispone de las características necesarias para llevar a cabo el trabajo que se desarrolla.
- Su lenguaje se considera como familiar en el Máster en Geomática y Geo información, ya que es un lenguaje de programación principal empleado durante el proyecto.

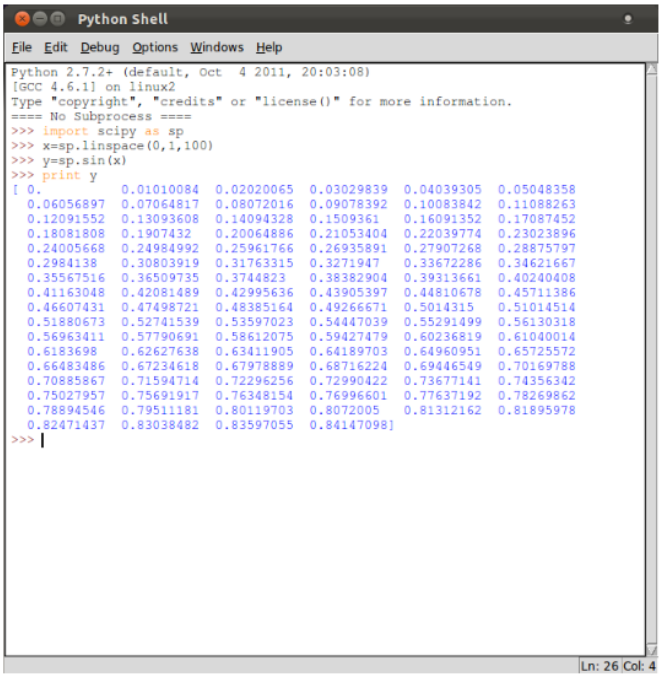


Figura 19. Entorno del lenguaje Python

Fuente: (Herrera & Sánchez, 2013)

2.5. Programas para programar Arduino

El Arduino tiene su software propio con lenguaje de programación y por supuesto es software libre, su lenguaje de programación está basado en C++. Arduino proporciona un software consistente en un entorno de desarrollo IDE (Entorno de Desarrollo Integrado) que implementa el lenguaje de programación de Arduino y el bootloader ejecutado en la placa. La principal característica del software de programación y del lenguaje de programación es su sencillez y facilidad de uso. (Crespo, 2016)

En la Figura 20 se muestra el entorno de programación compuesto por un conjunto de herramientas de programación el cual se encuentra empaquetado como un programa de aplicación; es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Además, incorpora las herramientas para cargar el programa ya compilado en la memoria flash del hardware. (Crespo, 2016)

Figura 20. Entorno Arduino 1.6.0

Fuente: (Crespo, 2016)

La aplicación Arduino es una plataforma de software para dispositivos móviles que incluye un Sistema Operativo y aplicaciones de base. Android es un conjunto de herramientas y aplicaciones vinculadas a una distribución Linux para dispositivos móviles. Android es una plataforma de código abierto para dispositivos móviles que está basada en Linux y desarrollada por Open Handset Alliance, se prevé que los primeros teléfonos con Android aparezcan en el segundo semestre de 2008 y compañías poderosas como LG, Motorola y HTC desarrollaron prototipos que incorporarán el Sistema Android.

Es un software para dispositivos móviles que incluye un Sistema Operativo, Middleware y aplicaciones de base. Los desarrolladores pueden crear aplicaciones para la plataforma se usa el SDK de Android. (Sandoval, 2015)

2.5.1. Versiones

Existen varias versiones del sistema operativo Android. El lanzamiento de la primera versión comercial (demo), Android versión 1.0, fue en el 2008. El

sistema operativo Android para móviles lo desarrollo Google y Open Handset Alliance. Desde su lanzamiento existen varias nuevas versiones con sus respectivas actualizaciones, de acuerdo con la necesidad.

Tabla 5: **Versiones de Android las más conocidas**

Nombre	Número de versión	Fecha de lanzamiento
Nougat	7.0 - 7.1	22 de agosto de 2016
Marshmallow	6.0–6.0.1	5 de octubre de 2015
Jelly Bean	4.1–4.3.1	9 de julio de 2012
Ice Cream Sandwich	4.0–4.0.4	18 de octubre 2011
Eclair	2.0–2.1	26 de octubre de 2009
Apple Pie	1.0	23 de septiembre 2008

Fuente: (Basterra Borello, 2017)

Estas actualizaciones típicamente corrigen fallos de programa y agregan nuevas funcionalidades. Las versiones de Android fueron desarrolladas bajo un nombre en clave y sus nombres siguen un orden alfabético como se detalla en la Tabla 5.

2.5.2. Características Especiales

En la Tabla 6, se muestra las características que tiene el sistema operativo Android.

Tabla 6: **Características Especiales de Android**

Código	Abierto
Núcleo basado	Kernel de Linux
Adaptable	A muchas pantallas y resoluciones
Utiliza	SQLite para el almacenamiento de datos
Navegador web	Basado en WebKit incluido
Soporte	Java y muchos formatos multimedia.
Soporte	HTML, HTML5, Adobe Flash Player, etc.
Incluye	Un emulador de dispositivos, herramientas para depuración de memoria y análisis del rendimiento del software.
Catálogo de aplicaciones	Gratuitas o pagas en el que pueden ser descargadas e instaladas (Google Play)

Bluetooth.	Comunicación bidireccional
Video llamadas.	Google Talk desde su versión HoneyComb,
Aplicaciones	Multitarea real

Fuente: (Basterra, 2012)

2.6. Aplicación Smartphone

2.6.1 App Inventor

App inventor es un sistema gráfico para desarrollar aplicaciones para Android como se muestra en la Figura 21, creándolas desde cero, aún sin tener conocimientos de programación, todo este proceso se reduce a enlazar diagramas de bloques que representan objetos o entidades en conjunto con otros que representan acciones, bien efectuadas por el usuario, bien por la aplicación, cuando se cumplen las condiciones que se definen durante el diseño. (González & García, 2015)

El diseño que se crea en la aplicación se lo carga de manera muy sencilla en el teléfono celular.

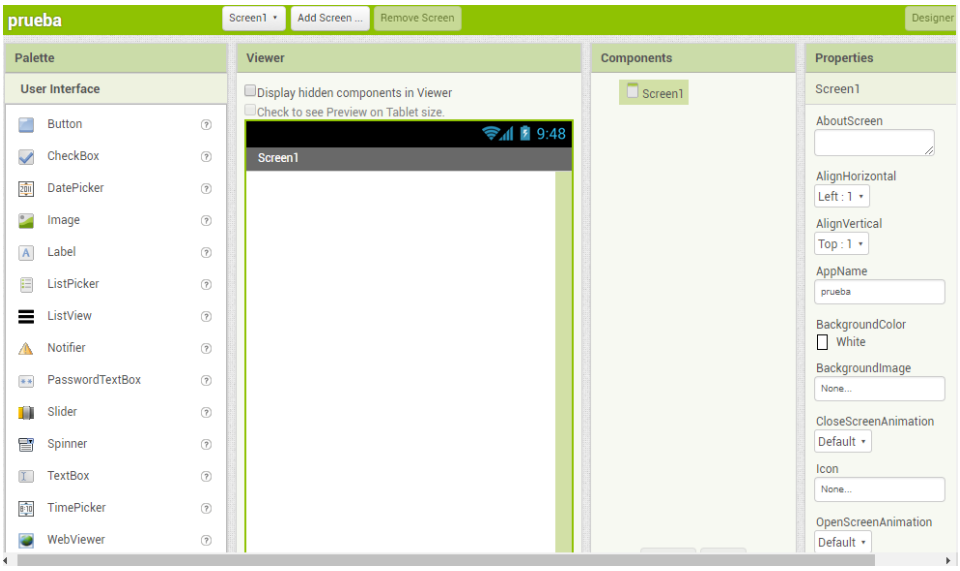


Figura 21. Entorno App Inventor

Fuente: (Sites, 2016)

Al final lo que se consigue es ejercitar la lógica, la inteligencia gráfica y funcional. Con esto se determina el uso de varios lenguajes de

programación en lo que no es preciso conocer su código, hay varios sitios donde podemos encontrar varios instaladores para la Raspberry Pi 3B. Tiene lógica si tiene en cuenta que se trata de desarrollar software para dispositivos que se manejan de forma gráfica y no textual. Con esto queda clara la utilidad del MIT App Inventor como instrumento de educación.

2.7. Metodología

La metodología es un instrumento que permite enlazar el sujeto con el objeto de la investigación. Sin la metodología es muy difícil llegar a la lógica que guiará al conocimiento científico.

En el presente trabajo se enfoca al conjunto de procedimientos lógicos a través de los cuáles se plantean los problemas científicos y se ponen a prueba el estudio y los materiales investigados, por tanto, los métodos a utilizarse son:

El método de investigación es un proceso de razonamientos que intentan no solamente describir los hechos sino también explicarlos y comprobarlos, sus etapas son:

- Percepción de un problema: en esta parte se encuentra el problema al cuál se enfoca el estudio.
- Identificación y definición del problema: es donde se observa para definir la dificultad del problema.
- Formulación de hipótesis: se busca las posibilidades de solución para el problema mediante previos estudios de los hechos.
- Adicionalmente en este método de investigación se realiza el estudio de las alternativas, y se describe cada una de las alternativas propuestas, así se describe los puntos fundamentales para posteriormente realizar el análisis adecuado y proseguir con la selección de la alternativa que mejor satisfaga al diseño.

- Para el método empleado hipotético-deductivo, la hipótesis propuesta se basa en las inferencias del conjunto de datos empíricos, de principios y leyes más generales. Lo que constituirá en la vía para arribar las conclusiones particulares a partir de la hipótesis y que después se puedan comprobar analíticamente.

- El método analítico empleado, se distinguen los elementos de la investigación y se procederá a revisar ordenadamente cada uno de ellos por separado, se utilizará este método a partir de la observación, encuestas y el análisis de algunos casos, se establecerán leyes y teorías.

Hay que tomar en cuenta que las operaciones a realizarse no son independientes una de la otra, el análisis de un objeto se realiza a partir de la relación que existe entre los elementos que conforman dicho objeto como un todo.

Se plantean las siguientes alternativas para el sistema de video vigilancia vehicular mediante el control de una aplicación Smartphone.

- Plataforma Raspberry Pi 3B modelo B.

Se realiza la selección de la plataforma y luego se seleccionará los componentes que mejor convenga para la aplicación, así como también los diferentes accesorios para la alternativa escogida.

CAPÍTULO II

PROPUESTA

3.1. Análisis de Requerimientos

A continuación, se describen las características de la tarjeta Raspberry Pi 3B para usar en el desarrollo del proyecto entre los cuales están: periféricos, pines de entrada y salida, capacidad de almacenamiento, así como las características de otros elementos como el módulo GPS, control remoto RF, cámara, sensor de ruido, modem USB 3G entre otros.

En el diseño del sistema de video vigilancia se utiliza una tarjeta madre denominada Raspberry Pi 3B que está basada en software libre la cual facilita el diseño del prototipo y el acople de los distintos dispositivos, se procederá a realizar el estudio de los parámetros para que el modem USB 3G tenga la habilidad de “conexión compartida a Internet” de manera tal que se genera una red interna entre los dispositivos como se indica en la Figura 22.

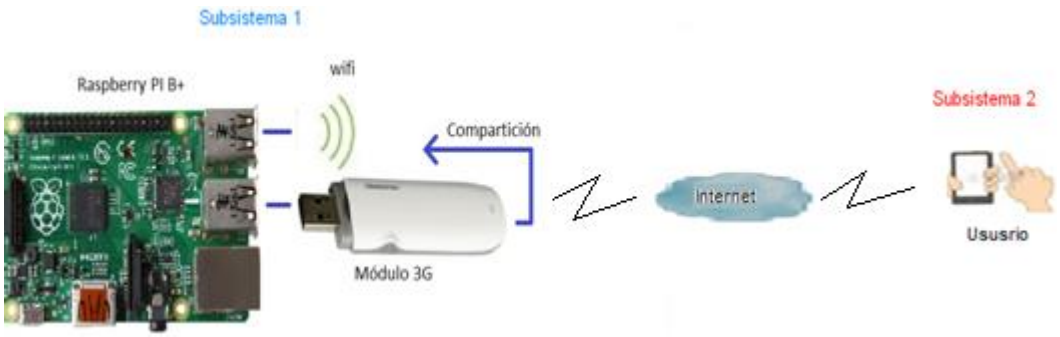


Figura 22. Red de comunicación entre dispositivos

Fuente: (Elaborado por el Autor)

El subsistema 1, está constituido por la tarjeta Raspberry Pi 3B y el módulo USB 3G que permite la comunicación inalámbrica, el acceso a internet y adicionalmente la comunicación bluetooth, este subsistema transmite la información al subsistema 2 que está constituido por un Smartphone.

En el subsistema 1 de la Figura 23, indicado los elementos que van conectados a la Raspberry Pi 3B, el cuál va a estar constar de un sensor de ruido, una cámara para la captura de imágenes y un modem USB 3G para el envío de datos. Adicionalmente la tarjeta Raspberry Pi 3B posee internamente un dispositivo de comunicación bluetooth entre la aplicación Android que está en el Smartphone y la tarjeta Raspberry Pi 3B. Mediante la aplicación Android instalada en el Smartphone se conecta a la tarjeta Raspberry Pi 3B para el encendido o apagado del sistema. Una vez que el sistema se haya encendido debido a la señal enviada por el sensor de ruido éste activará la cámara la cual comenzará a capturar imágenes en un período corto de tiempo, de esta manera las secuencias de imágenes serán enviadas al subsistema 2, el mismo que será encargado de monitorear por el usuario.

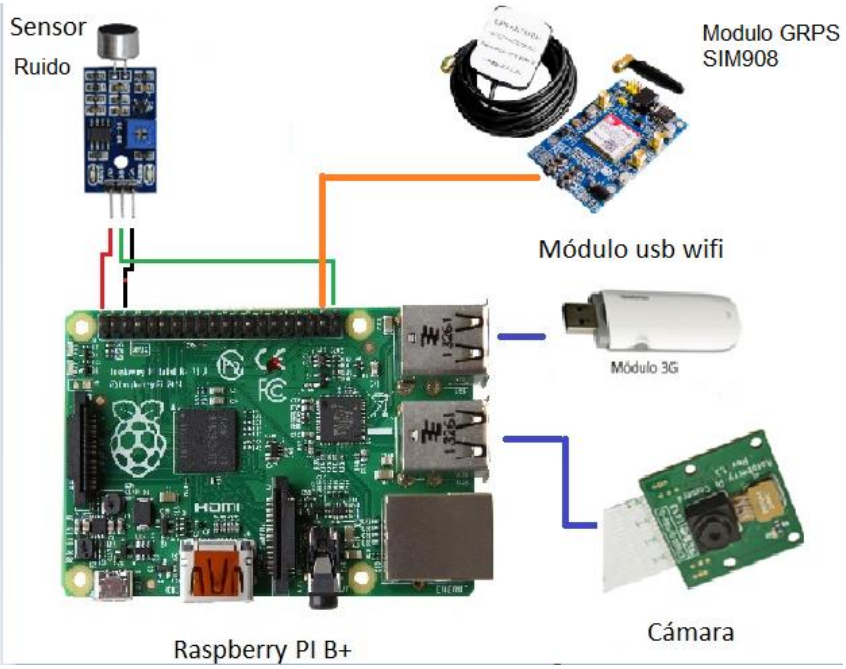


Figura 23. Componentes del subsistema 1

Fuente: (Elaborado por el Autor)

A continuación, en la Figura 24, se aprecia el diagrama del proceso de interacción, la comunicación entre los dos subsistemas de transmisión y la recepción del sistema prototipo.

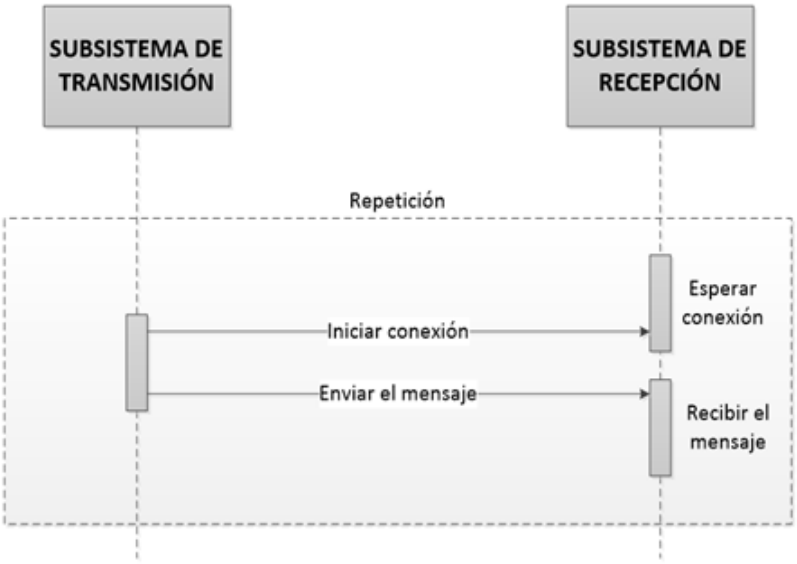


Figura 24. Proceso de comunicación de los subsistemas

Fuente: (Elaborado por el Autor)

Una vez descrita la funcionalidad de los subsistemas a continuación se describe la trama del subsistema 1 de transmisión que consta de la tarjeta Raspberry Pi 3B, esta es la encargada de capturar imágenes, detectar ruido, obtener la ubicación mediante módulo GRPS y recibir una llamada telefónica, el subsistema 2 recibe toda esta información mediante una red inalámbrica USB 3G. La Figura 25, muestra los bloques que forman parte del subsistema de transmisión y recepción con cada uno de los módulos. Toda la información que recibe el subsistema 2 es enviada al correo electrónico de esta forma se monitorea los eventos que se producen al ingresar a internet y abrir el correo electrónico.

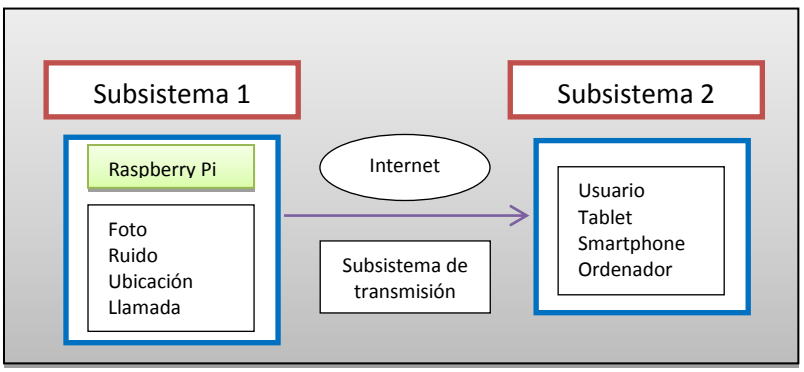


Figura 25. Subsistemas de transmisión y recepción.

Fuente: (Elaborado por el Autor)

3.2. Diseño

En esta sección se describe el diseño del sistema prototipo, se describe el software y adicionalmente se hace un análisis de todas las conexiones diagramas cálculos del hardware que se utiliza en el proyecto.

3.2.1. Arquitectura del sistema prototipo

Para analizar la arquitectura del sistema prototipo el cuál consta de dos subsistemas, en el subsistema 1 se caracteriza por tener conectada a la tarjeta Raspberry Pi 3B un sensor de ruido, una cámara, una batería de respaldo, un regulador de voltaje, un módulo GPS y un módulo de comunicación de radio frecuencia. La comunicación entre el módulo Raspberry Pi3B es mediante una red inalámbrica WIFI y la transmisión de datos entre los módulos es por bluetooth el mismo que se encuentra integrado en la tarjeta.

Adicionalmente a la tarjeta Raspberry Pi 3B está conectado un modem USB 3G para tener la habilidad de conexión compartida a internet y finalmente en la web se guarda las diferentes secuencias de imágenes y datos, para ser monitoreada por medio del subsistema 2 el cuál consta de un Smartphone, como se observa en la Figura 26.

3.2.2. Diagrama general de bloques del sistema prototipo

En la Figura 27, se muestra un diagrama de bloques que describe el funcionamiento de forma general del prototipo del sistema de localización y seguridad vehicular.

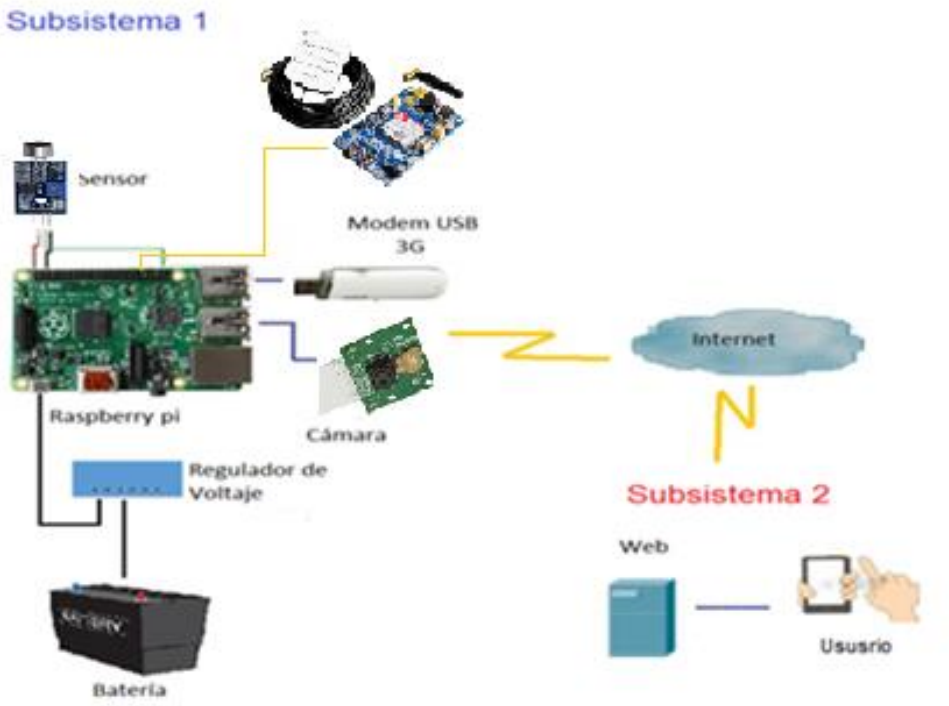


Figura 26. Arquitectura del sistema prototipo.

Fuente: (Elaborado por el Autor)

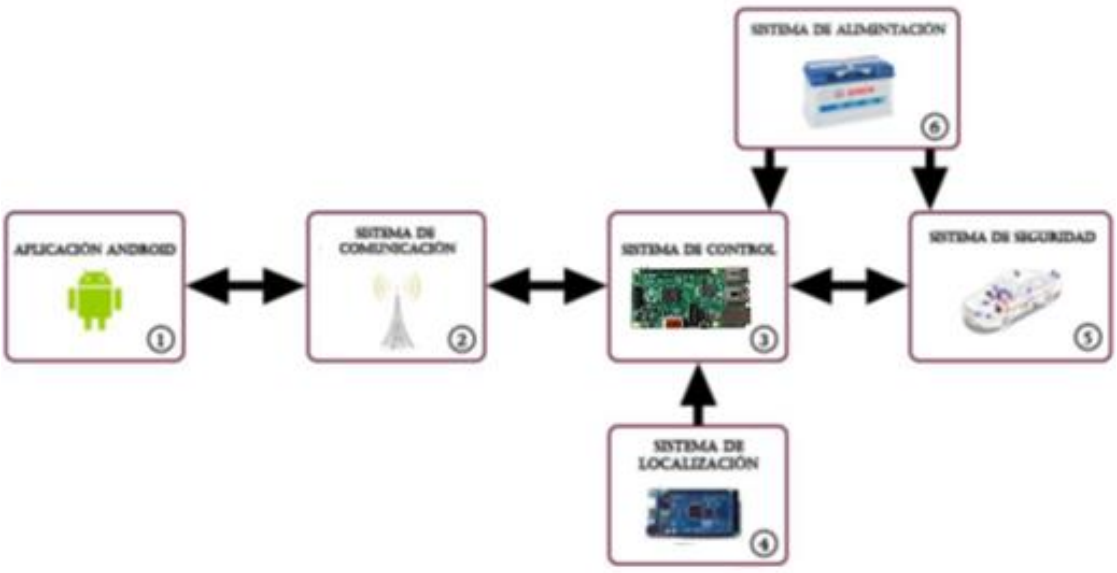


Figura 27. Diagrama de bloques general del sistema prototipo

Fuente: (Princy, 2015)

El diagrama de bloques mostrado en la Figura 27 muestra los dispositivos usados para el sistema prototipo, el cuál indica una parte del proyecto

distribuido en bloques, donde cada uno de estos representa una función específica dentro del prototipo a desarrollar como se muestra a continuación:

1. La Aplicación Android se desarrolla en APP Inventor, la cual se ejecuta en un Smartphone y permitirá el monitoreo de los eventos del sistema.

2. Los sistemas de comunicación tanto bluetooth como el módulo WIFI 3G son los encargados de transmitir toda la información para ser enviada y receptada por parte del usuario en la aplicación del Smartphone al Sistema de video seguridad que será instalado en el interior del vehículo.

3. El Sistema de Control es el encargado de procesar toda la información como el encendido y apagado del sistema, solicitudes y respuestas por parte del usuario, a través de la aplicación Smartphone mediante el uso de la tecnología bluetooth, WIFI 3G y GSM se encargan además de transmitir las señales provenientes del sistema GPS.

4. El Sistema de Localización GRPS es el encargado de receptar las señales provenientes de los satélites y trasladarlas al sistema de control, para luego ser procesadas dentro del mismo y posteriormente ser enviadas al usuario del sistema cuando este lo requiera.

5. El Sistema de video seguridad permite visualizar los eventos que se presentan dentro del vehículo dado el caso que se active la alarma.

6. El Sistema de Alimentación permite suministrar el voltaje y corriente necesarios para que el prototipo entre en funcionamiento; en el caso de que la alimentación de energía se lo haga por medio de la batería principal del vehículo y este falle el sistema cuenta con una etapa de alimentación de respaldo.

3.2.3. Bloque del sistema de control

Este sistema de control conformado por la tarjeta Raspberry Pi 3B que es el cerebro prototipo para recibir y enviar las instrucciones a realizar por cada uno de los bloques, los cuales se encuentran interconectados todos los módulos usados en el desarrollo del proyecto, para tener el control del encendido del vehículo, localización GPS, el envío y recepción de solicitudes y respuestas.

El diagrama de bloques mostrado en la Figura 28, conformado por la tarjeta Raspberry Pi 3B encuentra interconectado a todos los demás sistemas de control así permitiendo una comunicación bidireccional.

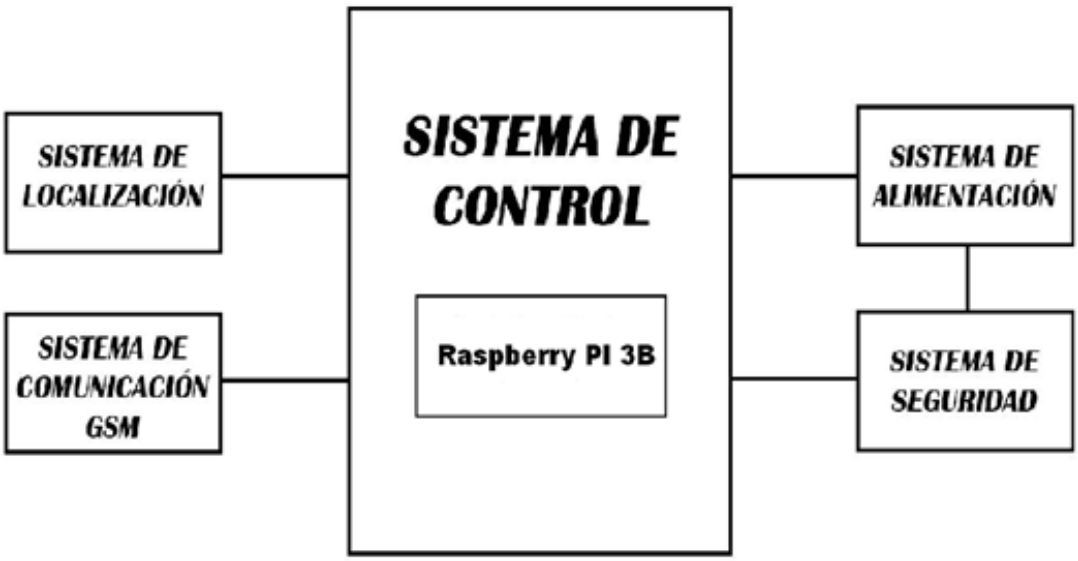


Figura 28. Diagrama de bloques del sistema de control

Fuente: (Elaborado por el Autor)

3.2.4. Diseño electrónico

En la Figura 29, se describe las etapas del hardware y software del sistema. Indican la estructura del software y hardware donde se toma en cuenta la interconexión y comunicación tanto entre los elementos físicos que confirman el sistema prototipo y el Smartphone. Cabe recalcar que se utilizó etapas de potencia para el mejor de control de sus partes móviles.

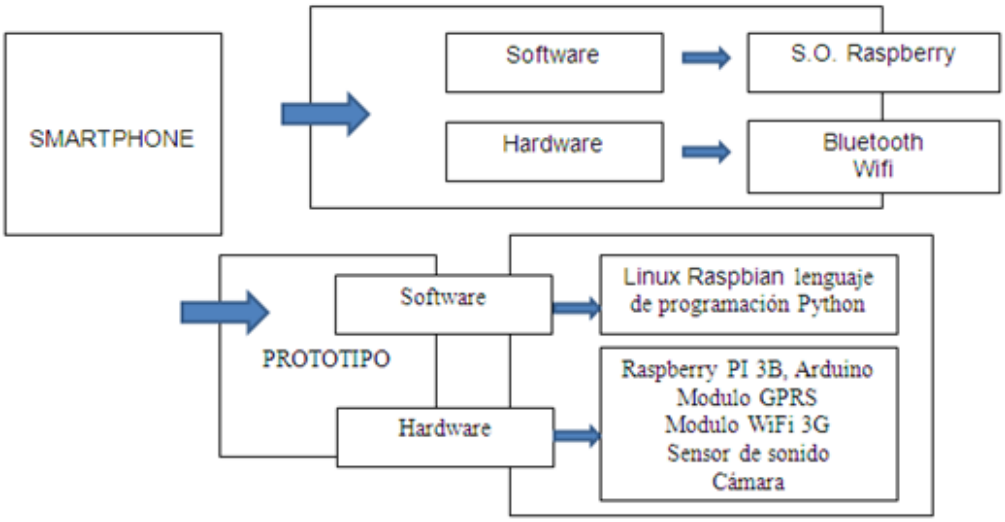


Figura 29. Etapas de software y hardware del Smartphone y del Prototipo

Fuente: (Princy, 2015)

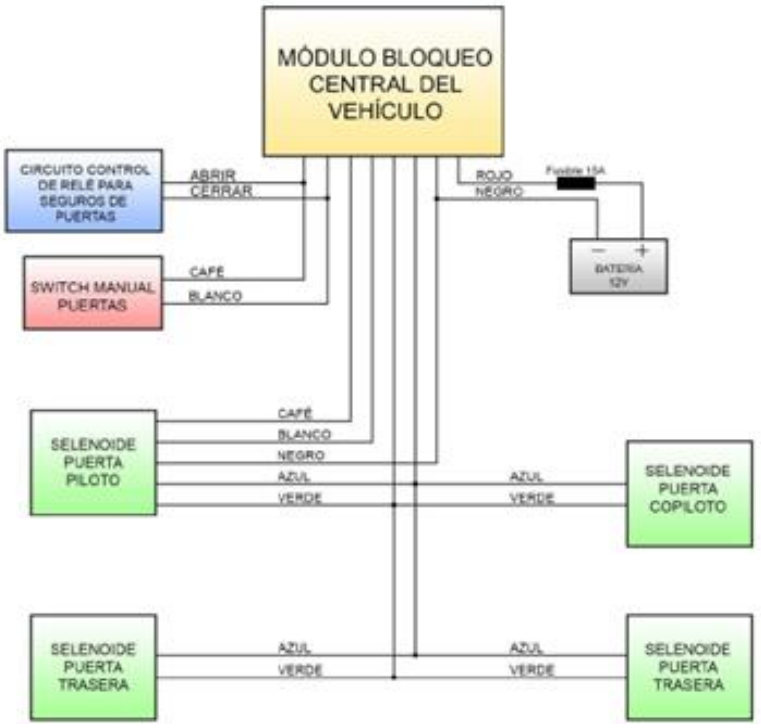


Figura 30. Esquema de conexión del circuito de control de relé de puertas al módulo de bloqueo central del vehículo.

Fuente: (Elaborado por el Autor)

En la Figura 30 se muestra el esquema propuesto para la conexión de los circuitos de control del relé al módulo de bloqueo central y al control de puertas del vehículo, se debe tomar en cuenta que este es un sistema pre

instalado en la mayoría de los vehículos por lo cual la conexión se realiza en paralelo para cada una de las señales de control provenientes de la tarjeta Raspberry Pi 3B, de tal forma que se tiene una señal para apertura de puertas y otra para el cerrado de las mismas.

3.2.5. Interfaz de Comunicación

3.2.5.1. Comunicación Raspberry y el ordenador

La conexión se la realiza a través de VNC (Virtual Network Computing), es un protocolo de comunicación que sirve para acceder y controlar una computadora de manera remota a través de la red, consiste que por medio de un ordenador se conecta a la terminal de la Raspberry Pi 3B para ejecutar comandos de manera remota como se aprecia en la Figura 31.

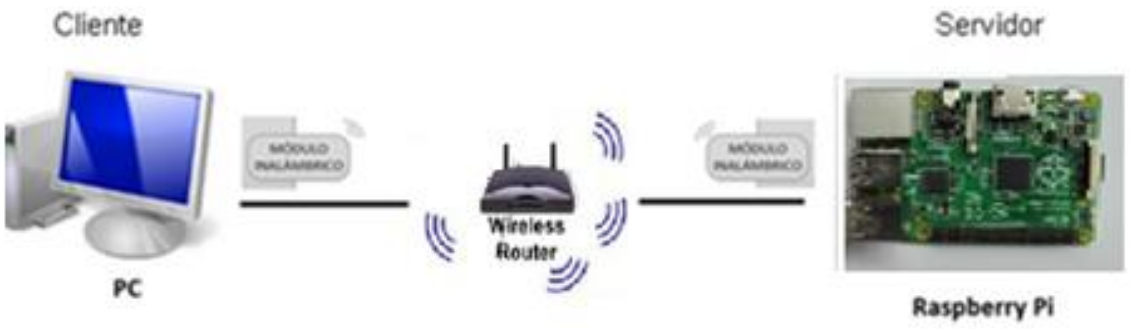


Figura 31. Interfaz de comunicación PC y Raspberry Pi.

Fuente: (Elaborado por el Autor)

3.2.5.2. Comunicación de los subsistemas

La interfaz de comunicación del subsistema 1 y subsistema 2 permite enviar o recibir señales entre si a través de una línea de transmisión. Este subproceso se encarga de iniciar la conexión entre la tarjeta Raspberry Pi 3B y el computador a través de una red inalámbrica mediante Wi-Fi, de esta forma realizar cambios dentro de la tarjeta Raspberry Pi 3B. El programa principal del prototipo se compone de dos subprocesos mostrados en el diagrama de flujo de la Figura 32.

Si por algún motivo se produjera algún error en la conexión, el algoritmo de programación realiza los intentos necesarios hasta que la comunicación entre los dispositivos sea la correcta.

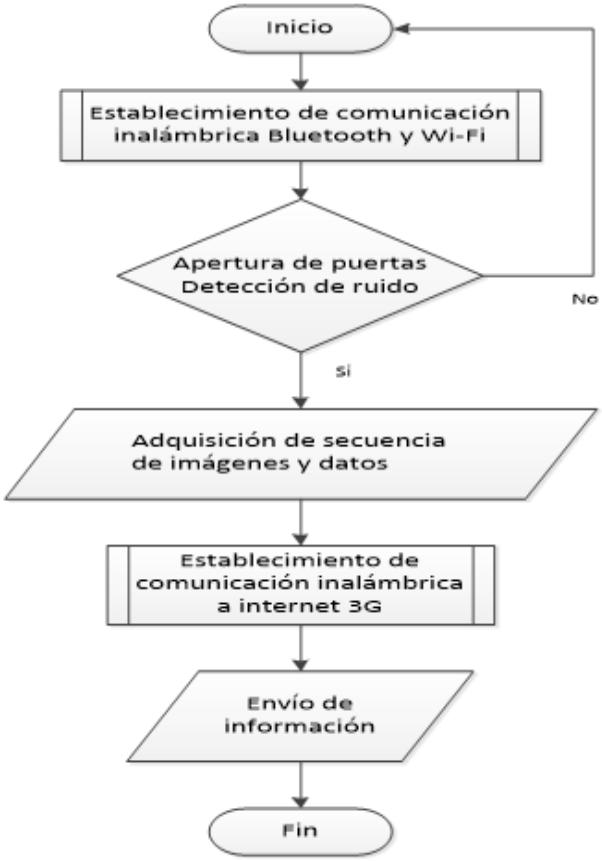


Figura 32. Diagrama de flujo de comunicación de los subsistemas.

Fuente: (Elaborado por el Autor)

3.2.6. Descripción de los componentes del diseño

En esta sección se va a diseñar las principales funciones que controlan los dispositivos y las acciones a realizar.

3.2.6.1. Cámara de Imagen Raspberry Pi 100003

La cámara utiliza el método *prvier_camera ()* la cuál captura una imagen y retorna el archivo en *file_path*.

En la Figura 33 se muestra el script del proceso y el tiempo de captura de imagen “ver Anexo 6”.

```
import time
import picamera

with picamera.PiCamera() as picam:
    picam.start_preview()
    time.sleep(5)
    picam.capture('foto.jpg')
    picam.stop_preview()
    picam.close()
```

Figura 333. Captura de imagen de la cámara

Fuente: (Elaborado por el Autor)

El momento que el sensor de sonido tiene un evento, la cámara se enciende y comienza a capturar imágenes, lo procesa y como paso siguiente envía la información hacía en correo electrónico, la Figura 34 muestra el diagrama de flujo para el proceso del encendido de la cámara.

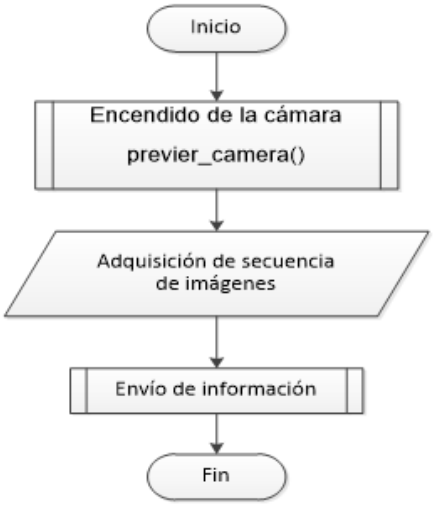


Figura 34. Diagrama de flujo para el encendido da la cámara.

Fuente: (Elaborado por el Autor)

3.2.6.2. Sensor de sonido

La Tabla 7 muestra los pines donde va conectado el sensor de sonido en la GPIO de la tarjeta Raspberry Pi 3B.

Tabla **7**: **Conexión sensor de sonido**.

Pin GPIO	Conectado
2 (5.0V)	Vcc
6 (Ground)	GND
4 (GPI04)	Signal

Fuente: (Elaborado por el Autor)

Para la detección de ruido se utiliza el pin 4 que envía la señal de activación a la tarjeta Raspberry Pi 3B. Se importa a las librerías de GPIO de Python para Raspberry; se inicializa el pin el cuál es el 4 de la GPIO y llama al método de inicialización de la detección de sonido “ver Anexo 6”.

En la Figura 35 se muestra parte del código de inicio para la detección de sonido el código completo se detalla en el “Anexo 6”, en este script se inicia y permanece en ejecución cada vez que se active el sistema. Cuando se activa la cámara inicia la captura de imágenes cada 15 segundos automáticamente.

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)
GPIO.setup(6, GPIO.IN)
GPIO.setup(17, GPIO.OUT)

while True:
    input_state = GPIO.input(6)
    if input_state == True:
        GPIO.output(17, GPIO.LOW)
        print('sonido detectado')
        time.sleep(0.5)
        GPIO.output(17, False)
```

Figura 35. Inicialización para la detección de sonido

Fuente: (Elaborado por el Autor)

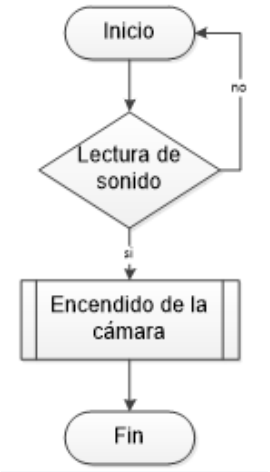


Figura 36. Diagrama de flujo lectura del sensor de sonido.

Fuente: (Elaborado por el Autor)

En la figura 36 se muestra el diagrama de flujo cuando existe un ruido el cual hace que se el sensor de sonido se active y actué enviando una señal que activa la cámara.

3.2.6.3. Modem USB 3G Huawei E3531

Cuando el modem se conecta a la ranura de la tarjeta Raspberry Pi 3B la conexión inicia automáticamente y se ejecuta el siguiente comando: *sudo apt-get install usb-modeswitch*, de esta manera se tiene acceso a internet, se chequea el estatus del modem como se ve en la Figura 37.

```
pi@raspberrypi:~ $ lsusb
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp. SMSC9512/9514 Fast
Ethernet Adapter
Bus 001 Device 002: ID 0424:9514 Standard Microsystems Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
pi@raspberrypi:~ $ lsusb
Bus 001 Device 005: ID 12d1:14dc Huawei Technologies Co., Ltd.
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp. SMSC9512/9514 Fast
Ethernet Adapter
Bus 001 Device 002: ID 0424:9514 Standard Microsystems Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

Figura 37. Conexión del modem 3G

Fuente: (Elaborado por el Autor)

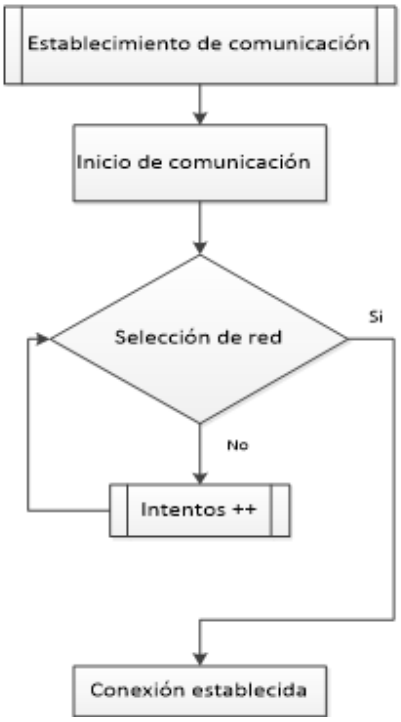


Figura 38. Diagrama de flujo del subproceso de comunicación

Fuente: (Elaborado por el Autor)

En la Figura 38 se ilustra el diagrama de flujo para el subproceso establecimiento de la comunicación a la red de internet.

3.2.6.4. Diagrama electrónico del sistema de comunicación GPRS

Se diseña el sistema para realizar la comunicación GSM con el fin de enviar mensajes de texto SMS al Smartphone, la conexión del módulo se realiza mediante la comunicación serial *UART*, a través de los pines de transmisión y recepción, en la Tabla 8 se muestra la distribución de pines usados para la conexión del módulo de comunicación GSM, así como la descripción de cada uno de ellos:

Tabla 8: **Distribución de pines del módulo gprs/gsm sim908.**

Pines	Conexión	Descripción
D7	Pin 19 de la placa Arduino Mega 2560 (Rx1)	Pin de recepción de la comunicación por software serial del módulo
D8	Pin 18 de la placa Arduino Mega 2560 (Tx1)	Pin de transmisión de la comunicación por software serial del módulo
VCC	Salida 5V de la Placa Arduino Mega 2560	Alimentación del módulo a 5V a través del propio Arduino
GND	Masa de la Placa Arduino Mega 2560	Tierra del módulo a través del propio Arduino

Fuente: (Elaborado por el Autor)

Figura 39. Esquema de conexión del módulo GPRS al Arduino Mega 2560.

Fuente: (Elaborado por el Autor)

Los pines de conexión usados en la placa Arduino Mega 2560 se conecta a los pines de transmisión y recepción del módulo de comunicación GPRS SIM908 “ver Anexo 5”, el pin usado es el 7 para transmisión el cuál se conecta a la Raspberry Pi 3B para la recepción, este sistema se encarga de enviar la ubicación por medio de un SMS, se muestra en la Figura 39.

3.2.6.5. Circuito regulador de voltaje

El circuito regulador de voltaje de 5V a 1 Amp, permite suministrar el voltaje y corriente necesarios a los módulos de la tarjeta Raspberry Pi 3B y el Arduino Mega 2560, para que no exista variación de voltaje y provoque daños a las tarjetas por lo cual el diseño se vuelve primordial dado que la corriente que proporciona las placas en sus pines de salida es de solamente de 50mA entre 5V a 3.3V “ver Anexo 4”.

3.2.6.6. Circuito de relé para seguros de puertas

Para controlar los seguros de las puertas del vehículo se diseñan dos circuitos de relés similares e independientes en caso de que uno de los dos se dañe, uno servirá específicamente para subir los seguros de las puertas y otro para bajarlos, de esta manera actúa como un switch electrónico “ver Anexo 7”. La activación de los relés se lo hace con un pulsador normalmente cerrado el cual envía la señal el instante que se abra una de las puertas haciendo que se active la alarma, el diseño de este circuito se lo ilustra en la Figura 40.

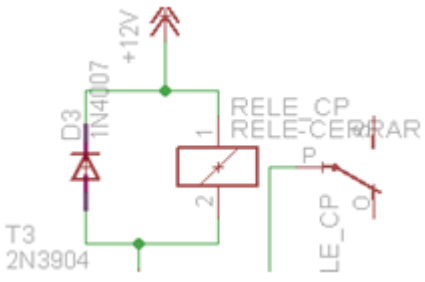


Figura 40. Circuito de relé para seguros de las puertas del vehículo.

Fuente: (Elaborado por el Autor)

3.2.6.7. Control de contacto a 12V

Esta etapa de control es la encargada del bloqueo del encendido del vehículo, el contacto a 12V se bloquea cuando se activa la alarma para impedir el paso de voltaje formado principalmente por el circuito de la llave de contacto. La tarjeta Raspberry Pi 3B envía la señal de activación la cual controla un relé al ser usado por el sistema de control.

El diseño del circuito de contacto está formado principalmente por el regulador de voltaje, el relé, le batería y la tarjeta Raspberry Pi 3B se lo puede observar en Figura 41.

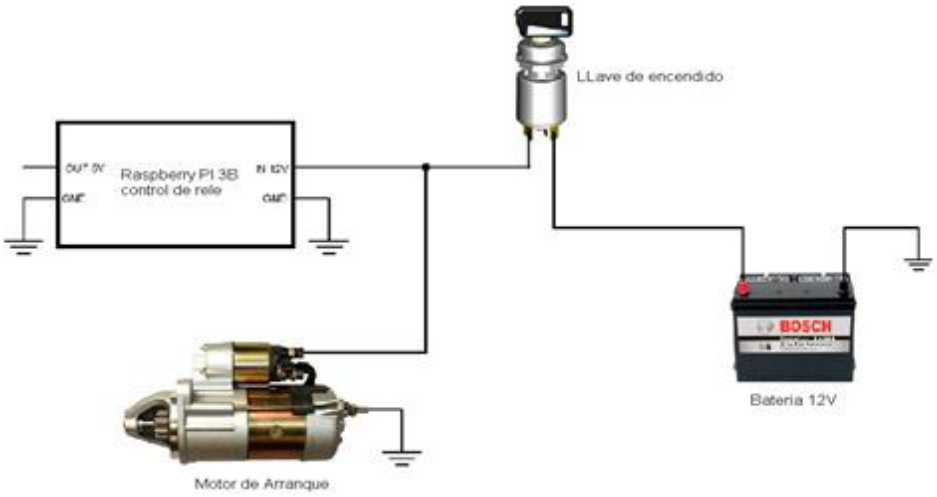


Figura 41. Esquema de conexión del circuito de encendido del vehículo.

Fuente: (Elaborado por el Autor)

3.2.6.8. Bloque del sistema de alimentación

Para el sistema de alimentación del prototipo se utiliza la conexión de baterías en paralelo, de tal manera de suministrar el voltaje necesario a cada uno de los dispositivos o módulos que conforman el sistema, este se encuentra distribuido en varias etapas que cumplen una función específica. Una característica especial es aquella que permite al prototipo continuar con su funcionamiento normal para la función de localización y el servicio de llamadas ya que este dispositivo tiene una alimentación independiente de cada una de las tarjetas, en el caso de que la batería del vehículo fuese desconectada.

El sistema de alimentación se conforma por dos etapas, la primera que es la alimentación por medio de la batería de vehículo y la segunda que es de alimentación de la fuente de respaldo, en la Figura 42 se observar el diagrama del sistema de alimentación con cada uno de los bloques que lo conforman.

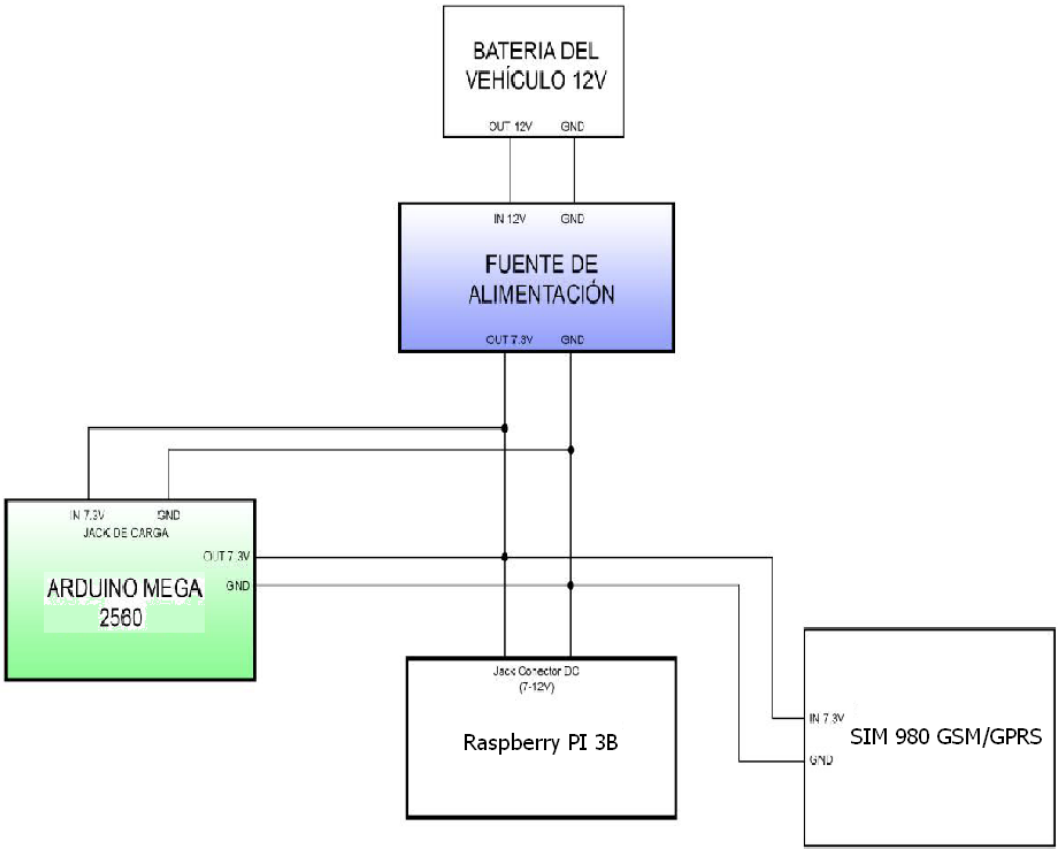


Figura 42. Diagrama de bloques del sistema de alimentación.

Fuente: (Elaborado por el Autor)

3.2.7. Diseño para las conexiones del sistema prototipo

En la figura 43 se ilustra el diseño realizado en Proteus donde se puede observar las conexiones respectivas del sistema prototipo, se utilizan los pines GPIO de propósito general de la tarjeta Raspberry Pi 3B los cuales tienen la señal en bajo y debido a que la señal del dispositivo es en alto se conecta un integrado inversor 7404 para negar la señal del sensor de ruido asi permitiendo la comunicación entre los mismos.

3.2.7.1. Diagrama general del sistema prototipo

Finalmente, después de haber realizado el diseño de cada una de las etapas que conforman el prototipo de sistema de seguridad vehicular, se procede a presentar el diagrama eléctrico total mostrado en la Figura 44, en donde se encuentra detallada cada uno de los dispositivos usados y la interconexión de los mismos, los cuáles conforman la tarjeta Raspberry Pi 3B, Arduino mega 2560, módulo SIM 908 GPRS, módulo de relés, regulador de voltaje y la batería de respaldo.

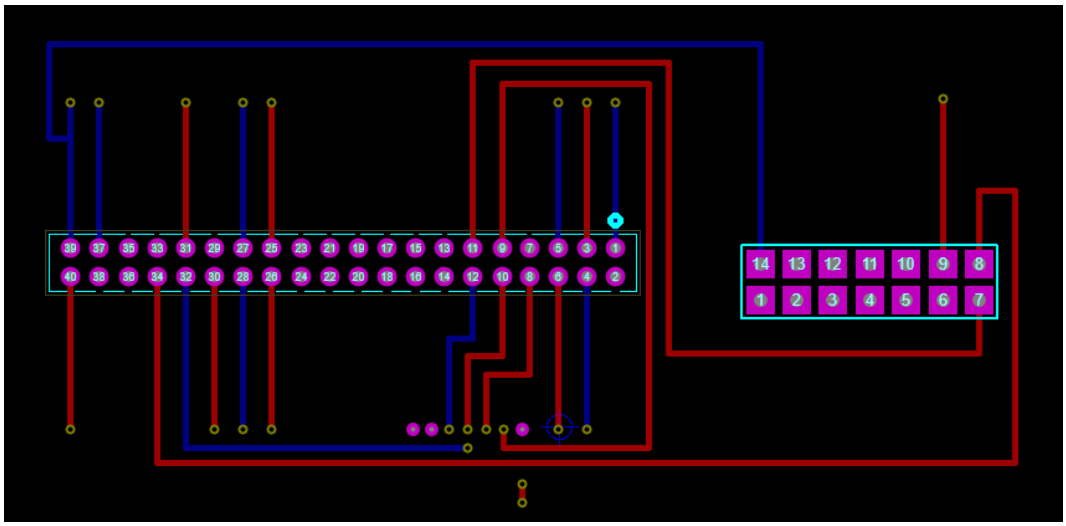


Figura 43. Diseño conexiones GPIO.

Fuente: (Elaborado por el Autor)

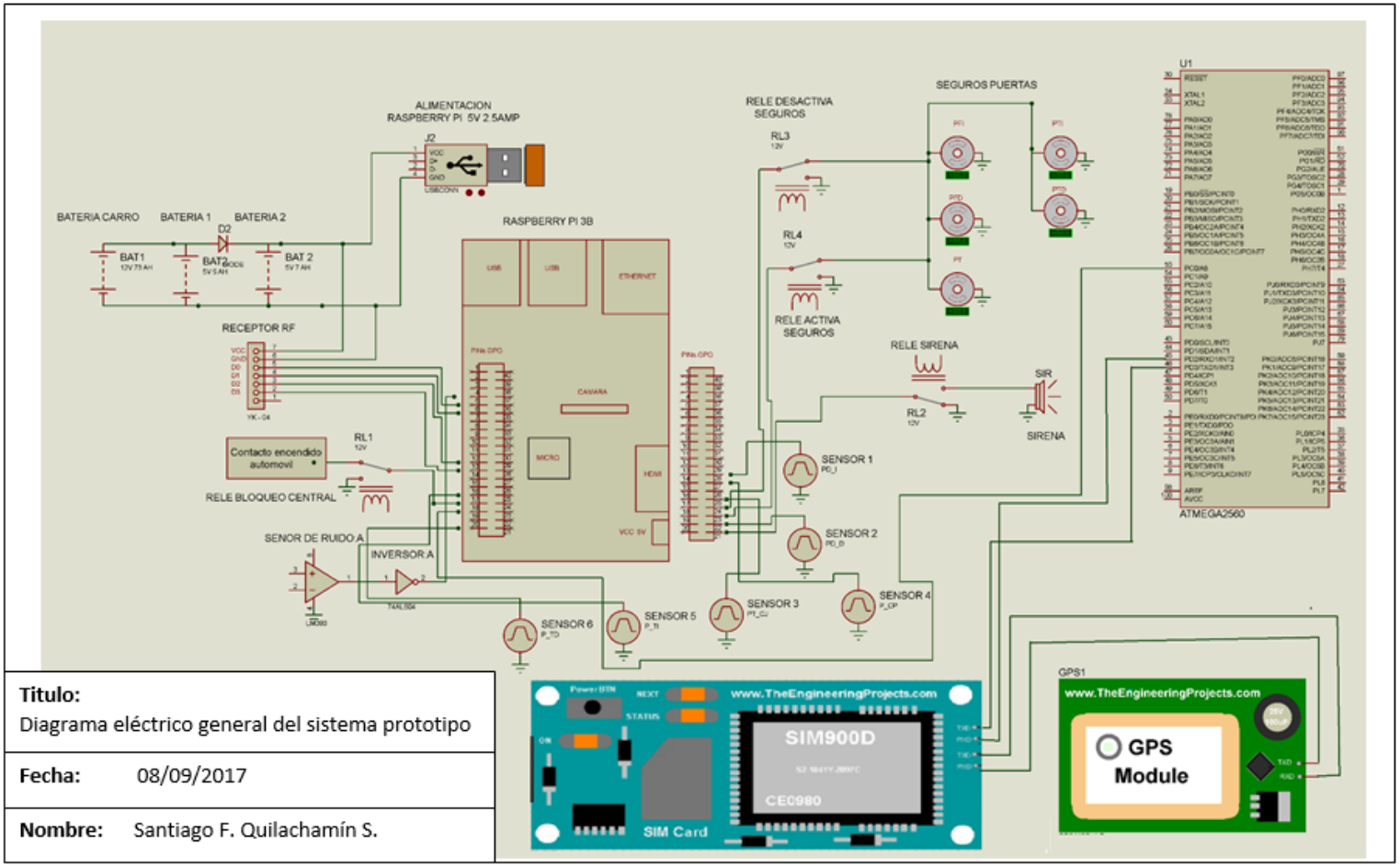


Figura 44. Diagrama eléctrico general del sistema prototipo.

Fuente: (Elaborado por el Autor)

3.2.8. Interfaz para Smartphone

La interfaz de usuario estará conformada por un botón de activación y desactivación del sistema, en la pantalla se listarán los datos además se visualiza el estado del vehículo si se encuentra con las puertas cerradas o abiertas. Cada grupo de sentencias tiene la función de realizar actividades específicas y facilitar el control de la herramienta de monitoreo, los elementos que conformarán la interfaz de usuario deben realizar las siguientes tareas.

- Control de la comunicación bluetooth.
- Control del inicio, pausa, stop/reset, continuar y salir.
- Pantalla de monitoreo y análisis de los datos recibidos.

La interfaz está desarrollada en la aplicación APP inventor cuyo proceso se detalla a continuación:

- Para crear la aplicación se inicia el entorno de trabajo al ingresar a la dirección URL (Appinventor, 2017). Una vez aquí se carga la página de inicio de la aplicación y se debe acceder a la opción MIT APP Inventor como se indica en la Figura 45.

- Es necesario tener una cuenta de correo en Gmail para acceder a la aplicación, en caso de no tener una se la debe crear una.

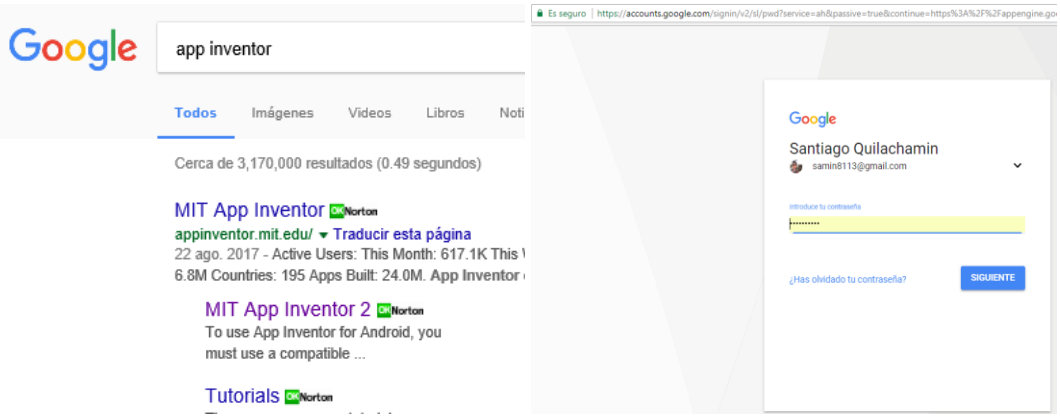


Figura 45. Estructura del entorno de edición

Fuente: (Google, 2017)

- Una vez confirmada la cuenta se tiene acceso al entorno de edición, en la cual se programará la aplicación como se muestra en la Figura 46.

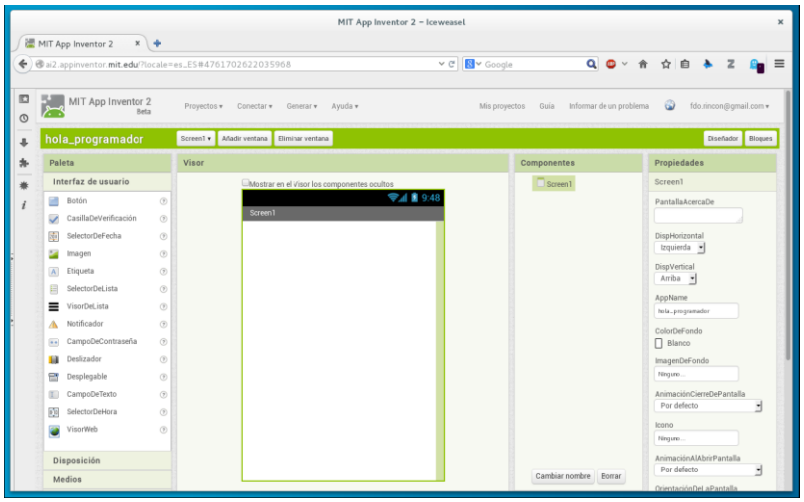


Figura 46. Estructura del entorno de edición

Fuente: (INVENTOR, 2017)

El funcionamiento de la aplicación se inspira en los bloques de construcción de distintas formas y tamaños los cuales únicamente encajan de cierta manera. Aunque estos bloques sean muy simples, mediante la combinación de muchos de ellos pueden construirse creaciones realmente complejas e impresionantes.

Para iniciar con la aplicación se busca a la izquierda del editor de bloques la columna Bloques, dentro de ella hay 3 categorías que son Integrados, screen1 y cualquier componente como se observa en la Figura 47.

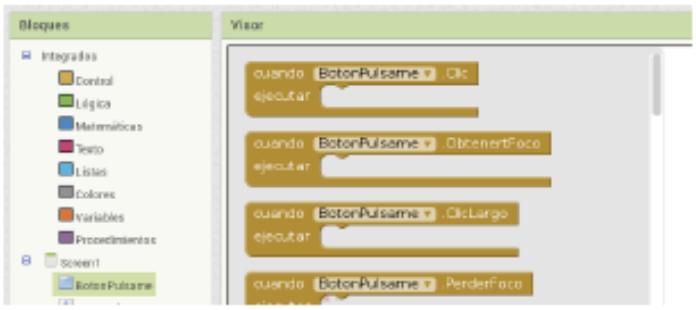


Figura 47. Bloques asociados al botón de la aplicación

Fuente: (Elaborado por el Autor)

En la categoría screen1 se encuentra los componentes correspondientes al botón y la etiqueta. Se debe hacer clic sobre screen y se observara a la derecha los bloques de colores, los cuales indican condiciones y procesos a seguir.

3.2.8.1. Diagrama de flujo para la aplicación Smartphone

El diagrama de flujo diseñado permite evidenciar la secuencia lógica que servirá para la posterior programación de la misma, en la Figura 48 se observa la secuencia que tiene la aplicación Android, la cual, al instante de ingresar a la aplicación, pide una clave de usuario y contraseña adicionalmente una vez ingresado a la aplicación se debe conectar mediante bluetooth así tener control del sistema de alarma.

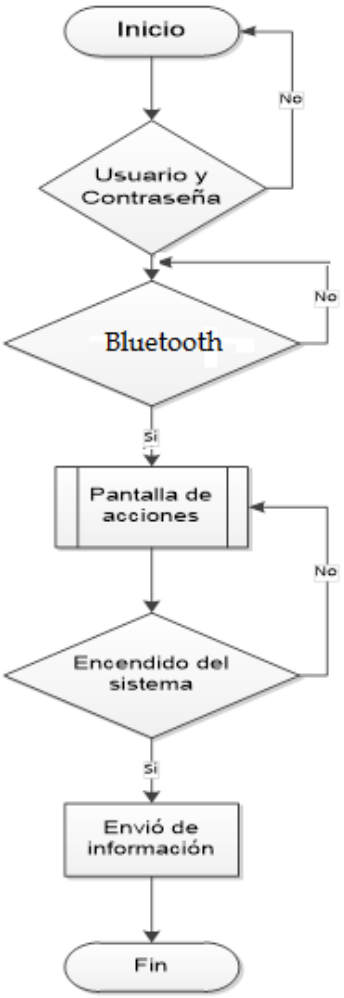


Figura 48. Diagrama de flujo de la aplicación Smartphone

Fuente: (Elaborado por el Autor)

3.2.8.2. Diseño de la aplicación Smartphone

En la Figura 49, se observa el tipo de tecnologías que se utilizó para establecer la comunicación y él envió bidireccional de datos entre los dispositivos. Tales como, bluetooth, Raspberry Pi 3B, Android, GRPS.

Figura 499. Tipo de tecnologías que se utilizaron

Fuente: (Elaborado por el Autor)

3.1.8.2.1. Pantalla de ingreso de datos

Esta pantalla principal permite al usuario tener control total de la seguridad mediante la asignación de un usuario y una contraseña. En la casilla de Usuario colocamos *(ALARMA)* y PASSWORD *(123456)*, de manera correcta, en caso de que uno de los dos campos no se llene adecuadamente y muestra un mensaje de error: Usuario/Contraseña incorrectos como se observa en la Figura 50 “ver Anexo 1”.

Figura 50. Pantalla de ingreso de usuario y contraseña

Fuente: (Elaborado por el Autor)

3.1.8.2.2. Pantalla de acciones y componentes

La Pantalla de Acciones permite al usuario hacer uso de las funciones que ofrece la aplicación para el Smartphone mediante el envío de comandos a la tarjeta Raspberry Pi 3B, las diferentes acciones son elegidas de la lista que se muestra en pantalla, cabe recalcar que solamente se puede elegir y confirmar una acción a la vez, en la Figura 51 se ilustra la pantalla de inicio de la aplicación y sus respectivos componentes.

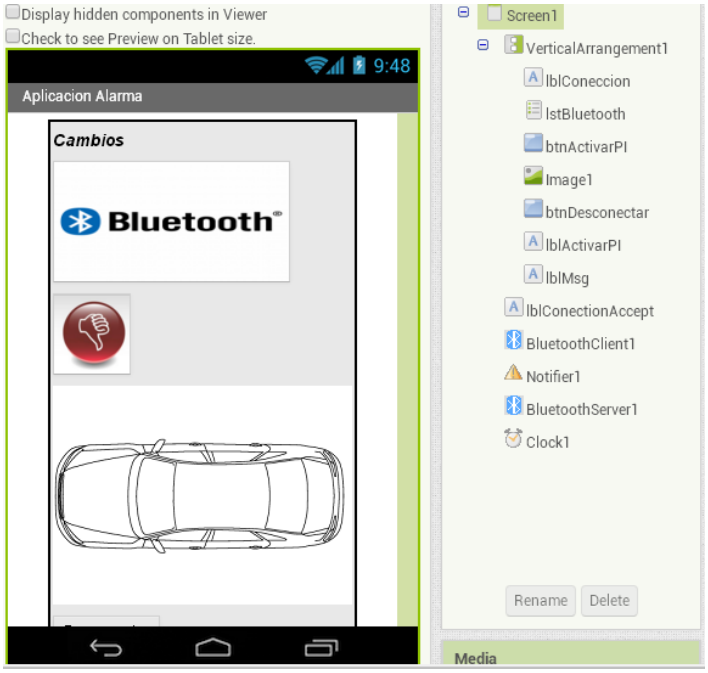


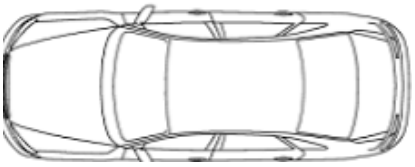
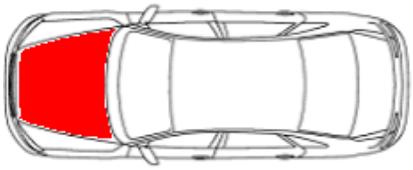
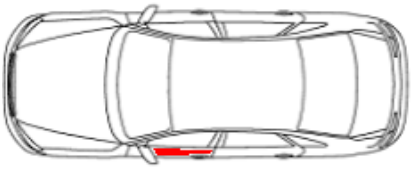
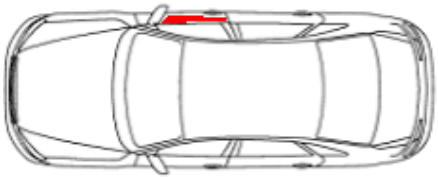
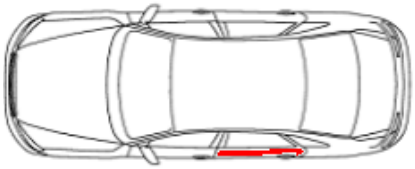
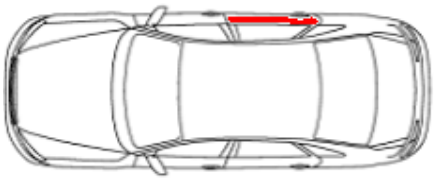
Figura 51. Pantalla de acciones y sus componentes

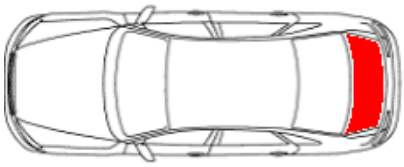
Fuente: (Elaborado por el Autor)

3.1.8.2.3. Descripción de aplicación

La Tabla 9 se lista los componentes usados para el desarrollo de la pantalla de acciones sus elementos son los botones de control, los cuadros de texto donde se visualiza los datos que envía los módulos y sensores. La visualización mediante cámara que permitirá determinar qué tipo de evento se está presentando dentro del vehículo. Así también acceso a envió de mensajes a redes sociales y también envío de correo electrónicos.

Tabla 9: **Funciones de la aplicación**

ELEMENTO	FUNCIÓN
	Conexión a bluetooth
	Desconexión a bluetooth
	Activación
	Desactivación
	Indicador de las puertas
	Indicador puerta capo abierta
	Indicador puerta derecha delantera abierta
	Indicador puerta izquierda delantera abierta
	Indicador puerta trasera derecha abierta
	Indicador puerta trasera izquierda abierta

	Indicador puerta cajuela abierta
Desconectado	Baja todos los servidores

Fuente: (Elaborado por el Autor)

La aplicación tiene las siguientes funcionalidades que permite mediante el botón de bluetooth la conexión del dispositivo al Smartphone y al sistema prototipo. Una vez que esta sincronizado permite visualizar un mensaje que indica conectado como se muestra en la figura 52, caso contrario muestra un mensaje de no conectado como se indicada en la Figura 53.

Una vez conectado al sistema prototipo, permite enviar datos mediante el botón de encendido, el cual hace que el sistema entre en operación y está en alerta ante cualquiera alarma, ya sea una abertura de una puerta del vehículo o la lectura del sensor de sonido, ante estos eventos el sistema toma una foto y la envía por correo electrónico, adicionalmente se visualiza mediante un SMS la ubicación geográfica del vehículo.

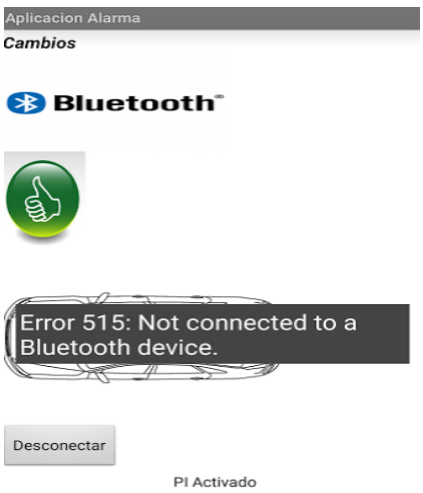


Figura 52. Botón de conexión bluetooth

Fuente: (Elaborado por el Autor)

Figura 53. Conexión bluetooth no disponible

Fuente: (Elaborado por el Autor)

CAPÍTULO III

IMPLEMENTACIÓN

4.1. Desarrollo

En esta sección se detalla el proceso de implementación del proyecto, así como también los pasos que se siguieron para instalación y ejecución.

4.1.1. Instalación del sistema operativo

En la PC se debe descargar el archivo *img.* del sistema operativo que va a instalar en la tarjeta Raspberry Pi 3B desde internet. Selecciono la *img.* que se encuentra en la PC como se muestra en la Figura 54, para seguidamente abrirla y elegir la tarjeta micro SD de 32Gb, que debe estar formateada de tal forma que sea *booteable*, se utiliza el programa *SDFormatter* el cuál lo modifica el orden de arranque del dispositivo. (LACERNA, 2015)

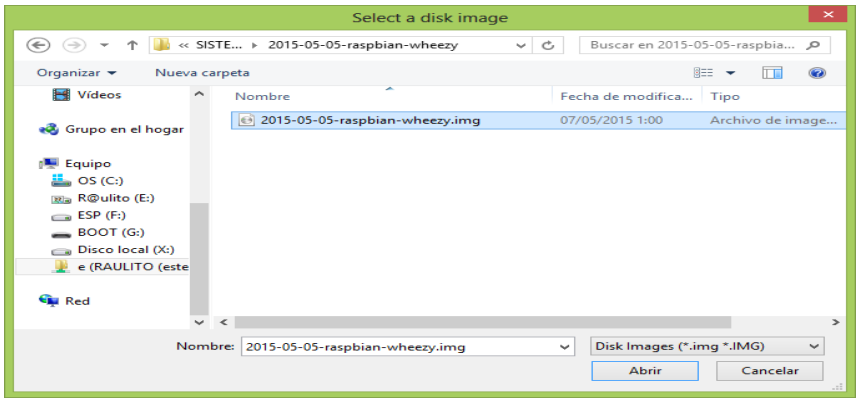


Figura 54. Imagen del archivo del sistema operativo a instalar

Fuente: (Elaborado por el Autor)

Para instalar la *img.* se utiliza el programa *Win32DiskImager* y lo ejecuta como administrador para que permita descargar todas las librerías el momento de la actualización del sistema operativo en la ventana de aplicación.

A continuación, como se muestra en la Figura 55 se escoge la opción *Write* para cargar el sistema operativo a la memoria micro SD, una vez finalizada la carga se procede a insertar en la ranura de la tarjeta Raspberry Pi 3B.

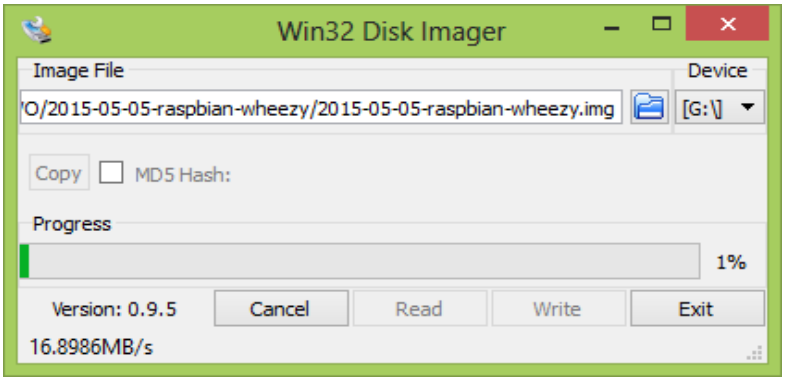


Figura 55. Carga de Win32Disklmanager en la SD

Fuente: (Elaborado por el Autor)

Ya instalado el sistema operativo e insertada en la ranura de la tarjeta Raspberry Pi 3B se procede a establecer comunicación con dicha tarjeta, para lo cual se utiliza el programa *advanced IP Scanner* este software es el encargado de buscar e identifico la IP de la Raspberry Pi 3B como se observa en la Figura 56, se consigue la IP de la Raspberry Pi 3B conectado un modem de una red interna y así estar en la misma red con la PC para establecer a comunicación remota.

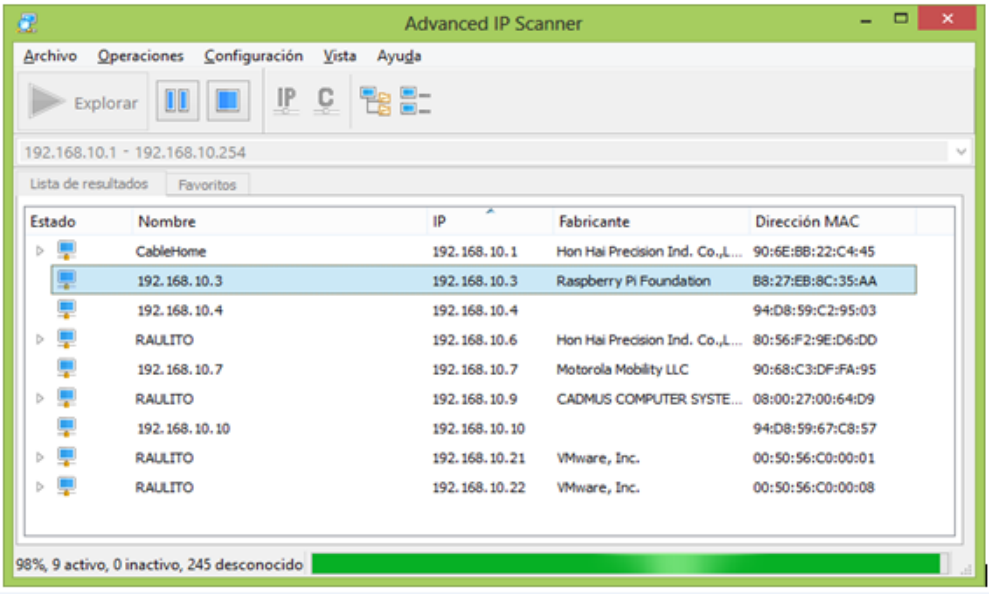


Figura 56. Identificación de la tarjeta Raspberry Pi 3B por IP

Fuente: (Elaborado por el Autor)

Una vez localizado la IP con el *software VNC (Virtual Network Computing)* se procede a ingresar por escritorio remoto, para ello se coloca la IP encontrada

por *Advanced IP Sanner* y se da clic en ok como se ilustra en la Figura 57 y comienza el proceso de conexión hacia la tarjeta Raspberry Pi 3B.

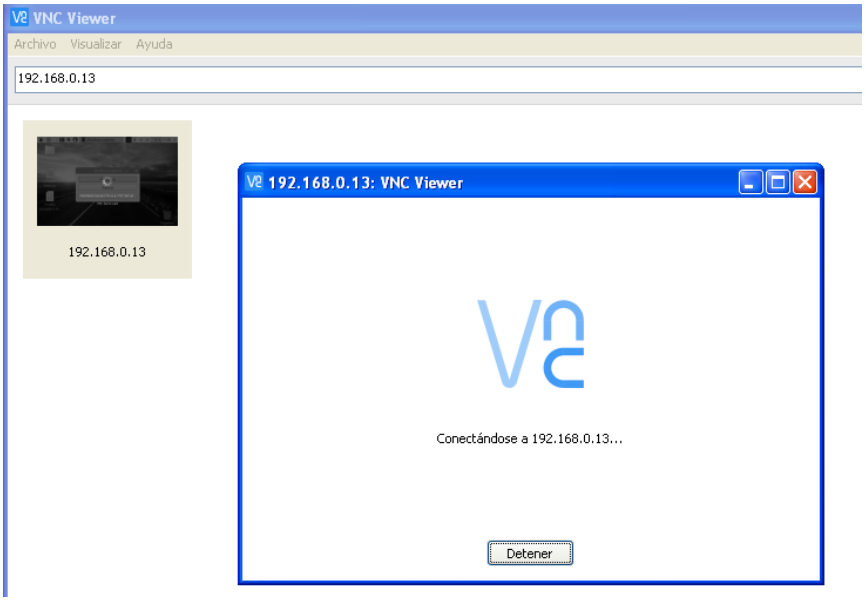


Figura 57. Conexión a escritorio remoto con la Raspberry.

Fuente: (Elaborado por el Autor)

Finalmente se ingresa con letras minúsculas el usuario *"pi",* y la contraseña *"raspberry",* para acceder a la tarjeta Raspberry Pi 3B, mediante VNC como se indica en la Figura 58.

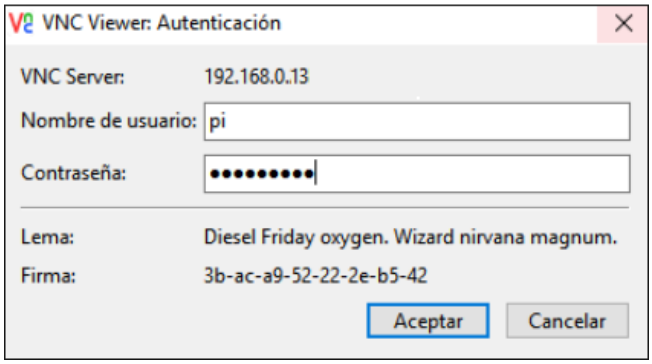


Figura 58. Ingreso de usuario y contraseña en VNC

Fuente: (Elaborado por el Autor)

Para acceder y conectarse mediante escritorio remoto con la tarjeta Raspberry Pi 3B, es necesario que el computador y la tarjeta Raspberry Pi 3B se encuentren en un mismo segmento de red.

A continuación, en la Figura 59 se muestra el entorno de comunicación entre una PC y la tarjeta Raspberry Pi 3B así también la venta de cmd de ejecución de los comandos de configuración de los módulos que controla.

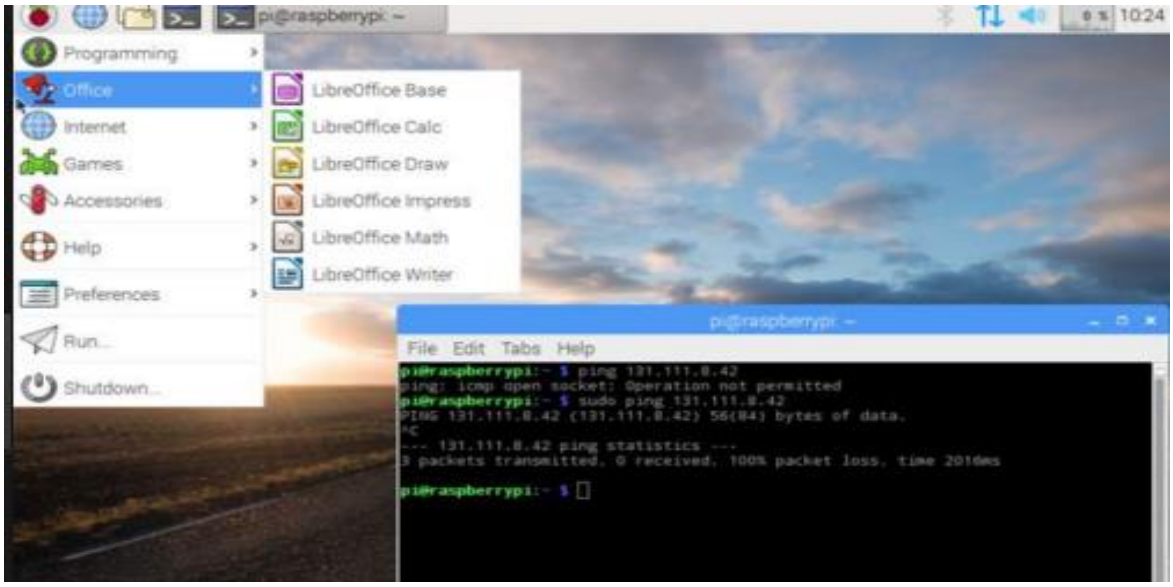


Figura 59. Entorno de comunicación PC y la Raspberry.

Fuente: (Elaborado por el Autor)

4.1.2. Configuración Bluetooth

Para establecer comunicación del Smartphone con la tarjeta Raspberry Pi 3B se debe configurar el bluetooth para lo cual se ingresa los siguientes comandos en el cmd:

1. *Comando hciconfig,* se utiliza para configurar dispositivos Bluetooth, también muestra el nombre e información básica sobre todos los dispositivos Bluetooth instalados en el sistema como se observa en la Figura 60.

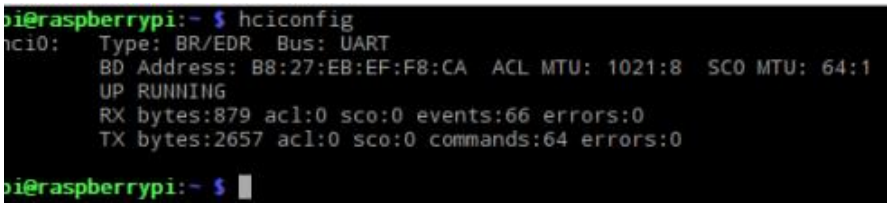


Figura 60. Comando hciconfig

Fuente: (Elaborado por el Autor)

2. *Comando sudo apt-get install python-dev libbluetooth-dev,* con este comando se instala todo el paquete completo de las librerías de Bluetooth como se observa en la Figura 61.

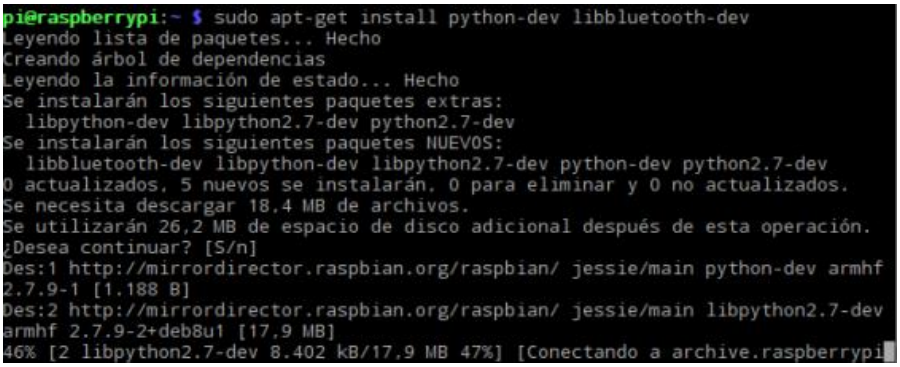


Figura 61. Comando de instalación python-dev

Fuente: (Elaborado por el Autor)

3. *Comando sudo service bluetooth status*, este comando permite ver el estado del bluetooth a través del terminal se observa que está habilitado o deshabilitado como se observa en la Figura 62.

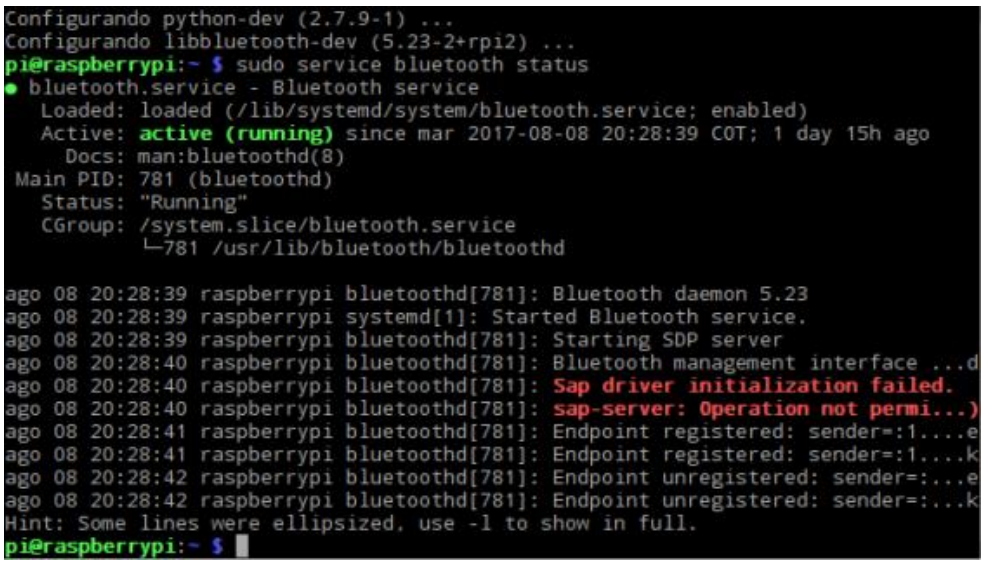


Figura 62. Comando de verificación del estado de bluetooth

Fuente: (Elaborado por el Autor)

4. *Comando sudo nano /lib/systemd/system/bluetooth.service*, inicia un archivo de configuración similar al de la Figura 61, el cual se habilita las

funciones experimentales agrego - *c* a la línea *ExecStart* como se observa en la Figura 63.

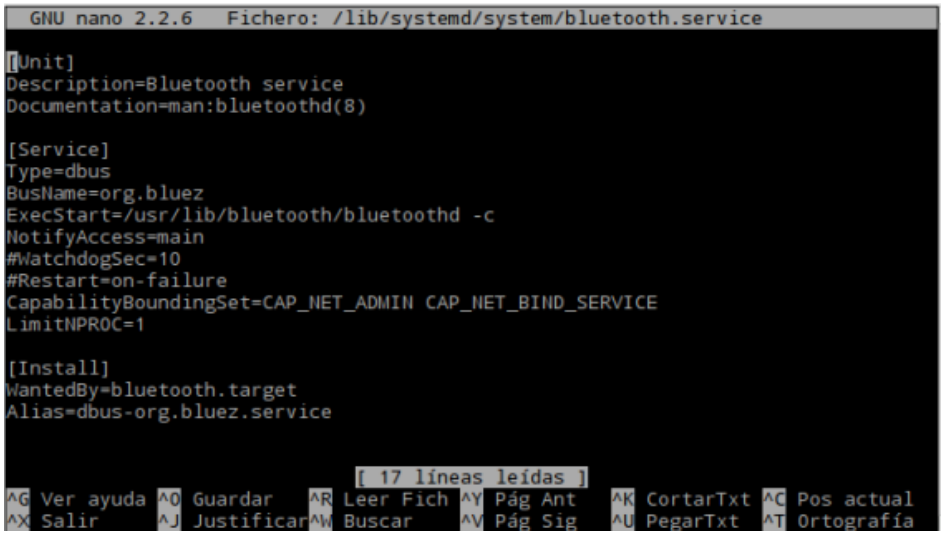


Figura 63. Edición bluetooth. service

Fuente: (Elaborado por el Autor)

5. Al finalizar la edición de *bluetooth.service* se cierra el archivo, se guarda los cambios efectuados y finalmente se reinicia el sistema con el comando *sudo reboot.*

6. Se verifica el estado de bluetooth al ingresar nuevamente el comando *sudo service bluetooth status* en el *cmd*, en el cual se observa los cambios efectuados como se ilustra en la Figura 64, el estado de bluetooth y el cambio efectuado con el comando anterior.

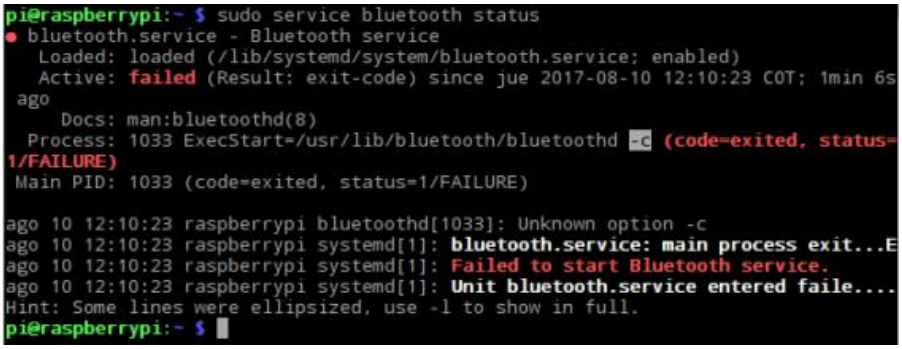


Figura 64. Estado de configuración efectuados

Fuente: (Elaborado por el Autor)

7. A continuación, se ejecuta el comando *sudo bluetoothctl*, con lo cual inicializa el bluetooth para ejecutar *power on* y a seguidamente establecer lo siguiente: *pairable on, discoverable on, agent on, defaul-agent;* como se observa en la Figura 65.

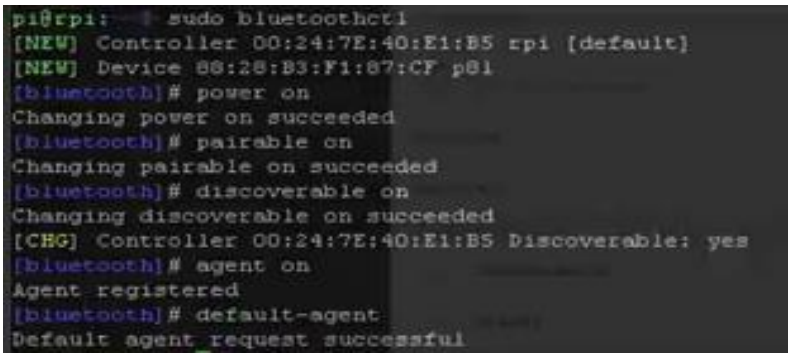


Figura 65. Comando sudo bluetoothct1

Fuente: (Elaborado por el Autor)

En la figura 66 se observa la descarga de las librerías desde internet, al salir de la anterior carpeta bluetooth se coloca *exit* y se ejecuta el comando root, luego *wget https://bootstrap.pypa.io/get-pip.py* para descargar todas las librerías desde internet, se ejecuta el archivo *get-pit.py* con el comando *sudo python get-pit.py*, a continuación se ejecuta *git clone https:// github.com/karulis/pybluez.git*, con el fin de descargar todos los archivos de *pyblue* y el ejecutable para posterior instalación en la tarjeta Raspberry Pi 3B.

Figura 66. Descarga de archivos desde internet

Fuente: (Elaboracdo por el Autor)

A continuación se ejecuta el comando *sudo python get-pip.py*, y se descarga *pybluez* de la página web se coloca el siguiente comando *git clone*

https://github.com/karulis/pybluez.git, con este comando se instala *pybluez* como se observa en la Figura 67, se ingresa a la carpeta *cd pybluez/* y se ejecuta *sudo python setup.py install*, a continuación se debe salir de la carpeta con el comando *cd*.

Figura 67. Instalación de pybluez

Fuente: (Elaborado por el Autor)

A continuación, se ejecuta *sudo nano bluetooth_adv,* y se agrega los siguientes comandos como se observa en la figura 68, se sale del archivo y se guarda los cambios ejecutados, con el fin de modificar el archivo se ingresa en la primera línea *hciconfig hci0 piscan* y en la segunda línea se ingresa *sdptool add SP*.

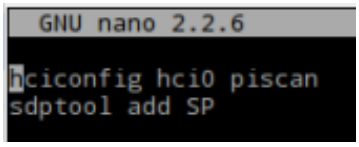


Figura 68. Modificación del archivo bluetooth_adv

Fuente: (Elaborado por el Autor)

Finalmente se ingresa por consola el siguiente comando *sudo chmod +x bluetooth_adv* y se ejecuta *sudo./bluetooth_adv*, con este último comando se debe registrar en el puerto serial como se ilustra en la Figura 69.

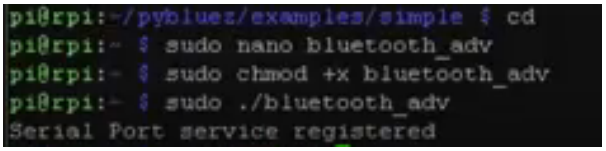


Figura 69. Registro del puerto serial

Fuente: (Elaborado por el Autor)

4.2. Implementación

En esta sección se muestra la implementación del sistema prototipo las conexiones en la tarjeta Raspberry Pi 3B y la ejecución de los *scripts* de cada dispositivo.

4.2.1. Elementos del sistema de seguridad vehicular

Durante el desarrollo del proyecto, se utilizan diversos dispositivos electrónicos que se listan a continuación:

4.2.1.1. Raspberry Pi 3B (Cerebro del sistema): Esta es la unidad que se encarga de procesar señales de entrada y de salida, las cuáles pueden ser las provenientes de sensores y además de permitir realizar un control sobre algunos dispositivos instalados dentro del vehículo como pueden ser los seguros eléctricos de las puertas, sirenas, etc.

4.2.1.2. Pulsadores: Se implementan 6 pulsadores con la finalidad de detectar si una puerta, capó o baúl se encuentran abiertos o cerrados, para ello, envían al cerebro del sistema el estado en que se encuentra cada uno de ellos. De tal forma al detectar que un pulsador está enviando alto se produzca una alerta sonora a través de la sirena.

4.2.1.3. Sensor de sonido. Se implementa un sensor de sonido dentro del vehículo, su función es detectar los impactos sonoros que pudieran producirse en el interior del vehículo producto de un ruido fuerte como en ventanas y puertas.

4.2.1.4. Control Remoto RF: Se implementa un control inalámbrico que funciona mediante radio frecuencia cuya función es la activar o desactivar la alarma del vehículo, también permite la apertura y cierre de los seguros de las puertas del vehículo.

4.2.1.5. Sirena: La función de este dispositivo es la de notificar mediante un sonido determinado la activación o desactivación de la alarma y la activación la alarma del vehículo frente a un evento.

4.2.1.6. Módulo Gprs: Se implementa un módulo de control GPRS Sim 908 con la finalidad de enviar la ubicación donde se encuentre el vehículo, esta ubicación se la envía mediante un mensaje SMS al teléfono celular registrado y realizará una llamada de notificación para alertar al usuario del sistema.

4.2.1.7. Cables Conectores: Se utiliza cables conectores con la finalidad de interconectar todos los dispositivos que están involucrados en la alarma vehicular para facilitar las comunicaciones entre ellos, son generalmente de material de cobre con revestimiento de plástico de diferentes colores, así distinguir su función.

4.2.2. Componentes usados en el sistema prototipo

Para la implementación se hizo uso de un vehículo Chevrolet modelo Grand Vitara en el cual se instalarán varios componentes de entrada y salida los mismos se detallan en la Tabla 10.

Tabla 10: **Componentes usados**

Cantidad	Componente
1	Raspberry Pi 3B
1	Arduino Mega 2560
1	Módulo de 4 Relés
1	SIM 908 GPRS 3G
1	Módulo RF YK04
1	Sensor de sonido
1	Cámara Pi 100003
1	Batería 12V – 7Ah
2	Regulador de voltaje
1	Sirena
1	Smartphone Samsung J7
3	Bus de datos cámara
1	Bus de datos GPIOS
6	Pulsadores

Fuente: (Elaborado por el Autor)

4.2.2.1. Sensor de sonido.

La tarjeta implementada se conecta al sensor de sonido a los pines de propósito general (GPIO), como se ilustra en la Figura 70, con la finalidad de detectar sonido a través del micrófono que tiene integrado en el dispositivo.
La característica de la resistencia ajustable sirve para controlar manualmente el límite de nivel sonoro o umbral de disparo del sensor, es decir se puede ajustar la sensibilidad del dispositivo, con esta resistencia variable.

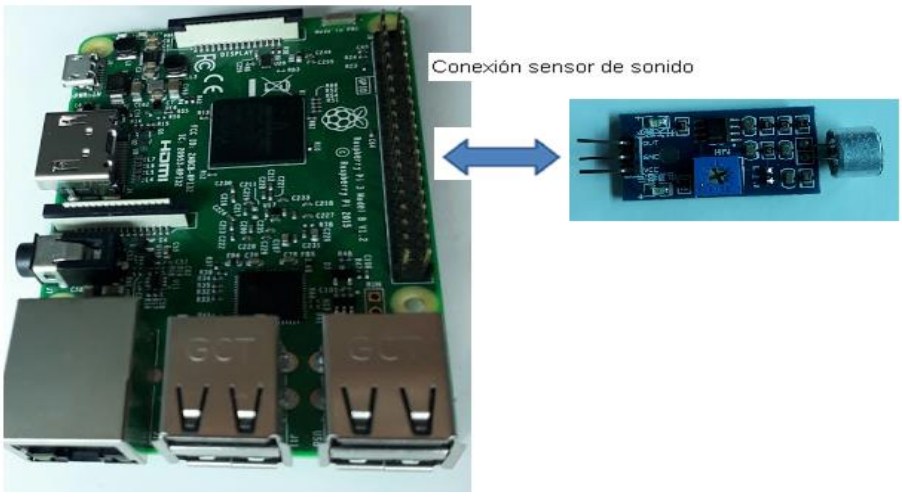


Figura 70. Conexión sensor de sonido a la Raspberry Pi 3B

Fuente: (Elaborado por el Autor)

A continuación, en la Figura 71 se ilustra el sensor de sonido en el vehículo Chevrolet Grand Vitara, colocado en el interior del vehículo en la esquina superior izquierda en la posición del lado del volante.

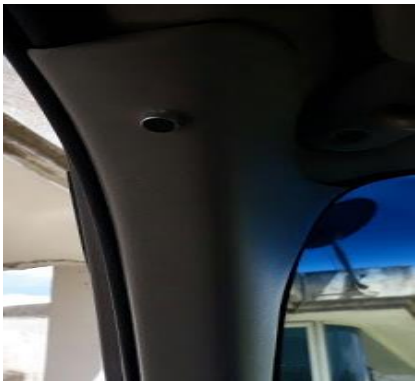


Figura 71. Implementación de sensor de sonido en el vehículo

Fuente: (Elaborado por el Autor)

4.2.2.2. Cámara

Este tipo de cámara se conecta directamente al socket del puerto de la cámara de la Raspberry Pi 3B, para los cual se utiliza una cámara de video Raspberry Pi 100003, como se ilustra en la Figura 72, este dispositivo se lo utiliza con la finalidad de procesar las imágenes que serán enviadas por secuencias al correo electrónico.

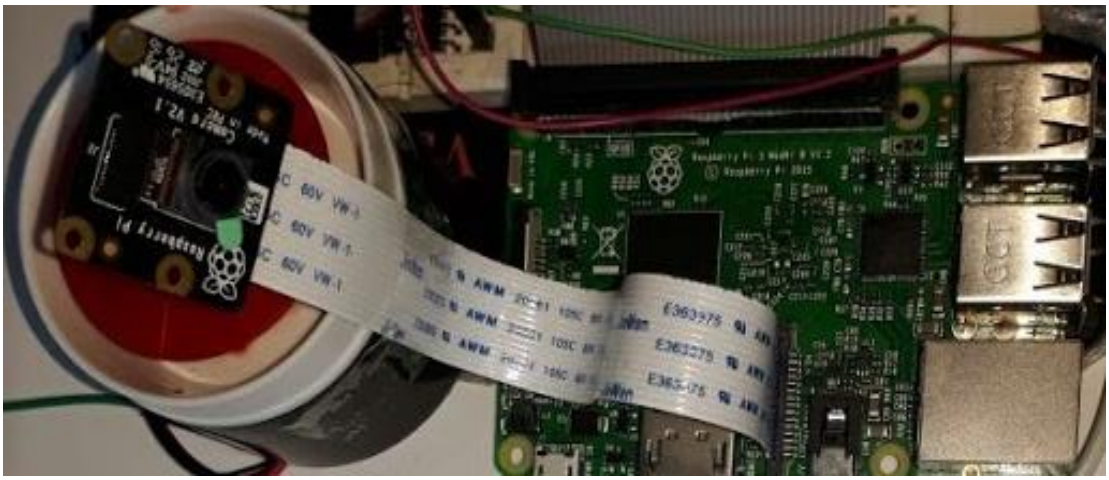


Figura 72. Conexión cámara a la Raspberry Pi 3B

Fuente: (Elaborado por el Autor)

Para la implementación de la cámara se utilizó un cable de15 cm de largo propio de la cámara, al cual se adaptó un cable tipo plano de 60 cm el largo como se muestra en la Figura 73 de esa manera permitiendo que se coloque la cámara en un lugar estratégico como se observa en la Figura 73 para la captura de imágenes.

Figura 73. Bus de datos adicional cámara Raspberry Pi 3B

Fuente: (Elaborado por el Autor)

Figura 74. Implementación cámara al vehículo

Fuente: (Elaborado por el Autor)

4.2.2.3. Control remoto RF

Se utilizo un módulo de control a distancia de radio frecuencia de cuatro canales para activar y desactivar el sistema, este dispositivo está conectado a los pines de propósito general GPIO de la tarjeta Raspberry Pi 3B que realizan diferentes acciones, como se indica la Tabla 11.

Tabla 11: **Pines de conexión módulo YK04 RF**

Módulo RF YK04	Señal	Función	GPIO
D0	B	Apagar	6
D1	D	Apagar Emergencia	13
D2	A	Encender	5
D3	C	Encender modo silencio	12

Fuente: (Elaborado por el Autor)

En la Figura 75 se muestra la tarjeta Raspberry Pi 3B, el módulo RF receptor que va conectado a los pines GPIO de la tarjeta Raspberry Pi 3B y el control RF transmisor que envía la señal mediante la pulsación de los botones A, B, C y D.

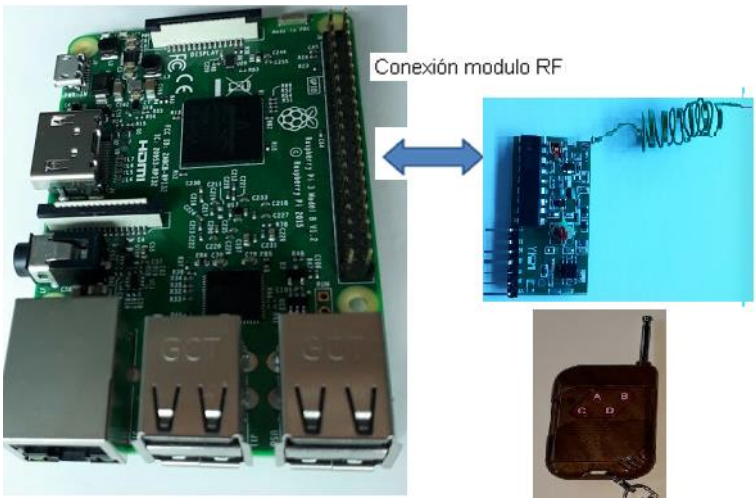


Figura 75. Tarjeta Raspberry Pi 3B, modulo RF y control RF

Fuente: (Elaborado por el Autor)

4.2.2.4. Modem USB 3G Huawei E3531

El modem USB 3G Huawei e3531 está conectado al puerto USB de la tarjeta Raspberry Pi 3B, como muestra la Figura 76, esto se realiza con la finalidad de establecer comunicación para el acceso a Internet.

Figura 76. Conexión USB 3G a la Raspberry Pi 3B

Fuente: (Elaborado por el Autor)

4.2.2.5. Módulo GPRS SIM 908 y el Arduino Mega 2560

Se utiliza un módulo GPRS SIM 908 y la tarjeta Arduino Mega 2560 como se muestra en la Figura 77. Para enviar un mensaje de texto con la ubicación, este dispositivo está conectada a la tarjeta Arduino Mega 2560, el cual está conectada a los puertos seriales de dicha tarjeta, la señal de entrada es gobernada por el pin 7 de la tarjeta Arduino Mega 2560 esta señal es enviada por la salida de la tarjeta Raspberry Pi 3B del pin 8 de la GPIO.

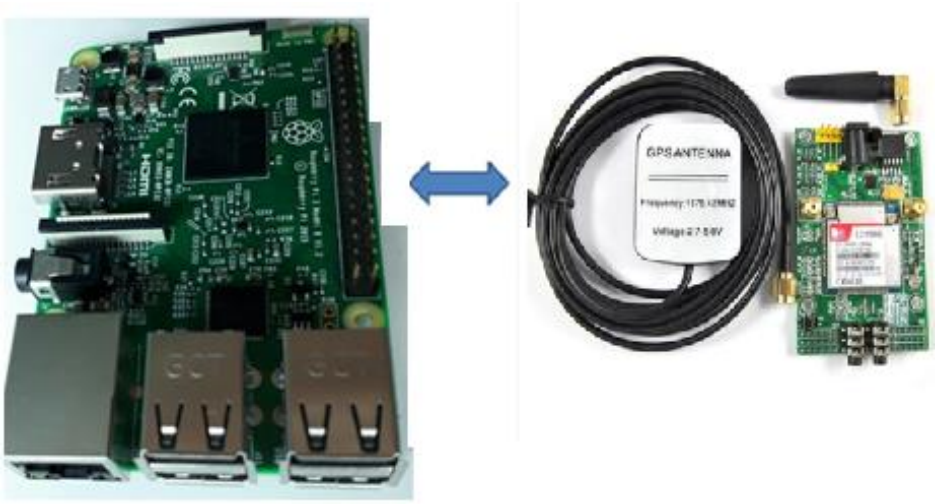


Figura 77. Módulo GPRS Sim 908 y Raspberry Pi 3B

Fuente: (Elaborado por el Autor)

4.2.2.6. Pulsadores para puertas

Para detectar si una puerta está abierta o cerrada se hace uso de pulsadores que están conectados a la GPIO de la tarjeta Raspberry Pi 3B, en la Tabla 12 se muestra los pines a los cuáles están conectadas los pulsadores para cada puerta del vehículo.

Tabla 12: **Pines de conexión puertas del vehículo**

Pulsador	PIN (GPIO)
Capo	26
Puerta delantera derecha	22
Puerta delantera izquierda	27
Puerta trasera derecha	20
Puerta trasera izquierda	16
Cajuela	19

Fuente: (Elaborado por el Autor)

En la Figura 78 se muestra la tarjeta Raspberry Pi 3B junto con a los pulsadores de que están instalados en los topes de las puertas del vehículo, los cuales son normalmente cerrados y envían una señal de alto el momento que son activados hacia las entradas GPIO de la tarjeta Raspberry Pi 3B.

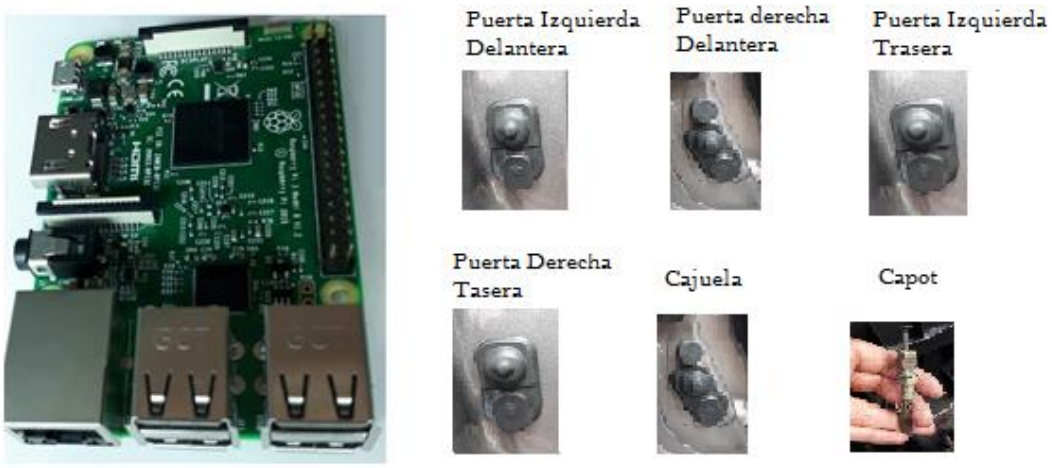


Figura 78. Tarjeta Raspberry Pi 3B y pulsadores de cada puerta

Fuente: (Elaborado por el Autor)

4.2.2.7. Conexión de la Raspberry Pi 3B y el Arduino Mega 2560

La señal de activación es enviada por la tarjeta Raspberry Pi 3B con el pin de salida 8, la cual envía un pulso para que se active el módulo GPRS y envíe un SMS al Smartphone, el Arduino recibe la señal en el pin 7. A continuación como se muestra en la Figura 79 las tarjetas Raspberry Pi 3B y el módulo Arduino Mega 2560 “ver Anexo 10”.

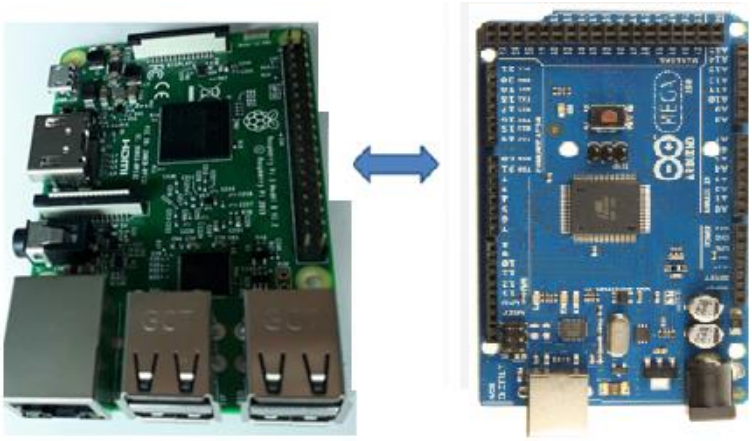


Figura 79. Tarjeta Raspberry Pi 3B y Arduino Mega 2560

Fuente: (Store, 2017)

4.2.2.8. Conexión de los componentes de salida

Los componentes que conforman esta conexión es la sirena, seguro de las puertas y bloqueo del encendido, los cuales están conectadas a la GPIO de la tarjeta Raspberry Pi 3 B y se los indica en la Tabla 13.

Tabla 13: **Pines de conexión componentes de salida**

Componente	PIN (GPIO)
Sirena	21
Seguro de las puertas para abrir	23
Seguro de las puertas para serrar	24
Puerta trasera	25
Bloqueo del encendido	17

Fuente: (Elaborado por el Autor)

Para la etapa de potencia se utilizó un módulo de relés de cuatro canales, los cuales están conectados los seguros de las puertas que activa y desactiva la sirena y el bloqueo de encendido, en la Tabla 14 se muestra las entradas y salidas a las cuales están conectadas. Para conectar los dispositivos de entrada y salida del sistema prototipo se utilizó el plug de 5 pines macho conector hembra como se observa en la Figura 80.

Tabla 14: **Pines de conexión componentes de salida**

Componente	RELE
Bloqueo del encendido	IN 1 (NC)
Sirena	IN 2 (NA)
Seguro de las puertas para abrir	IN 3 (NA)
Seguro de las puertas para cerrar	IN 4 (NA)

Fuente: (Elaborado por el Autor)

Figura 80. Plug 5 Pines Macho Conector Hembra en el Panel de Alambre De Metal 16mm GX16-5

Fuente: (Elaborado por el Autor)

Se utilizaron 4 plug que están enumeradas en la caja a continuación, en la Tabla 15 se muestra los pines de conexión de los dispositivos de entrada y salida del sistema prototipo “ver Anexo 2”.

Tabla 155: **Pines de conexión de entrada y salida**

PIN DEL PLUG	PLUG 1	PLUG 2	PLUG 3	PLUG 4
Pin	***Relé***	***Dispositivo***	***E/S***	***Señal puertas***
1	IN 4 (C)	Sensor +	Señal capo	Cajuela
2	IN 4 (NA)	Señal sensor	IN 1 (NC)	Chofer
3	IN 3 (C)	----	Común sirena IN 2 (+12V)	Copiloto
4	IN 3 (NA)	Batería -12V	Señal sirena IN 2 (NA)	P. Trasera izquierda
5	IN 1 (C)	Batería +12V	Sensor GND	P. Trasera derecha

Fuente: (Elaborado por el Autor)

4.2.2.9. Batería y regulador de voltaje.

La batería es la encargada proporcionar energía eléctrica al prototipo, la cual es recargada por el generador del vehículo o de la batería propia del vehículo con un voltaje de 12 voltios a 73 Amperios. La Batería propia del prototipo tiene un voltaje 12 Voltios a 7 Amperios a la cual se conecta a un regulador de voltaje para disminuir al voltaje a 5 voltios “ver Anexo 3” como se indica en la Figura 81 y conectarla a la tarjeta Raspberry Pi 3B.

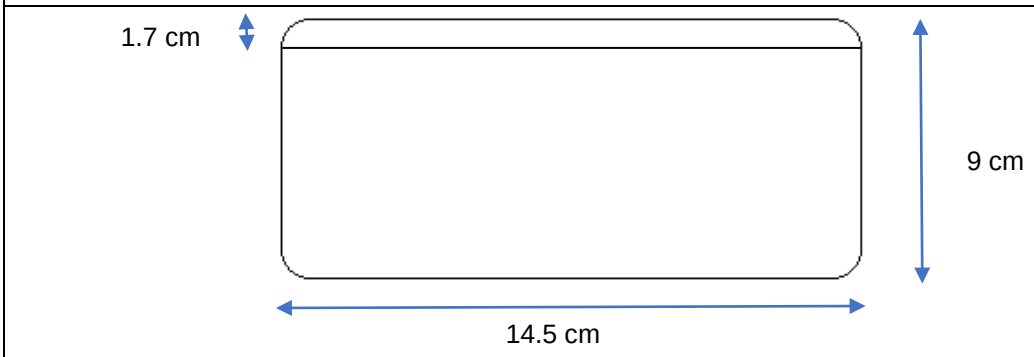
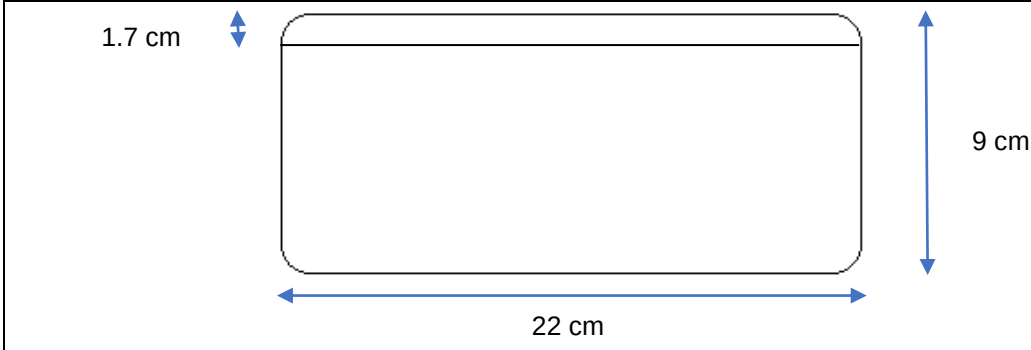
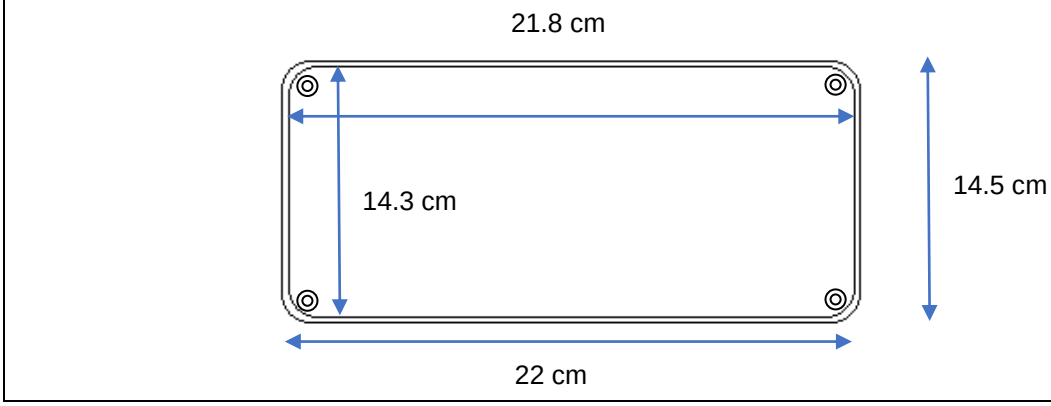
Figura 81. Conexión batería y regulador de voltaje

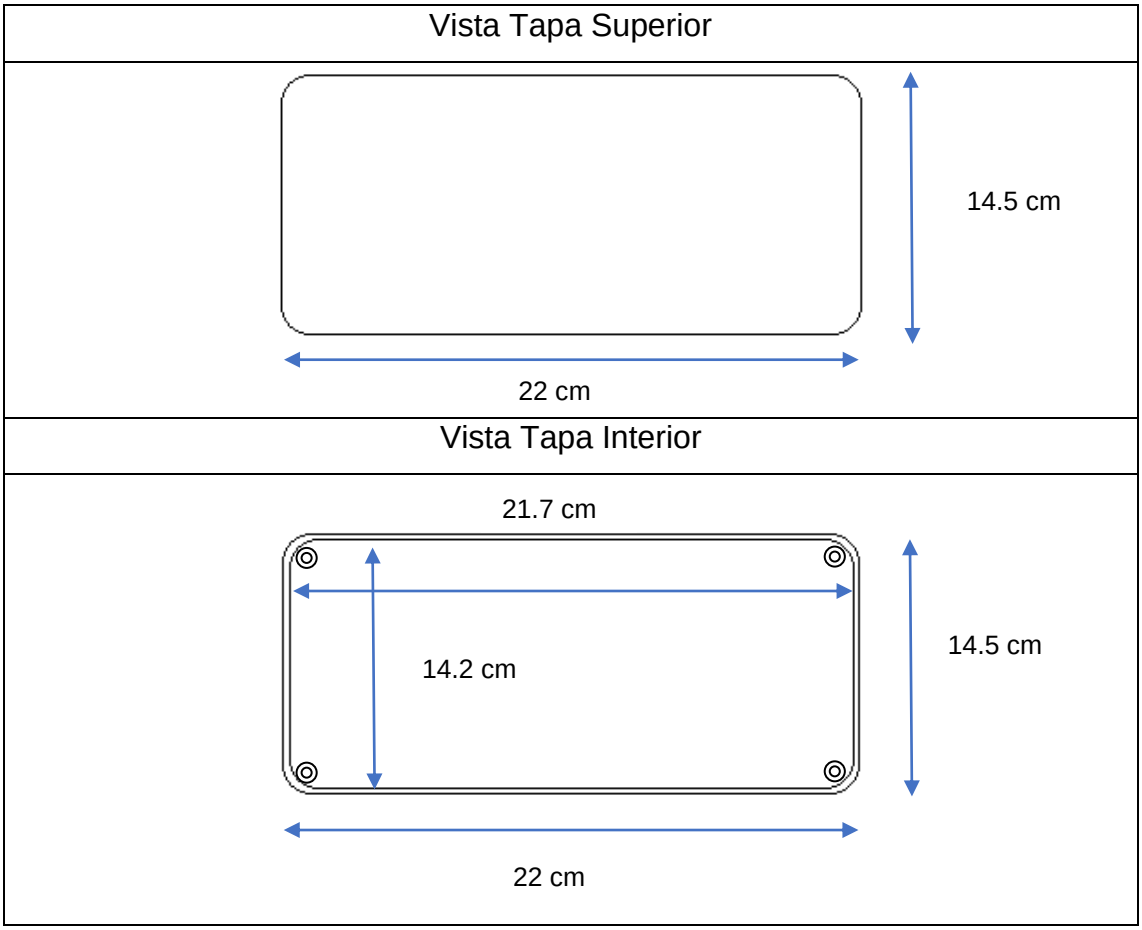
Fuente: (Elaborado por el Autor)

4.2.2.10. Caja del prototipo

Es de material pastico PVC, resistente al agua, con un grosor de 1.5 milímetros, se ajusta con cuatro tornillos a los extremos que aseguran la tapa, para evitar la manipulación; en la cual se instalan todos los elementos, eléctricos, electrónicos, etapas de potencia, bus de datos, cableado de alimentación etc., se muestra en la Tabla 16.

Tabla 16: **Pines de conexión de entrada y salida**

<table>
<tr><td>Vista Frontal</td></tr>
<tr><td> 
</td></tr>
<tr><td>Vista Lateral</td></tr>
<tr><td> 
</td></tr>
<tr><td>Vista Interior</td></tr>
<tr><td> 
</td></tr>
</table>



Fuente: (Elaborado por el Autor)

El sistema prototipo general se implementa con todos sus componentes se lo ilustra en la Figura 82, con el cual se va a realizar las respectivas pruebas de funcionamiento para posterior instalación en el vehículo Chevrolet Grand Vitara para más detalle “ver Anexo 11”.

Figura 82. Sistema general implementado para pruebas

Fuente: (Elaborado por el Autor)

4.2.3. Implementación de la aplicación Smartphone

Al implementar la interfaz gráfica de cada una de las pantallas en la aplicación se hace mediante App Inventor basada su programación en bloques, en la Figura 83 se observa la manera de ingresar a la herramienta de programación que se basa en bloques.

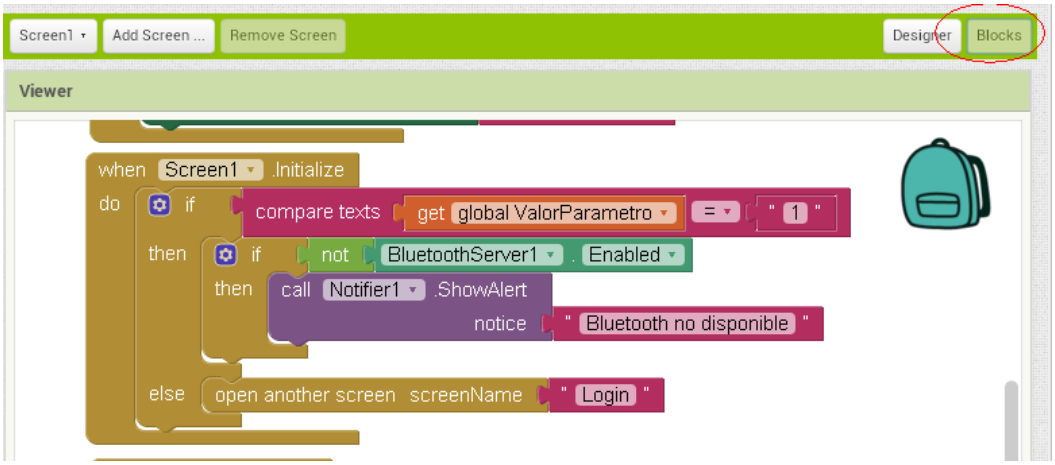


Figura 83. Ingreso a la herramienta para implementar

Fuente: (Elaborado por el Autor)

En el primer paso se realiza la programación de inicio del Screen1 que es la pantalla de inicio y que corresponde a la opción de *Login*. Al ingresar a la aplicación, la pantalla se inicializa en la cual se muestran cuadros de textos donde se debe llenar los campos de usuario y la PASSWORD.

En el segundo paso se procede hacer una lectura de la base de datos *TinyBD1,* donde se verifica que se encuentra vacío el tag contraseña y el usuario, si es correcto entonces se almacena una clave por defecto que servirá si en alguna ocasión se decidiera reinstalar la aplicación o se instala en un Smartphone diferente pero se conserva el mismo número celular el cuál es registrado al sistema del prototipo, en la Figura 84 se observa la programación de bloques para la pantalla de inicio de la aplicación Smartphone.

En el instante de hacer click el programa compara los datos ingresados de usuario y password con los almacenados en la base de datos, si la validación

de los datos es correcta, primero se limpia el campo de texto, luego se oculta el teclado en pantalla y finalmente se abre el Screen1 que corresponde a la pantalla de acciones como se muestra en la Figura 85, si fuese incorrecta la comparación se muestra en pantalla una alerta con el mensaje Contraseña incorrecta.

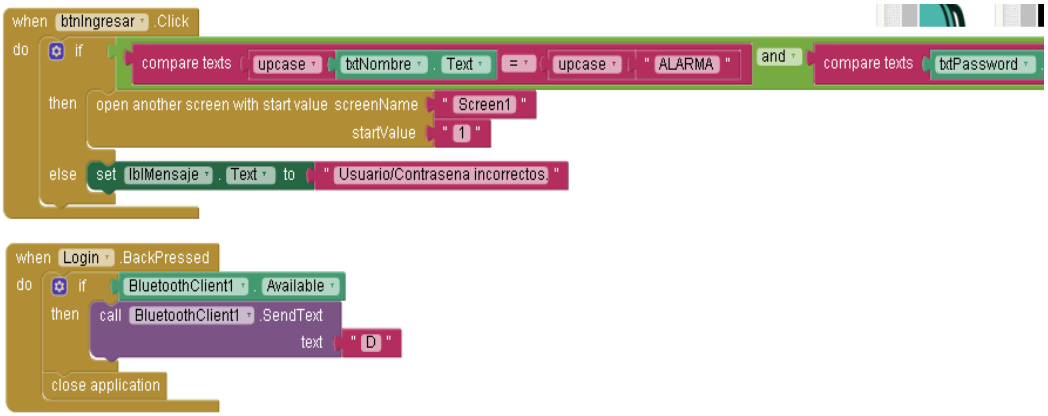


Figura 84. Implementación de la pantalla principal de inicio

Fuente: (Elaborado por el Autor)

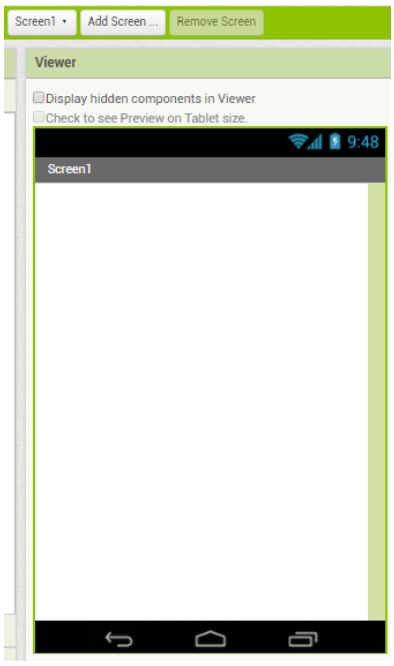


Figura 85. Pantalla Screen 1

Fuente: (Elaborado por el Autor)

El siguiente paso es la programación de la pantalla de screen1, en la opción de screen1 que corresponde a la pantalla de acciones como se muestra en la

Figura 86, se procede a programar de manera ordenada y se elige cada uno de los bloques correspondientes a cada acción que se desee que cumpla la aplicación, como se describe a continuación:

Cuando se inicializa la pantalla de acciones muestra el diseño de la pantalla secundaria, esto se hace debido a que anteriormente se ingresó una contraseña y el usuario en pantalla con ayuda del teclado virtual, la Figura 86 muestra el programa de bloques de la conexión a bluetooth y los botones de encendido y apagado del sistema, al hacer clic en el botón de encendido y apagado hace una comparación entre todas las casillas de verificación correspondientes a cada una de las acciones que ofrece el prototipo, entonces al momento de elegir una acción, la aplicación muestra una alerta en pantalla y así se confirma la acción seleccionada.

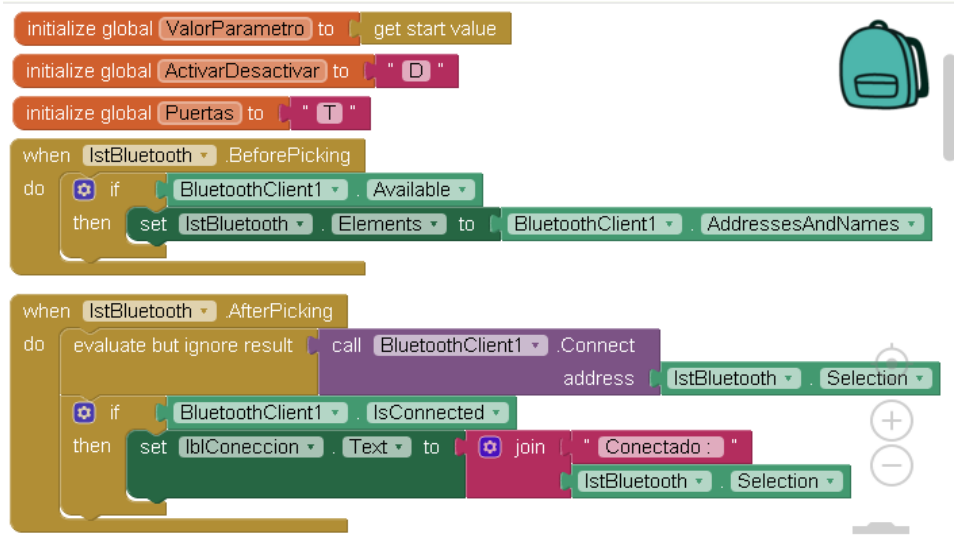


Figura 86. Implementación del botón de activación y desactivación

Fuente: (Elaborado por el Autor)

Una vez que el sistema está activado entra en funcionamiento el monitoreo de las puertas del vehículo en el estado que se encuentre si es cerrado o abierto, lo cual se ilustra en la Figura 87 el bloque de programación para esta función.

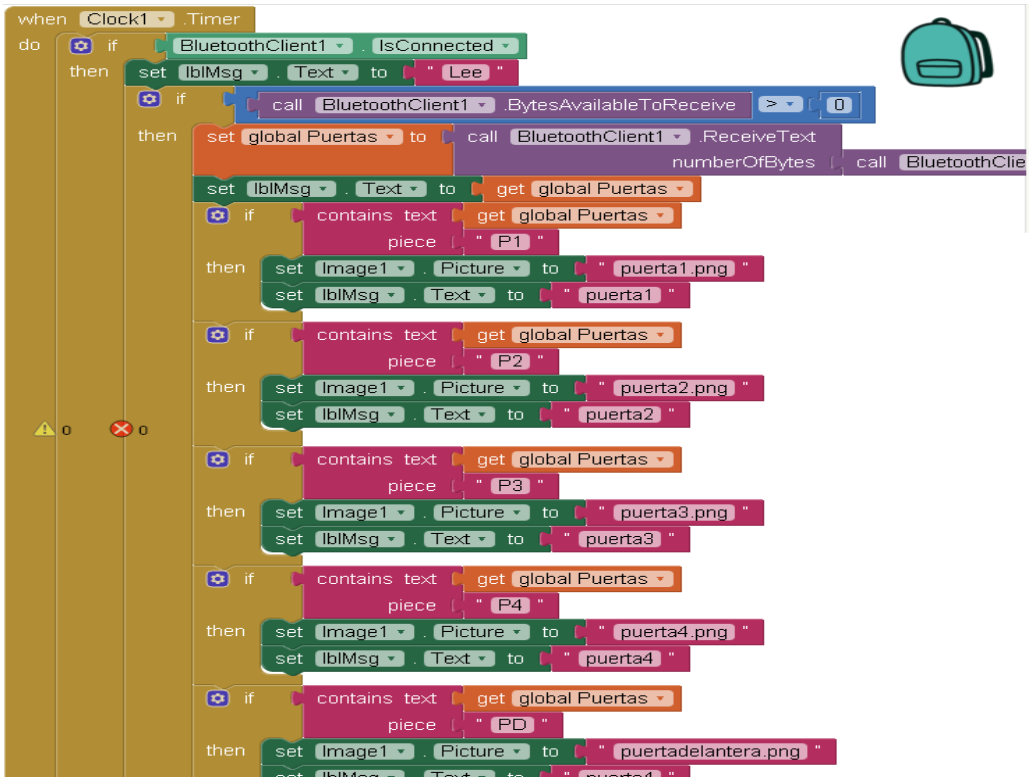


Figura 87. Implementación del estado de las puertas

Fuente: (Elaborado por el Autor)

Finalmente se programa del botón desactivar, se crea la función de desconectarse con el uso de bluetooth y con la tarjeta Raspberry Pi 3B mantiene en la pantalla de acciones “ver Anexo 8”, el proceso de programación se observa en la Figura 88.

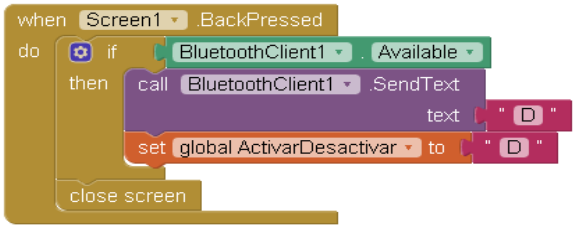


Figura 88. Implementación del botón desactivar

Fuente: (Elaborado por el Autor)

4.3. Pruebas de funcionamiento

4.3.1. Monitoreo.

Las pruebas se realizan en una determinada área sin obstáculos debido a que la tecnología bluetooth es de corto alcance, se considera que es un ambiente

controlado, ya que permite realizar las pruebas descritas en esta sección, en la Figura 89 se describe el escenario para las pruebas del prototipo inalámbrico, para posterior ser implementado.

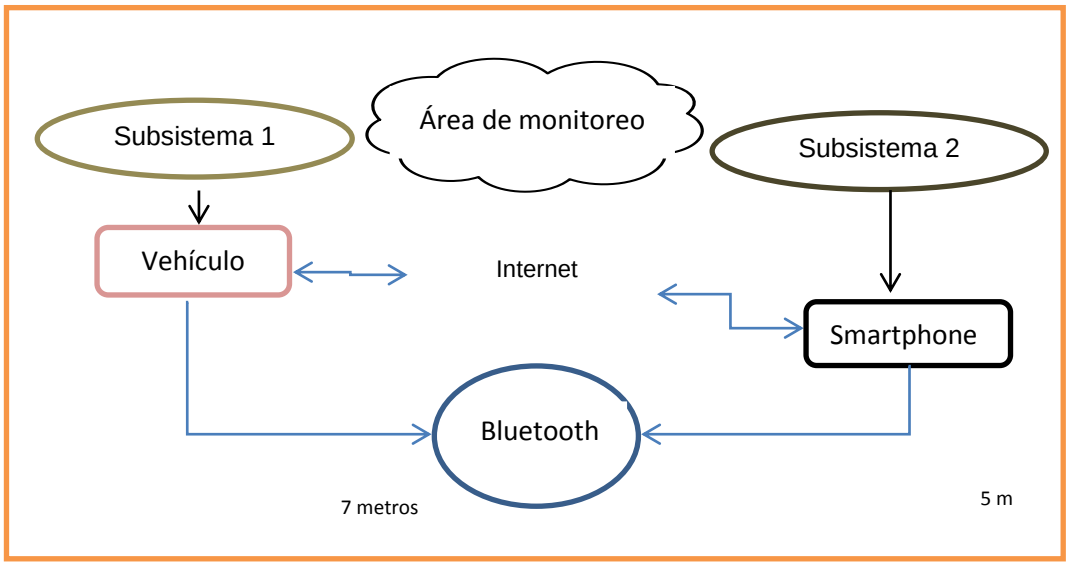


Figura 89. Escenario para las pruebas de funcionamiento

Fuente: (Elaborado por el Autor)

4.3.2. Pruebas de encendido del sistema

Al sistema se lo enciende utilizando dos tecnologías, el primero mediante la aplicación del Smartphone por medio de la conexión bluetooth y la segunda opción es mediante el control remoto inalámbrico del módulo RF al pulsar el botón de encendido del sistema la alarma se activa en una distancia entre 1 a 1.50 metros como límite de transmisión “ver Anexo 10”.

4.3.2.1. Prueba de funcionamiento en base a la aplicación del Smartphone

En la Figura 90 se valida el encendido del sistema mediante la aplicación en el Smartphone y la ejecución del programa de inicio. Para esto se deben inicializan todas las sentencias que lo forman, del tal forma que es necesario activar el sistema primero mediante el bluetooth del Smartphone y sincronizarlo con el bluetooth de la tarjeta Raspberry Pi 3B, caso contrario el sistema no se activará.

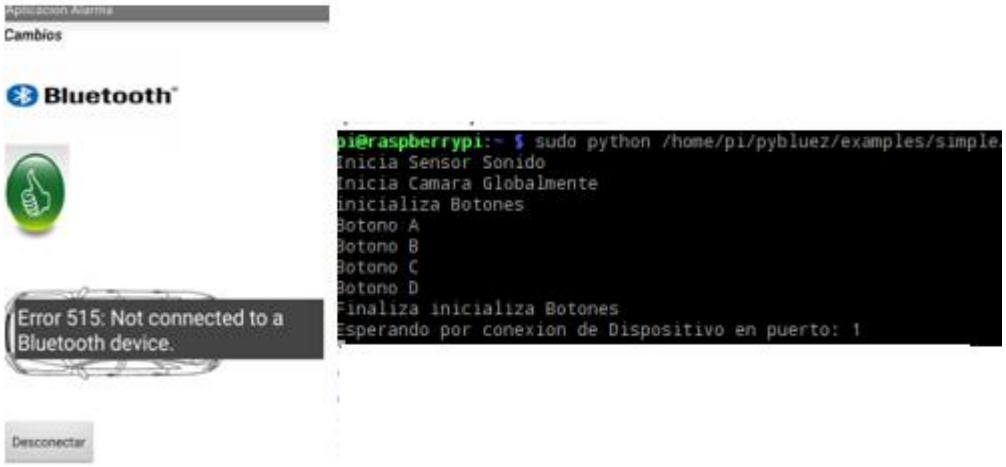


Figura 90. Entorno de programación y encendido del sistema mediante el Smartphone

Fuente: (Elaborado por el Autor)

Una vez encendido el sistema, se abren los seguros de las puertas y se bloquea el encendido del vehículo.

4.3.2.2. Prueba de funcionamiento en base al módulo RF YK04

La Figura 91 muestra el entorno de programación, como parte de sus procesos se detallan los nombres de los botones del módulo RF. Se valida las siguientes acciones: mediante el botón A se activa, se desactiva mediante el botón C, el botón B activa el modo silencio y el botón D es un respaldo del botón A como emergencia.

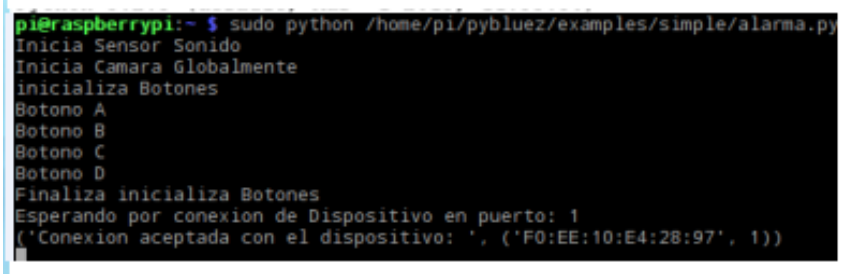


Figura 91. Entorno de programación y botones del módulo RF

Fuente: (Elaborado por el Autor)

4.3.2.3. Prueba de funcionamiento del sensor de sonido

Una vez activado el sistema se valida que entre en estado de alerta, inmediatamente se verifica el comportamiento todos los sensores de las puertas ante cualquier eventualidad. Si el sensor de sonido detecta ruido se

alarma el sistema, procede a ejecutar del programa como se observa en la Figura 92, se activa la sirena y la cámara el cuál toma una foto para posterior se enviada por correo electrónico, de la misma manera se enviada un SMS con la ubicación geográfica.

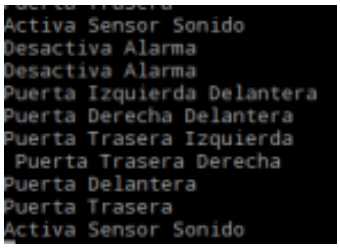


Figura 92. Prueba de funcionamiento del sensor de sonido

Fuente: (Elaborado por el Autor)

4.3.2.4. Prueba de funcionamiento pulsadores de las puertas

Se utilizan 6 pulsadores para las puertas del vehículo Chevrolet Grand Vitara, se valida que al instante de abrir una puerta se activa la sirena y la cámara la misma que captura una imagen y la envía por correo electrónico, así como también por medio de un mensaje de texto SMS se manda el detalle la ubicación geográfica del vehículo. En la Figura 93 se ilustra el funcionamiento de la apertura del capot, la cajuela y los proceso que se ejecutan.

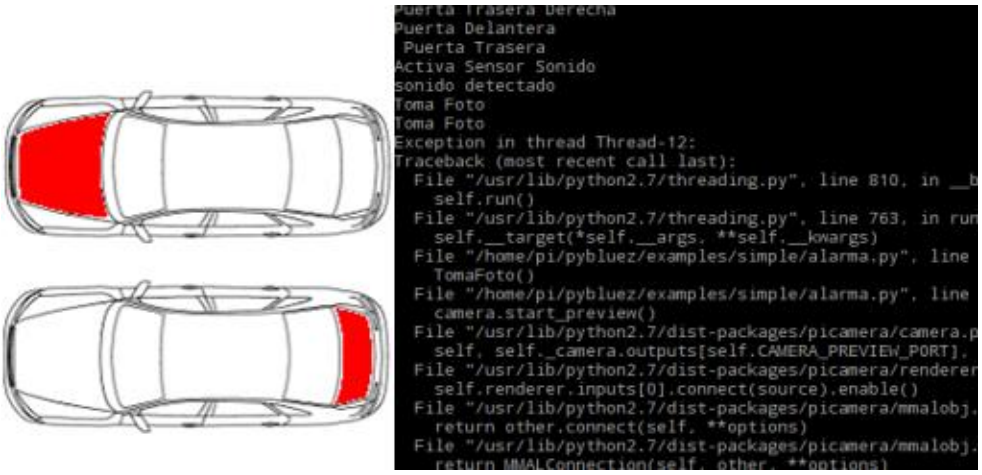


Figura 93. Prueba de funcionamiento del capot y cajuela

Fuente: (Elaborado por el Autor)

En la siguiente Figura 94 se valida el comportamiento de los procesos cuando se activa las puertas laterales delanteras y las puertas laterales traseras.

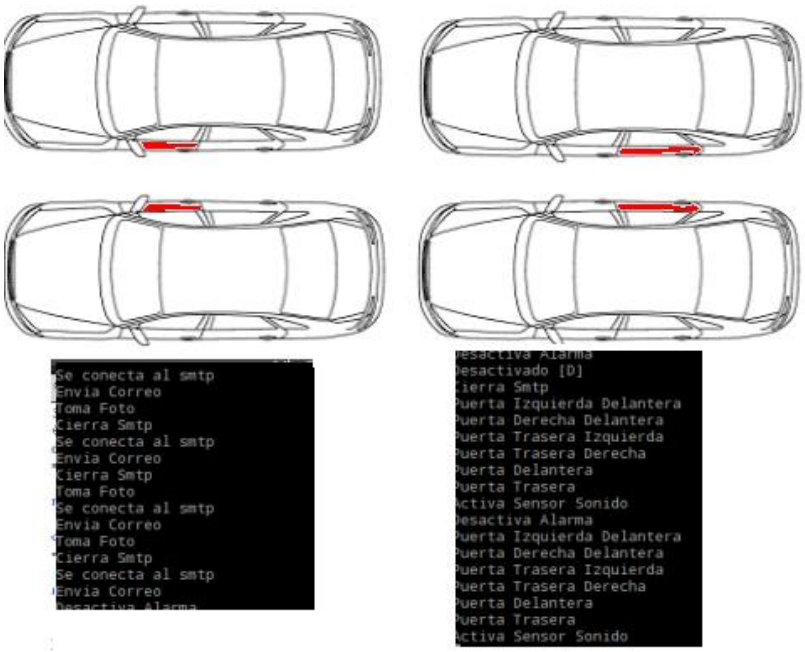


Figura 94. Prueba de funcionamiento puertas laterales delanteras y laterales traseras

Fuente: (Elaborado por el Autor)

4.3.2.5. Prueba de funcionamiento cámara

En el instante que ocurre una alarma del sistema se ingresa a la web por medio de un Smartphone con conexión a internet, en el cuál el usuario va a observar la imagen capturada del evento interno que esté ocurriendo en ese momento, en la Figura 95 se observa el mensaje al ingresar al correo electrónico asignado.

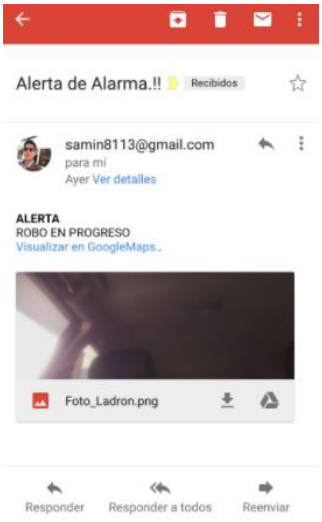


Figura 95. Monitoreo de la cámara

Fuente: (Elaborado por el Autor)

4.3.2.6. Prueba de funcionamiento modulo SIM 908 GPRS 3G

De la misma manera se valida que la cámara en el instante que se alarma el prototipo envía un SMS con la ubicación geográfica donde se encuentre el vehículo el cuál se puede monitorear atreves del Smartphone, en la figura 96 se observa el mensaje enviado el cuál esta como un link para visualizar con la aplicación de google map.

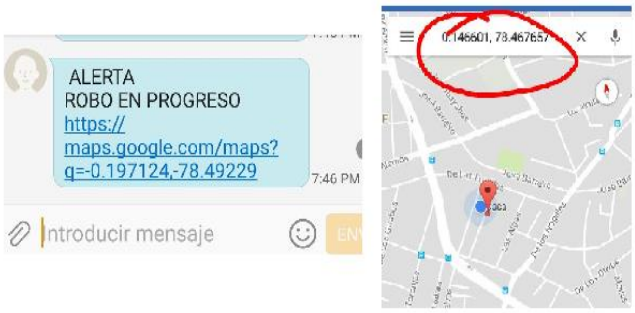


Figura 96. Monitoreo GPS por medio de SMS

Fuente: (Elaborado por el Autor)

4.4. Análisis de resultados

A continuación, en la Tabla 17 se detalla las observaciones del análisis y comportamiento de los componentes del prototipo inalámbrico.

Tabla 17: **Resultados del sistema prototipo**

Componente	Encendido	Comunicación	Observación
Cámara	OK	OK	Para mejor visualización la cámara debe de estar en un lugar despejado
Sensor de sonido	OK	OK	Para una mejor detección el sensor de sonido debe estar ubicado en un lugar sin bloqueos.
Respaldo de Baterías 12 voltios 7 amp	OK	OK	48 horas al 100%.
Raspberry Pi 3B	OK	OK	Ninguna

Módulo RF YK04	OK	OK	La señal del módulo es media debido a las características propias del módulo.
Llamada telefónica	OK	OK	Depende de la operadora.
Correo electrónico	OK	OK	Se debe tener acceso a internet
Mensaje de texto	OK	OK	Depende de la cobertura de la operadora.
Smartphone	OK	OK	Ninguna
Modulo GPS	OK	OK	Depende del espacio libre, exteriores
Modulo Relés	OK	OK	Ninguna

Fuente: (Elaborado por el Autor)

En la Figura 97, se analiza el ciclo de interaccion entre el prototipo y las diferentes acciones a realizar por el sistema de video vigilancia. Una ves sincronizado la aplicación android con la tarjeta Rapsbery Pi 3B, este permite la ejecucion deteccion de sonido, realizar el monitoreo del estado de las puertas, envio de imagen en tiempo real y de la ubicación donde se encuentre.

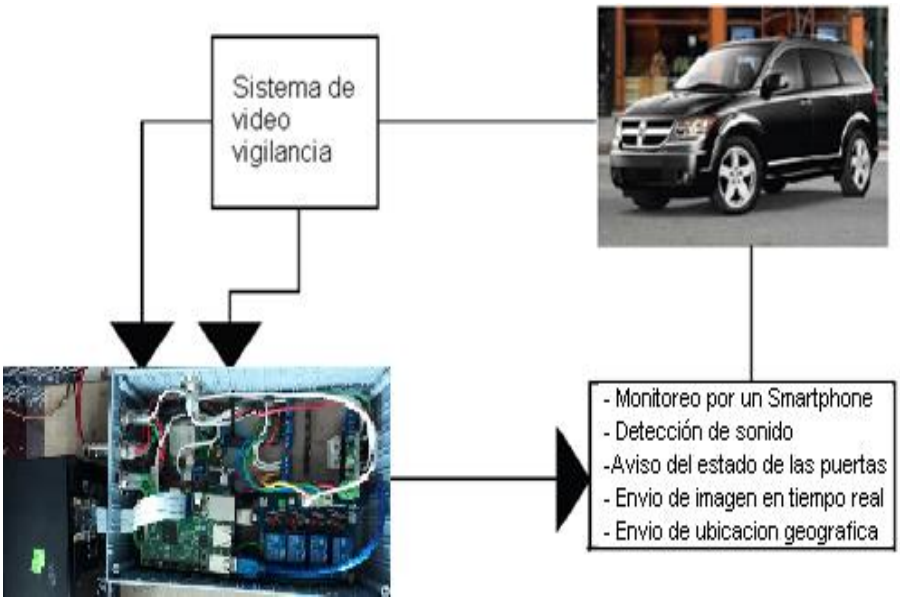


Figura 97. 98Análisis del sistema de video vigilancia para el vehículo

Fuente: (Elaborado por el Autor)

En la Tabla 18, se muestra las pruebas de funcionamiento de los elementos que forman parte del prototipo, dichas pruebas se hacen a varias distancias ya que las tecnologías que se utiliza son de corto alcance.

Tabla 18: **Pruebas de funcionamiento de los elementos.**

No	Prueba	Pb 1	Pb 2	Pb 3	Pb 4	Pb 5	Pb 6	Pb 7	Pb 8	Pb 9	Pb 10	Porcentaje
1	¿Conecta – desconecta bluetooth?	√	√	√	√	√	√	√	√	√	√	100
3	¿Botón activa?	√	√	√	√	√	√	√	√	√	√	100
4	¿Botón desactiva?	√	√	√	√	√	√	√	√	√	√	100
5	¿Puerta izquierda?	√	√	√	√	√	√	√	√	√	√	100
6	¿Puerta derecha?	√	√	√	√	√	√	√	√	√	√	100
7	¿Puerta p izquierda?	√	√	√	√	√	√	√	√	√	√	100
8	¿Puerta p derecha?	√	√	√	√	√	√	√	√	√	√	100
9	¿Puerta capot?	√	√	√	√	√	√	√	√	√	√	100
10	¿Puerta cajuela?	√	√	√	√	√	√	√	√	√	√	100
11	¿Mensaje de texto?	√	√	√	√	√	√	√	√	√	√	100
12	¿Llamada?	√	√	√	√	√	√	√	√	√	√	100
13	¿Correo electrónico?	√	√	√	√	√	√	√	√	√	√	100
14	¿Batería?	√	√	√	√	√	√	√	√	√	√	100
15	¿Sistema Automático?	√	√	√	√	√	√	√	√	√	√	100
16	¿Desconectar?	√	√	√	√	√	√	√	√	√	√	100
17	¿Modulo RF?	√	√	x	X	x	x	X	X	x	X	100
18	¿Modulo GPS?	√	√	√	√	√	√	√	√	√	√	100

	100,00%

Fuente: (Elaborado por el Autor)

En la Tabla 19, se indica el número de pruebas de funcionamiento que se realizó al sistema así validando el porcentaje de efectividad de procesos que realizan acciones determinadas.

Tabla 19: **Pruebas de funcionamiento del sistema.**

No	Prueba	Pb 1	Pb 2	Pb 3	Pb 4	Porcentaje
1	¿Abre la aplicación en el teléfono inteligente?	√	√	√	√	100
3	¿La aplicación permite ingresar con usuario y contraseña?	√	√	√	√	100
4	¿El teléfono inteligente se sincroniza con el prototipo?	√	√	√	√	100
5	¿Existe envió y recepción de datos entre el teléfono inteligente y el prototipo?	√	√	√	√	100
6	¿El control RF permite el encendido, apagado y modo silencio de la alarma?	√	√	√	√	100
7	¿Al activarse la alarma, se enciende el bloqueo central del vehículo impidiendo que este arranque?	√	√	√	√	100
8	¿El instante que se activa la alarma, se enciende la cámara y toma fotos de los eventos internos los mismos que son enviados por correo electrónico?	√	√	√	√	100
9	¿El momento que se activa la alarma, el prototipo realiza la llamada hacia el teléfono inteligente?	√	√	√	√	100
10	¿Cuándo se corta la energía de la fuente principal del prototipo, entra funcionar la batería de respaldo?	√	√	√	√	100

100%

Fuente: (Elaborado por el Autor)

4.5. Presupuesto económico

A continuación, se detalla el costo económico de cada uno de los materiales utilizados y también del costo del tiempo utilizado para el diseño y la implementación “ver Anexo 12”.

Tabla 20: **Costos referenciales del sistema prototipo.**

EQUIPO	CANTIDAD	COSTO UNITARIO ($)	COSTO TOTAL ($)
Tarjetas Raspberry Pi 3B Modelo B	1	45	45
Cámara de video Raspberry Pi 100003 / extensión cable	1	31	31
Sensor de ruido	1	7	7
Modem USB 3G Huawei	1	14	14
Arduino Mega 2560	1	16	16
SIM 908 GPRS	1	80	80
Control remoto RF módulo YK04	1	7	7
Módulo de relés de 4 canales	1	5	5
Regulador de voltaje	3	6	18
Batería 12V a 7Ah	1	21	21
Tarjeta micro SD	1	15	15
Bornes de unión	9	2	18

Caja plástica	1	12		12
Chip CNT	2	3		6
Cable plano 40 pines	1	6		6
Otros	1	5		5
Tiempo de diseño	1	50		50
Tiempo de implementación	1	50		50
Computadora	1	50		50
			TOTAL	526.00$

Fuente: (Elaborado por el Autor)

CONCLUCIONES Y RECOMENDACIONES

Conclusiones

- Durante el estudio se determinó que en el Ecuador entre el año 2016 y 2017, se tuvieron un crecimiento de índice de robos. En el 2016 el índice de robos fue de 35% versus el año 2017, que al corte del mes de marzo fue del 45% esto es un problema latente. Basado en este resultado y revisando que le diseño como tal tendrá un costo inferior a sistemas actuales en el mercado y con las mismas características técnicas operativas y demás. Se concluye que tanto el estudio arroja la necesidad como en el ámbito técnico y económico el proyecto es viable.

- Dentro de la etapa de diseño se utilizó una tarjeta Raspberry Pi 3B como cerebro del sistema, este a su vez se lo complementa con la inserción de sensores, actuadores etc., de tal forma, se prevee que, dentro del alcance del proyecto, se disponga de varias posibilidades para conectar los seguros y la apertura de las puertas, el bloque central, sensores externos, sirena y módulo GPRS, que suman siete elementos electrónicos. Dichos elementos electrónicos versus los dispositivos disponibles en el mercado, se tienen algunas ventajas. Por lo tanto, el diseño está orientado ampliar el sistema, con mejores características.

- Dentro de la etapa de implementación se tiene que manejar dos niveles de voltaje. Se tiene que dejar un voltaje para el sistema de alarma e integrarlo al sistema del vehículo, lo cual generó un inconveniente. Por lo tanto, se determina el uso de una batería externa independiente para alimentación del sistema y está a su vez se carga a mediante el sistema eléctrico del vehículo, en el caso de algún corte el sistema anti robo sigue teniendo autonomía de energía. Mediante cálculos se

obtiene un tiempo de funcionamiento mínimo de 41,26 horas y un máximo de 53.05 horas, con un límite de las 48 horas.

- Se realizan pruebas de cobertura 3G de las operadoras presentes en el Ecuador, en la cual movistar con una cobertura del 85% en exteriores y 45% en interiores, claro tiene una cobertura de 95% en exteriores y de 65% en interiores, cnt tiene una cobertura de 95% en exteriores y de 78% en interiores. Por lo tanto, se determina que al optar el uso de la operadora cnt no se tendría problemas de cobertura al momento de implementar esta solución dentro del sistema.

- En la etapa de pruebas inicial se recogieron algunos problemas, los cuales se modificaron posteriormente como son: ajuste de antenas, ajuste de sensibilidad del sensor de ruido, sincronismo en a comunicación tanto vía bluetooth como en radio frecuencia, etapas de control con los relés, etc. Con lo el sistema estaba a un 70% de operación. De lo cual en el último proceso de pruebas se alcance un 98% de operabilidad por el problema de GPS, siendo este 2% de diferencia respecto al 100% debido a conectividad celular que son factores externos al sistema. Por lo tanto, el sistema está operando de forma correcta.

- Se concluye que, mediante el monitoreo del sistema, con la aplicación Android en un teléfono inteligente, que permite la visualización de los eventos internos al momento que se presenten, tales como los estados de los seguros de las puertas, posición del vehículo y así también permite activar y desactivar los seguros.

- El monitoreo con la aplicación móvil tuvo resultados favorables, ya que en pruebas apenas una de ellas tuvo intermitencia. Por lo tanto, de estas pruebas se rescata que la disponibilidad de la aplicación como tal bajo ciertas características en normal o aproximada al estándar con la que se maneja en las aplicaciones móviles.

Recomendaciones

- Se recomienda ubicar el sensor de sonido en un lugar que sea estático, libre de obstáculos, lo más próximo de una ventana, con esto se asegura que las lecturas del mismo sean precisas el momento de monitorear.

- Se recomienda utilizar un regulador de voltaje de 5 voltios el mismo que va a enviar constantes 5 voltios a la tarjeta Raspberry Pi 3B. Con lo cual se evitará fluctuaciones tanto de la tarjeta como de los sensores, lo que previene que los mismos tengan daños irreversibles.

- Para el caso de las Tablet se recomienda tener una versión superior de Android 3.0 (honeycomb), y para smartphones se recomienda una versión superior de Android 4.0 (ice cream sandwich). En versiones anteriores no se garantiza el buen funcionamiento de la aplicación.

- Es recomendable mantener un plan de datos contratado para acceder a internet mediante el módulo USB 3G, de esa manera se podrá enviar los mensajes al correo electrónico por medio de la Raspberry pi modelo 3B.

REFERENCIAS BIBLIOGRAFÍAS

(2012-2017). Obtenido de https://es.aliexpress.com/?aff_platform=link-c-tool&cpt=1503889857105&sk=Qr7YZfi&aff_trace_key=1d58b9b0f23448cc82c4de2c4f3ec067-1503889857105-08481-Qr7YZfi

(2017). Obtenido de https://articulo.mercadolibre.com.ec/MEC-412846597-alarma-nemesis-bluetooth-controlado-desde-el-celular-_JM

4gltemall. (25 de Marzo de 2017). *HUAWEI HiLink E3251 DC-HSPA+ USB Stick.* Recuperado el 26 de Julio de 2017, de https://www.4gltemall.com/huawei-hilink-e3251-dc-hspa-usb-stick.html

Appinventor. (30 de junio de 2017). *MIT App Inventor + Internet of Things*. Recuperado el 30 de Julio de 2017, de http://appinventor.mit.edu

Aráuz, F. H. (1 de Octubre de 2017). Obtenido de http://www.ecuadorinmediato.com/index.php?module=Noticias&func=news_user_view&id=199981

Arduino. (27 de Junio de 2014). *Arduino Mega 2560*. Recuperado el 20 de Julio de 2017, de http://arduino.cc/en/Main/ArduinoBoardMega2560

Arduino. (26 de September de 2015). Obtenido de https://create.arduino.cc/projecthub/bigboystoys13/diymall-gprs-gps-sim908-module-at-commands-6a8dda?ref=platform&ref_id=424_trending___&offset=95

ARDUINO MEGA 2560. (2017). Obtenido de ARDUINO: https://www.arduino.cc/en/Main/arduinoBoardMega2560

Arduino, V. S.-1. (1995 - 2017). *http://www.ebay.es*. Obtenido de http://www.ebay.es: http://www.ebay.es/itm/Vibration-Sensor-Module-SW-18015P-sensor-vibracion-Arduino-/272275064743

Basterra. (2012). *Android OS*. Obtenido de http://androidos.readthedocs.io: http://androidos.readthedocs.io/en/latest/data/caracteristicas/

Basterra Borello, C. V. (2017). *Android OS Documentation.* readthedocs.

Brown, P. (2016). *Raspberry Pi GPIO Pinout.* Recuperado el 12 de Junio de 2016, de https://github.com/Microsoft-Build-2016/CodeLabs-IoTDev/blob/master/Module4-OpenHack/GPIO.md

Bureau, I. A. (8 de Agosto de 2016). Obtenido de http://www.iabspain.net/wp-content/uploads/downloads/2014/07/Informe-coches-conectados-2014.pdf

Cachiguango, Y. (2010). *Diseño de una red de video vigilancia local y remota sobre ip en tiempo real para una hosteria aplicanod el concepto de Green IT .* Quito: EPN.

CARSEG. (2011). Obtenido de http://www.hunter.com.ec/inicio.aspx

Cedatos. (27 de Septiembre de 2017). Obtenido de http://www.cedatos.com.ec/detalles_noticia.php?Id=86

Cerda, D., & Pazmiño, I. (2011). *Diseño e implementación de un sistema con GPS y control de seguridad vehicular con comunicación GSM. Diseño e implementación de un sistema con GPS y control de seguridad vehicular con comunicación GSM.* Latacunga, Ecuador.

CHEVROLET. (s.f.). Obtenido de http://www.chevrolet.com.ec/chevystar/mi-chevystar.html

COMERCIO, E. (20 de Diciembre de 2012). Obtenido de http://www.elcomercio.com/actualidad/seguridad/robo-de-vehiculos-registro-incremento.html

COMERCIO, E. (10 de Febrero de 2016). Obtenido de http://www.elcomercio.com/actualidad/barrios-robos-vehiculos-quito-delincuencia.html

Corporation, W. D. (2017). Obtenido de https://www.sandisk.es/home/memory-cards/microsd-cards/ultra-microsd-48mbs

Crespo, E. (25 de Septiembre de 2016). *Aprendiendo Arduino*. Recuperado el 27 de Julio de 2017, de https://aprendiendoarduino.wordpress.com/category/software/

Cuadros, D. (2016). *Diseño de un sistema de seguridad antirrobo de video vigilancia vehicular mediante el control de una aplicación smartphone.* Quito: UTE.

Dalmegnu. (21 de Septiembre de 2016). *Iniciación a la programación en Lenguaje C*. Recuperado el 30 de Julio de 2107, de http://dalmegnu.amayaos.com/2016/09/21/iniciacion-a-la-programacion-en-lenguaje-c/

Ecuador, M. (2017). Obtenido de https://listado.mercadolibre.com.ec/accesorios-autos/laminas-de-seguridad

Ecuador, M. (2017). Obtenido de https://articulo.mercadolibre.com.ec/MEC-412589324-alarma-nemesis-serie-max-manejo-basico-y-avanzado-_JM

ECUADORINMEDIATO. (27 de Septiembre de 2017). Obtenido de http://www.ecuadorinmediato.com/index.php?module=Noticias&func=news_user_view&id=149111

Ecured. (25 de Junio de 2017). *Lenguaje de Programación Ruby*. Recuperado el 30 de Julio de 2017, de https://www.ecured.cu/Lenguaje_de_Programaci%C3%B3n_Ruby

Elcajondeardu. (20 de Septiembre de 2015). *Guía básica de utilización de sensor de sonido con Arduino* . Recuperado el 20 de Julio de 2017, de http://elcajondeardu.blogspot.com/2015/09/tutorial-sensor-de-sonido.html

Electronics, A. (2014).

Faican, R. (2016). *Diseño e implementacion de un prototipo inalambrico de monitoreo a travez de seuaencias de imagenes utilizando hadware y software libre.* Quito: EPN.

Faranux. (27 de Septiembre de 2016). *4 Channels RF Remote Control Module YK04.* Recuperado el 30 de Julio de 2017, de http://www.faranux.com/product/4-channels-rf-remote-control-module-yk04/

García, D. M. (2016). *Sistema de navegación para robots móviles basado en un ordenador de placa simple.* Valencia-España: Universidad Politécnica de Valencia.

Globaltronic. (2014). Obtenido de http://nemesisla.com/alarmas.html

Gómez, J. (2011). *Sistema de ecuperacion de información sobre trabajos terminales SpinnenneTTz .* México: ESCOM.

González, C., & García, S. (2015). *Curso de App Inventor.* Ciudad Real - España: Universidad de Castilla-La Mancha.

Google. (17 de Abril de 2017). Obtenido de https://www.google.com.ec/search?dcr=0&source=hp&q=app+inventor&oq=app+inventor&gs_l=psy-ab.3...685.2879.0.2942.16.9.1.0.0.0.349.349.3-1.2.0.dummy_maps_web_fallback...0...1.1.64.psy-ab..13.3.693.6..35i39k1j0i10i67k1j0i67k1.333.9ATQev-Lsb4

Herrera, J., & Sánchez, L. (2013). Obtenido de http://pendientedemigracion.ucm.es/info/aocg/python/puesta_marcha/index.html

Inchausti, D. (27 de Octubre de 2014). *Configuración del GPS.* Recuperado el 30 de Julio de 2017, de

https://boscosteleco.wordpress.com/2014/10/27/dron-bosco-configuracion-del-gps/

Instituto de Automática Industrial. (22 de Marzo de 2016). *Centre for Automation and Robotics*. Recuperado el 9 de Junio de 2017, de Instituto de Automática Industrial.

INVENTOR, M. A. (18 de Septiembre de 2017). Obtenido de http://ai2.appinventor.mit.edu/?locale=en#5063462222561280

Itead. (12 de Febrero de 2017). *SIM900 GSM/GPRS SHIELD ICOMSAT PREVIEW*. Recuperado el 5 de Julio de 2017, de https://www.itead.cc/blog/sim900-gprs-shield-icomsat-preview

Jon Holton, T. F. (2012). *Raspberry Pi Architecture.* N/E: Ars Technica.

Krit Janard, W. M. (2015). *Accelerating Real-time Face Detection on a Raspberry Pi Telepresence Robot .* Pathum Thani: IEEE.

LACERNA, J. R. (2015). *SISTEMA DE OPERACIÓN Y MONITOREO PARA UN VEHÍCULO MEDIANTE RASPBERRY PI.* Pereira: UNIVERSIDAD TECNOLÓGICA DE PEREIRA.

Marin, J. G. (2015). *Prototipo de sistema de seguridad en vehículos de transporte publico que permite la captura de imagen fotográfica, posicionamiento global y almacenamiento en base de datos.* Bogotá-Colombia: Universidad Distrital Francisco José de Caldas.

Marko Viitanen, A. K. (2015). *Kvazaar HEVC Still Image Coding on Raspberry Pi 2 for Low-Cost Remote Surveillance.* Finland: IEEE.

Mata, G., & Javier, F. (2010). *Videovigilancia: CCTV usando vídeos IP.* Málaga-España: Vértice.

Mushoq. (19 de Septiembre de 2017). Obtenido de https://lahora.com.ec/noticia/1101993411/home

NOREN, N. (2017).

Orbegozo, R. (03 de Enero de 2015). *Python 2.x y Python 3.x.* . Recuperado el 15 de Julio de 2017, de https://ricardo705.wordpress.com/2015/01/03/python-2-x-y-python-3-x-diferencias-de-sintaxis-en-solo-4-paginas/

Pablo, M. (Abril de 2012). Obtenido de http://bibdigital.epn.edu.ec/bitstream/15000/4593/1/cd-4215.pdf

Perez, E. (8 de Diciembre de 2011). Obtenido de http://aplicaciones-web-lenguajes-programaci.blogspot.com/

PI-Supply. (2 de Abril de 2017). *Raspberry Pi Camera Board v1.3 (5MP, 1080p).* Recuperado el 25 de Julio de 2017, de https://www.pi-supply.com/product/raspberry-pi-camera-board-v1-3-5mp-1080p/

Premkumar, K. (2015). *Smart Phone Based Robotic Arm Control Using Raspberry Pi, Android and Wi-Fi .* Coimbatore: IEEE.

Princy, S. E. (2015). *Implementation of Cloud Server for Real Time Data Storage using Raspberry Pi.* Coimbatore: IEEE.

Productosled. (15 de Abril de 2015). *Batería 12V 7A*. Recuperado el 30 de Julio de 2017, de http://www.productosled.com.ar/?product=bateria-12v-7a

Proenium. (30 de Agosto de 2016). Obtenido de https://www.proenium.com/carcentinel/

RASPBIAN. (27 de mayo de 2016). *RASPBIAN*. Recuperado el 4 de junio de 2016, de RASPBIAN: https://www.raspberrypi.org/downloads/raspbian/

Rodríguez, C. (2016). *Diseño e implementación de un sistema de control de acceso domiciliaria vía SMS por celular.* Sangolqui - Ecuador.

Sandisk, D. S. (2012). *Micro SD Data Sheet.* N/E: APR.

Sandoval, D. (2015). *Sistemas Operativos Moviles*. Obtenido de http://introtecnologia-pc.wixsite.com: http://introtecnologia-pc.wixsite.com/tecnologia-pc/sistemas-operativos-moviles

Sevilla, G. (Mayo de 2016). *SISTEMA OPERATIVO PARA RASPBERRY PI.* Recuperado el 8 de Julio de 2016, de http://raspberryhack.com/feed/xinu-otro-sistema-operativo-para-raspberry-pi

Siemon. (5 de Marzo de 2016). *CCTV y Vigilancia por Video sobre 10G ip.* Recuperado el 7 de Septiempbre de 2016, de NETWORK CABLING SOLUTIONS: https://www.siemon.com/la/white_papers/SD-03-08-CCTV.asp

Simbaña, W. (2015). *Diseño e implementacion de un prototipo inalembrico de monitoreo a traves de secuencias de imagenes utilizando hadware y software libre.* Quito: EPN.

Sites. (20 de Diciembre de 2016). *App Inventor en Español*. Recuperado el 15 de Julio de 2017, de https://sites.google.com/site/appinventormegusta/instalacion

Store. (30 de Abril de 2017). *ARDUINO MEGA 2560 REV3*. Recuperado el 30 de Julio de 2017, de https://store.arduino.cc/usa/arduino-mega-2560-rev3

Tecnologia-informatica. (07 de Julio de 2014). *Control remoto con Arduiono.* Recuperado el 30 de Julio de 2017, de http://www.tecnologia-informatica.es/control-remoto-con-arduiono/

Telegrafo, E. (12 de Diciembre de 2016). Obtenido de http://www.eltelegrafo.com.ec/noticias/ecuador/3/el-parque-automotor-sube-el-57-en-ecuador

TELÉGRAFO, E. (30 de JULIO de 2017). Obtenido de http://www.eltelegrafo.com.ec/noticias/judicial/13/en-2-anos-y-medio-la-policia-registro-el-robo-de-13-271-carros

Teslabem. (3 de Marzo de 2017). *Módulo sensor de sonido con micrófono electret 10mm.* Recuperado el 20 de Julio de 2017, de http://teslabem.com/modulo-sensor-de-sonido-con-microfono-electret.html

Tinajero, J. L. (2014). *108T0115 Evaluación de la eficiencia de los controladores Arduino Mega y Seimens Logo 23ORC en procesos industiales.* Obtenido de Repositorio Institucional de la Escuela Superior Politécnica de Chimborazo: dspace.espoch.edu.ec/bitstream/123456789/3805/1/108T0115.pdf

Torrelaguna. (18 de Octubre de 2009). Obtenido de http://recursostic.educacion.es/observatorio/web/es/software/programacion/745-introduccion-a-la-programacion-con-el-lenguaje-c

Velasco, N. (2005). *Sistema embebido para la conexión de un PLC Siemens S7-200 a la red GSM.* Sevilla, España.

Vele, O. (2016). *Guia de estudio programacion C.* Cuenca-Ecuador.

Webantics. (4 de Febrero de 2017). *Huawei E3533 21.6MBPS HSPA+ 3G USB Modem.* Recuperado el 26 de Julio de 2017, de https://www.webantics.com/huawei-e3533-21mbps-hspa-3g-usb-modem

ANEXO 1
MANUAL DE USUARIO

1. PRECAUCIONES DE SEGURIDAD

- **Encendido del dispositivo**

Para el encendido del prototipo se debe colocar el switch en la posición ON que se encuentra a un costado del mismo y pulsar el botón de encendido del SIM908.

Adicional se debe encender el bluetooth del teléfono inteligente, para permitir el sincronismo y seguido la comunicación.

- **Interferencias**

El módulo Rf y el dispositivo móvil Smartphone puede recibir interferencias externas.

- **No mojes el dispositivo**

El prototipo no se lo debe mojar, debido a que contiene dispositivos electrónicos que se pueden dañar y dejar de funcionar.

- **Úselo correctamente**

El prototipo se lo debe usar con el fin con él cual fue construido, así no hacer un mal uso del mismo.

- **Acerca del dispositivo**

El prototipo es un sistema de seguridad antirrobo vehicular, mediante el cual se puede tener el control de la seguridad del vehículo. Se lo puede manejar mediante teléfono inteligente o módulo de radio frecuencias. Además, tiene la opción de enviar la posición por medio de mensaje de texto y al correo electrónico, así también envía una imagen con los eventos que se presentan dentro del vehículo. Se activa mediante un sensor de ruino o al momento de forzar el seguro de las puertas.

- **Conexión/desconexión del dispositivo**

La conexión del prototipo se lo realiza mediante el módulo bluetooth que permite la comunicación bidireccional.

El módulo RF del prototipo controla en todo momento el estado de las puertas mediante comunicación unidireccional.

2. INSTALACION DEL SOFTWARE

- **Requerimientos básicos**

Para el aplicativo que controla el prototipo

- Smartphone o Tablet con SO Android 2.3.4 o mayores.
- RAM 512 Mb.
- Procesador de 1 Ghz.
- Pantalla táctil.
- Memoria interna de 1Gb.

- **Instalación de software**

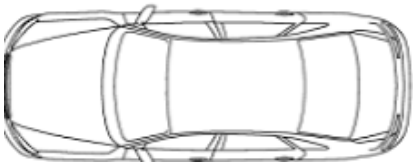
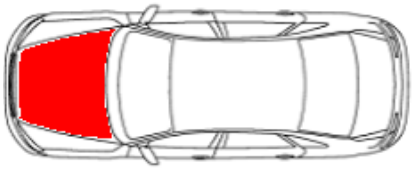
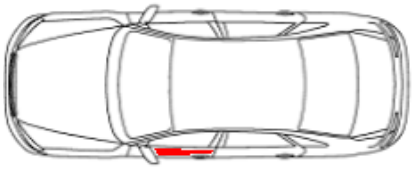
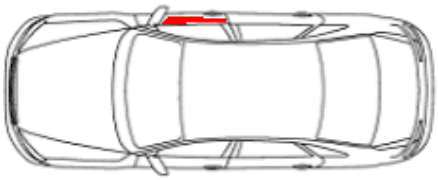
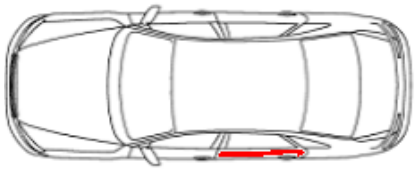
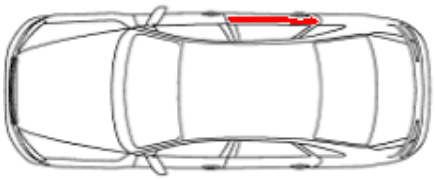
Para la instalación se debe copiar el Apk de la aplicación que viene incluido en el DC de instalación. El Apk se lo debe ejecutar en un smartphone o en una Tablet. El mismo que tiene el nombre de SISTEMA DE SEGURIDAD ANTIRROBO VEHICULAR.

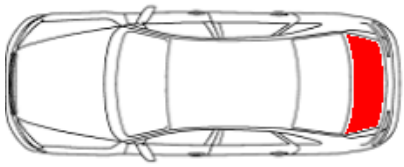
3. DESCRIPCION DE PARTES

- **Modulo Radio Frecuencia**

Señal Módulo RF YK04	Función
B	Apagar
D	Emergencia Apagar.
A	Encender
C	Encender modo silencio

- **Interfaz Android**

ELEMENTO	FUNCIÓN
Bluetooth	Conexión a bluetooth
Desconectar	Desconexión a bluetooth
	Activación
	Desactivación
	Indicador de las puertas
	Indicador puerta capo abierta
	Indicador puerta derecha delantera abierta
	Indicador puerta izquierda delantera abierta
	Indicador puerta trasera derecha abierta
	Indicador puerta trasera izquierda abierta

	Indicador puerta cajuela abierta
Desconectado	Baja todos los servidores

4. GUIA DE USO

Se debe seguir los siguientes pasos que se detalla a continuación:

- **Teléfono Inteligente**

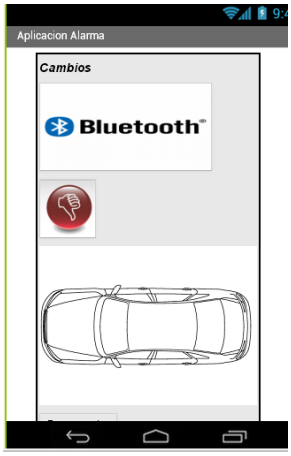
  1- El Smartphone se lo debe encender.
  2- Se debe encender el bluetooth y buscar al prototipo.
  3- Se debe sincronizar con el prototipo.
  4- Importante se tendrá control total de la seguridad del prototipo.

- **Interface Android**

  1. Se debe descargar Se debe abrir el aplicativo en el teléfono
  2. Se debe activar el Bluetooth.
  3. Abrir la aplicación que previamente se instalo
  4. Escribir USUARIO (ALARMA)
  5. Escribir PASSWORD (123456)

6. Si se ingresa mal los datos se despliega un mesaje de Usuario/Contraseña incorrecta

7. Una ves ingresa los datos, se visualiza la interfaz

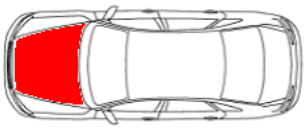


8. Buscamos el prototipo a seleecionar y se lo debe sincronizar.

9. En la imagen de color rojo, se debe pulsar cambia a verde y se activa la alarma, es decir los seguros de las puertas.

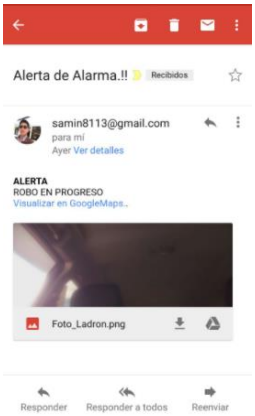
10. En la imagen de color verde, se debe pulsar y cambia a rojo y se activa la alarma, es decir los seguros de las puertas.

11. En caso de que una puerta de encuentre abierta nos aparece la siguiente imagen.

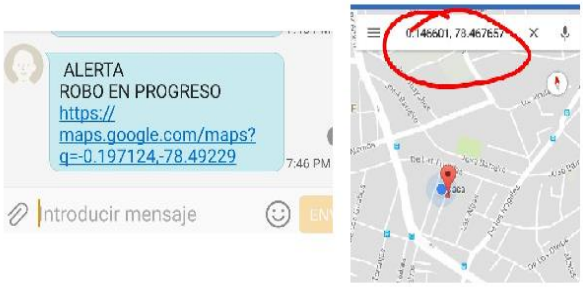


12. Como identifica que se abrió una puerta realiza el siguiente proceso:

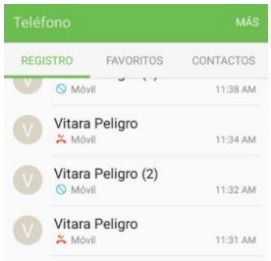
    a. Toma foto y envía al correo electrónico.



    b. Envía mensaje de texto con la posición.



c. Hace una llamada al celular (timbra).



❖ **Prototipo Físico**

Se lo debe encender con su botón on y el pulsador, los mismos que están en la caja.

5. Transmisor Radio Frecuencia

✓ Se toman el control y se extender su antena.

✓ Cada boton tiene una accion, la misma que se ejecuta al pulsar uno de ellos de manera individual.

6. Solución de problemas más comunes.

✓ Apagar el bluetooth y nuevamente encenderla.

✓ Cierre la aplicación y nuevamente ábrala.

✓ Ingrese bien la contraseña y el usuario.

✓ Si tiene más problemas consulte los números de soporte.

7. E-mail y teléfonos de la organización

Santiago Quilachamín	samin8113@gmail.com	0958742945	Área de Soporte 7x24

ANEXO 2
MANUAL TÉCNICO

CONTENIDO

1. OBJETIVOS

Detallar la información necesaria para el manejo del funcionamiento del sistema prototipo.

Específicos

- Representar la funcionalidad técnica de la estructura, diseño y definición del aplicativo.
- Describir las herramientas utilizadas para el diseño y desarrollo de conexiones del prototipo.

2. ALCANCE

Este manual describe los pasos necesarios para cualquier persona que tenga ciertas bases de conocimientos de telecomunicaciones pueda realizar la instalación y configuración del prototipo.

3. RESUMEN

La seguridad en el Ecuador es muy crítica, por consecuente el alto índice de robos a vehículos de gama baja, media y alta, hace que las persona busque sistemas que les permita proteger de una manera u otra sus vehículos.

4. REQUISITOS DEL SISTEMA

- Batería principal del vehículo de 12V a 73 Ah
- Batería de respaldo 12V a 7Ah
- Espacio físico de 20x20X5 cm aproximadamente
- Módulo de bloqueo central de puertas preinstalado
- Modem 3G
- Raspberry Pi 3B

- GPRS SIM 908
- Arduino Mega 2560
- Cámara de video Raspberry Pi 100003
- Sensor de ruido de impacto
- Control remoto módulo YK04
- Regulador de voltaje
- Smartphone con un chip instalado de cualquier operadora
- Caja de plástico 20x20X5 cm aproximadamente

5. DIAGRAMA DEL CABLEADO

La mala operación de instalación como conexiones incorrectas puede llegar a ocasionar daños irreparables, así como también una incorrecta operación del prototipo.

Antes de aplicar la energía al sistema, primero inspeccionar el cableado y asegurar que todas las conexiones se encuentren realizados de manera correcta.

La figura A permite visualizar el diagrama del cableado de los pines GPIO de la tarjeta Raspberry Pi 3B, donde se detalla de manera gráfica la instalación de cada uno de los componentes que conforman el sistema de manera simultánea se detalla cada uno de los pines con los componentes que lo conforman.

La alimentación de la tarjeta Raspberry Pi 3B está conformada por el ping de carga mini USB principal de la tarjeta está a su vez está regulada por el variador de voltaje.

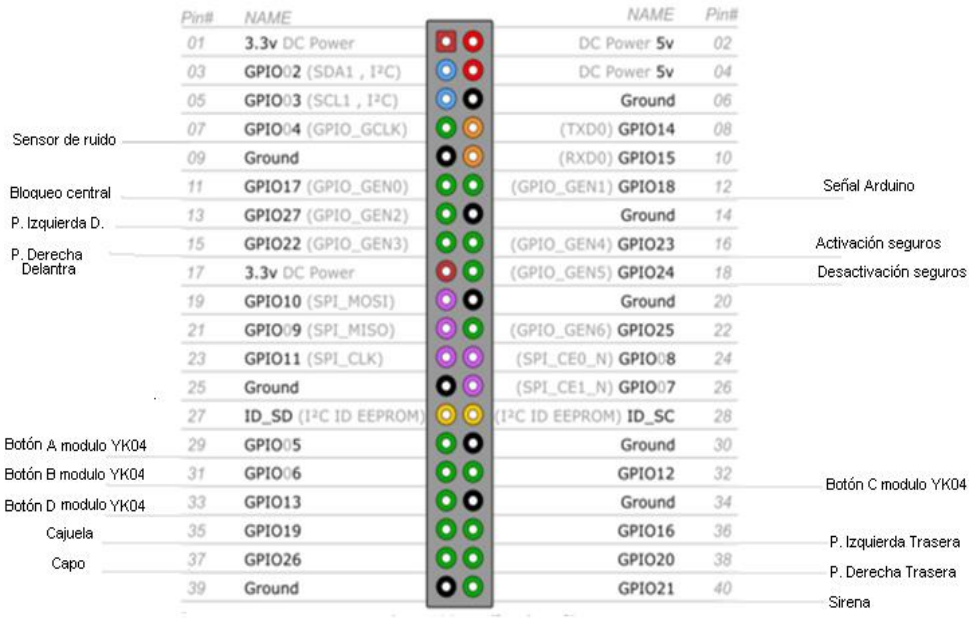


Figura A. Diagrama cableado GPIO

- ✓ **Pin 2 y 4.** Salida de alimentación de voltaje 5V los cuales van conectados la alimentación del sensor de ruido y la alimentación de integrado 7404.
- ✓ **Pin 6**. Salida tierra común GND, este pin está conectado el sensor de ruido al pin de tierra y al 7404 adicionalmente es la señal de activación de las puertas del vehículo.
- ✓ **Pin 4.** Entrada de la señal del sensor de ruido de impacto
- ✓ **Pin 17.** Salida de la señal del bloqueo central este pin se activa cuando la alarma se enciende.
- ✓ **Pin 27.** Señal de entrada de la puerta izquierda delantera esta señal se activa en bajo
- ✓ **Pin 22.** Señal de entrada de la puerta derecha delantera
- ✓ **Pin 5.** Señal de entrada del botón A del control remoto del módulo YK04.
- ✓ **Pin 6.** Señal de entrada del botón B del control remoto del módulo YK04.
- ✓ **Pin 13.** Señal de entrada del botón D del control remoto del módulo YK04.
- ✓ **Pin 19.** Señal de entrada de la cajuela

- ✓ **Pin 26.** Señal de entrada del capo
- ✓ **Pin 18.** Señal de salida para la activación del Arduino mega 2560 el cual esta envía la ubicación geográfica y un mensaje SMS.
- ✓ **Pin 23.** Señal de salida para la activación de los seguros de las puertas del vehículo.
- ✓ **Pin 24.** Señal de salida para la desactivación de los seguros de las puertas del vehículo.
- ✓ **Pin 12.** Señal de entrada del botón C del control remoto del módulo YK04.
- ✓ **Pin 16.** Señal de entrada de la puerta izquierda trasera
- ✓ **Pin 20.** Señal de entrada de la puerta derecha trasera
- ✓ **Pin 21.** Señal de salida para la activación de la sirena.

A continuación, en la figura B se ilustra el diagrama del cableado de la tarjeta Arduino mega 2560.

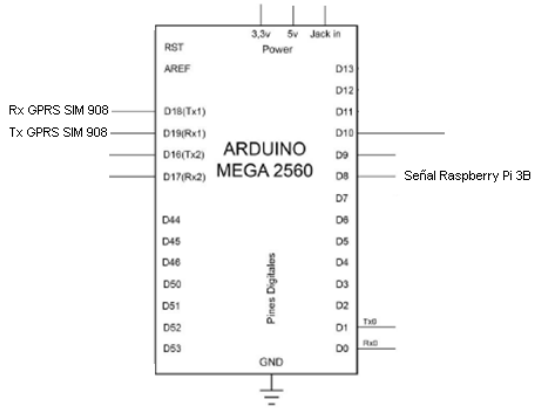


Figura B. Diagrama cableado Arduino mega 2560

- ✓ **Pin 8.** Señal de entrada de la señal de activación por parte de la Raspberry Pi 3B.
- ✓ **Pin 18(Tx_1).** Señal de recepción (Rx), de la tarjeta GPRS SIM 908.
- ✓ **Pin 19(Rx_1).** Señal de trasmisión (Tx), de la tarjeta GPRS SIM 908.

En la figura C se ilustra el diagrama de los pines del control remoto módulo YK04.

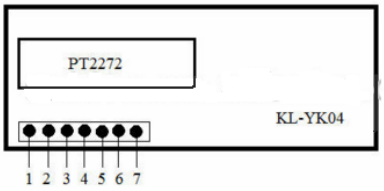


Figura C. Diagrama pines módulo YK04

- ✓ **Pin 7.** Entrada de la tierra común GND.
- ✓ **Pin 6.** Entrada de alimentación de voltaje 5V, proveniente de la batería de respaldo de 12V a 7Ah, el cual está regulado por el variador de voltaje.
- ✓ **Pin 5 (Do).** Señal se salida del botón B, este pin a su vez está conectada al GPIO pin 6.
- ✓ **Pin 4 (D1).** Señal se salida del botón D, este pin a su vez está conectada al GPIO pin 13.
- ✓ **Pin 3 (D2).** Señal se salida del botón A, este pin a su vez está conectada al GPIO pin 5.
- ✓ **Pin 2 (D3).** Señal se salida del botón C, este pin a su vez está conectada al GPIO pin 12.
- ✓ **Pin 1 (D4).** Señal de activación del control remoto RF.

6. ETAPA DE POTENCIA

Para la etapa de potencia se utiliza un módulo de relés de 4 canales, en la figura D se ilustra el diagrama de cableado de la etapa de potencia.

7. SEÑAL DE ENTRADA

- ➢ **Jumper J5.**

- **Pin 1.** Entrada de alimentación principal de voltaje proveniente del GPIO pin 2, 5V.
- **Pin 2.** Entrada de la señal del bloqueo central proveniente de la GPIO pin 17.
- **Pin 3.** Entrada de la señal de la sirena proveniente de la GPIO pin 21.
- **Pin 4.** Entrada de la señal para la desactivación de los seguros de las puertas del vehículo proveniente de la GPIO pin 24.

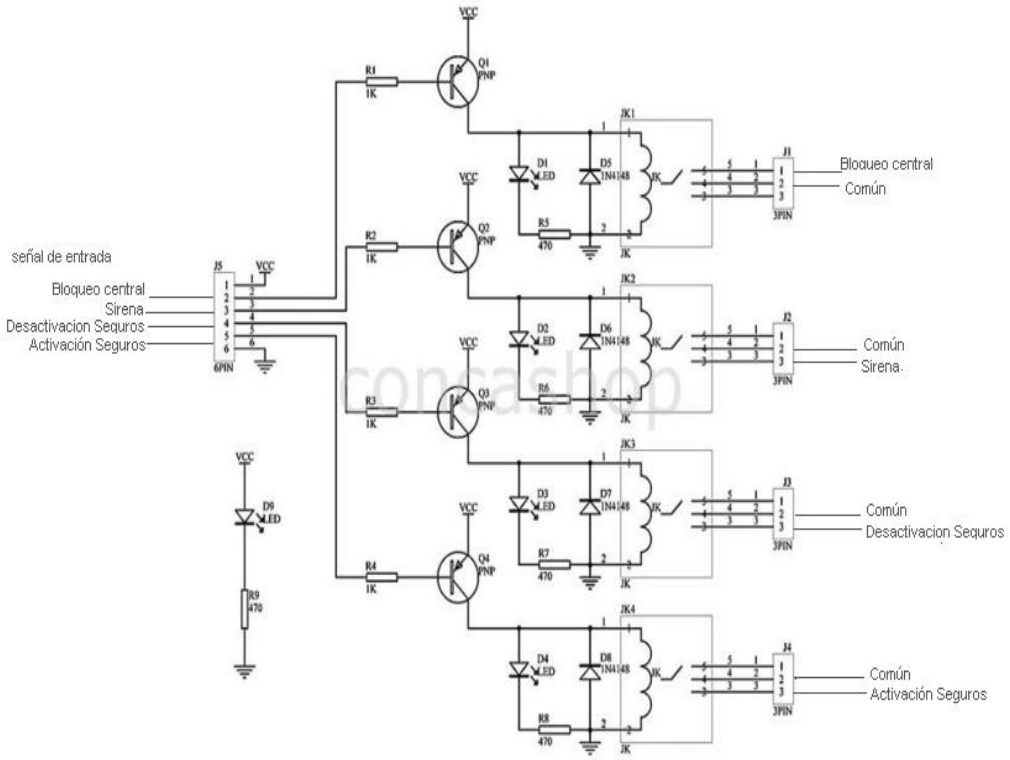


Figura D. Diagrama etapa de potencia

- **Pin 5.** Entrada de la señal para la activación de los seguros de las puertas del vehículo proveniente de la GPIO pin 23.
- **Pin 6.** Entrada de alimentación tierra común proveniente del GPIO pin 6.

8. SEÑALES DE SALIDA

➢ Jumper J1.

- **Pin 1 (NC).** Señal de salida normalmente cerrada para el bloqueo central del vehículo, este pin se conecta de manera paralela al sistema del bloqueo central preinstalado en el vehículo.
- **Pin 2 (C).** Señal común del sistema del bloqueo centra pre-instalado en el vehículo.

➢ Jumper J2.

- **Pin 2 (C).** Señal común de la sirena, proveniente de la batería principal del vehículo 12V.

- **Pin 3 (NA).** Señal de salida normalmente abierta para la activación de la sirena del vehículo.

  - **Jumper J3.**

- **Pin 2 (C).** Señal común para la desactivación de los seguros de las puertas del vehículo, este pin se conecta de manera paralela al sistema de los seguros de la desactivación de las puertas preinstalado en el vehículo
- **Pin 3 (NA).** Señal de salida normalmente abierta para la desactivación de las puertas del vehículo.

  - **Jumper J4.**

- **Pin 2 (C).** Señal común para la activación de los seguros de las puertas del vehículo, este pin se conecta de manera paralela al sistema de los seguros de activación de las puertas preinstalado en el vehículo
- **Pin 3 (NA).** Señal de salida normalmente abierta para la activación de las puertas del vehículo.

9. CONEXIONES

Finalmente, para la conexión de los dispositivos de entrada y salida del sistema prototipo se usó 4 plug de 5 pines macho conector hembra como se observa en la figura E.

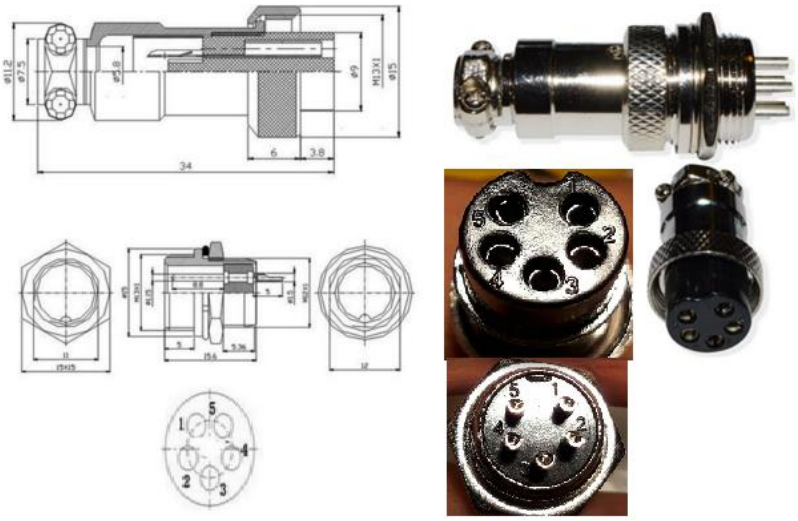


Figura D. Plug 5 Pines Macho Conector macho hembra

- **Plug 1.**

- **Pin 1.** Conexión de relé canal J4 proveniente del pin 2.
- **Pin 2.** Conexión de relé canal J4 proveniente del pin 3.
- **Pin 3.** Conexión de relé canal J3 proveniente del pin 2.
- **Pin 4.** Conexión de relé canal J3 proveniente del pin 3.
- **Pin 5.** Conexión de relé canal J1 proveniente del pin 2.

- **Plug 2.**

- **Pin 1.** Conexión del sensor de ruido de impacto para la alimentación 5V proveniente de la GPIO pin 4.
- **Pin 2.** Conexión para el sensor de ruido de impacto alimentación tierra común GND proveniente de la GPIO pin 6.
- **Pin 4.** Entrada de alimentación para el sistema está conectada al regulador de voltaje +12V, proveniente de la batería de respaldo 12V
- **Pin 5.** Entrada de alimentación tierra común GND para el sistema está conectada al regulador de voltaje -12V, proveniente de la batería de respaldo 12V a 7Ah.

- **Plug 3.**

- **Pin 1.** Conexión del capo del vehículo proveniente del pulsador normalmente cerrado ubicado en el capo del vehículo.
- **Pin 2.** Conexión del bloqueo central del vehículo proveniente del sistema del bloqueo central preinstalado en el vehículo.
- **Pin 3.** Conexión para la sirena del vehículo proveniente de la batería principal del vehículo +12V.
- **Pin 4.** Conexión de la sirena proveniente del canal J2 del relé pin 3.
- **Pin 5.** Conexión para el sensor de ruido proveniente de la GPIO pin 6.

- **Plug 4.**

- **Pin 1.** Conexión de la cajuela del vehículo proveniente del pulsador normalmente cerrado ubicado en la cajuela del vehículo.

- **Pin 2.** Conexión de la puerta delantera izquierda del vehículo proveniente del pulsador normalmente cerrado ubicado en la puerta izquierda delantera del vehículo.
- **Pin 3.** Conexión de la puerta derecha delantera del vehículo proveniente del pulsador normalmente cerrado ubicado en la puerta derecha delantera del vehículo.
- **Pin 4.** Conexión de la puerta izquierda trasera del vehículo proveniente del pulsador normalmente cerrado ubicado en la puerta izquierda trasera del vehículo.
- **Pin 5.** Conexión de la puerta derecha trasera del vehículo proveniente del pulsador normalmente cerrado ubicado en la puerta derecha trasera del vehículo.

Se utiliza este tipo de plug mostrado en la figura E para facilitar la conexión y desconexión del sistema prototipo de esta manera se puede realizar cualquier tipo de mantenimiento al desconectar todo el cableado del sistema.

Figura E. Plug de 5 pines para la conexión y desconexión del sistema

A continuación, se muestra en la figura F el diagrama de conexión de cada una de las etapas que conforman el prototipo del sistema de seguridad vehicular, se procede a presentar el diagrama eléctrico de conexión total, en donde se encuentra detallada cada uno de los dispositivos usados y la interconexión de los mismos, los cuáles conforman la tarjeta Raspberry Pi 3B, Arduino mego 2560, GPRS SIM 908, módulo de relés, regulador de volteja y la batería de respaldo.

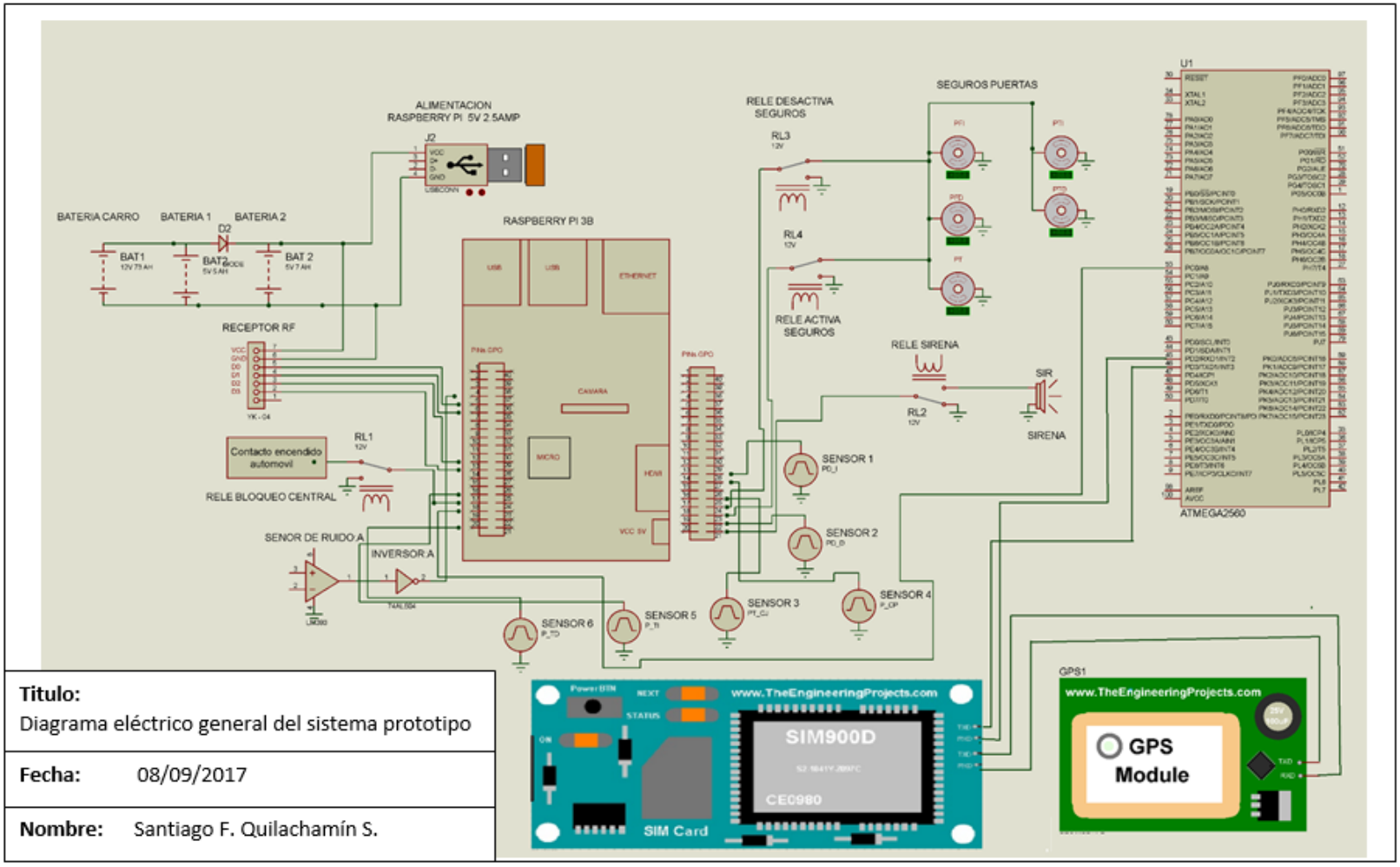


Figura F. Diagrama de conexión general del sistema prototipo

10. Detalles del Correcto Arranque del Sistema

A continuación, se indicará el proceso de inicio del sistema, estos pasos servirán para verificar que el sistema se encuentre en correcto funcionamiento, si algunos de estos pasos fallaren se debe tomar otras medidas.

Los pasos de arranque del sistema se describen a continuación:

a) Como primer paso después de haber realizado las todas las conexiones de cableado de manera correcta, es suministrar con energía al sistema con la alimentación principal proveniente de la batería del vehículo conectado en paralelo la batería de respaldo de 12V a 7Ah.

b) Al conectar la batería al sistema, verificar que de manera instantánea se enciende la sirena con un doble pitido después de 10 segundos, esto indica que el sistema arrancó de manera correcta.

c) Esperar aproximadamente 5 segundos y verificar que se haya cerrado las puertas, el cual es el tiempo que el sistema demora en arrancar.

d) Después de que el circuito ha arrancado de manera correcta, todas las funciones del sistema pueden ser ejecutadas, la cual se deberá comprobar su correcto arranque, esto se lo hace luego de que el sistema haya dado un doble pitido breve y continuo.

e) Finalmente, después de haber comprobado todos los pasos, el sistema se encuentra instalado correctamente y operable de completamente.

11. SOLUCION DE PROBLEMAS

- ✓ El prototipo no enciende, revisar las conexiones de las 4 borneras que no estén sueltas o desconectadas.
- ✓ El prototipo no enciende, apagar el prototipo de la fuente de energía y encenderla.

- ✓ En caso de que, de no encender el prototipo, cambiar de batería por una nueva.
- ✓ Si no recibe llamadas y mensajes de texto SMS, validar que exista saldo en el chip que se utiliza.
- ✓ En caso de que los mensajes de texto SMS no contenga la posición del vehículo, puede ser que el vehículo se encuentre en un lugar cerrado como un sótano, cochera o parqueadero. Tomar en cuenta que el GPS recibe la señal del satélite en exteriores.
- ✓ No se recibe email con la foto ni la ubicación, validar que el chip tenga paquete de datos.
- ✓ Si existe un corto circuito en el vehículo, apague el vehículo por unos segundos y vuelva a encenderlo.
- ✓ El prototipo se moja, apagarlo y secarlo hasta que todos sus componentes no contengan agua o la sustancia con la cual se lo roció, y contactar a los números de soporte.
- ✓ El módulo GPS no recibe señal, y no se envía en el correo la posición, saque el chip y vuélvalo a colocar.
- ✓ Si tiene más problemas consulte los números de soporte.

12. E-mail y teléfonos de la organización

Santiago Quilachamín	samin8113@gmail.com	0958742945	Área de Soporte 7x24

ANEXO 3

DATASHEET DEL CIRCUITO INTEGRADO LM2596

LM2596 SIMPLE SWITCHER® Power Converter 150 kHz 3A Step-Down Voltage Regulator

Check for Samples: LM2596

FEATURES

- 3.3V, 5V, 12V, and Adjustable Output Versions
- Adjustable Version Output Voltage Range, 1.2V to 37V ±4% Max Over Line and Load Conditions
- Available in TO-220 and TO-263 Packages
- Ensured 3A Output Load Current
- Input Voltage Range Up to 40V
- Requires Only 4 External Components
- Excellent Line and Load Regulation Specifications
- 150 kHz Fixed Frequency Internal Oscillator
- TTL Shutdown Capability
- Low Power Standby Mode, I_Q Typically 80 µA
- High Efficiency
- Uses Readily Available Standard Inductors
- Thermal Shutdown and Current Limit Protection

APPLICATIONS

- Simple High-Efficiency Step-Down (Buck) Regulator
- On-Card Switching Regulators
- Positive to Negative Converter

DESCRIPTION

The LM2596 series of regulators are monolithic integrated circuits that provide all the active functions for a step-down (buck) switching regulator, capable of driving a 3A load with excellent line and load regulation. These devices are available in fixed output voltages of 3.3V, 5V, 12V, and an adjustable output version.

Requiring a minimum number of external components, these regulators are simple to use and include internal frequency compensation , and a fixed-frequency oscillator.

The LM2596 series operates at a switching frequency of 150 kHz thus allowing smaller sized filter components than what would be needed with lower frequency switching regulators. Available in a standard 5-lead TO-220 package with several different lead bend options, and a 5-lead TO-263 surface mount package.

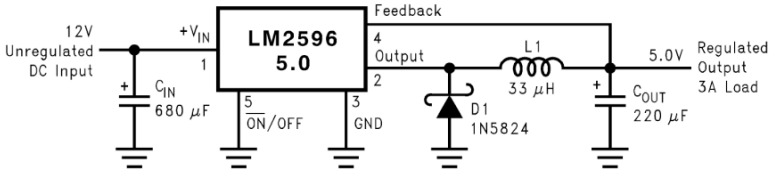
A standard series of inductors are available from several different manufacturers optimized for use with the LM2596 series. This feature greatly simplifies the design of switch-mode power supplies.

Other features include a ensured ±4% tolerance on output voltage under specified input voltage and output load conditions, and ±15% on the oscillator frequency. External shutdown is included, featuring typically 80 µA standby current. Self protection features include a two stage frequency reducing current limit for the output switch and an over temperature shutdown for complete protection under fault conditions. (1)

(1) † Patent Number 5,382,918.

Typical Application

(Fixed Output Voltage Versions)



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

SIMPLE SWITCHER is a registered trademark of Texas Instruments.
All other trademarks are the property of their respective owners.

Connection Diagrams

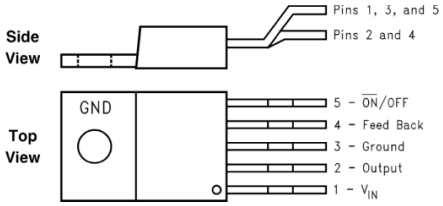


Figure 1. 5-Lead Bent and Staggered Leads, Through Hole TO-220 (T) Package See Package Number NDH0005D

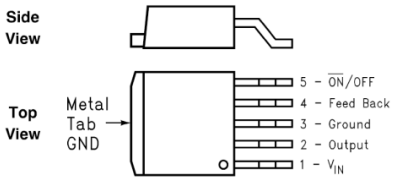


Figure 2. 5-Lead DDPAK/TO-263 (S) Package See Package Number KTT0005B

These devices have limited built-in ESD protection. The leads should be shorted together or the device placed in conductive foam during storage or handling to prevent electrostatic damage to the MOS gates.

Absolute Maximum Ratings (1)(2)

Maximum Supply Voltage	45V
$\overline{ON}$ /OFF Pin Input Voltage	−0.3 ≤ V ≤ +25V
Feedback Pin Voltage	−0.3 ≤ V ≤+25V
Output Voltage to Ground (Steady State)	−1V
Power Dissipation	Internally limited
Storage Temperature Range	−65°C to +150°C
ESD Susceptibility	
Human Body Model (3)	2 kV
Lead Temperature	
DDPAK/TO-263 Package	
Vapor Phase (60 sec.)	+215°C
Infrared (10 sec.)	+245°C
TO-220 Package (Soldering, 10 sec.)	+260°C
Maximum Junction Temperature	+150°C

(1) Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. Operating Ratings indicate conditions for which the device is intended to be functional, but do not ensure specific performance limits. For ensured specifications and test conditions, see the Electrical Characteristics.
(2) If Military/Aerospace specified devices are required, please contact the Texas Instruments Sales Office/ Distributors for availability and specifications.
(3) The human body model is a 100 pF capacitor discharged through a 1.5k resistor into each pin.

Operating Conditions

Temperature Range	−40°C ≤ T_J ≤ +125°C
Supply Voltage	4.5V to 40V

http://www.ti.com/lit/ds/symlink/lm2596.pdf

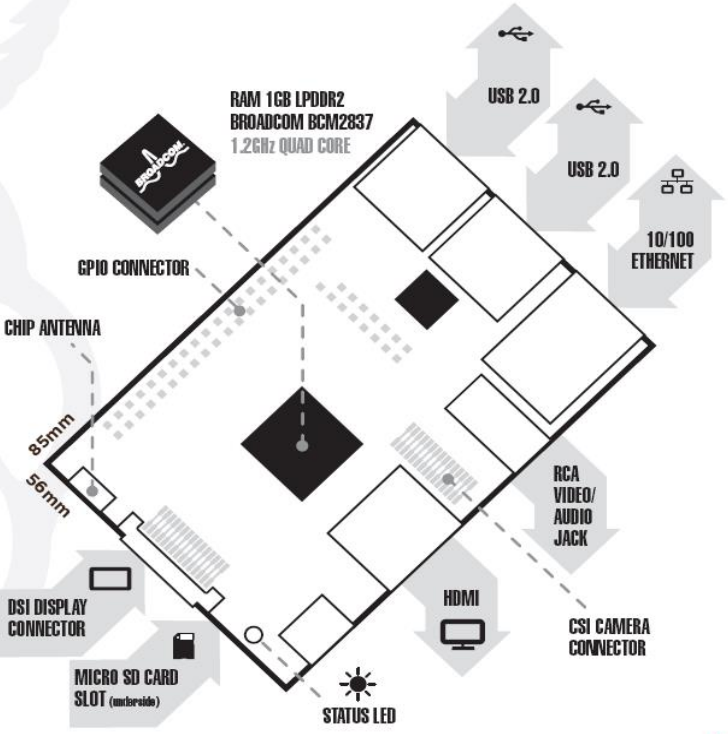
ANEXO 4

DATASHEET RASPBERRY PI

Raspberry Pi

Raspberry Pi 3 Model B

Product Name	**Raspberry Pi 3**
Product Description	The Raspberry Pi 3 Model B is the third generation Raspberry Pi. This powerful credit-card sized single board computer can be used for many applications and supersedes the original Raspberry Pi Model B+ and Raspberry Pi 2 Model B. Whilst maintaining the popular board format the Raspberry Pi 3 Model B brings you a more powerful processer, 10x faster than the first generation Raspberry Pi. Additionally it adds wireless LAN & Bluetooth connectivity making it the ideal solution for powerful connected designs.
RS Part Number	**896-8660**



RS

Raspberry Pi

Raspberry Pi 3 Model B

Specifications

Processor	Broadcom BCM2387 chipset. 1.2GHz Quad-Core ARM Cortex-A53 802.11 b/g/n Wireless LAN and Bluetooth 4.1 (Bluetooth Classic and LE)
GPU	Dual Core VideoCore IV® Multimedia Co-Processor. Provides Open GL ES 2.0, hardware-accelerated OpenVG, and 1080p30 H.264 high-profile decode. Capable of 1Gpixel/s, 1.5Gtexel/s or 24GFLOPs with texture filtering and DMA infrastructure
Memory	1GB LPDDR2
Operating System	Boots from Micro SD card, running a version of the Linux operating system or Windows 10 IoT
Dimensions	85 x 56 x 17mm
Power	Micro USB socket 5V1, 2.5A

Connectors:

Ethernet	10/100 BaseT Ethernet socket
Video Output	HDMI (rev 1.3 & 1.4 Composite RCA (PAL and NTSC)
Audio Output	Audio Output 3.5mm jack, HDMI USB 4 x USB 2.0 Connector
GPIO Connector	40-pin 2.54 mm (100 mil) expansion header: 2x20 strip Providing 27 GPIO pins as well as +3.3 V, +5 V and GND supply lines
Camera Connector	15-pin MIPI Camera Serial Interface (CSI-2)
Display Connector	Display Serial Interface (DSI) 15 way flat flex cable connector with two data lanes and a clock lane
Memory Card Slot	Push/pull Micro SDIO

Key Benefits

- Low cost
- 10x faster processing
- Consistent board format
- Added connectivity

Key Applications

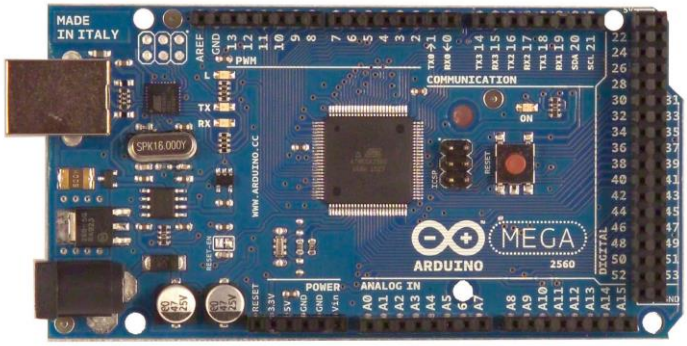
- Low cost PC/tablet/laptop
- Media centre
- Industrial/Home automation
- Print server
- Web camera
- Wireless access point
- IoT applications
- Robotics
- Server/cloud server
- Security monitoring
- Gaming
- Environmental sensing/monitoring (e.g. weather station)

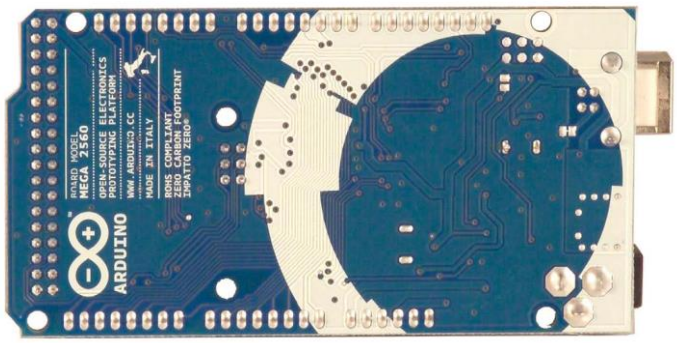
RS

http://docs-europe.electrocomponents.com/webdocs/14ba/0900766b814ba5fd.pdf

ANEXO 5

DATASHEET ARDUINO MEGA 2560





The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 (datasheet). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

Power

The Arduino Mega2560 can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. T can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip .
- **External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2).** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- **PWM: 0 to 13.** Provide 8-bit PWM output with the analogWrite() function.
- **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Duemilanove and Diecimila.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **I^2C: 20 (SDA) and 21 (SCL).** Support I^2C (TWI) communication using the Wire library (documentation on the Wiring website). Note that these pins are not in the same location as the I^2C pins on the Duemilanove.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and analogReference() function.

There are a couple of other pins on the board:

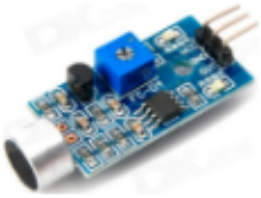
- **AREF.** Reference voltage for the analog inputs. Used with analogReference().
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

http://www.mantech.co.za/datasheets/products/A000047.pdf

http://www.atmel.com/Images/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.pdf

ANEXO 6

SENSOR DE SONIDO LM 393



Vcc, GND y OUT. Vcc es el pin que se debe de conectar a la fuente de tensión (5V), GND es la toma a tierra y por último OUT es el pin de salida al que va a conectar el sensor (en este caso se ha conectado a **A0**). Cabe destacar que este sensor recoge datos analógicos, por lo que debe de conectarse a los pines hembra de la parte analógica, no en la digital (cosa fácilmente confundible, pues se puede pensar que simplemente recoge sonido (1) o no recoge sonido (0), pero no es así).

http://www.iescamp.es/miarduino/2016/02/11/sensor-de-sonido-con-arduino/

ANEXO 7

MÓDULO RELE DE 4 CANALES

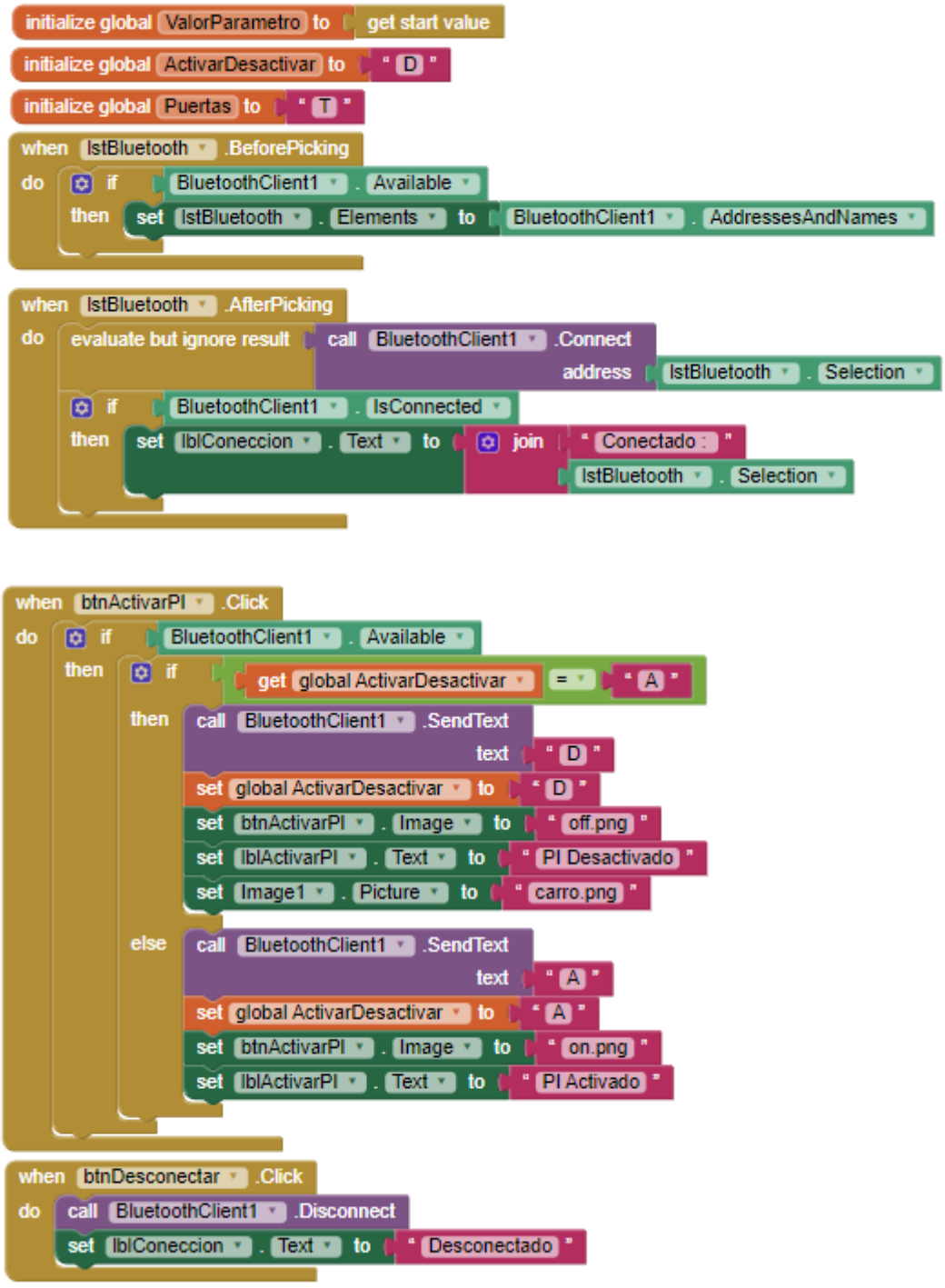
Los relés están montados en una placa con 6 pines que pueden conectarse se utiliza cables dupont para conectarlo directamente a tu placa Arduino. Basta con alimentar el módulo en los pines VCC y GND con 5VDC directamente desde tu placa Arduino y tienes disponibles los pines IN1 a IN4, para controlar los cuatro relés. Un nivel alto conmutará el relé correspondiente. Se dispone de las salidas COM, NC y NO al utilizar un bloque terminal con tornillos. La salida NC junto con la COM, está siempre cerrada en el momento que el pin IN está a nivel bajo, y conmuta a NO y COM cuando el pin IN pasa a nivel alto. Incluye un LED por cada relé que indica el instante que el relé está cerrado (pin IN a nivel alto). Los pines IN1 a IN4 están protegidos con optoacopladores para evitar que cualquier fallo en el relé pueda quemar tu placa Arduino. Además, la placa dispone de una entrada de alimentación adicional marcada como JD-VCC que permite alimentar el módulo con una fuente externa que no sea la misma que la de la placa Arduino, hacer un aislamiento máximo. Simplemente quita el puente que une los pines JD-VCC y VCC, conecta JD-VCC y GND a una fuente externa de 5VDC y el pin VCC a los 5VDC de la placa Arduino.

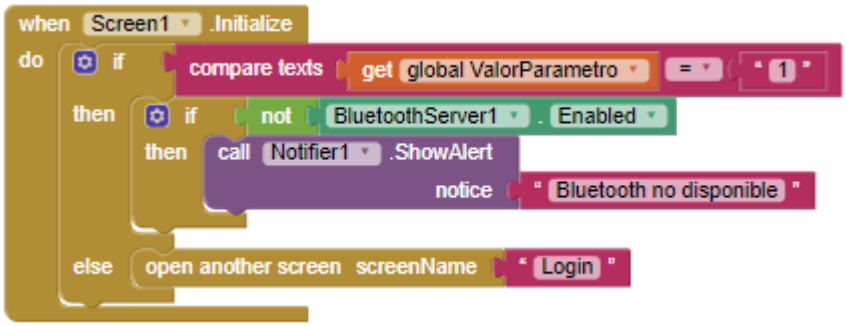
Al usar la fuente externa, los pines JD-VCC y GND permite además usar señales de entrada de solo 3.3VDC para activar los relés.

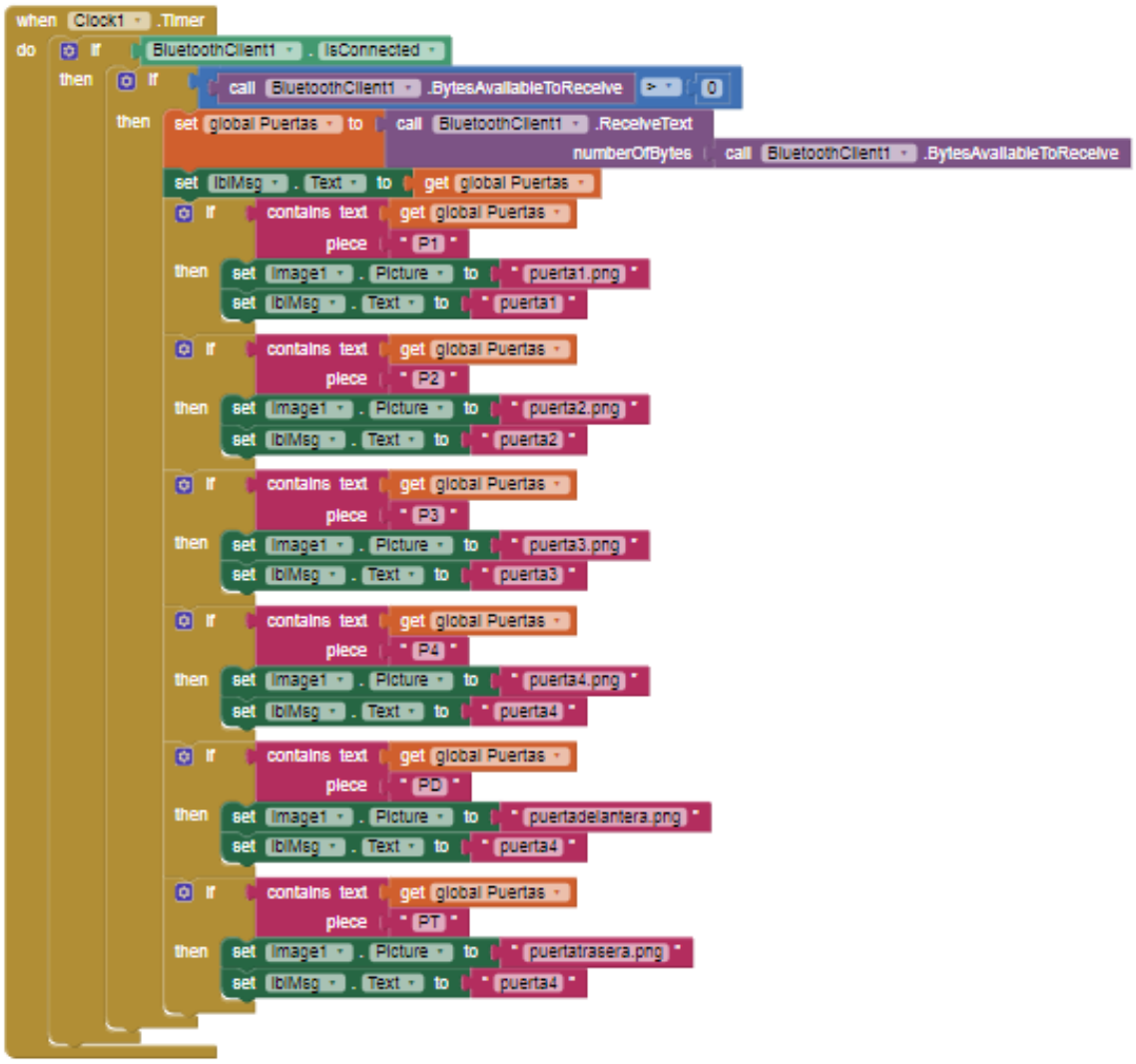
http://www.tiendatec.es/arduino/reles/523-modulo-rele-4-canales-5v-para-arduino-8405231430008.html

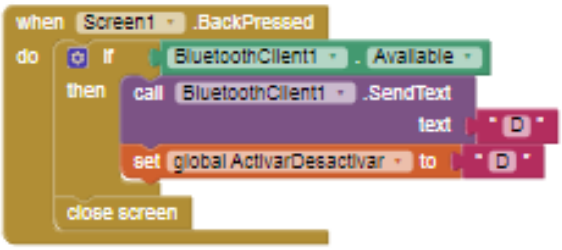
ANEXO 8

CODIGO DE PROGRAMACION EN APP INVENTOR




when Screen1 .Initialize
do if compare texts get global ValorParametro = " 1 "
then if not BluetoothServer1 . Enabled
then call Notifier1 .ShowAlert
notice " Bluetooth no disponible "
else open another screen screenName " Login "


when Clock1 .Timer
do if BluetoothClient1 . IsConnected
then if call BluetoothClient1 .BytesAvailableToReceive > 0
then set global Puertas to call BluetoothClient1 .ReceiveText
numberOfBytes call BluetoothClient1 .BytesAvailableToReceive
set lblMsg . Text to get global Puertas
if contains text get global Puertas
piece " P1 "
then set Image1 . Picture to " puerta1.png "
set lblMsg . Text to " puerta1 "
if contains text get global Puertas
piece " P2 "
then set Image1 . Picture to " puerta2.png "
set lblMsg . Text to " puerta2 "
if contains text get global Puertas
piece " P3 "
then set Image1 . Picture to " puerta3.png "
set lblMsg . Text to " puerta3 "
if contains text get global Puertas
piece " P4 "
then set Image1 . Picture to " puerta4.png "
set lblMsg . Text to " puerta4 "
if contains text get global Puertas
piece " PD "
then set Image1 . Picture to " puertadelantera.png "
set lblMsg . Text to " puerta4 "
if contains text get global Puertas
piece " PT "
then set Image1 . Picture to " puertatrasera.png "
set lblMsg . Text to " puerta4 "


when Screen1 .BackPressed
do if BluetoothClient1 . Available
then call BluetoothClient1 .SendText
text " D "
set global ActivarDesactivar to " D "
close screen

ANEXO 9

CÓDIGO DE PROGRAMACIÓN PARA RASPBERRY PI 3B

El código de programación desarrollado en Python usado para el prototipo de sistema de video vigilancia vehicular se muestra a continuación:

```
import RPi.GPIO as GPIO
import threading
import picamera
from time import sleep
from bluetooth import *
from gpiozero import Motion Sensor
from datetime import datetime
#Correo
import smtplib
import mimetypes
from email. MIMEMultipart import MIMEMultipart
from email.MIMEImage import MIMEImage
from email.Encoders import encode_base64
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
#Puertas
#Puertas delanteras
GPIO.setup(27, GPIO.IN, pull_up_down=GPIO.PUD_UP)
GPIO.setup(22, GPIO.IN, pull_up_down=GPIO.PUD_UP)
#Puertas Traseras
GPIO.setup(16, GPIO.IN, pull_up_down=GPIO.PUD_UP)
GPIO.setup(20, GPIO.IN, pull_up_down=GPIO.PUD_UP)
#CAPO
GPIO.setup(26, GPIO.IN, pull_up_down=GPIO.PUD_UP)
#CAJUELA
GPIO.setup(19, GPIO.IN, pull_up_down=GPIO.PUD_UP)
#Control remoto RFID
#Bolton A
GPIO.setup(5, GPIO.IN)
#Boton B
GPIO.setup(6, GPIO.IN)
#Boton C
GPIO.setup(12, GPIO.IN)
#Boton D
GPIO.setup(13, GPIO.IN)
GPIO.setup(25, GPIO.OUT)
GPIO.output(25, False)
#Pulso Salida Arduino
```

```
GPIO.setup(18, GPIO.OUT)
#Activacion Seguros Puertas
GPIO.setup(23, GPIO.OUT)
#Bloqueo Central y Vidrios
GPIO.setup(17, GPIO.OUT)
#Desactivacion Seguros
GPIO.setup(24, GPIO.OUT)
#Sirena
GPIO.setup(21, GPIO.OUT)
#Sensor de Movimiento
print("Inicia Sensor Movimiento")
pir = MotionSensor (4)
#Camara
print("Inicia Camara Globalmente")
camera = picamera.PiCamera()
#Bluetooth Configurations
#Bluetooth Sockets
server_sock = BluetoothSocket( RFCOMM )
server_sock.bind(("",PORT_ANY))
server_sock.listen(1)
#Abrir Socket Bluetooth
port = server_sock.getsockname()[1]
#Identificador de servicio
uuid = "94f39d29-7d6d-437d-973b-fba39e49d4ee"
#Variables globales para ejecucion de hilos
gData = "T"
gBotonA = "D"
#Funcion para emitir sonido pasando el tiempo de actividad
def Sonido(tiempo):
   GPIO.output(21, True)
   sleep(tiempo)
   GPIO.output(21, False)
#Funciones para cada boton del control remoto
def ControlA():
   print("Botono A")
   while True:
                if(GPIO.input(5)):
                        Activar(True)
                        sleep(0.1)
def ControlB():
   print("Botono B")
   while True:
                if(GPIO.input(6)):
                        Desactivar()
                        sleep(0.1)
def ControlC():
```

```
    print("Botono C")
    while True:
                if(GPIO.input(12)):
                        sleep(0.3)
                        Activar(False)
                        sleep(0.1)
def ControlD():
    print("Botono D")
    while True:
                if(GPIO.input(13)):
                        GPIO.output(25, True)
                        Sonido(0.3)
                        sleep(0.1)
                        Sonido(0.3)
                        sleep(0.1)
                        Sonido(0.3)
                        sleep(3)
                        GPIO.output(25, False)
                        sleep(0.1)
#Funcion para activar todos los controles de las puertas
def Activar(conSonido):
        gData = "A"
        if conSonido == True:
        Sonido(0.3)
        #Crea hilos de ejecucion
        #Puertas
        p1 = threading.Thread(target=Puerta1)
        p2 = threading.Thread(target=Puerta2)
        p3 = threading.Thread(target=Puerta3)
        p4 = threading.Thread(target=Puerta4)
        pd = threading.Thread(target=PuertaDelantera)
        pt = threading.Thread(target=PuertaTrasera)
      ss = threading.Thread(target=sensorsonido)
        #Ejecuta Hilos
        p1.start()
        p2.start()
        p3.start()
        p4.start()
        pd.start()
        pt.start()
        ss.start()
      #Bloque Central
        GPIO.output(17, True)
        #Seguros Puertas
        GPIO.output(23, True)
        sleep(3)
```

```
        GPIO.output(23, False)
#Funcion para desactivar todos los hilos
def Desactivar():
      print("Desactiva Alarma")
        gData = "D"
        Sonido(0.3)
        sleep(0.1)
        Sonido(0.3)
        GPIO.output(17, False)
        GPIO.output(24, True)
        sleep(3)
        GPIO.output(24, False)
#Camara
def TomaFoto():
        print("Toma Foto")
        camera.start_preview()
        sleep(2)
        camera.capture('/home/pi/pybluez/examples/simple/fotoLadron.jpg')
        camera.stop_preview()
#Envio de Correo
def EnvioCorreo():
        #Datos para el smtp de correo gmail.
        to = "samin8113@gmail.com"
        gmail_user = "samin8113"
        gmail_pwd = "santhy8113"
        smtp = "smtp.gmail.com"
        # Creamos objeto Multipart, quien sera el recipiente que enviaremos
        msg = MIMEMultipart()
        msg['From'] = gmail_user
        msg['To'] = to
        msg['Subject'] = "Alarma robo en proceso??"
        # Adjuntamos Imagen
        file = open("/home/pi/pybluez/examples/simple/fotoLadron.jpg", "rb")
        attach_image = MIMEImage(file.read())
        attach_image.add_header('Content-Disposition', 'attachment; filename =
"fotoLadron.png"')
        msg.attach(attach_image)
        print("Se conecta al smtp")
        smtpserver = smtplib.SMTP(smtp, 587)
        smtpserver.ehlo()
        smtpserver.starttls()
        smtpserver.ehlo
        smtpserver.login(gmail_user, gmail_pwd)
        print("Envia Correo")
        smtpserver.sendmail(gmail_user, to, msg.as_string())
        smtpserver.close()
```

```
            print("Cierra Smtp")
#Funcion Pulsador
def Puerta1():
    print("Puerta Izquierda Delantera")
    while True:
        input_state = GPIO.input(27)
        if input_state == False:
                #Envio datos por bluetooth
                try:
                 client_sock.send("P1")
                except: # catch *all* exceptions
                 e = sys.exc_info()[0]
                       print( "<p>Error Envio Datos Bluetooth: %s</p>" % e )
            GPIO.output(18, True)
            sleep(3)
            GPIO.output(18, False)
            Sonido(2)
            TomaFoto()
            EnvioCorreo()
        if gData == "D":
            break

def Puerta2():
    print("Puerta Derecha Delantera")
    while True:
        input_state = GPIO.input(22)
        if input_state == False:
            #Envio datos por bluetooth
            try:
                client_sock.send("P2")
            except: # catch *all* exceptions
                e = sys.exc_info()[0]
                print( "<p>Error Envio Datos Bluetooth: %s</p>" % e )
            GPIO.output(18, True)
            sleep(3)
            GPIO.output(18, False)
            Sonido(2)
            TomaFoto()
            EnvioCorreo()
        if gData == "D":
            break
def Puerta3():
    print("Puerta Trasera Izquierda")
    while True:
        input_state = GPIO.input(16)
        if input_state == False:
```

```
            #Envio datos por bluetooth
            try:
               client_sock.send("P3")
            except: # catch *all* exceptions
               e = sys.exc_info()[0]
               print( "<p>Error Envio Datos Bluetooth: %s</p>" % e )

            GPIO.output(18, True)
            sleep(3)
            GPIO.output(18, False)
            Sonido(2)
            TomaFoto()
            EnvioCorreo()
         if gData == "D":
            break
   def Puerta4():
      print("Puerta Trasera Derecha")
      while True:
         input_state = GPIO.input(20)
         if input_state == False:
            #Envio datos por bluetooth
            try:
               client_sock.send("P4")
            except: # catch *all* exceptions
               e = sys.exc_info()[0]
               print( "<p>Error Envio Datos Bluetooth: %s</p>" % e )

            GPIO.output(18, True)
            sleep(3)
            GPIO.output(18, False)
            Sonido(2)
            TomaFoto()
            EnvioCorreo()
         if gData == "D":
            break
   def Puerta Trasera():
      print("Puerta Trasera")
      while True:
         input_state = GPIO.input(19)
         if input_state == False:
            #Envio datos por bluetooth
            try:
               client_sock.send("PT")
            except: # catch *all* exceptions
               e = sys.exc_info()[0]
               print( "<p>Error Envio Datos Bluetooth: %s</p>" % e )
```

```
            GPIO.output(18, True)
            sleep(3)
            GPIO.output(18, False)
            Sonido(2)
            TomaFoto()
            EnvioCorreo()
        if gData == "D":
            break
def PuertaDelantera():
    print("Puerta Delantera")
    while True:
        input_state = GPIO.input(26)
        if input_state == False:
            #Envio datos por bluetooth
            try:
                client_sock.send("PD")
            except: # catch *all* exceptions
                e = sys.exc_info()[0]
                print( "<p>Error Envio Datos Bluetooth: %s</p>" % e )

            GPIO.output(18, True)
            sleep(3)
            GPIO.output(18, False)
            Sonido(2)
            TomaFoto()
            EnvioCorreo()
        if gData == "D":
            break
def sensorsonido():
    print('Activa Sensor Sonido')
    while True:
        input_state = GPIO.input(4)
        if input_state == True:
            print('sonido detectado')
            GPIO.output(18, True)
            sleep(3)
            GPIO.output(18, False)

            Sonido(2)
            TomaFoto()
            EnvioCorreo()
        if gData == "D":
            break
#Creo hilos por cada boton del Control remoto
ba = threading.Thread(target=ControlA)
bb = threading.Thread(target=ControlB)
```

```
bc = threading.Thread(target=ControlC)
bd = threading.Thread(target=ControlD)
print("inicializa Botones")
ba.start()
bb.start()
bc.start()
bd.start()
print("Finaliza inicializa Botones")
#Inicializo Servidor Bluetooth
advertise_service( server_sock, "Servidor",
                service_id = uuid,
                service_classes = [ uuid, SERIAL_PORT_CLASS ],
                profiles = [ SERIAL_PORT_PROFILE ],
                                        #protocols = [ OBEX_UUID ]
                 )
print("Esperando por conexion de Dispositivo en puerto: %d" % port)

client_sock, client_info = server_sock.accept()
print("Conexion aceptada con el dispositivo: ", client_info)
#Hilo de ejecucion
try:
   while True:
      data = client_sock.recv(1024)
       if len(data) == 0: break
      if data == "A":
                           print("Activado [%s]" % data)
                           Activar(True)
      if data == "D":
                           Desactivar()
                           GPIO.output(17, False)
                           print("Desactivado [%s]" % data)
         gData = data
except IOError:
   GPIO.cleanup()
   pass
print("Desconectado")
GPIO.cleanup()
client_sock.close()
server_sock.close()
print("Todo ok.!")
```

ANEXO 10

CÓDIGO DE PROGRAMACIÓN PARA ARDUINO

El código de programación desarrollado en el IDE para el Arduino usado para el GPRS de la localización se muestra a continuación:

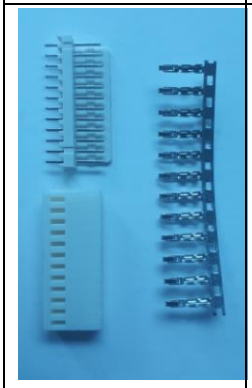
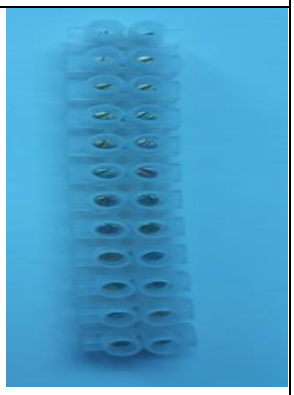
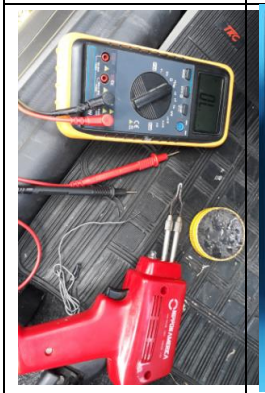
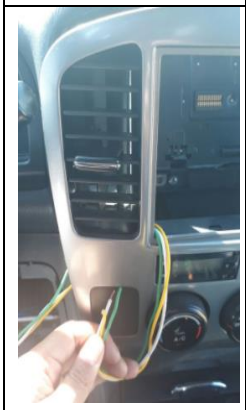
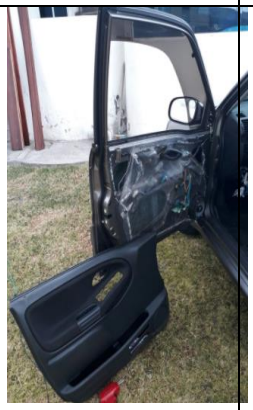
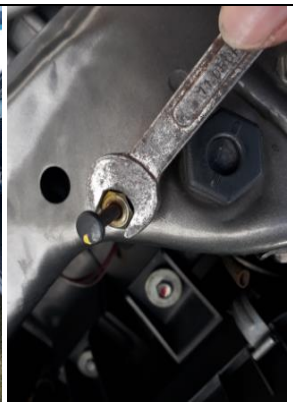
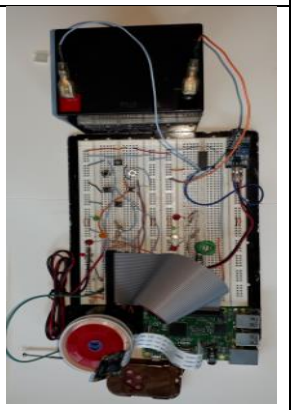
```
#include "Streaming.h"
#include <SoftwareSerial.h>
#include "GPRS.h"
#define GPRS Serial
#define GPRS_BAUD 9600

SoftwareSerial DEBUG(6, 7);
CGPRS_SIM gprs(GPRS, DEBUG);

void setup()
{
  while (!DEBUG);
  DEBUG.begin(9600);
  delay(150);
  Serial1.begin(9600);
  DEBUG << "SIM800 TEST" << endl;
  while (!gprs.init(GPRS_BAUD, SERIAL_8N1))
  {
    DEBUG.write('.');
  }
    DEBUG.println("OK");
  while (!gprs.checkNetwork());
  gprs.SMS_number = "0958742945";
  DEBUG.println("SETUP FINISH");
  //gprs.sendSMS("ALARMA");
  pinMode(8, INPUT);
  digitalWrite(8, HIGH);
  gprs.initGPS();
//  gprs.callID();
}
void loop(){
  while (1)
  {
    if (digitalRead(8))
    {
      gpsIN();
      gprs.callID();
      String SMS_data = "";
```

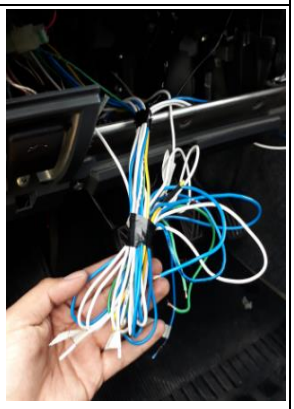
```
   SMS_data = " ALERTA " "\n" "ROBO EN PROGRESO "" \n"
          "https://maps.google.com/maps?z=12&t=m&q=loc:" + gprs.GPS_lat + "," +
          gprs.GPS_long;
      gprs.sendSMS(SMS_data);
      delay(5000);
    }
    //DEBUG.println("HOLA");
    delas(200);
    gpsIN();
    Serial1.print("LAT: ");
    Serial1.print(gprs.GPS_lat);
    Serial1.print("&LON: ");
    Serial1.println(gprs.GPS_long);
    //
    }
}
void gpsIN() {
  bool gps_true = false;
  gprs.inputStringGps = "";

  uint32_t t_gps = millis();
  gprs.getGPS();
  gprs.inputStringGps = gprs.Buffer_gprs;
  DEBUG << "gps: " << gprs.inputStringGps << endl;
  int begin_idx = 0, idx = 0;
  String ar[15] = "";
  for (uint8_t w = 0; w < 15; w++) {
    ar[w] = "";
    idx = gprs.inputStringGps.indexOf(",", begin_idx + 1);
    ar[w] = gprs.inputStringGps.substring(begin_idx + 1, idx);
    //DEBUG << "ar[" << w << "]: " << ar[w] << endl;
    begin_idx = idx;
  }

  if (ar[0] != "") {
    gprs.GPS_Time = ar[1];
    gprs.GPS_lat = ar[3];
    gprs.GPS_lat_NS = ar[4];
    gprs.GPS_long = ar[5];
    gprs.GPS_long_WE = ar[6];

    String aux;
      if (gprs.GPS_lat_NS == "S")
       {
         aux = gprs.GPS_lat;
         gprs.GPS_lat = "";
```

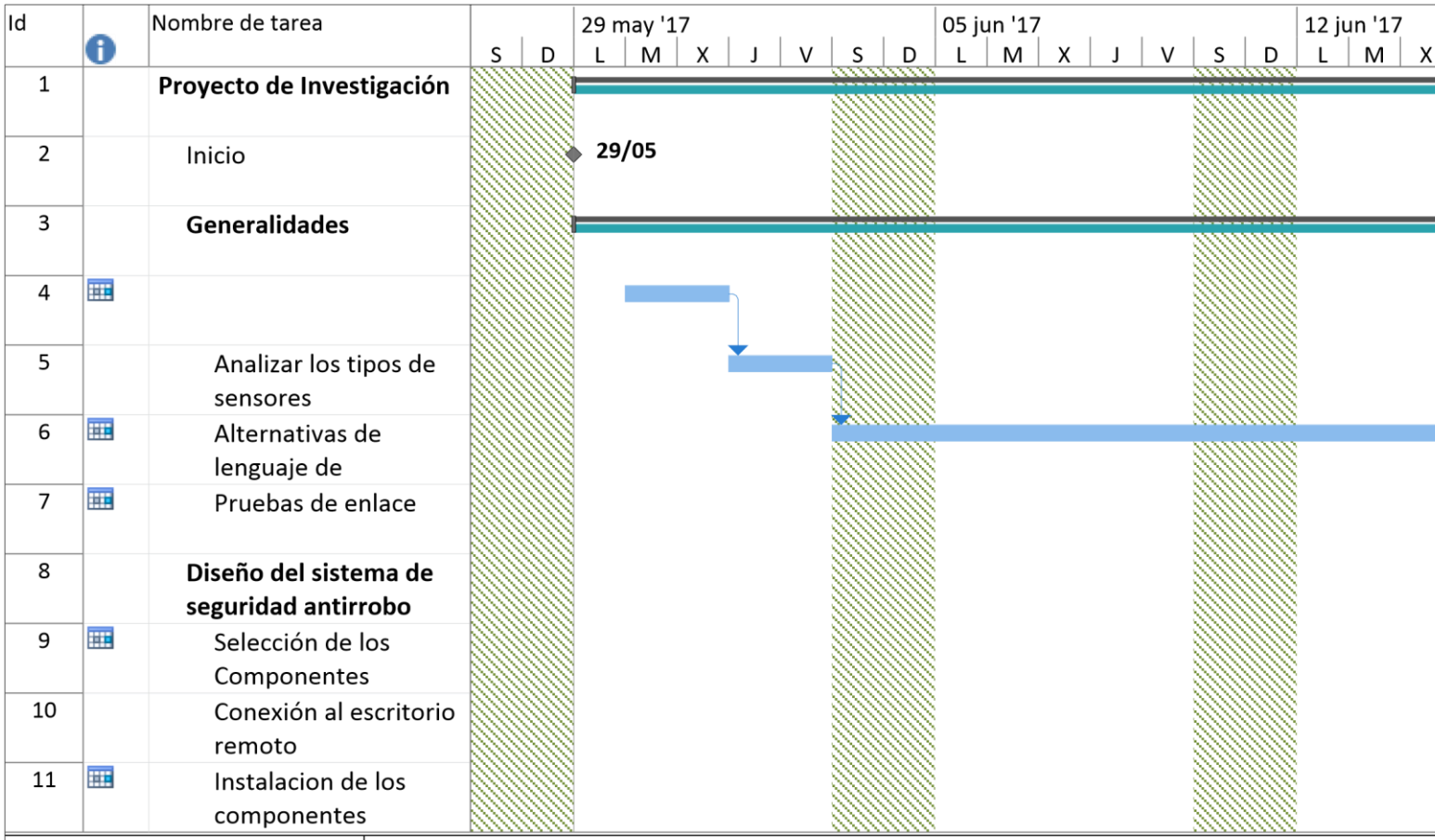
```
          gprs.GPS_lat = "-" + aux;
          }
        if (gprs.GPS_long_WE == "W")
         {
          aux = gprs.GPS_long;
          gprs.GPS_long = "";
          gprs.GPS_long = "-" + aux;
          }
    DEBUG << "gps: " << endl;
    DEBUG << gprs.GPS_Time << endl;
    DEBUG << gprs.GPS_lat << endl;
    DEBUG << gprs.GPS_long << endl;
  }
}
```

ANEXO 11

FOTOGRAFÍAS

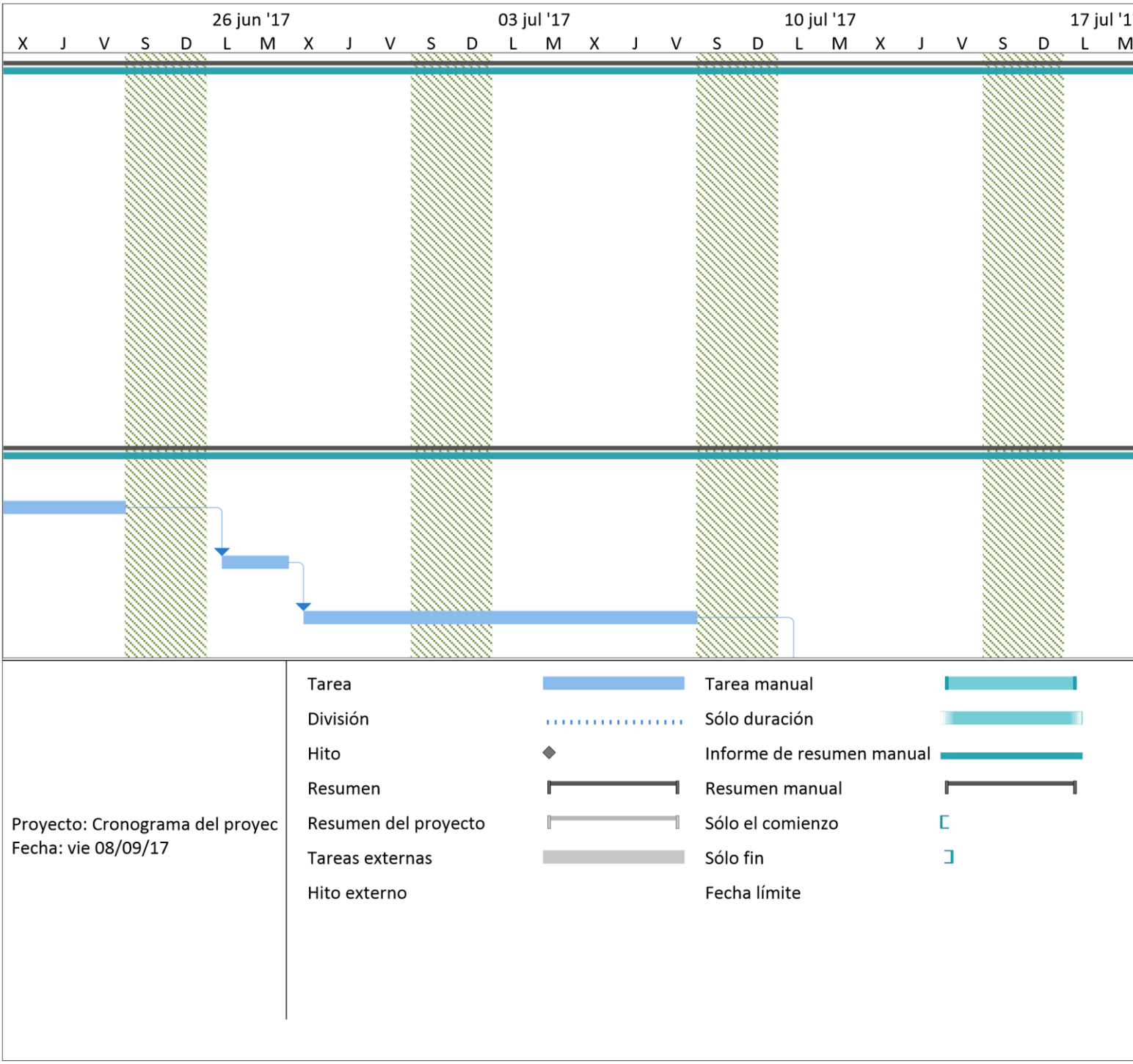
Sockets	Rele	Varios	Borneras
			
Pruebas equipos	Caja prototipo	Raspberry Pi 3B	Cable plano 40 pines
			
Cableado camara	Cableado señal de puertas	Sensor de capo	Circuito de Prueba
			
Conexiones	Modulo de señal de puertas		Cableado puertas

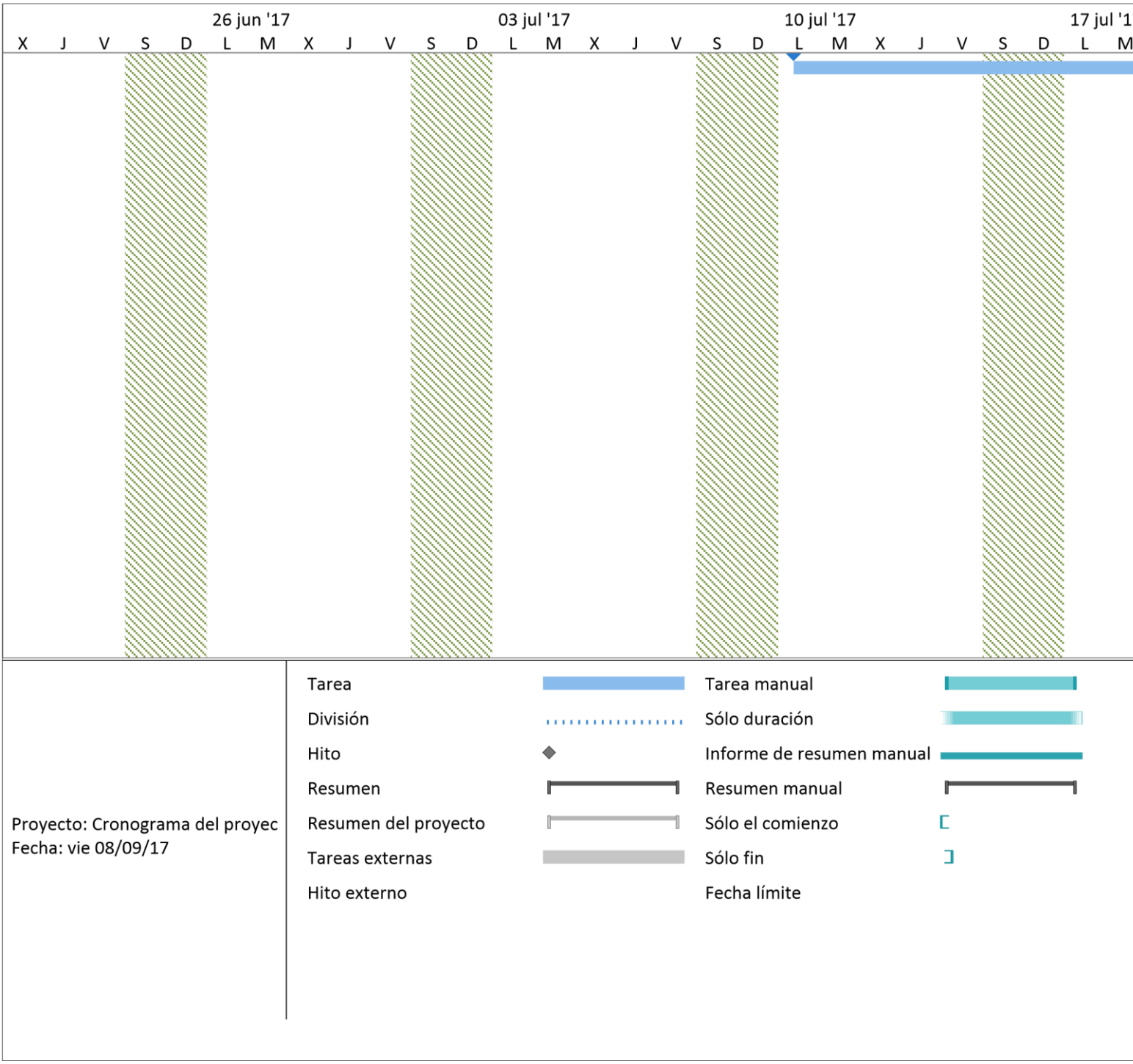
prototipo			traseras
			
Instalacion camara	Cabledo sensor puerta	Armado de prototipo	Cableado terminal
			
Prototipo posterior	Prototipo frontal	Prototipo talera	Sensor de ruido
			

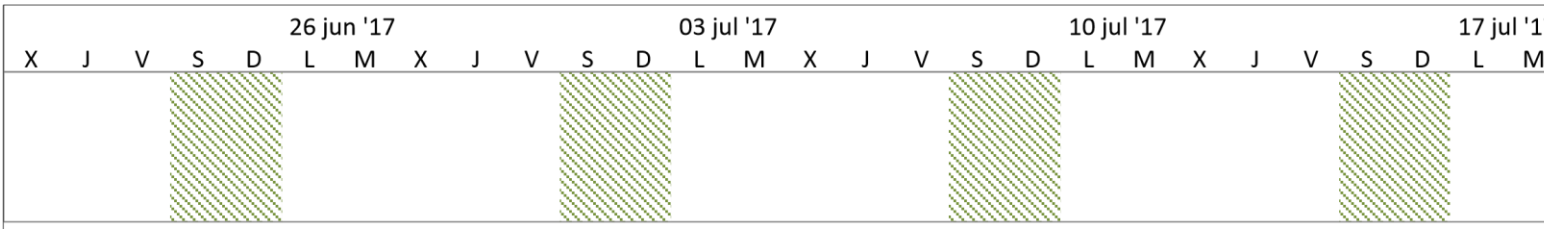


Proyecto: Cronograma del proyec
Fecha: vie 08/09/17

Tarea		Tarea manual	
División		Sólo duración	
Hito		Informe de resumen manual	
Resumen		Resumen manual	
Resumen del proyecto		Sólo el comienzo	
Tareas externas		Sólo fin	
Hito externo		Fecha límite	
Tarea inactiva		Progreso	
Hito inactivo		Progreso manual	
Resumen inactivo			

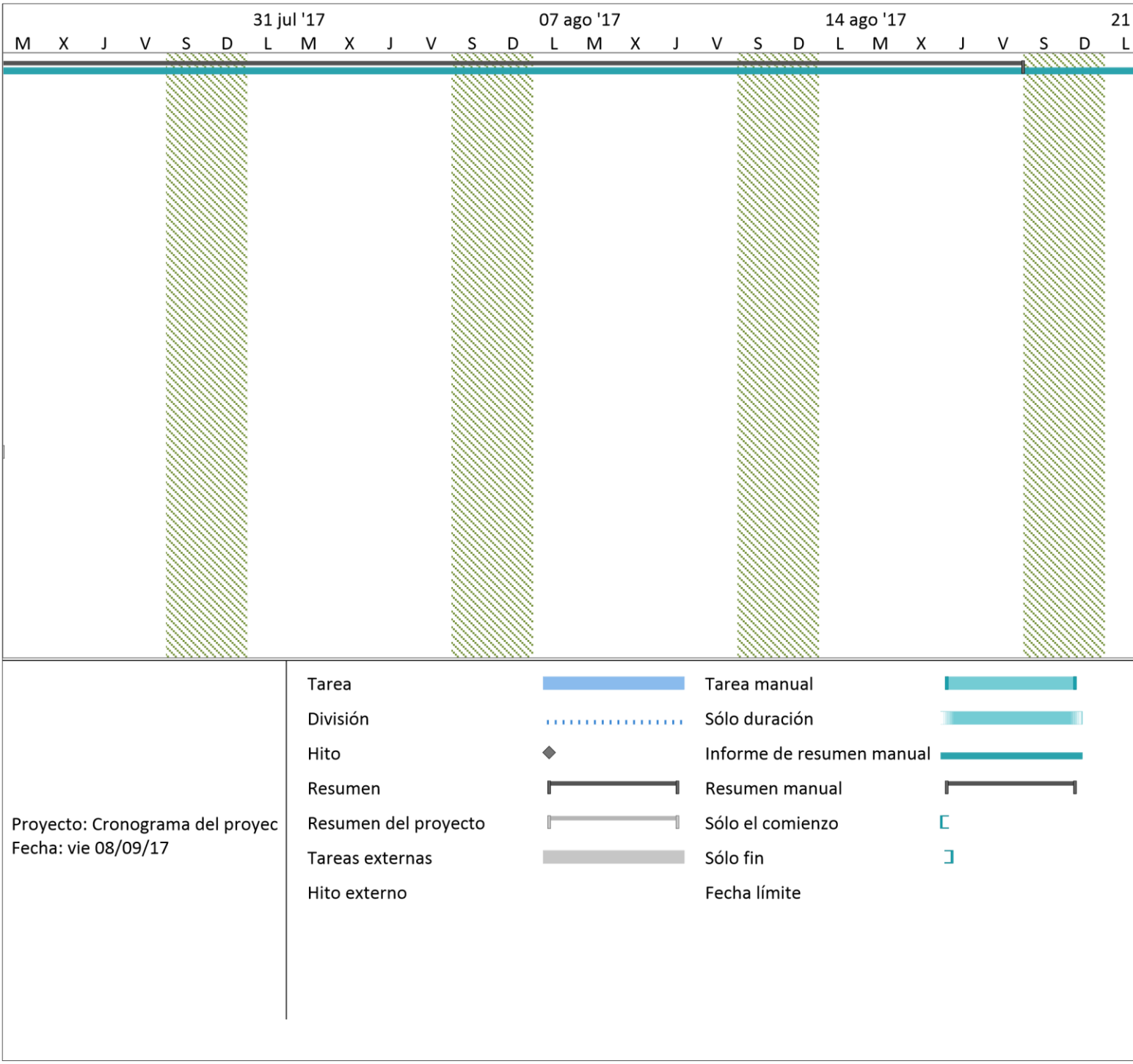

26 jun '17
03 jul '17
10 jul '17
17 jul '1
X J V S D L M X J V S D L M X J V S D L M X J V S D L M
Proyecto: Cronograma del proyec
Fecha: vie 08/09/17
Tarea
División
Hito
Resumen
Resumen del proyecto
Tareas externas
Hito externo
Tarea manual
Sólo duración
Informe de resumen manual
Resumen manual
Sólo el comienzo
Sólo fin
Fecha límite

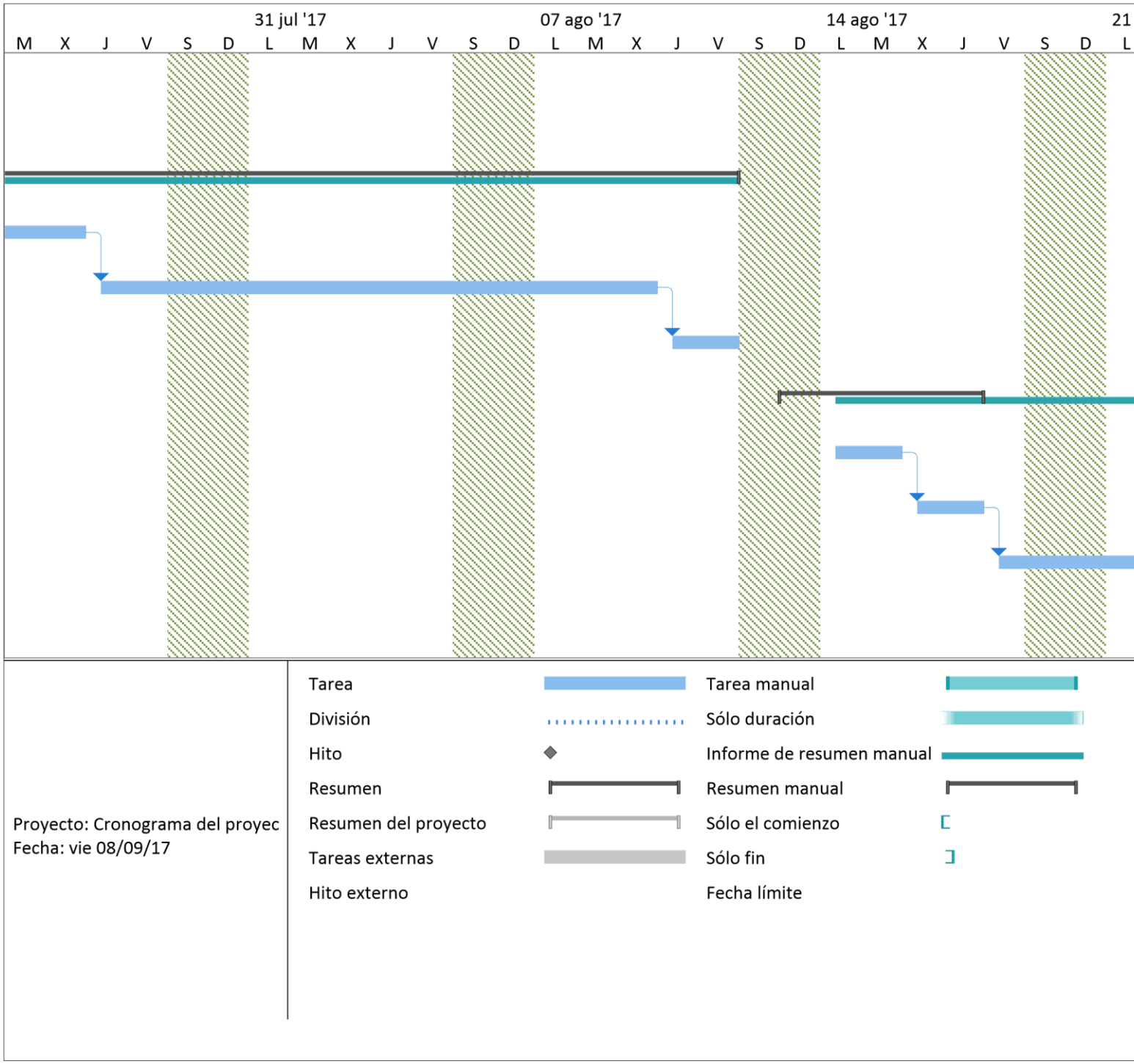

26 jun '17
03 jul '17
10 jul '17
17 jul '1
X J V S D L M X J V S D L M X J V S D L M X J V S D L M
Proyecto: Cronograma del proyec
Fecha: vie 08/09/17
Tarea
División
Hito
Resumen
Resumen del proyecto
Tareas externas
Hito externo
Tarea manual
Sólo duración
Informe de resumen manual
Resumen manual
Sólo el comienzo
Sólo fin
Fecha límite

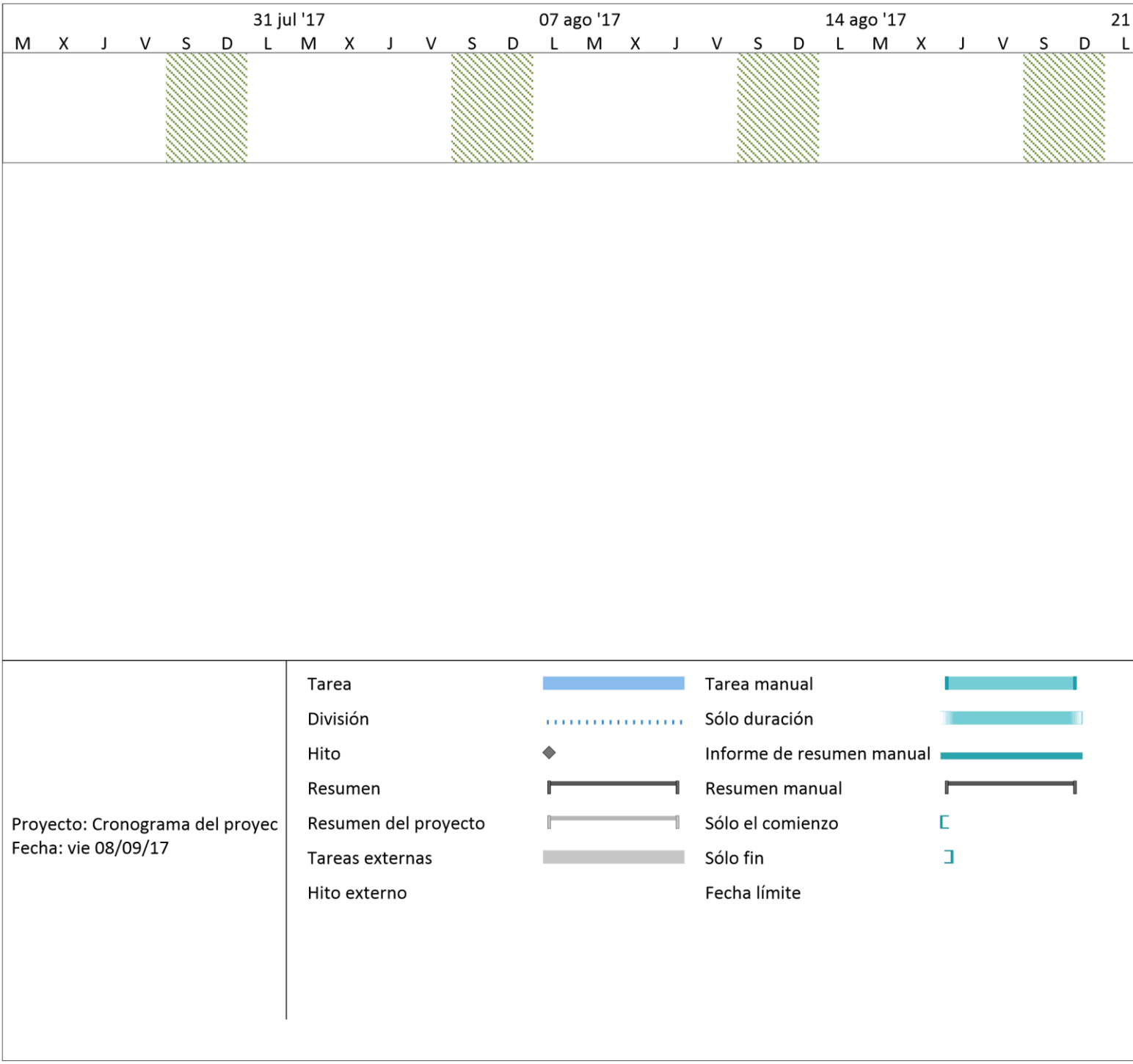


Proyecto: Cronograma del proyec
Fecha: vie 08/09/17

Tarea		Tarea manual	
División		Sólo duración	
Hito		Informe de resumen manual	
Resumen		Resumen manual	
Resumen del proyecto		Sólo el comienzo	
Tareas externas		Sólo fin	
Hito externo		Fecha límite	


31 jul '17
07 ago '17
14 ago '17
21
M X J V S D L M X J V S D L M X J V S D L M X J V S D L
Proyecto: Cronograma del proyec
Fecha: vie 08/09/17
Tarea
División
Hito
Resumen
Resumen del proyecto
Tareas externas
Hito externo
Tarea manual
Sólo duración
Informe de resumen manual
Resumen manual
Sólo el comienzo
Sólo fin
Fecha límite


31 jul '17
07 ago '17
14 ago '17
21
M X J V S D L M X J V S D L M X J V S D L M X J V S D L
Proyecto: Cronograma del proyec
Fecha: vie 08/09/17
Tarea
División
Hito
Resumen
Resumen del proyecto
Tareas externas
Hito externo
Tarea manual
Sólo duración
Informe de resumen manual
Resumen manual
Sólo el comienzo
Sólo fin
Fecha límite


31 jul '17
07 ago '17
14 ago '17
21
M X J V S D L M X J V S D L M X J V S D L M X J V S D L
Proyecto: Cronograma del proyec
Fecha: vie 08/09/17
Tarea
División
Hito
Resumen
Resumen del proyecto
Tareas externas
Hito externo
Tarea manual
Sólo duración
Informe de resumen manual
Resumen manual
Sólo el comienzo
Sólo fin
Fecha límite