

**UNIVERSIDAD NACIONAL TECNOLÓGICA DE LIMA SUR**

**FACULTAD DE INGENIERÍA Y GESTIÓN**

**ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA Y  
TELECOMUNICACIONES**



**“DESARROLLO DE UN PROTOTIPO, UTILIZANDO MÓDULOS  
ELECTRÓNICOS Y TECNOLOGÍA GSM PARA MONITOREO DE  
VIVIENDAS EN ZONAS RURALES EN EL DISTRITO DE MOLINOPAMPA  
– AMAZONAS, 2019”**

**TRABAJO DE SUFICIENCIA PROFESIONAL**

Para optar el Título Profesional de

**INGENIERO ELECTRÓNICO Y TELECOMUNICACIONES**

**PRESENTADO POR EL BACHILLER**

CARRION SALAZAR, ANTHONY GINO

**Villa El Salvador  
2019**

## **DEDICATORIA**

El presente trabajo se la dedico a toda mi familia en especial a mis padres que gracias a su apoyo he podido desarrollar este proyecto.

## **AGRADECIMIENTO**

Agradezco a Dios, por darme las fuerzas necesarias para lograr cada una de mis metas, también a mis compañeros que a lo largo de la vida universitaria hemos desarrollado innumerables proyectos, al grupo de robótica que hemos compartido triunfos y derrotas en cada competencia.

## ÍNDICE

<b>INTRODUCCIÓN</b> .....	1
<b>CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA</b> .....	3
1.1. Descripción de la Realidad Problemática .....	3
1.2. Justificación del Problema .....	5
1.3. Delimitación del Proyecto .....	7
1.3.1. Teórica .....	7
1.3.2. Temporal .....	7
1.3.3. Espacial .....	7
1.4. Formulación del Problema .....	7
1.4.1. Problema General .....	7
1.4.2. Problemas específicos .....	8
1.5. Objetivos .....	8
1.5.1. Objetivo General .....	8
1.5.2. Objetivos Específicos .....	8
<b>CAPÍTULO II: MARCO TEÓRICO</b> .....	9
2.1. Antecedentes .....	9
2.1.1. Antecedentes Internacionales .....	9
2.1.2. Antecedentes Nacionales .....	10
2.2. Bases Teóricas .....	12
2.2.1. Comunicaciones Móviles .....	12
2.2.2. Composición de las comunicaciones móviles .....	12
2.2.3. Tecnología GSM .....	13
2.2.4. MODULO SIM 800L .....	14
2.2.5. Comandos AT .....	15
2.2.6. ARDUINO .....	17
2.2.6.1. Arduino Nano .....	17

2.2.6.2. IDE Arduino .....	18
2.2.7. SESORES INFRAROJOS.....	19
2.2.8. MODULO PIR HC-SR501 .....	20
2.2.9. MODULO KY-038 .....	21
2.2.10. STEP UP XL6009 .....	21
2.2.11. EasyEDA.....	22
2.3. Definición de términos básicos .....	23
<b>CAPÍTULO III: DESARROLLO DEL TRABAJO DE SUFICIENCIA PROFESIONAL .....</b>	<b>25</b>
3.1. Modelo de solución propuesto .....	25
3.2. Diseño de la PCB para el soporte de los módulos electrónicos .....	26
3.3. Lógica de programación del prototipo.....	30
3.3.1. Adquisición de datos de los sensores .....	31
3.3.2. Emisión - recepción de SMS y llamadas telefónicas.....	34
3.3.3. Procesamiento de los datos .....	34
3.4. Implementación del proyecto .....	46
3.5. Consolidación de Resultados .....	47
3.5.1. Pruebas del algoritmo .....	48
3.6. Presupuesto del proyecto .....	52
3.7. Diagrama de Gantt .....	53
<b>CONCLUSIONES .....</b>	<b>54</b>
<b>RECOMENDACIONES .....</b>	<b>55</b>
<b>BIBLIOGRAFÍA .....</b>	<b>56</b>
<b>ANEXOS .....</b>	<b>59</b>

## LISTADO DE FIGURAS

Figura 1: Porcentaje de víctimas de hechos delictivos y percepción de inseguridad en lima entre los meses marzo y agosto 2019. ....	1
Figura 2: Equipamiento TIC del hogar 2012-2018.....	3
Figura 3: Tasa de Hogares con Telefonía Móvil.....	4
Figura 4: Porcentaje de personas que Han sido víctimas de robos de sus viviendas en el distrito de Molinopampa.....	6
Figura 5: líneas telefónicas funcionales en el lugar de las viviendas del distrito Molinopampa.....	6
Figura 6: Estructura de una red móvil. ....	13
Figura 7: Arquitectura de la tecnología GSM.....	14
Figura 8: Módulo SIM 800L. ....	15
Figura 9: Esquema de pines de Arduino nano. ....	18
Figura 10: Entorno de programación de Arduino.....	19
Figura 11: Modulo FC-51.....	20
Figura 12 : Sensor PIR HC-SR501.....	21
Figura 13: Módulo KY-038. ....	21
Figura14: Modulo STEP UP XL6009.....	22
Figura 15: Ventana principal de EasyEDA. ....	22
Figura 16: Diagrama de Bloques- modelo de solución propuesta.....	25
Figura 17: Diseño del circuito en EasyEDA.....	27
Figura 18: Entrada de alimentación del circuito.....	28
Figura 19: Esquema del circuito para alimentación del ARDUINO y el módulo SIM800L.....	28
Figura 20: Jack para la entrada de señal de los sensores. ....	29
Figura 21: Borneras auxiliares del circuito. ....	29
Figura 22: PCB cara superior y cara inferior. ....	30
Figura 23: Esquemático del módulo IR FC -51.....	31
Figura 24: (1) Sensor no detecta obstáculo. (2) sensor detecta obstáculo. ....	32
Figura 25: Posicionamiento del sensor en el marco de la puerta. ....	32
Figura 26: Modulo PiR y lente de Fresnel. ....	33
Figura 27: Detector PRI con lente de fresnel. ....	33
Figura 28: Posición del sensor PIR dentro de la vivienda. ....	34

Figura 29: Diagrama de flujo global del prototipo de monitoreo de viviendas. ....	35
Figura 30: Declaración de las variables globales y librerías necesarias. ....	36
Figura 31: Variables direccionadas al pin de entrada de los sensores.....	37
Figura 32: Función setup, activación del monitor serial y declaración de los pines de los sensores como entradas.....	38
Figura 33: Comandos AT para configurar el SIM800L. ....	38
Figura 34: Función loop(). ....	39
Figura 35: Guardar información de entrada del SIM800L. ....	39
Figura 36: Condicional de acuerdo al mensaje de entrada del SIM800L. ....	40
Figura 37: Código para mostrar en pantalla lo almacenado en las variables. ....	41
Figura 38: Código para cargar información de número telefónico, hora, fecha y mensaje de la información obtenida del SIM800L.....	41
Figura 39: Se guarda las respuestas de los comandos AT enviados al módulo SIM800L.....	42
Figura 40: Se muestra en pantalla la las respuestas a los comandos AT enviadas a la SIM800L. ....	42
Figura 41: Funciones de verificación de usuario, llamada éntrete y SMS entrante. ....	42
Figura 42: Código para comparar número de celulares que llaman con los números pre establecidos. ....	43
Figura 43: Configuración para realizar acciones de acuerdo al mensaje interpretado. ....	43
Figura 44: Programación para escribir y enviar un mensaje desde el sistema al teléfono pre establecido ....	44
Figura 45: Función que mantiene la información de los sensores actualizadas...	45
Figura 46. Algoritmo para enviar SMS que advierte al usuario de algún evento de peligro de la vivienda.....	46
Figura 47: PCB en físico, y PCB con los módulos electrónicos.....	46
Figura 48: Modelado en 3D e impreso de las carcasas de los sensores. ....	47
Figura 49: Maqueta de pruebas. ....	47
Figura 50: Entrada de SMS en Formato ASCCI y TEXTO. ....	48
Figura 51: Mensaje de entrada para registrar un nuevo número de usuario. ....	49
Figura 52: Respuesta del mensaje SMS para registra un numero Celular.....	50
Figura 53: Respuesta para saber el estado de la puerta.....	50

Figura 54: Mensaje enviado luego de procesar el mensaje de entrada. ....	51
Figura 55: mensaje de alerta, visto en el monitor serial y el celular. ....	51
Figura 56: Diagrama de Gantt, muestra las actividades que se realizó. ....	53

## LISTADO DE TABLAS

Tabla 1 Comandos AT más utilizados con el módulo SIM800L. ....	16
Tabla 2 Tabla de presupuesto.....	52

## INTRODUCCIÓN

El presente proyecto hace referencia a la inseguridad de nuestro país, precisa, en las viviendas de las zonas rurales, la inseguridad es un tema que se viene tratando durante muchos años, según las investigaciones realizadas por el INE I en el año 2013 un 35.9% fueron víctimas de algún hecho delictivo y para el año 2018 las cifra bajó a 25.7% (INEI I. N., 2019). Notándose que para el 2019 las víctimas han ido bajando, en la misma investigación del INEI, solo en la provincia de lima entre los meses marzo y agosto del 2019 un 30.3% fueron víctimas de algún hecho delictivo y un 89.4% cree que será víctima de algún hecho delictivo.



Figura 1: Porcentaje de víctimas de hechos delictivos y percepción de inseguridad en lima entre los meses marzo y agosto 2019.

Fuente INEI.

Un caso particular de esta inseguridad son los robos de viviendas de los que muchos tememos, ya que en ella se encuentran gran parte de nuestras pertenencias, según la Encuesta Nacional de Programas presupuestales la cual también investiga a las viviendas de áreas urbanas que fueron afectadas por robos o intento de robos (INEI I. N., 2019), en la cual menciona que en el año 2018, a nivel nacional urbano, el 9.5% de la viviendas fueron afectadas por robos o intento de robos.

Al pasar de los años las personas han tratado de cierta manera, prevenir los intentos de robos utilizando la tecnología a su favor; instalando cámaras de video vigilancia, cercas eléctricas perimetrales, etc. Pero muchas veces los costos no son accesibles a ciertos sectores de nuestro país, y es allí donde la delincuencia actúa con más fuerza.

Elaborar un prototipo electrónico de monitoreo para viviendas rurales ayudara a estar alerta frente a posibles robos de las pertenencias que se encuentran dentro de las viviendas, utilizando la tecnología GSM como medio de comunicación inalámbrica ya que es el estándar de telefonía móvil más extendido en el mundo y de bajo coste.

## CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA

### 1.1. Descripción de la Realidad Problemática

No es de sorpresa saber que la mayoría de personas han sido víctimas de algún acto delictivo, actualmente nuestro país está pasando hechos delictivos y con mayor frecuencia en las viviendas, provocando grandiosas pérdidas económicas y humanas.

En vista de ello la tecnología ha ido avanzado y evolucionando para satisfacer las necesidades del ser humano, los sistemas de telecomunicaciones cada vez han ido mejorando y la transmisión de datos ahora son mucho más accesibles, en una encuesta de OSIPTEL, muestra que en el año 2018 más de un 73% de hogares peruanos ya tiene por lo menos un Smartphone para conectarse a internet (OSIPTEL, 2019).

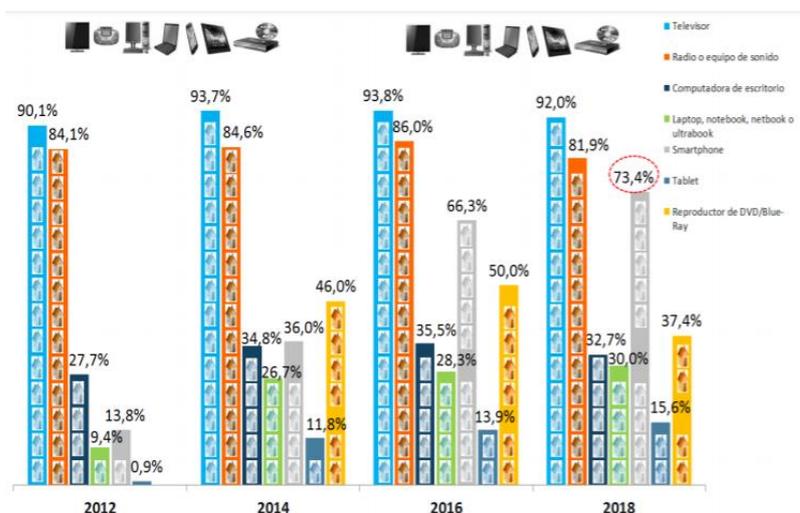
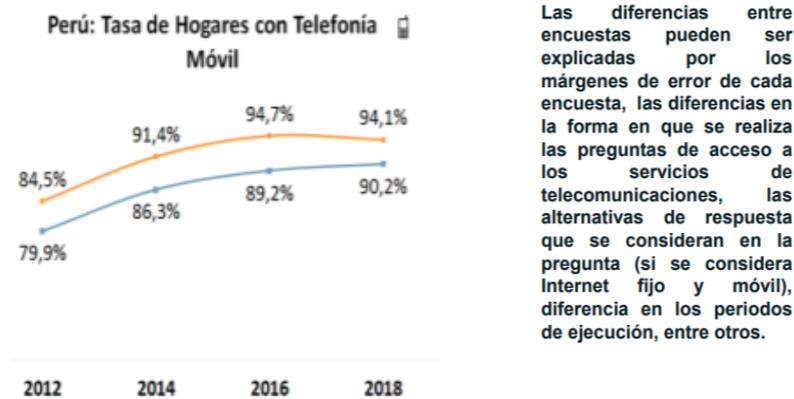


Figura 2: Equipamiento TIC del hogar 2012-2018.

Fuente OSIPTEL: "Encuesta Residencial de Servicios de Telecomunicaciones (ERESTEL 2018)".

Así mismo se puede notar que hay un alto crecimiento de usuarios que cuentan con algún tipo de teléfono móvil en el hogar, en la encuesta realizada por ERESTEL 2018, el 94.1% de usuarios cuentan con un teléfono móvil y la encuesta realizada por ENAHO, muestra que un 90,20% de los usuarios cuentan con un teléfono móvil (OSIPTEL, 2019). En la figura 3 se puede apreciar de forma gráfica los resultados de las dos encuestas.



*Figura 3:* Tasa de Hogares con Telefonía Móvil.  
Fuente OSIPTEL “Encuesta Residencial de Servicios de Telecomunicaciones (ERESTEL 2018)”.

En las zonas rurales, tomando como referencia en el Distrito de Molinopampa, Provincia de Chachapoyas, Departamento de Amazonas, zona en la que la mayoría de las familias que lo pueblan, se dedican exclusivamente a la crianza de ganado vacuno, y en diferentes puntos geográficos de la localidad, convirtiéndose en la principal fuente de ingresos económicos de las familias; dichos lugareños, tienen su hato ganadero lejos del pueblo, y en una de las partes de las canchas de pastoreo, tienen una casa, la misma que, aparte de usarlo como habitación y/o acampamento, es usado como almacén de insumos, materiales, herramientas, equipos y máquinas, elementos indispensables para que realicen labores productivas. Cabe destacar que en su mayoría, este tipo de viviendas cuentan con sistemas de alumbrado fotovoltaico y cercas eléctricas para el manejo de pastoreo en vacunos, todo el kit se encuentra en estas casas (paneles solares baterías reguladores de voltajes, impulsores eléctricos, varillas de metal para soporte de filamentos de los cercos eléctricos, etc.), estos ganaderos, en muchas de la veces, van algunas horas del día para hacer labores como rodeo de sus ganados, ordeño de las vacas o alguna otra actividad de corto tiempo, retornando a sus casas en el pueblo, para complementar sus labores en otra actividad o trasladándose a otra

zona en donde también cuentan con más crianzas en otro potrero, quedándose estos inmuebles en la vulnerabilidad de inseguridad y a merced de los delincuentes, por lo tanto los actos de robos son más constantes por lo mismo que, la mayoría de las horas del día, esas viviendas se encuentran sin personal, y en mucho de los casos estas personas no tienen los recursos económicos suficientes para comprarse los equipos de seguridad de viviendas, como lo tienen en otros lugares, mucho menos para contratar guardianía, además estos medios tecnológicos convencionales, no se adaptan a sus necesidades.

Estos actos delictivos, son barrera que imposibilita a otras personas que desean invertir, no exclusivamente en ganadería, sino en distintos proyectos de desarrollo productivo, ya que es una incertidumbre de perder en algún momento todo lo invertido, restando de esta manera el aparato productivo de la zona y porque no decir del país. Sumado a todo lo explicado, el sistema de justicia, que en alguna manera está a favor del delincuente, como: pruebas de lo ocurrido, sospechas, fecha y hora, no existe suficiente personal policial, falta de combustible, a cuánto asciende lo perdido, etc. mientras tanto, el delincuente se está preparando para volver a robar.

## **1.2. Justificación del Problema**

De acuerdo a una publicación del diario Gestión en enero del 2016 en una entrevista a Gustavo Velasco, gerente general de la empresa de seguridad Verisure, menciono que el 90% de los robos a hogares se lleva a cabo cuando están desocupadas.

Una encuesta realizada a 50 pobladores del distrito de Molinopampa demuestra que a un 68% de los entrevistados han sido víctimas de robo en su casa de campo, demostrando de esa manera la inseguridad que muchas veces afecta al crecimiento económico de las personas y del mismo distrito. En la figura 4 se puede apreciar un gráfico porcentual.



Figura 4: Porcentaje de personas que Han sido víctimas de robos de sus viviendas en el distrito de Molinopampa.  
Fuente: Elaboración propia.

Cada uno de los entrevistados tiene su casa de campo en distintos puntos del distritito de Molinopampa, y para el propósito de este proyecto el 100% cuenta con un teléfono celular, solo 5 personas no cuentan con línea telefónica en el lugar donde se encuentra su vivienda, de las 45 personas restante 40 de ellos cuentan con el servicio telefónico de Claro y 5 personas con el de Movistar, tal como se muestra en la figura 5.



Figura 5: líneas telefónicas funcionales en el lugar de las viviendas del distrito Molinopampa.  
Fuente: Elaboración propia

Por ello en este proyecto se pretende interconectar un sistema de seguridad de viviendas rurales monitoreada por un celular (no es precisos que tenga conexión a internet), para logra el monitoreo del hogar se utilizara el servicio de mensajes simple (SMS), lo cual facilitaría de gran manera a los lugares que no cuentan con internet pero que si tienen algún tipo de línea telefónica ya se de Movistar, Claro, Bitel, Entel, etc.

### **1.3. Delimitación del Proyecto**

#### **1.3.1. Teórica**

Para minimizar costos en su fabricación del presente prototipo se utilizaron dispositivos electrónicos como el Arduino nano, lenguaje de programación C++, módulos infrarrojos, módulos PIR, software de diseño de circuitos online EasyEDA, Modulo SIM800L el cual utiliza el protocolo GSM , comandos AT para su codificación, con modo de comunicación Full-duplex.

#### **1.3.2. Temporal**

El desarrollo del proyecto comprende el periodo desde agosto 2019 a noviembre 2019.

#### **1.3.3. Espacial**

La elaboración del prototipo se desarrollara en los laboratorios de la Universidad Nacional Tecnológica de lima Sur, en las pruebas desarrolladas se tomaran en cuenta al modelo de las viviendas del distrito de Molinopampa, departamento de amazonas.

### **1.4. Formulación del Problema**

#### **1.4.1 Problema General**

- ¿Cuál es el prototipo electrónico más favorable para mejorar la seguridad y monitorear el control de acceso de las viviendas en las zonas rurales del distrito de Molinopampa-departamento de Amazonas?

#### **1.4.2. Problemas específicos**

- ¿Qué tipo de tecnología se utilizará para desarrollar el prototipo electrónico que ayudará a mejorar la seguridad de las viviendas en zonas rurales en el distrito de Molinopampa?
- ¿Cómo elaborar el PCB del prototipo electrónico que ayudará a mejorar la seguridad de las viviendas en zonas rurales del distrito de Molinopampa?
- ¿De qué manera este prototipo electrónico servirá de base para elaborar otros de mayor envergadura y mejorar la seguridad de las viviendas en las zonas rurales?

#### **1.5. Objetivos**

##### **1.5.1. Objetivo General**

- Desarrollar un prototipo, utilizando módulos electrónicos y tecnología GSM para el monitoreo del control de acceso y mejorar la seguridad de las viviendas en zonas rurales en el distrito de Molinopampa departamento de Amazonas.

##### **1.5.2. Objetivos Específicos**

- Utilizar la tecnología GSM para el desarrollo del prototipo electrónico para monitoreo y control de acceso de viviendas en zonas rurales.
- Diseñar la placa electrónica que soporte a los módulos que se utilizaran en el desarrollo del prototipo electrónico para monitoreo y control de acceso de viviendas en zonas rurales.
- Diseñar un prototipo escalable, para poder ajustarle más tecnología en el futuro.

## CAPÍTULO II: MARCO TEÓRICO

### 2.1. Antecedentes

Como antecedentes que ayudan a enriquecer el presente trabajo, en la utilización de la tecnología GSM como vía de monitoreo inalámbrico tenemos:

#### 2.1.1. Antecedentes Internacionales

Zambrano (2012) en su trabajo de investigación que se titula “*Sistema de alarma para mejorar la seguridad de la Empresa AUPLATEC ubicada en el Cantón Pelileo*”, teniendo como objetivo principal Rediseñar el sistema de alarmas para mejorar la seguridad de la empresa AUPLATEC, y al final del estudio concluyo:

- En base a las necesidades de protección y seguridad que requiere la empresa AUPLATEC, se plantea un sistema de alarma vía GSM que permita notificar eventos, relacionados con la operacionalidad y funcionamiento del sistema; estableciendo un control a distancia.
- Las pérdidas económicas para la empresa AUPLATEC se pueden disminuir notablemente contando con el sistema de alarmas vía GSM propuesta en el presente proyecto, garantizando la seguridad total para toda la empresa.

Ordoñez (2010) en su trabajo titulado: *“Monitoreo y control mediante mensajes de texto (GSM) para un sistema automatizado de propagación vegetativa de platas de ornato”*, cuyo objetivo principal fue diseñar e implementar un sistema de automatización para mejorar la propagación de plantas ornamentales, que además se pueden controlar y monitorear mediante mensaje de textos GSM (SMS) de la cual concluye que sus sistema logro:

- El mejor arraigue de las plantas permitiendo un trasplante más fácil, mejorando la calidad de las plantas y en consecuencia la prestación de las mismas, aumentando la buena imagen ya establecida del invernadero.
- consiguió una propagación más rápida y por consecuencia acelerar y mejorar el sistema de producción del invernadero mediante una automatización.

Se puede precisar en esta investigación, que el uso de la tecnología GSM mediante la monitorización del invernadero por mensajes de texto (SMS), ayudo en el monitorea inalámbrico frente alguna eventualidad no pronosticada en el sistema con una rápida respuesta del usuario.

### **2.1.2. Antecedentes Nacionales**

Zúñiga (2018) en su trabajo de investigación titulado: *“Diseño E Implementación De Un Timbre Inteligente Basado En El Internet De Las Cosas (IOT) Para Fortalecer La Seguridad Contra Robos En Viviendas Sociales”*, al término de su proyecto concluyó:

- Se estudiaron los diferentes sistemas de monitoreo como cámaras directamente conectadas a un VDR, cámaras con conexión a internet y cámaras de video vigilancia supervisadas por empresas públicas o privadas

- Se diseñó y se logró interconectar con interfaz directa el timbre y la Raspberry por medio del pin 23, luego a través del conversor de USB a TTL utilizando el protocolo RS-232 conectamos la Raspberry con el Sim800L. Luego se logró conectar el teléfono móvil con nuestro sistema, a través del aplicativo AndroidStreamingMediaPlayer.
- Realizando pruebas en diferentes escenarios a diferentes distancias, se logró que el sistema funcione, logrando transmitir video por streaming.

En dicho proyecto mostro el funcionamiento de la tecnología GSM para poder vincular su sistema con el usuario, así mismo logro hacer Streaming vinculando las cámaras de seguridad con la plataforma YouTube en la que se puede acceder al video en vivo y con una cuenta privada, todo esto logrando tener acceso a internet

Mendes & Campos (2016) en su tesis titulada “*Desarrollo del prototipo de un sistema de seguridad basado en tecnología GSM para la empresa familia CASTRO Y BACA S.A.C.*” cuyo objetivo general es el configurar e implementar un Prototipo de un Sistema de Seguridad basado en tecnología GSM para la empresa FC&B S.A.C. de la cual concluyó lo siguiente:

- Se logró la correcta instalación de los dispositivos de adquisición datos, lo que permitió la correcta comunicación entre estos; además superar las dificultades que se presentaron, como eliminar las señales falsas que activaban al sistema.
- Se obtuvo una lógica de programación eficiente que permitió la integración en conjunto entre los dispositivos y el módulo GSM. Para lograr ello se priorizó la organización de la programación y se determinó las condiciones generales.

## 2.2. Bases Teóricas

### 2.2.1. Comunicaciones Móviles

Las comunicaciones móviles han permitido que los usuarios puedan utilizar el servicio de telecomunicaciones para intercomunicarse mientras se desplacen a lo largo del territorio, esta movilidad de los usuarios que se encuentran en los extremos de la comunicación ha permitido la utilización de radiofrecuencia para su comunicación por encima de los hilos o cables telefónicos.

### 2.2.2. Composición de las comunicaciones móviles

#### **Estaciones fijas**

Son las estaciones diseñadas para la interconexión de la comunicación sin tener que moverse, entre ellas se encuentran (Padilla, 2011).

#### **Estaciones base BS.** (Base Station).

Son las estaciones fijas por la cual se maneja comunicaciones de baja, media o alta bidireccional, utilizado para la comunicación de uno o más radios móviles o teléfonos celulares, sirviendo como punto de acceso para que dos terminales se comuniquen a través de la estación base (Padilla, 2011).

Bajo el contexto de la telefónica móvil se designa a una estación base como **BTS** (Base Transceiver Station), el cual está equipado por transmisores y receptores de radio, normalmente se conectan las antenas en lo más alto de la torre para una mejor cobertura o línea de vista con las demás antenas (Hermando, 2015).

#### **Estación de control CS.**

Son utilizadas para gestionar a una estación base, o una estación repetidora (RS) que son las encargadas de retransmitir las señales recibidas, logrando de esa manera una mayor cobertura (Padilla, 2011).

## Estaciones móviles

Son las estaciones radioeléctricas previstas para su utilización en vehículos en marcha o que efectúen paradas en puntos determinados para aliar el rango de la comunicación, las estaciones también pueden ser equipos portátiles (Handy) (Padilla, 2011).

## Equipos de control

Los equipos de control son necesarios para el funcionamiento de las estaciones base, generan y receptiona llamadas, localiza e identifica usuarios, equipos y vehículos, desarrolla la transferencia de llamadas, etc (Padilla, 2011).

En la figura 6 se muestra un esquema con los componentes utilizados en las comunicaciones móviles

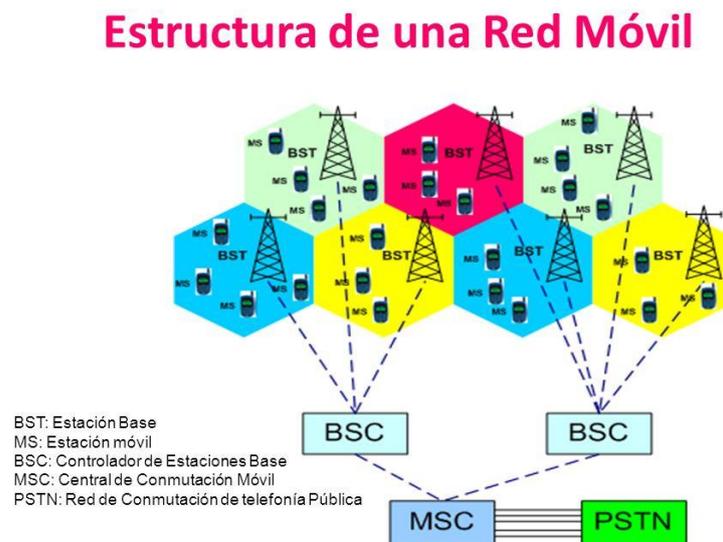


Figura 6: Estructura de una red móvil.  
Fuente: <https://slideplayer.es/slide/1101541/>

### 2.2.3. Tecnología GSM

Sistema global para telecomunicaciones móviles (*Global System for Mobile Telecommunications, GSM*), surgió por motivos de estudio para el desarrollo de un sistema móvil terrestre inicialmente para las telecomunicaciones europeas, con una serie de normativas que fueran adoptadas mundialmente, originalmente estuvo al mando de la CEPT (Conference of European Posts and Telegraphs), para 1989

fue transferida a ETSI (European Telecommunications Standards Institute ), y en 1991 se permitió que esta tecnología se utilizase no solo en Europa sino también en otros países (GSMA, 2017).

Debido a la alta demanda de usos de la tecnología GSM esta misma cuenta con 4 versiones principales de las cuales tenemos la GSM-850, GSM 900, GSM1800, GSM 1900, GSM es uno de los estándares digitales inalámbricos de la segunda generación (2G), con más importancia en el mundo, con presencia en más de 160 países, más de 7.9 mil millones de conexiones móviles en el mundo (GSMA, 2017).

En Perú se utiliza GSM 850-1900Mhz con el que podemos tener acceso a transmisión de voz, servicio de mensajes cortos (SMS), renvío de llamadas para otro número, transmisión y recepción de datos y fax, etc (OSCIPTTEL, 2015). En la figura 7 se muestra los principales componentes de la arquitectura GSM.

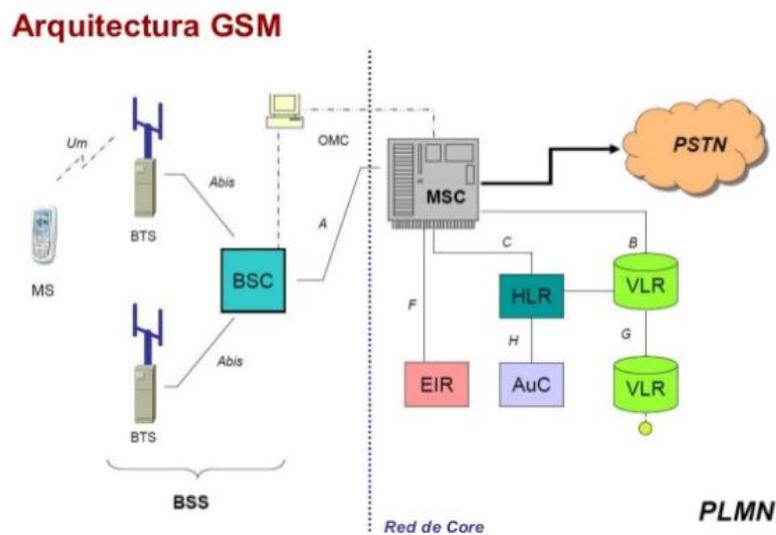


Figura 7: Arquitectura de la tecnología GSM.

Fuente: <https://es.slideshare.net/davidortegag/capitulo-07-comunicaciones-moviles>

#### 2.2.4. MODULO SIM 800L

SIM800L es un módulo celular en miniatura que permite la transmisión GSM/GPRS, el envío y la recepción de SMS y la realización y recepción de llamadas de voz, un módulo de tamaño reducido y de bajo costo, puede trabajar en cuatro bandas de frecuencia como la, GSM850Mhz, EGSM900Mhz, DSC1800Mhz y PCS 1900Mhz, utiliza comandos AT para su programación (SIMCom, 2013).

Características más resaltantes.

- Nivel Lógico de 3V a 5V
- Voltaje de Operación: 3.4V ~ 4.4V DC
- Consumo de corriente (sleep): 0.7 mA
- Consumo de corriente (max): 500 mA
- Hacer y recibir llamadas de voz usando un auricular o un altavoz de 8Ω externo + micrófono electret.
- Quad-band 850/900/1800/1900MHz
- Enviar y recibir mensajes SMS
- Controlado por Comandos AT (3GPP TS 27.007, 27.005 y SIMCOM enhanced AT Commands)
- Interfaz de comandos AT con detección “automática” de velocidad de transmisión
- Soporta USSD
- Soporta Reloj en tiempo real (RTC)
- Velocidades de transmisión serial desde 1200bps hasta 115200bps
- Tamaño de la SIM: Micro SIM

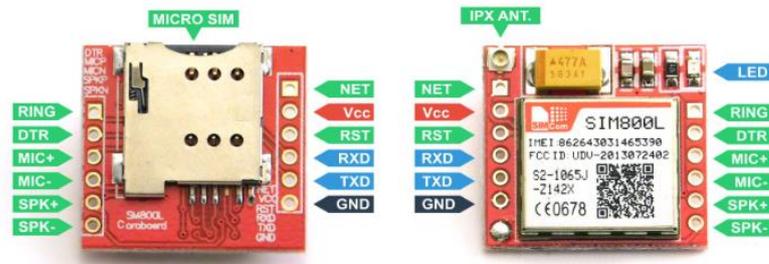


Figura 8: Módulo SIM 800L.  
Fuente: www. Etqan.sa

### 2.2.5. Comandos AT

AT deriva de la abreviatura de “*attention*”, son instrucciones codificadas para la comunicación entre el hombre y un terminal modem, fue desarrollado en 1977 por Denis Hayes, para poder configurar un modem y darle instrucciones como marcar un número telefónico (Bluehack, 2005).

Al principio los comandos AT fueron creados para la comunicación con módems, pero la telefonía móvil GSM tomo como lenguaje estándar para poder comunicarse con sus terminales, la notación de los comandos AT empiezan con las letras AT+ seguida de una serie de instrucciones que luego es procesado y posteriormente se obtiene un respuesta. En la tabla 1 se muestra los comandos más utilizados en GSM.

Tabla 1  
*Comandos AT más utilizados con el módulo SIM800L.*

COMANDO AT	DESCRIPCION
AT+IPR=?	Muestra la velocidad que acepta el modulo.
AT+IPR=9600	Define la velocidad en 9600 baudios.
AT+CGSN	Devuelve el IMEI (Identidad de Equipo de Estación Móvil Internacional), del terminal móvil.
AT+CPIN?	Comprobamos que la tarjeta SIM esté funcionando.
AT+CGREG?	Comprobar el estado de registro del dispositivo.
AT+COPS?	El dispositivo devuelve la lista de redes que se encuentran disponible para conectarse.
AT+CSQ	Medir el rendimiento de la fuerza de la señal.
AT+CMGF?	Seleccionar formato de mensaje SMS (0=PDU, 1=Texto).
AT+CMGL= "ALL"	Mirar todos los mensajes.
AT+CMGL= "REC UNREAD"	Mirar solo mensajes no leídos.
AT+CMGL= "REC READ"	Mirar solo mensajes leídos.
AT+CMGL= "STOP UNSENT"	Mirar solo mensajes no enviados.
AT+CMGL= "STO SENT"	Mirar solo mensajes enviados.
AT+CCALR?	Dispositivo listo para llamada (1 significa listo, 0 significa que no está listo).
AT+CCID	Obtiene el número de la tarjeta SIM.
AT+CLIP?	Parámetro para activar o desactivar la identificación de llamada entrante.

Nota. Comandos sacadas de la hoja de datos del fabricante SIM 800L. Fuente: SIMcom

## **2.2.6. ARDUINO**

Arduino es una plataforma de hardware y software libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios.

Arduino fue inventado en el año 2005 por el entonces estudiante del instituto IVRAE Massimo Banzi, quien, en un principio, pensaba en hacer Arduino por una necesidad de aprendizaje para los estudiantes de computación y electrónica del mismo instituto, ya que en ese entonces, adquirir una placa de micro controladores eran bastante caro y no ofrecían el soporte adecuado; no obstante, nunca se imaginó que esta herramienta se llegaría a convertir en años más adelante en el líder mundial de tecnologías DIY (Do It Yourself) (Arduino, 2019).

Tras su evolución de Arduino se han desarrollado múltiples placas para distintos proyectos entre ellos tenemos al Arduino Uno, que el Arduino más usado, Arduino nano cuyo Arduino se utilizó en este proyecto, Arduino mini, Arduino Leonardo, Arduino Due, Arduino Mega, Arduino Ethernet, etc.

### **2.2.6.1. Arduino Nano.**

Podríamos decir que es la versión pequeña de Arduino Uno, ya que posee un microcontrolador basado en ATmega 328P el mismo que Arduino Uno, tiene 14 pines que pueden ser configurados como entradas o salidas digitales, 6 entradas analógicas y un cristal de 16 MHz, conexión mini USB para conectarse a la computadora y un botón de reseteo.

Características resaltantes:

- Microcontrolador: ATmega328.
- Voltaje de operación: 5V.
- Voltaje de alimentación (Recomendado): 7-12V.
- Entradas Analógicas: 8
- Memoria Flash: 32KB.
- EEPROM: 1KB.

- SRAM: 2 KB
- Frecuencia de trabajo: 16MHz.
- Dimensiones: 18,5mm x 43,2mm
- I/O Digitales: 14 (6 son PWM) , las cuales se describen en la figura7

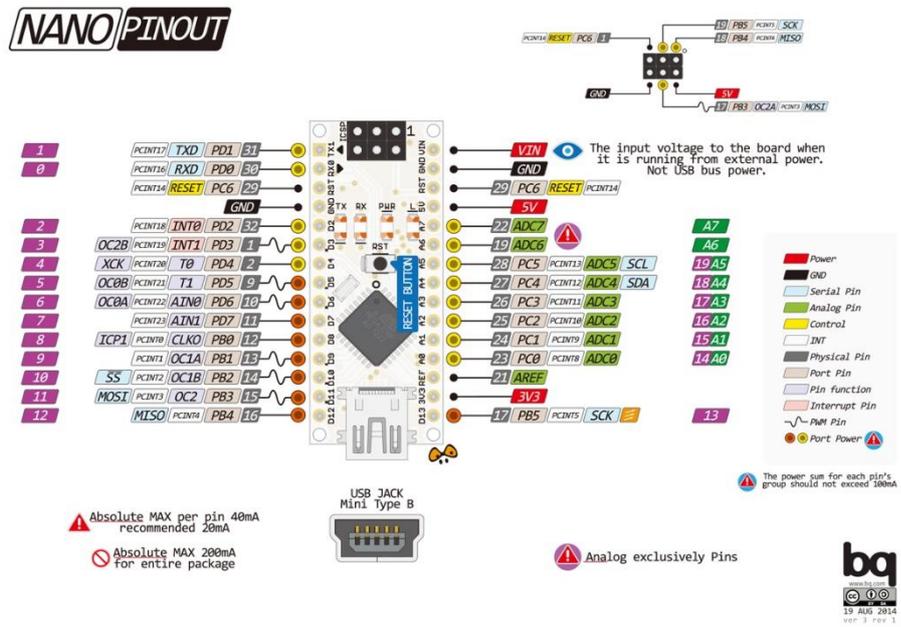


Figura 9: Esquema de pines de Arduino nano.  
Fuente: <https://github.com/arduino/Arduino/issues/7739>.

### 2.2.6.2. IDE Arduino

Entorno de Programación Integrado- Arduino IDE (Integrated Developer Environment), como hace referencia el nombre es un entorno de programación utilizado para la programación de las placas Arduino; consta de un editor de código, un depurador, un compilador, y un constructor de interfaz gráfica.

La IDE-Arduino proporciona todas las herramientas para poder pasar el programa compilado a la memoria flash de las placas, se puede descargar directamente de su página oficial (Arduino, 2019).

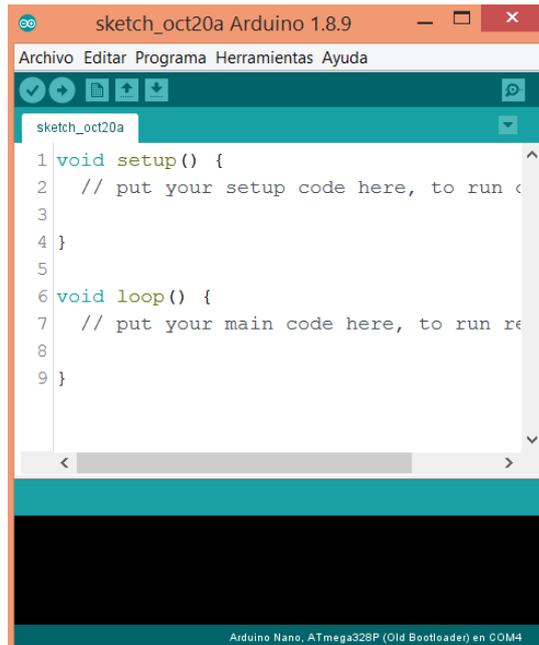


Figura 10: Entorno de programación de Arduino.  
Fuente: [www.Arduino.cc](http://www.Arduino.cc)

## 2.2.7. SESORES INFRAROJOS

Los sensores infrarrojos son dispositivos optoelectronicos cabases de medir distancias al percibir la radiación infrarroja (IR) de los objetos.

El modulo IR -FC-51, es un módulo infrarrojo activo, consta de un emisor y un receptor, donde el emisor es un led infrarrojo (emite luz no visible al ojo humano) y el receptor es un fototransistor sensible a la luz infrarroja, incorpora en su placa un regulador de distancia de la que se puede modificar para detectar por reflejo de la luz infrarroja emitida por el diodo en un rango de 2 cm a 30 cm, la cual hace factible a las necesidades de este proyecto.

Aspectos importantes a considerar:

- VCC - Entrada de alimentación de DC de 3.3V-5V
- GND -0V Pin de alimentación
- OUT - Pin de salida digital
- Ángulo de detección -35 grados
- Salidas Nivel lógico bajo cuando se detecta un obstáculo

- Rango de detección: 2cm - 30cm (Ajustable usando un potenciómetro) tal como se muestra en la figura 9.



Figura 11: Modulo FC-51.

Fuente: [http://www.dmf.unisalento.it/~denunzio/allow\\_listing/ARDUINO/FC51.pdf](http://www.dmf.unisalento.it/~denunzio/allow_listing/ARDUINO/FC51.pdf)

### 2.2.8. MODULO PIR HC-SR501

Debido a la temperatura que se encuentran los cuerpos ya sea de seres vivos o incluso de objetos, emiten radiación electromagnética infrarroja, la radiación aumenta a mayor temperatura, debido a estas características de los cuerpos se han creado sensores que detectan estas radiaciones en una longitud de onda alrededor de 9,4 micrones, el cual permite la detección de movimiento, dichos sensores son conocidos como sensores PIR (Pyroelectric Infrared) (Punto flotante S.A., 2017).

EL módulo PIR HC-SR501 contiene un sensor piroelectrico pasivo, capaz de detectar cambios de radiación infrarroja, dichas señales son procesadas por el circuito integrado BISS0001, el cual está constituido por amplificadores operacionales obteniendo una salida TTL la cual nos ayuda a precisar el cambio de radiación o común mente el movimiento de una persona, para luego poder programarlo a nuestra conveniencia en un microcontrolador (Adafruit, 2018).

Aspectos importantes a considerar del módulo.

- Voltaje de operación: 5VDC
- Consumo de corriente en reposo: <math><50\mu\text{A}</math>
- Rango de detección: 3 a 6 metros, ajustable mediante Trimmer (Sx)
- Angulo de detección: <math><110^\circ</math>
- Tiempo de retardo: 5-200 S (puede ser ajustado (Tx), por defecto 5S +-3%)
- Temperatura de trabajo: -20°C hasta 80°C

- Dimensión: 3.2 cm x 2.4 cm x 1.8 cm (aprox.), tal como se aprecia en la figura 10.

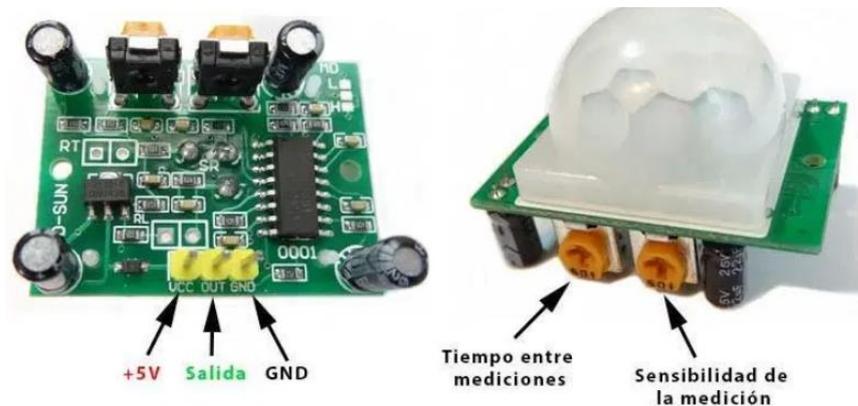


Figura 12 : Sensor PIR HC-SR501.  
Fuente: Adafruit learning system

### 2.2.9. MODULO KY-038

Es un módulo que incorpora un micrófono CZN153 para detección de sonido un amplificador operacional LM393 para obtener salida digitales al alcanzar al Umbral establecido, y una salida analógica propia del micrófono que servirá en este proyecto para recepción de voz al momento de realizar una llamada telefónica, en la figura 11 se muestra el módulo completo.



Figura 13: Módulo KY-038.  
Fuente: <https://www.alldatasheet.com/datasheet-pdf/pdf/1138845/ETC2/KY-038.html>

### 2.2.10. STEP UP XL6009

Es un módulo convertido de voltaje DC-DC soporta 2.5A, puede entregar un voltaje de salida constante de 7v a 32v, frente a variaciones de voltajes de entrada entre 5v a 32v, todo ello gracias al regulador XL6009 que se encuentra incorporado en la placa, se trata de un MOSFET de potencia de canal N, con frecuencia de conmutación fija de 400khz, la salida de voltaje se ajusta gracias a un potenciómetro multivuelta como se muestra en la figura12 (Kylinchip, 2015).



Figura14: Modulo STEP UP XL6009.  
Fuente: www.pixelelectric.com

### 2.2.11. EasyEDA

EasyEDA es una plataforma online para la simulación y creación de placas electrónicas, desarrollado con un conjunto de herramientas que permite a los ingenieros de hardware diseñar, simular, compartir de forma privada o pública sus trabajos electrónicos, otra de las características importantes es que incluye con una amplia lista de materiales, archivos Gerber y diseños de PCB (EasyEDA, 2019).

La plataforma es de fácil uso, La capacidad de importar esquemas y símbolos, proporciona una forma útil de portar esquemas al diseño de PCB sin tener que volver a dibujarlos desde cero, en la figura 13 se muestra la ventana principal de EasyEDA.

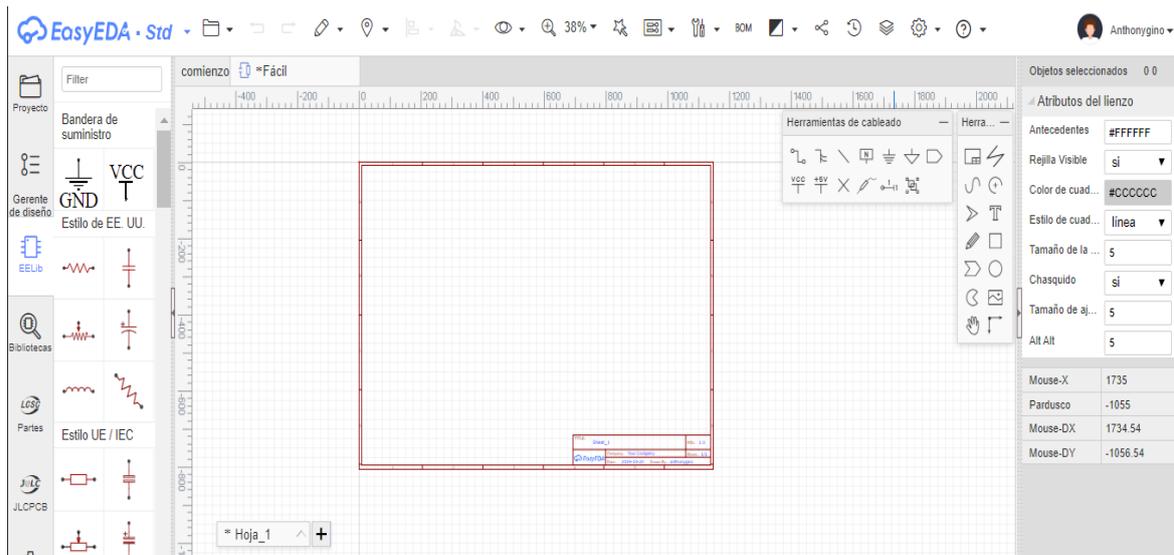


Figura 15: Ventana principal de EasyEDA.  
Fuente: www.easyeda.com

### 2.3. Definición de términos básicos

- Microcontrolador: es una pequeña computadora inyectada en un chip, están diseñadas principalmente para el control industrial y no para el procesamiento de grandes cantidades de datos, su principal ventaja es la alta inmunidad al ruido, el bajo costo y su tamaño. (Ecured, 2018).
- PIN: es una palabra inglesa cuyo significado es *clavija*, en electrónica se hace referencia a las patillas de los elementos electrónicos, cuyo funcionamiento en particular de cada uno de los pines es necesario consultar a la hoja de datos de cada dispositivo.
- Vcc: deriva de la palabra Colector supply voltaje, se utiliza para referenciar el pin de alimentación de un circuito. (Hernandez, 2015).
- Gnd: se utiliza clásicamente para referirse a un punto en específico del circuito cuya tensión de corriente es 0v. (Hernandez, 2015).
- Sensor pasivo: son aquellos sensores que necesitan una fuente auxiliar para poder generar señales representativas de las magnitudes a medir ejemplo: resistencias variable, inductancia variable, capacidad variable, etc. (Palomino, 2017).
- Algoritmo: Se refiere al conjunto de instrucciones que representan un modelo de soluciones para un problema en específico, en la que debe de ser preciso definido y finito. (Villalobos, 2008).
- Lente de Fresnel: El nombre deriva del físico francés Augusting Fresnel, quien construyó un lente convexo que ayuda a focalizar los rayos de luz que entran de forma paralela al eje óptico, pueden ser vidrio tallados o plásticos semiconductores (Meyrán, 2008).

- Comunicación serial: es un tipo de comunicación entre dispositivos cuya particularidad es el envío de datos de un bit a la vez, de forma secuencial, utiliza un canal de comunicación o también llamado bus (Perez, 2015).
- Baudios: se emplea como la unidad de medida de la velocidad de transmisión de señales expresadas en símbolos por segundos, en una transmisión binaria en la que solo existen 2 valores "0" y "1" un baudio equivale a un bit por segundo (Gonzales, 2017).
- Variables: en programación una variable hace referencia a un área reservada en la memoria principal del ordenador, que puede cambiar de contenido en cualquier momento de la ejecución del programa.  
Tiene por lo general un identificador (nombre) y asignado un tipo de dato que se está utilizando, puede ser numérico, alfanumérico, lógico, booleano, etc. (Villalobos, 2008, p.33)
- Buffer: en informática un buffer es un espacio de memoria, ayuda a almacenar datos de forma temporal, su uso principal es mantener la información en memoria cuando se optimen datos con muy alta velocidad de la que no podemos procesarlo al instante o cuando llegan datos de manera inesperada que no resulta beneficioso dedicar tiempo y recursos de sistema a esperarlos (Carlos, 2015).

## CAPÍTULO III: DESARROLLO DEL TRABAJO DE SUFICIENCIA PROFESIONAL

### 3.1. Modelo de solución propuesto

El modelo de solución propuesto se muestra en la figura 16, el manejo de los módulos electrónicos que se presentan fueron teniendo en cuenta su valor económico y fiabilidad, ya que el prototipo será para las zonas rurales, se trató de conseguir módulos económicamente accesibles pero que sean fiables en su uso, en el modelo de solución propuesta para el desarrollo del proyecto, se diferencia 4 partes, la alimentación del dispositivo, sensores, Arduino y Comunicación GSM.

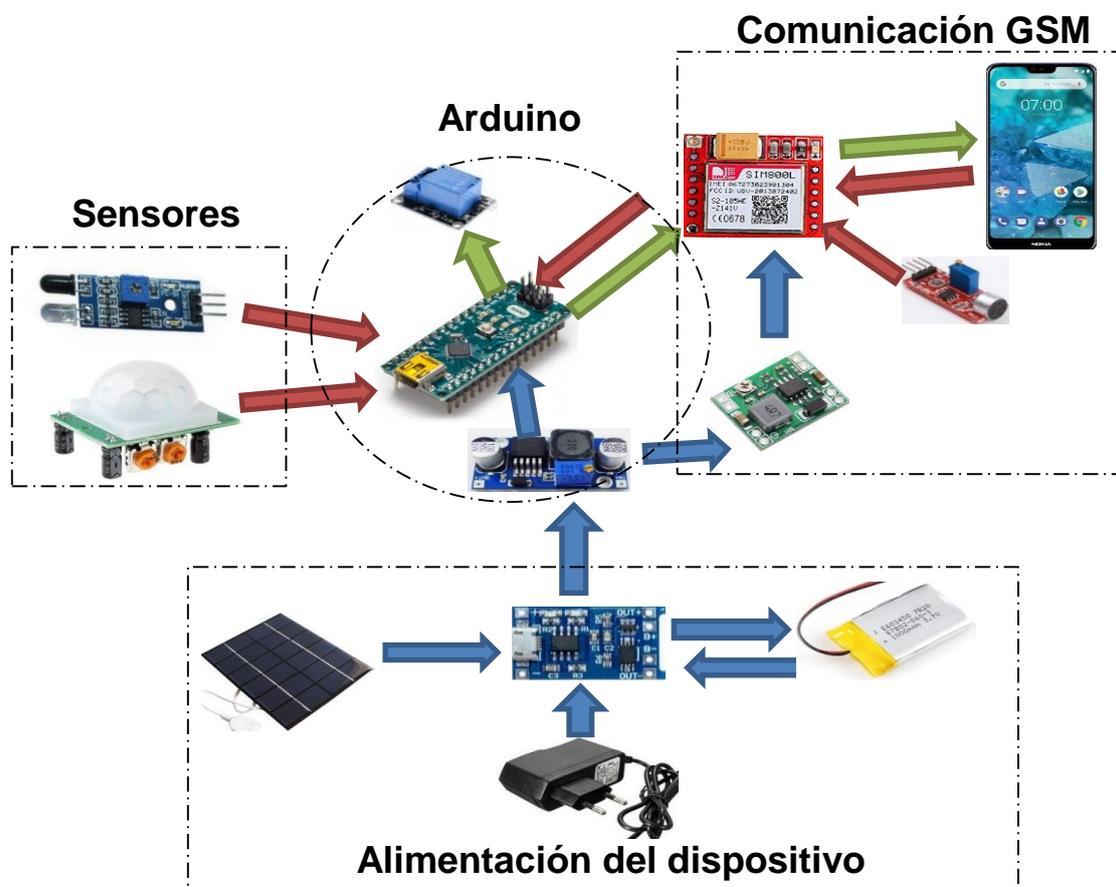


Figura 16: Diagrama de Bloques- modelo de solución propuesta.  
Fuente: Elaboración propia.

El prototipo a elaborar será alimentado ya sea por una fuente de 5 volts o un panel solar de 5v que irán directamente conectado al cargador de baterías TP4056 el cual cargara una batería Lipo de 3.7v 2.5ah, el módulo TP4056 protege la batería frente a un sobre descarga, contra cortocircuito y tiene la función de Pass-through que permite utilizar el proyecto conectado mientras se está cargado la batería.

Para alimentar al Arduino primero se eleva el voltaje que nos proporciona el módulo TP4056 hasta unos 9v con el módulo Step-up XL6009, y conectamos al pin “vin” que es el pin de alimentación de Arduino.

El módulo SIM800L funciona con una alimentación de 3.3v a 4.4v para ello reducimos el voltaje proporcionado por el módulo Step-up a unos 4.3v con el módulo Step-Dow MP1584 que proporciona hasta 3 A salida, la cual es importante porque el módulo SIM800L puede llegar a picos de 2 A cuando realiza llamadas o envió de SMS.

En la parte de los sensores tenemos a los módulos IR-FC-51 y al módulo PIR HC-CR501 los cuales serán encargados de proporcionar la información de lo que ocurre en la vivienda para luego ser procesado por Arduino y tomar decisiones.

### **3.2. Diseño de la PCB para el soporte de los módulos electrónicos**

El proyecto a presentar es el desarrollo de un prototipo utilizando módulos electrónico y tecnología GSM para monitoreo de viviendas, ya que se trata de un prototipo que podría ser el principio para el desarrollo de más innovaciones en cuando a seguridad de viviendas en zonas rurales, se utiliza módulos electrónicos para su funcionamiento debido al cual la placa quede quedar de dimensiones grandes.

El diseño de la placa se desarrolló con el programa EasyEDA. En la figura 17 se puede apreciar el diseño del circuito.

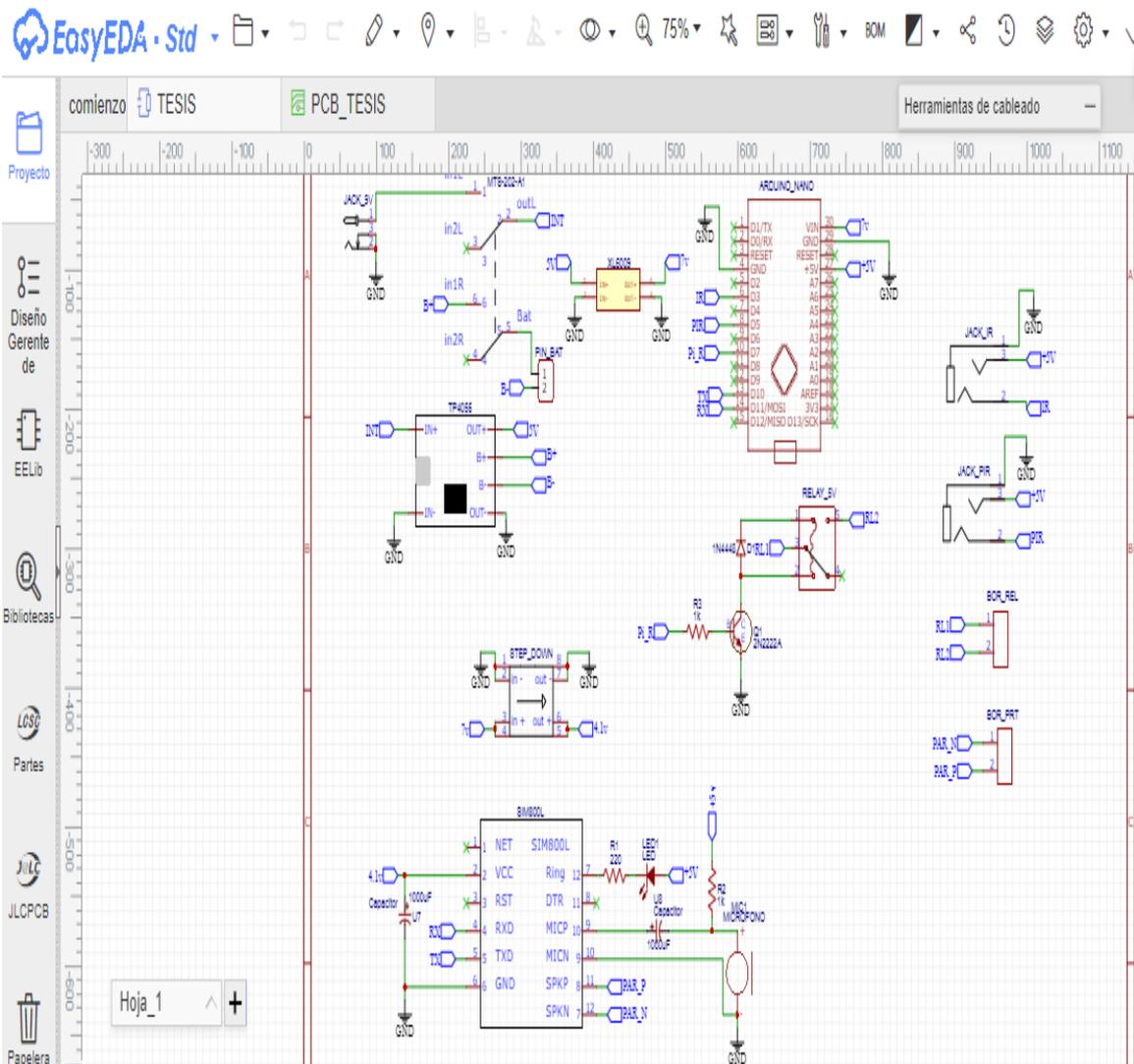


Figura 17: Diseño del circuito en EasyEDA.  
Fuente: Elaboración propia.

Primero tenemos la entrada de alimentación del proyecto, que será mediante un mini Jack para DC de 5x2.1mm la cual va directamente a un interruptor de palanca de 6 pines este se utiliza para prender o apagar el sistema al mismo tiempo el mismo interruptor conecta la batería con el módulo TP4056 de esa manera evitamos descargar la batería cuando no se esté utilizando el proyecto o se necesite trasladar el modulo, se puede apreciar de una mejor manera en la figura 18.

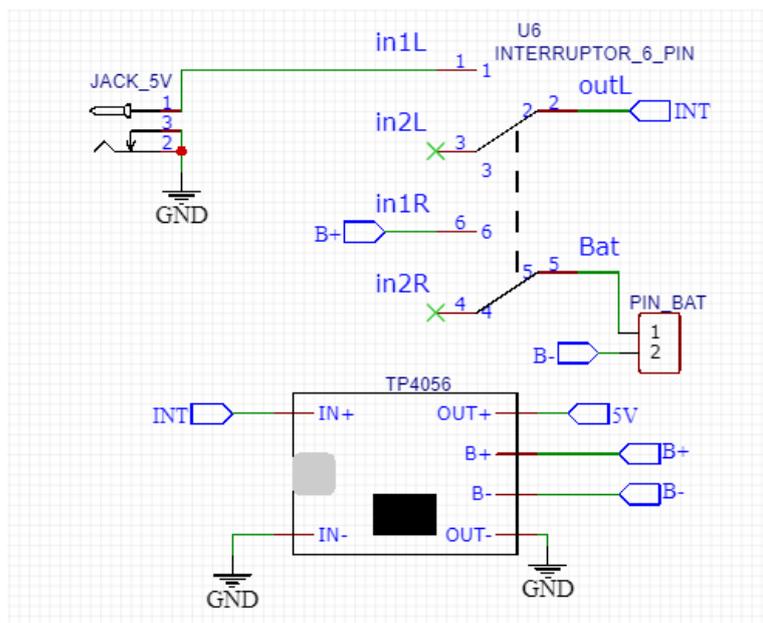


Figura 18: Entrada de alimentación del circuito.  
Fuente: Elaboración propia.

Del módulo TP4056 solo proporciona una máximo de 4.4 volts para poder alimentar correctamente al Arduino necesitamos entre 7v a 12v por el pin "Vin" para ello utilizamos el módulo Step-up XL6009 el cual eleva el voltaje y el módulo StepDow MP1584 baja el voltaje hasta lo requeridopara la alimentación del módulo SIM800L, tanto el módulo StepUp como el módulo StepDown mantiene el voltaje especificado a pesar de que la fuente de alimentación pueda bajar o subir su voltaje, la figura 19 se representa el esquema.

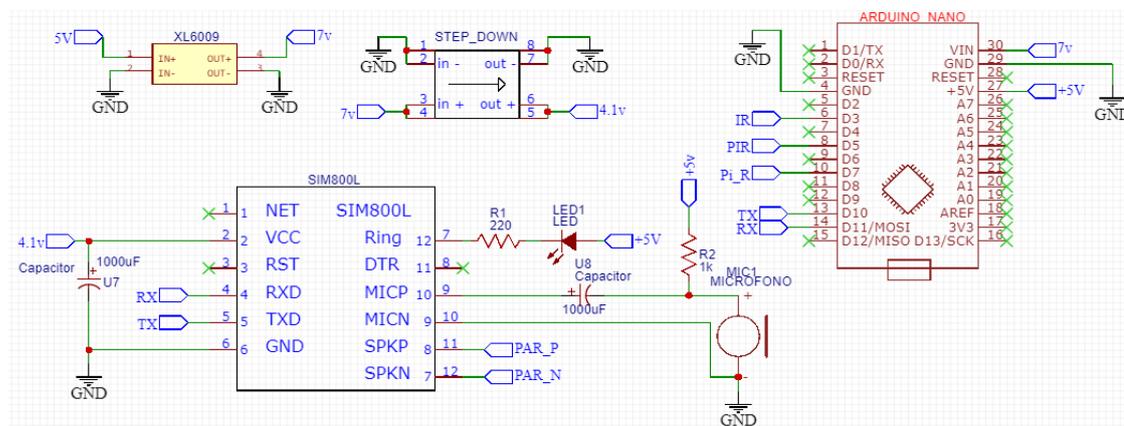


Figura 19: Esquema del circuito para alimentación del ARDUINO y el módulo SIM800L.  
Fuente: Elaboración propia.

Luego tenemos los módulos de entrada de señal que proporciona los sensores, se utiliza dos mini Jacks hembras de audio de 3.5mm, estos Jack tienen 5 pines de los cuales utilizaremos 3 pines, uno para GND, el segundo para entrada de señal de los sensores y el tercero para Vcc que sería 5v cuya tensión es el voltaje de funcionamiento de los mismos, el esquema se muestra en la figura 20.

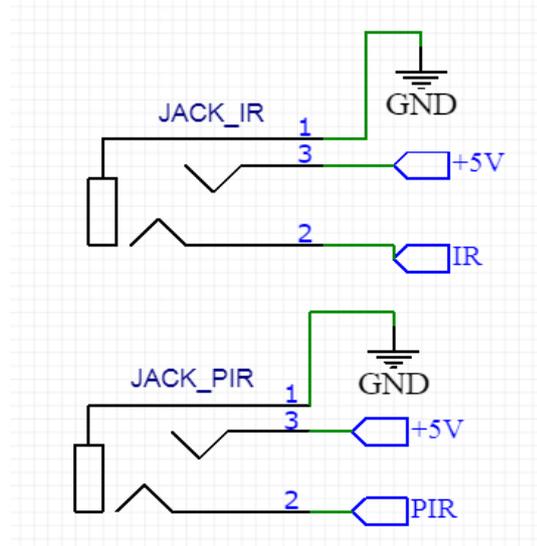


Figura 20: Jack para la entrada de señal de los sensores.  
Fuente: Elaboración propia.

Finalmente tenemos las borneras de dos pines para conectar un pequeño parlante de un máximo de 8ohms que se vincula directamente al módulo SIM800L el cual se podría utilizar para escuchar llamadas, y la otra bornera está vinculada a un relay, que podría prender una bocina o cualquier otro artefacto electrónico que se desee. El esquemático se aprecia en la figura 21.

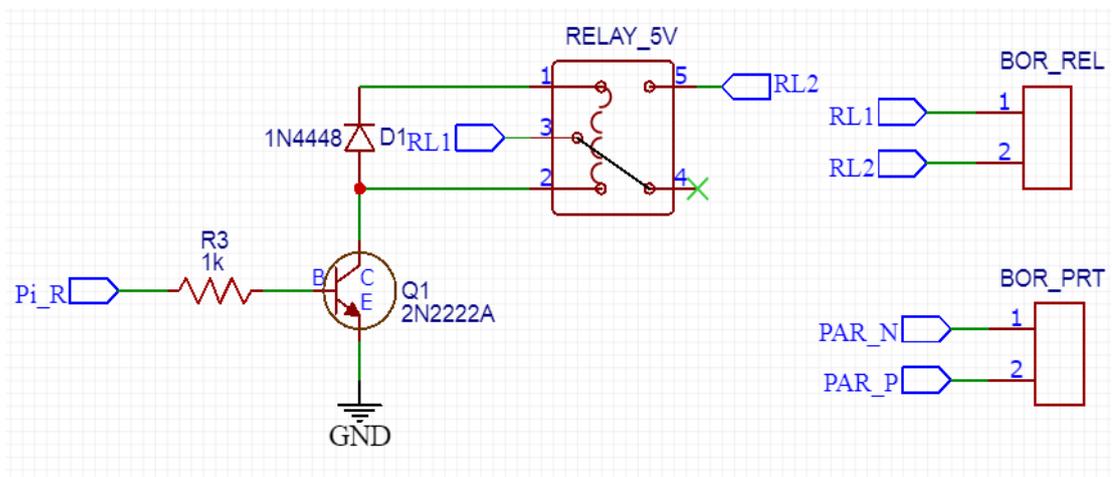


Figura 21: Borneras auxiliares del circuito.  
Fuente: Elaboración propia.

Una vez diseñado el circuito pasamos a modelar la PCB en el mismo programa EasyEDA, la PCB se diseñó de doble capa para evita montajes de pistas y posibles cortocircuito, en la figura 22 se muestra las dos caras de la PCB.

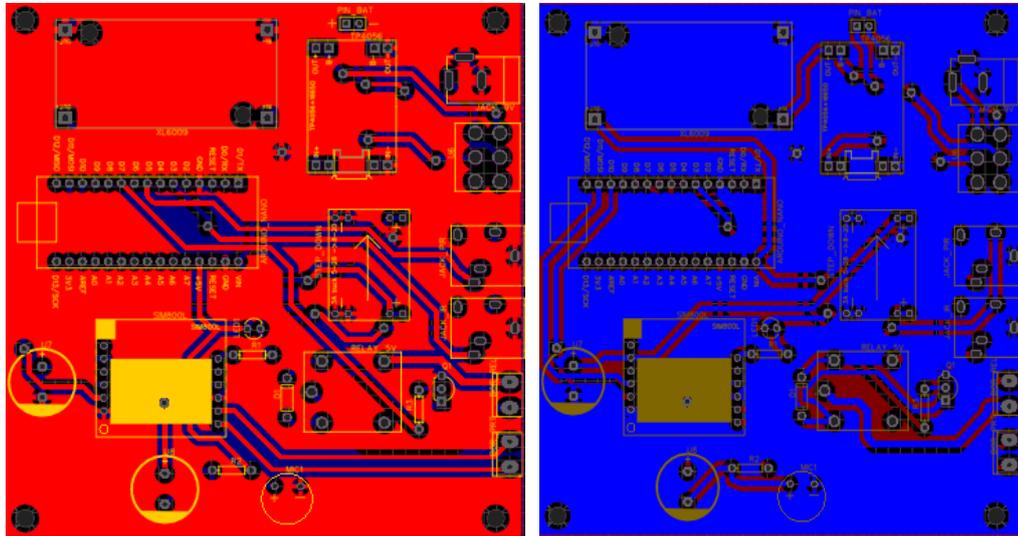


Figura 22: PCB cara superior y cara inferior.  
Fuente: Elaboración propia.

### 3.3. Lógica de programación del prototipo

Para el desarrollo de la lógica de programación tenemos 3 esquemas importantes:

#### ✓ **Adquisición de datos de los sensores**

En esta sección se presentarán los sensores y el funcionamiento de los mismos, los cuales son indispensables para el funcionamiento del prototipo.

#### ✓ **Emisión y recepción de sms y llamadas**

Para la recepción de SMS y llamadas, así como la transmisión de las mismas, se utilizará el módulo GSM SIM 800L, el cual utiliza una tarjeta SIM para realizar las mismas operaciones como si fuera un celular común, gracias a los comandos AT se podrá programar el módulo para la adquisición de los datos a utilizar.

#### ✓ **Procesamiento de los datos**

Para procesar todos los datos ingresados por los sensores y el módulo SIM800L, nos ayudaremos de la placa ARDUINO NANO, que es controlada por un chip

ATMEL 328p de suficientes capacidades para el desarrollo del prototipo, básicamente será como la memoria de nuestro prototipo.

### 3.3.1. Adquisición de datos de los sensores

El robo de las viviendas en la mayoría de los casos se logra por el forcejeo de las chapas de las puertas, en vista de ello para el caso de este proyecto manejaremos el modulo IR -FC-51, que consta principalmente de un led de emisión infrarroja y un fototransistor sensible a la luz infrarroja, cuyo funcionamiento será de ayuda para saber si la puerta se encuentra cerrada o si se encuentra abierta.

El modulo se IR -FC-51 consta de un OPAM LM393 que está en modo comparador el cual nos proporciona una salida digital, en la figura 23 se muestra el esquemático del módulo.

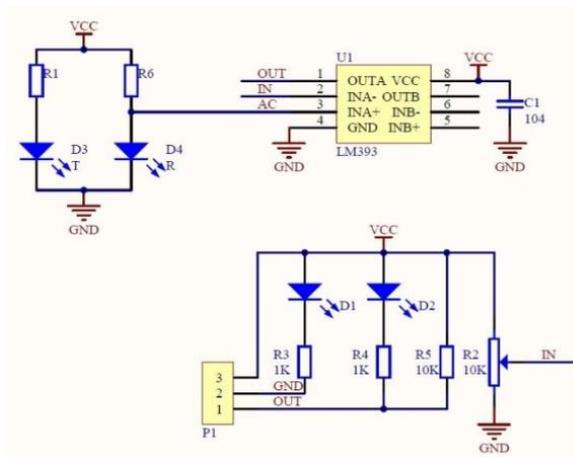


Figura 23: Esquemático del módulo IR FC -51.  
Fuente: [www.ti.com.cn](http://www.ti.com.cn)

Para determinar la salida del OPAM en modo comparador utilizamos la formula siguiente.

$$V_{out} = (VIN+) - (VIN-) \quad (1)$$

Donde:

Vout= voltaje de salida de OPAM

VIN+=voltaje en la entraba no inversora del OPAM

VIN-=voltaje en la entrada inversora del OPAM

Para establecer la distancia de detección del sensor se ajusta por un potenciómetro de 10kΩ, en la figura 24 se aprecia que al detectar el sensor un obstáculo se enciende un led indicador.

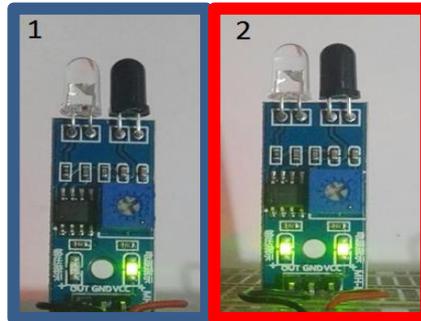


Figura 24: (1) Sensor no detecta obstáculo. (2) sensor detecta obstáculo.  
Fuente: Elaboración propia

Este sensor se mantendrá en la parte del marco de la puerta de esa manera podremos obtener las lecturas digitales, según el diseño de la PCB cuando el sensor detecte a la puerta cerrada obtendremos a la salida un “0” lógico y cuando la puerta este abierta obtendremos a la salida un “1” lógico. En la figura 25 se muestra un esquema de como iría posicionado el sensor infrarrojo.

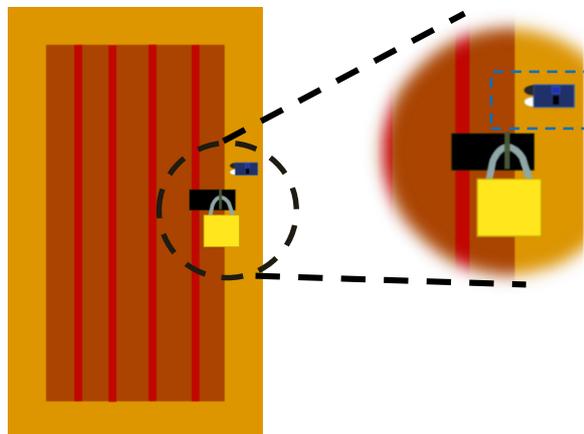


Figura 25: Posicionamiento del sensor en el marco de la puerta.  
Fuente: Elaboración propia

Otro de los sensores a utilizar en este proyecto es el sensor de movimiento PIR que viene en un módulo llamado Modulo PIR HC-SR501 detecta el calor producida por los cuerpos en forma de radiación infrarroja, que ayuda a obtener datos dentro de la viviendas si se detecta algún movimiento, el módulo PIR a utilizar cuenta con un lente llamado lente de fresnal que nos es más que una carcasa ovoide que se coloca encima del sensor para lograr centrar o focalizar las radiaciones en el

sensor (Adafruit, 2018). En la figura 26 se muestra el sensor PIR y el lente de Fresnel.

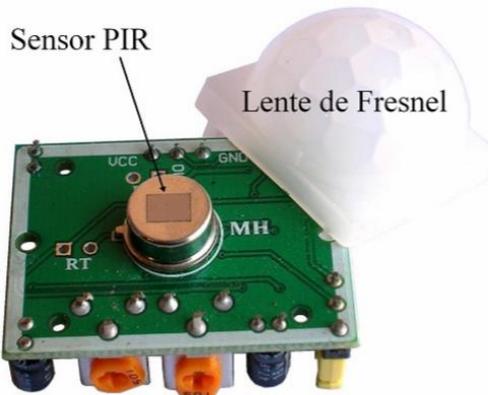


Figura 26: Modulo PiR y lente de Fresnel.  
Fuente: puntoflotante.net

Gracias al lente de Fresnel el módulo PIR puede detectar cambio de radiación en un rango máximo de  $110^\circ$  lo cual es ajustable por un potenciómetro incluido en su diseño PCB, así como la distancia ajustable hasta unos 6m de detección, tal como se muestra en la figura 27.

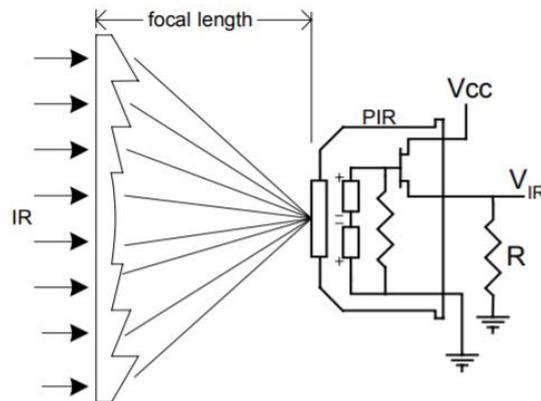


Figura 27: Detector PIR con lente de fresnel.  
Fuente: CYPRESS MICROSYSTEM

Para del proyecto dicho Modulo PIR ira en la parte opuesta a la puerta de entrada de la vivienda, para obtener lecturas precisas, cuando detecte movimiento mandara por el pin de salida un 5v lo cual tomaremos como un "1" lógico y cuando no detecte movimiento el sensor mandara un 0v lo cual tomaremos como "0" lógico, posteriormente estos datos serán programados en el microcontrolador Arduino, en la figura 28 se muestra un ilustrativo de la posición del sensor.

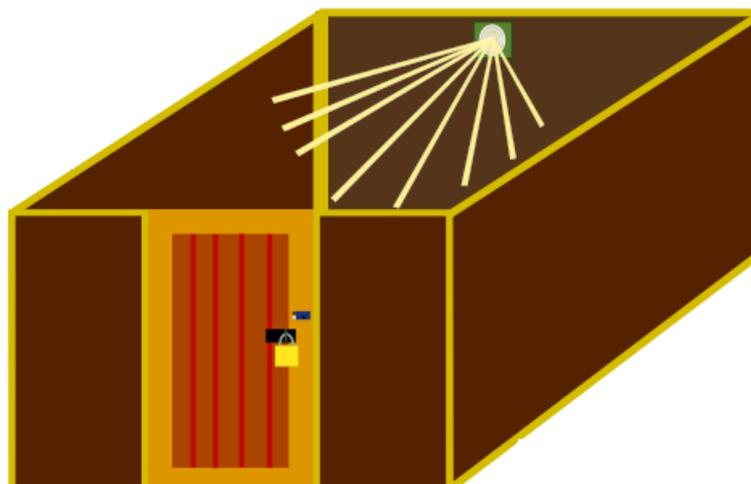


Figura 28: Posición del sensor PIR dentro de la vivienda.  
Fuente: Elaboración propia

### 3.3.2. Emisión - recepción de SMS y llamadas telefónicas

El prototipo para monitoreo de viviendas esta constituido por módulos electrónicos adaptados a la necesidad del usuario, para la interacción del módulo y el dueño de la vivienda se considero utilizar la tecnología GSM, la mayor parte de la población del distrito de Molinopampa, cuenta con un teléfono celular siendo actualmente el medio tecnológico más utilizado para la comunicación a distancia, en vista de ello, por el bajo coste y una amplia cobertura se decidió utilizar tecnología GSM en este proyecto, desplazando a comunicaciones por radiofrecuencia ya que estos módulos no garantizan una amplia cobertura de funcionamiento.

El SIM 800L es un módulo de conectividad inalámbrica que utiliza tecnología GSM para su funcionamiento, utilizaremos la funcionalidad de mensajes de texto SMS y llamada telefónica para conectarse a nuestro prototipo, gracias a su amplia gama de comandos AT podremos programar el modulo y adaptarlos a nuestras necesidades.

### 3.3.3. Procesamiento de los datos

Debido a su amplia funcionalidad fácil programación y tamaño reducido, se utilizara la placa Arduino nano como cerebro de nuestro prototipo, en el cual se programara una serie de instrucciones que faciliten el funcionamiento de los

periféricos conectados a la placa, en la figura 29 se muestra el diagrama de flujo global del funcionamiento del prototipo que luego se ira desglosando uno por uno.

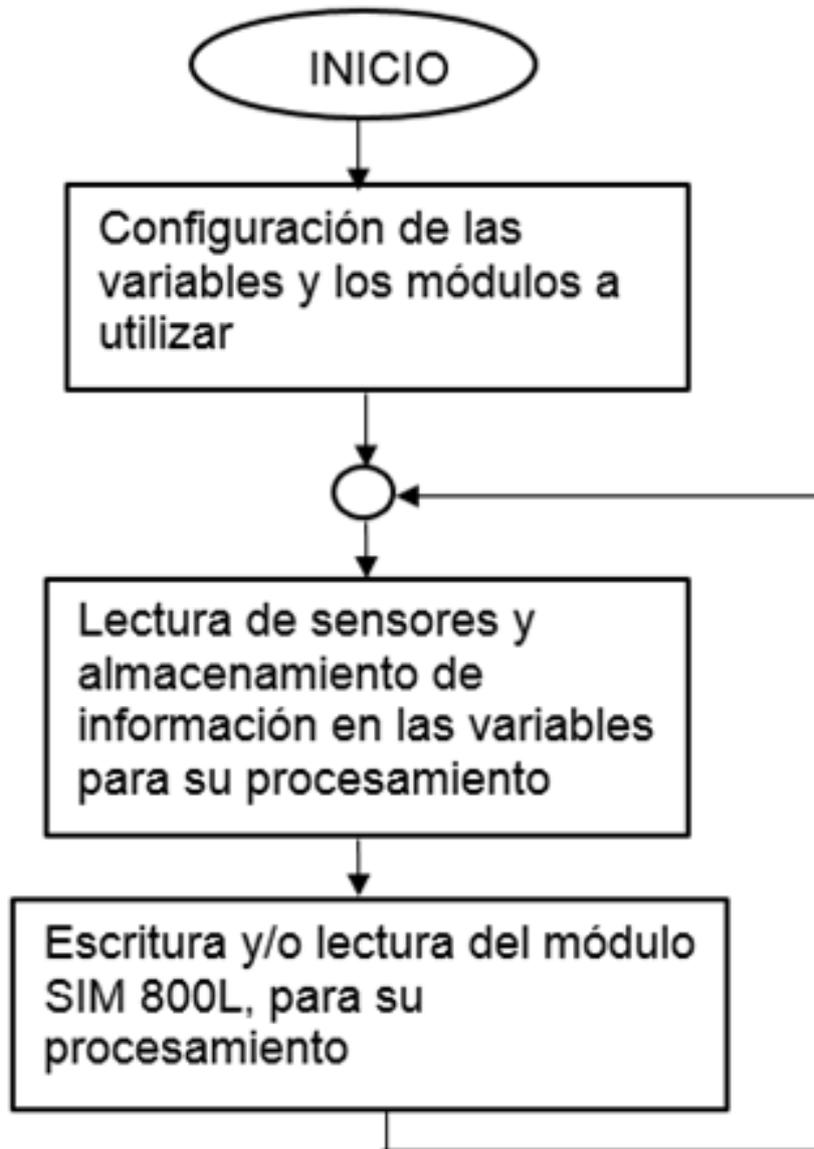


Figura 29: Diagrama de flujo global del prototipo de monitoreo de viviendas.  
Fuente: Elaboración propia.

El programa se desarrollara en la IDE de Arduino el cual utiliza un lenguaje de programación C++, iniciaremos programando nuestras variables globales y librerías a utilizar, tal como se muestra en la figura 30 y 31.

```
#include <SoftwareSerial.h>
#include <EEPROM.h>
#define espacioEEPROM 1000
SoftwareSerial serialGSM(10, 11);
bool llegoSMS = false;
bool llamada = false;
String tlfSMS;
String tlfLlamada;
String telefonoCliente = "932333942";
String telefonoSMSCliente = "932333942";
String telefonoSMSCliente51 = "+51932333942";
String telf1Usuario;
String HoraFechaSMS;
String mensajeSMS;
String comandoGSM = "";
```

Figura 30: Declaración de las variables globales y librerías necesarias.  
Fuente: Elaboración propia.

La placa Arduino nano, cuenta sola con una entrada y salida serial (tx, rx) para poder cargar el código desde la computadora a la placa, lo cual impide manejar este mismo puerto para la lectura del módulo GSM SIM 800L, pero se puede utilizar una librería para configurar otros pines de Arduino que funcionen como pines seriales, para ello se incluye la librería SoftwareSerial.h para luego vincularles los pines 10 como “rx” y 11 como “tx” este nuevo software seria será llamado serialGSM.

El prototipo es capaz de guardar los números de celular que el usuario le proporciona, por ello se incluye la librería EEPROM.h de la cual se utilizara 1kbits de capacidad, líneas siguientes tenemos 2 variables booleanas las cuales empiezan en falso o cero lógico, las variables se cambian de valor solo si se produce algún evento para la cual están creadas (“llegoSMS” cambia de valor solo si llego un SMS y “llamada” cambia de valor solo si se detecta una llamada).

Las variables siguientes son de tipo String o secuencia ordenada de caracteres, en la cuales se carga con la información obtenida del SIM 800L. “tlfSMS” carga el

numero telefónicos del celular que realizo el mensaje de texto, “tlfLlamada” carga el número telefónico del celular que realizó la llamada, “telefonoCliente, telefonoSMSCcliente, telefonoSMSClinet51”, están cargados con el teléfono celular del dueño de la vivienda, utilizado para realizar la llamada y mensajes de texto respectivamente, en “teléf1Usuario” se cargara un teléfono, que el usuario mande por medio de un SMS. En los siguiente String se carga la hora, fecha, y el mensaje de entrada al prototipo.

Luego creamos las variables que están vinculadas a los pines que manejan la información de entrada de los sensores, esta información recibida de los sensores son de tipo booleano por lo tanto serán cargadas en las variables booleanas las cuales empiezan en bajo (cero lógico) tal como se muestra en la figura 31.

En este trabajo se pretende diseñar un prototipo electrónico y como primero paso tendríamos que verificar si nuestros sensores están funcionando correctamente y poderlos visualizar en la pantalla del computador, para ello activamos la

```
int IR=3;
bool senIR=LOW;
int PIR=5;
bool senPIR=LOW;

bool flanRF=HIGH;
bool flanPIR=HIGH;
```

Figura 31: Variables direccionadas al pin de entrada de los sensores.  
Fuente: Elaboración propia

visualización del monitor erial con los comandos Serial.begin(9600) y serialGSM.begin(9600), donde “9600” es la velocidad en baudios que el Arduino y el computador cambiarían información, si en el caso que el sueño de la casas aya proporcionado otro número telefónico opcional para que el sistema mande mensaje en caso de producirse un evento anómalo en el sistema, lo primero que hace el código es cargar ese número en “telf1Usuario” buscando en la memoria EEPROM desde la posición cero, tal como se muestra en la figura 32.

Dentro de la función setup, se declara los pines correspondiente a los sensores como entradas con los comandos pinMode(IR,INPUT), pinMode(PIR,INPUT).

```
void setup()
{
  Serial.begin(9600);
  serialGSM.begin(9600);
  telf1Usuario = EEPROM_LECTURA(0);

  pinMode(IR, INPUT);
  pinMode(PIR, INPUT);
  cnfGSM();
}
```

Figura 32: Función setup, activación del monitor serial y declaración de los pines de los sensores como entradas.

Fuente: Elaboración propia.

En la figura 32 se muestra “cnfGSM();” el algoritmo llama a esta función en la cual se encuentran los comandos AT indispensables para poder configurar el módulo GSM SIM 800L, “AT+CMGF=1”, configura al módulo SIM800L para que el formato de los mensajes sea en modo texto, “AT+CNMI=2,2,0,0” configura el modo de entrada de los SMS al Arduino, “ATX4”, proporciona información de todos los códigos velocidad de información y estado de llamadas, “AT+COLP=1”, establece el modo de respuesta inmediatamente después de realizar la marcación por ejemplo puede mostrar (BUSY, NO DIAL TONE, NO CARRIER), en la figura 33 se muestra el comando utilizado para enviar estos comandos AT al SIM800L.

```
void cnfGSM()
{
  serialGSM.print("AT+CMGF=1\n;AT+CNMI=2,2,0,0\n;ATX4\n;AT+COLP=1\n;AT+CLIP=1\n");
}
```

Figura 33: Comandos AT para configurar el SIM800L.

Fuente: Elaboración propia.

Siguiendo en la programación llegamos a la función loop(), es un bucle cerrado, donde se programa la lectura de los sensores, interpretación de las lecturas, mensajes de entrada y codificación de mensajes, en la figura 34 empieza llamando

a la función “leerGSM();” en la que se pregunta si hay algo en la entrada serial del SIM800L para poder guardar en alguna de las variables.

```
void loop()
{
    leerGSM();

    if (comandoGSM != "")
    {
        Serial.println(comandoGSM);

        comandoGSM = "";
    }
}
```

*Figura 34:* Función loop().  
Fuente: Elaboración propia.

La información de entrada se almacena en un buffer interno del microcontrolador para asegurarnos de no tener pérdidas de lectura, toda información de entrada del SIM800L esta codificada en formato ASCII, esta información cargada en el buffer se vuelve a cargar en una variable tipo String llamada “textoEnt” el cual hace referencia al texto de entrada pero ya en formato texto, luego se borra la información del buffer. En la figura 35 se aprecia los comandos utilizados.

```
void leerGSM()
{
    static String textoEnt = ""; // Cadena llamada TextoEnt para almacenar el texto de entrada del SIM.
    static unsigned long tmp1 = 0; // tiempo de control de escritura
    static int count=0; // variable interna contador
    static unsigned char buffer[64];
    if (serialGSM.available())
    {
        while(serialGSM.available())
        {
            buffer[count++] = serialGSM.read();
            if(count == 64)break;
        }
        textoEnt += (char*)buffer;
        tmp1 = millis();
        for (int i=0; i<count; i++)
        {
            buffer[i]=NULL; // Resetea el fuffer.
        }
        count = 0;
    }
}
```

*Figura 35:* Guardar información de entrada del SIM800L.  
Fuente: Elaboración propia.

Una vez registrado el mensaje de entrada se procede a su interpretación, si el mensaje empieza con +CLIP: significa que la entrada es una llamada telefónica entonces se registra el número telefónicos en “tlfLlamada” según el mensaje el número telefónico debe de estar entre la posición 18 y 27 de la información en “textoEnt”, si la información de estrada empieza con +CMT: significa que esta entrado un SMS por lo tanto se procede a activar la variable “llegoSMS”, en la figura 36 se muestra los códigos a utilizar de acuerdo al mensaje de entrada que proporciona el SIM800L.

```
if ( (millis() - tmp1) > 100) && textoEnt != "" ) // Condicional empezar a descifrar lo que se guardó en textoEnt.
{
  if ( textoEnt.substring(10,16) == "+CLIP:" )
  {
    tlfLlamada=textoEnt.substring(18,27);
  }
  if ( textoEnt.substring(2,7) == "+CMT:" )
  {
    llegoSMS = true;
  }

  if (llegoSMS) // Condicional en la que es verdadera solo si se recepciona un SMS
  {
```

Figura 36: Condicional de acuerdo al mensaje de entrada del SIM800L.  
Fuente: Elaboración propia

En caso de que el SIM800L mande información y sea interpretado como SMS, se realiza una serie de arreglos para poder cargar en la variables correspondientes información necesaria que se encuentra dentro del mensaje SMS, con “textoEnt.charAt(nli)” se logra leer carácter por carácter dentro de la información que tiene “textoEnt” si es que encuentra la primera comilla “ ” ” se aumenta en uno una variable llamada “citas” para luego poder utilizarla como condición que nos ayudara tener la certeza de haber encontrado dentro del mensaje el número telefónico de la persona que mando el SMS, luego se carga carácter por carácter en la variable tipo String llamada “tlfSMS y al encontrar la 5ta comilla se carga en “HoraFechaSMS” la hora y la fecha cuando se mandó el SMS, al producirse el segundo salto de línea en el mensaje es ahí donde encontramos el mensaje en si del SMS, este mensaje le cargamos en la variable String llamada “mensajeSMS” la figura 38 muestra el código utilizado para realizar todas estas lecturas y almacenarlas en las variables correspondientes.

Una vez almacenada la información en las variables desarrollamos un código para mostrarlo en pantalla de esa manera verificamos si todo está correcto, en la figura 37 se muestra en código.

```

if (tlfSMS!=" " && HoraFechaSMS!=" " && mensajeSMS!=" ")// Condicional que es verdadera solo si las tres variables están cargadas.
{
    Serial.print("tienes un nuevo mensaje de:"); // se imprime en pantalla.
    Serial.println(tlfSMS); // el numero de celular.
    Serial.print("a las: ");
    Serial.println(HoraFechaSMS ); // la hora y la fecha.
    Serial.print("SMS: ");
    Serial.println(mensajeSMS); // y el mensaje de texto.
}

```

*Figura 37: Código para mostrar en pantalla lo almacenado en las variables.  
Fuente: Elaboración propia.*

```

if (llegoSMS) // Condicional en la que es verdadera solo si se recepciona un SMS
{
    byte linea = 0; // Variable que verifica los saltos de línea.
    byte citas = 0; //
    int nLi; // Variable entera para saber el número de posición de la letra dentro del String.
    for (int nLi=1; nLi < textoEnt.length(); nLi++) // contar hasta el número máximo presente en la cadena textoEnt
    {
        if (textoEnt.charAt(nLi) == '"') // busca comilla (") dentro del texto almacenado en textoEnt
        {
            citas++; //Aumenta en una unidad la variable entera cita.
            continue;
        }
        if ( (linea == 1) && (citas == 1) )
        {tlfSMS += textoEnt.charAt(nLi); // Guarda en tlfSMS el caracter guardado en textoEnt en el indice nLi.
        }
        if ( (linea == 1) && (citas == 5) ) //condicional que es verdadera si línea es 1 y lleguemos a la 5ta comilla del texto.
        {HoraFechaSMS += textoEnt.charAt(nLi); // Se carga en HoraFechaSMS la hora y la fecha del mensaje.
        }
        if ( linea == 2 ) //Condicional en la que es verdadera si se produjo el segundo salto de línea.
        {mensajeSMS += textoEnt.charAt(nLi); //Se almacena en mensajeSMS, el texto de entrada del SMS.
        }
        if (textoEnt.substring(nLi - 1, nLi + 1) == "\r\n") ////Condicional en la que es verdadera si se produce un salto de línea.
        {linea++; // Aumenta en uno la variable linea.
        }
    }
}

```

*Figura 38: Código para cargar información de número telefónico, hora, fecha y mensaje de la información obtenida del SIM800L.  
Fuente: Elaboración propia.*

El código que se muestra en la figura 39, sirve solo para guardar en “comandoGSM” las respuestas a los comandos AT que se enviaron para configurar el módulo SIM800L.

```

else // muestra solo las respuestas de comando AT enviados en la funcion cnfGSM(); .
{
  comandoGSM = textoEnt; // se guarda en comandoGSM lo que se cargó en textoEnt, respuestas de comando AT.
}
textoEnt = ""; // blanquear la cadena textoEnt,

```

*Figura 39: Se guarda las respuestas de los comandos AT enviados al módulo SIM800L.  
Fuente: Elaboración propia.*

Posteriormente visualizo las respuestas cargadas en la variable llamada “comandoGSM” solo si esta se encuentra cargada con información, obtendremos una respuesta “OK” si es que el SIM800L ha logrado configurarse adecuadamente, en la figura 40 podemos ver le código a utilizar.

```

if (comandoGSM != "")
{
  Serial.println(comandoGSM);
  comandoGSM = "";
}

```

*Figura 40: Se muestra en pantalla la las respuestas a los comandos AT enviadas a la SIM800L.  
Fuente: Elaboración propia.*

La programación anterior ayudo a interpretar el mensaje que fueron recibidas por el Sim800L y enviadas al Arduino nano, en la figura 41 se muestra 2 funciones que nos ayudaran a establecer seguridad a nuestro sistema (LlamadEntran() y SMSEntrante()).

```

void loop()
{
  leerGSM();
  if (comandoGSM != "")
  {
    Serial.println(comandoGSM);
    comandoGSM = "";
  }
  mensajeSMS.trim(); // Elimina espacio en blanco de la cadena de mensaje.
  llegoSMS = false;

  SENSOR();

  LlamadEntran();
  SMSEntrante();
}

```

*Figura 41: Funciones de verificación de usuario, llamada éntrate y SMS entrante.  
Fuente: Elaboración propia.*

La función “LlamadEntran();” es utilizada para comparar el número de celular de la persona que llama al sistema, con el número de celular guardado en el código

que es el número de celular del dueño de la vivienda (el numero guardado en el código se muestra en la figura 30 ), en el caso de que el número de celular de entrada este verificado dentro de nuestro código se procederá a responder con el comando “ATA” el cual es un comando AT que se envía al módulo GSM SIM 800L para responder llamadas telefónicas, en el caso de que el número de celular entrante no este preestablecido en nuestro algoritmo se procederá a colgar la llamada con el comando “ATH” que un comando AT para colgar llamadas, en la figura 42 se muestra el código a desarrollar.

```

void LlamadEntran()
{
  if (tlfLlamada == telefonoCliente || tlfLlamada==telf1Usuario.substring(3,12))
  {
    serialGSM.print ("ATA\n");
    Serial.println("LLAMADA ENLAZADA ");
    tlfLlamada= "";
  }
  if (tlfLlamada!=" " && tlfLlamada !=telefonoCliente || tlfLlamada!=" " && tlfLlamada!=telf1Usuario.substring(3,12))
  {
    serialGSM.print ("ATH\n");
    Serial.println("NUMERO DESCONOCIDO ");
    tlfLlamada= "";
  }
}

```

*Figura 42:* Código para comparar número de celulares que llaman con los números pre establecidos.

Fuente: Elaboración propia.

Con la función “SMSEntrante()” mostrada en la figura 32 primero se compara el número de celular del cual llego el SMS con el número pre establecido en el

```

void SMSEntrante()
{
  if (tlfSMS!=" " && tlfSMS==telefonosMSCliente51 || tlfSMS!=" " && tlfSMS==telf1Usuario)
  {
    if(mensajeSMS.substring(0,5)=="*#*1@")
    {
      EEPROM_ESCRITURA(0, mensajeSMS.substring(5,17));
      telf1Usuario = EEPROM_LECTURA(0);
      enviaSMS(telefonoSMSCliente,"El nuevo numero ingresado es:" + telf1Usuario);
    }
    if (mensajeSMS=="puerta"||mensajeSMS=="Puerta"||mensajeSMS=="PUERTA")
    {
      if (senIR==LOW)
      {
        enviaSMS(telefonoSMSCliente,"La puerta esta CERRADA");
      }
      if (senIR==HIGH)
      {
        enviaSMS(telefonoSMSCliente,"La puerta esta ABIERTA");
      }
    }
  }
}

```

*Figura 43:* Configuración para realizar acciones de acuerdo al mensaje interpretado.

Fuente: Elaboración propia.

código, si es que los números son iguales se procede a realizar una serie de instrucciones, en la figura 43 muestra alguna de esas instrucciones.

Desarrollamos el código el cual realiza una pregunta, si es que el mensaje empieza con los símbolos `*#*1@` significa que el usuario quien mando el SMS quiere añadir un numero de usuario más al sistema, entonces se escribe en la memoria del microcontrolador el número que se ingresó en el mensaje, llamado a la función `EEPROM_ESCRITURA (0,tlfSMS)`.

Si el mensaje de entrada es (puerta, Puerta o PUERTA), el algoritmo interpretará que el usuario quiere saber en qué estado se encuentra la puerta de su vivienda, al realizarse ese evento, se procede a desarrollar la lectura del sensor infrarrojo IR-FC-51 cuya información esta almacenada en la variable "senIR", si la información almacenada en la variable es un uno lógico (dicho de otra forma un alto o HIGH), se interpreta como puerta cerrada de lo contrario si en la variable esta almacenada un cero lógico (dicho de otra forma bajo o LOW), se interpreta como puerta abierta, siendo uno de los dos casos se procede a responder el mensaje utilizando la función "enviarSMS();".

En la figura 44 se observa el código que se vincula al momento de llamar la función "enviarSMS()", con "serialGSM.print()" se imprime o se escribe los códigos que ingresaran en forma serial al módulo GSM SIM800L, "AT+CMGS" en un comando AT para poder empezar a escribir el mensaje y "(char)26" es el comando para enviar el mensaje

```
void enviaSMS(String telefono, String Mensaje)
{
  serialGSM.print("AT+CMGS=\"" + telefono + "\"\n");
  serialGSM.print(Mensaje + "\n");
  serialGSM.print((char)26);
}
```

Figura 44: Programación para escribir y enviar un mensaje desde el sistema al teléfono pre establecido. Fuente: Elaboración propia.

Para cargar los valores de entrada de los sensores y mantenerlos siempre actualizando en el void loop se creó la función "SENSOR ()" la cual está

constantemente verificando los valores de entrada para por almacenarlos en variables preestablecidas, dicha función se lee primero antes de las funciones “LlamadEntrante y SMSentrante”, porque se requiriere información de los sensores actualizados para poder responder los mensajes, en la figura 45 se muestra la función “SENSOR()”.

```
void loop()
{
  leerGSM();
  if (comandoGSM != "")
  {
    Serial.println(comandoGSM);
    comandoGSM = "";
  }
  mensajeSMS.trim(); // Elimina espacio en blanco de la cadena de mensaje.
  llegoSMS = false;
  SENSOR();

  LlamadEntran();
  SMSentrante();
}
```

Figura 45: Función que mantiene la información de los sensores actualizadas.  
Fuente: Elaboración propia.

Una vez llamada a la función “SENSOR()”, lo primero que ejecuta son los comandos para poder leer la información en forma digital de los pines preestablecidos correspondiente tanto para el sensor IR y el sensor PIR y luego cargarlas en las variables “senIR y senPIR”, respectivamente, luego desarrolla dos flancos de subida tanto para el sensor INFRAROJO que estará en la puerta de la vivienda como para el sensor PIR que estará en la pared de la vivienda, estos flancos ayudan a enviar los mensajes de alerta solo si se activa el paso de un estado de cero lógico “0” a uno lógico “1” y la condición de flanco este en bajo, ya que los sensores transfieren la información de cero lógico si no se produce ninguna evento que les pueda alterar, de esta manera el sistema solo envía mensajes al producirse el flanco de lo contrario estaría mandado reiteradamente un mensaje solo al cambio de estado de los sensores, en la figura 46 se muestran el algoritmo para entrar a un flanco de subida.

```

void SENSOR()
{
  senIR=digitalRead(IR);
  senPIR=digitalRead(PIR);

  if (senIR == HIGH && flanRF == LOW)
  {
    enviaSMS(telefonoSMSCliente,"PELIGRO!! alguien abrio la puerta de ACASH");

    if(telf1Usuario.substring(0,3)=="+51")
    {
      String NumAlt=telf1Usuario.substring(3,12);
      delay(4000);
      enviaSMS(NumAlt,"PELIGRO!! alguien abrio la puerta de ACASH");
    }
  }
  flanRF = senIR;

  (senPIR == HIGH && flanPIR == LOW)
  {
    enviaSMS(telefonoSMSCliente,"Alguien esta dentro de la casa de ACASH");
    if(telf1Usuario.substring(0,3)=="+51")
    {
      String NumAlt=telf1Usuario.substring(3,12);
      delay(4000);
      enviaSMS(NumAlt,"Alguien esta dentro de la casa de ACASH");
    }
  }
}

```

Figura 46. Algoritmo para enviar SMS que advierte al usuario de algún evento de peligro de la vivienda.  
Fuente: Elaboración propia

### 3.4. Implementación del proyecto

Debido que el prototipo a diseñar en este proyecto utiliza módulos electrónicos, la PCB que se diseñó es de 10cm \* 10 cm, se soldó espadines en la PCB para montar los módulos esto ayuda a regular el voltaje de los módulos Sep down y Stem up antes de utilizarlos, de lo contrario se podría quemar los demás módulos, en la figura 47 se muestra la PCB con los componentes montados en los lugares correspondientes.

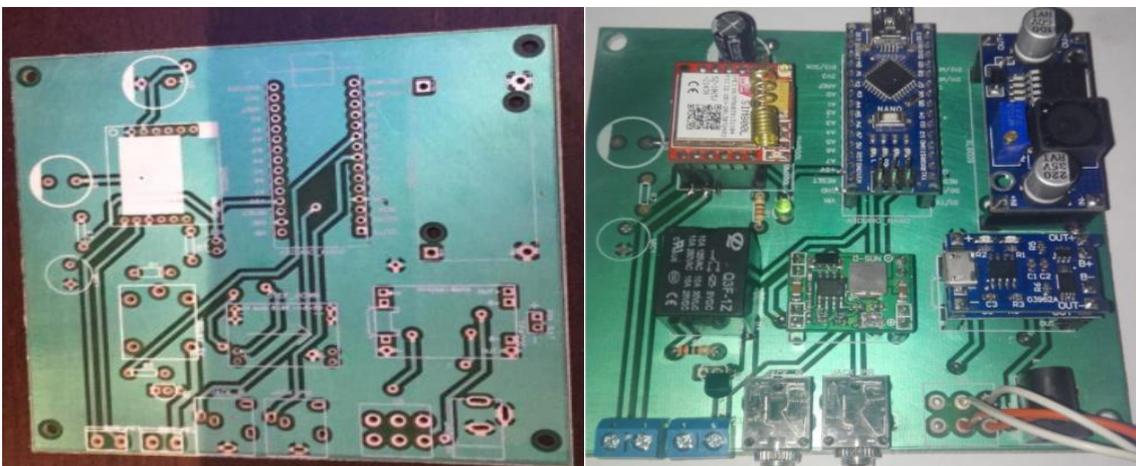


Figura 47: PCB en físico, y PCB con los módulos electrónicos.  
Fuente: Elaboración propia

Como los módulos electrónicos tienen a simple vista la placa y las pistas, para proteger de la intemperie se diseñó sus carcasas utilizando el software Blender, para luego imprimirlas en una impresora 3d, en la figura 48 se aprecia el modelado de las carcasas y las mismas impresas en 3d.

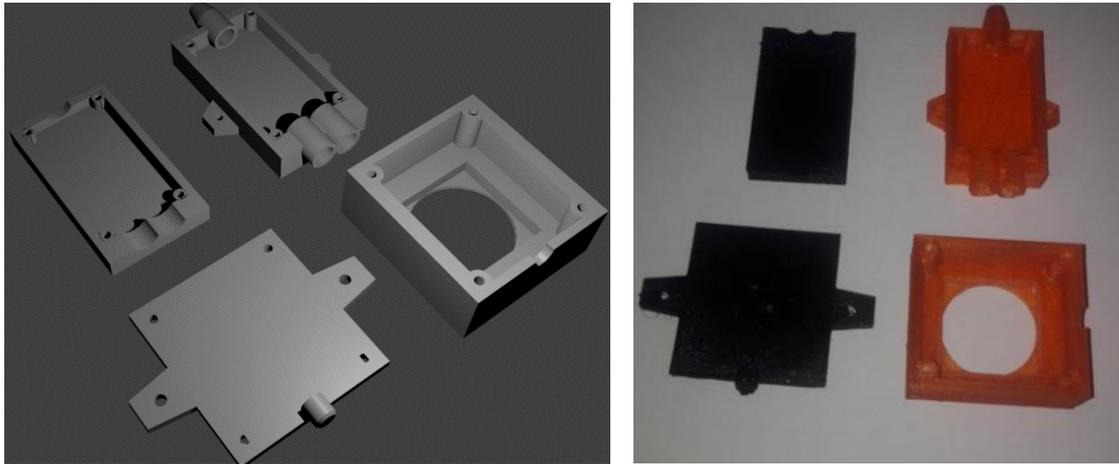


Figura 48: Modelado en 3D e impreso de las carcasas de los sensores.  
Fuente: Elaboración propia.

### 3.5. Consolidación de Resultados

Para probar el prototipo se elaboró una maqueta que simula una casa convencional de las viviendas de campo del distrito de Molinopampa. Se colocaron los sensores tal cual se muestran en las figuras 25 y 28. En la figura 49 tenemos la maqueta montada para las pruebas.

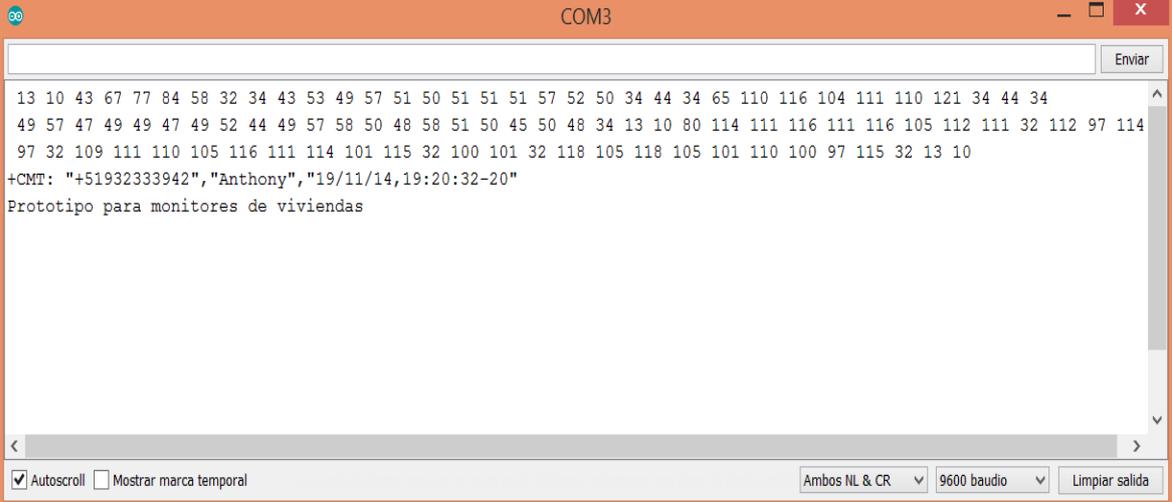


Figura 49: Maqueta de pruebas.  
Fuente: Elaboración propia

### 3.5.1. Pruebas del algoritmo

Para desarrollar las pruebas del algoritmo es necesario activar el monitor serial para poder visualizar si lo que se pretende realizar se está ejecutando correctamente, en la figura 32 se muestra el código para poder activar el monitor serial.

Abrimos el monitor serial de la IDE de Arduino y visualizamos la información de entrada cuando se alerte del ingreso de un SMS, en la figura 50 se puede ver la información proporcionada por el módulo SIM800L, cuando llega una mensaje de texto SMS, las primeras 3 filas es la información en bruto o información sin procesar, la cual está en código ASCII, para mayor comodidad en la programación se convirtió el código ASCII a formato texto la cual se visualiza en la 4ta y 5ta fila.



```
COM3
13 10 43 67 77 84 58 32 34 43 53 49 57 51 50 51 51 51 57 52 50 34 44 34 65 110 116 104 111 110 121 34 44 34
49 57 47 49 49 47 49 52 44 49 57 58 50 48 58 51 50 45 50 48 34 13 10 80 114 111 116 111 116 105 112 111 32 112 97 114
97 32 109 111 110 105 116 111 114 101 115 32 100 101 32 118 105 118 105 101 110 100 97 115 32 13 10
+CMT: "+51932333942", "Anthony", "19/11/14,19:20:32-20"
Prototipo para monitores de viviendas
```

Figura 50: Entrada de SMS en Formato ASCII y TEXTO.  
Fuente: Elaboración propia

La cadena de entrada empieza con +CMT: la cual significa que la información de entrada se trata de un mensaje de texto, seguido de la primera comilla (") y el número telefónico de la persona quien envió el mensaje luego cierra con la segunda comilla para luego abrir una tercera comilla, seguido de un nombre, este nombre solo se hará presente si es que en la tarjeta SIM se agregó el contacto previamente, luego cierra con la 4ta comilla, para abrir una 5ta comilla y cerrar con la 6ta comilla dentro de las cuales se encuentra la información de hora y fecha de envió del

mensaje, luego se produce un salto de línea y se visualiza el mensaje en enviado "Prototipo para monitoreo de viviendas", toda esta información se tiene que guardar para luego procesarla, en la figura 38 se puede ver el código para guardar esta información.

Lo importante para nosotros es el número de celular de quien mande el mensaje y el mensaje en sí, esta información ayudara en la seguridad del prototipo ya que eliminara cualquier mensaje de entrada que no se haya programado previamente tampoco admitirá un mensaje de un número que no se haya programado previamente, en la figura 30 se aprecia la declaración de las variables globales en las cuales se incluyó el número de usuario, en este caso el número de celular del dueño de la vivienda, este número será comparado con el número de celular del mensaje éntrate si en caso sean iguales se procederá a comparar el mensaje enviado con los mensajes pre establecidos, en la figura 43 se muestra el algoritmo para realizar dicho proceso.

Como ya se dijo con anterioridad para poder registra un nuevo número en el sistema, solo lo puede hacer el Súper usuario en este caso el dueño de la vivienda cuyo número de celular este pre establecido en la programación, en la figura 51 se aprecia el mensaje de entrar para poder registra un nuevo número de usuario el cual servirá para que el sistema envíe los mensajes de alerta, tanto al dueño de la vivienda como al nuevo usuario ingresado.

```
+CMT: "+51932333942", "Anthony", "19/11/15,09:18:08-20"  
*#*1@987654321
```

*Figura 51: Mensaje de entrada para registrar un nuevo número de usuario.  
Fuente: Elaboración propia.*

Para que el algoritmo reconozca que el mensaje de entrada se trata de un mensaje para grabar el número de celular de un nuevo usuario, primero analiza las 5 primeras líneas de la trama las cuales deberían ser `*#*1@` si es correcto pasa a grabar el numero en la memoria EEPROM. En la figura 52 se muestra la lógica para procesar el SMS de entrada, se digito también para poder visualizar en el monitor serial, de esa manera poder estar seguros que el código funciona correctamente.

Podemos notar en pantalla que al ingresar el mensaje de registro el algoritmo responde correctamente.

```
void SMSentrante()
{
  if (tlfSMS!=" " && tlfSMS==telefonoSMSCliente51 || tlfSMS!=" " && tlfSMS==telfiUsuario)
  {
    Serial.print("\n");
    Serial.print("primeras 5 lineas:");
    Serial.println(mensajeSMS.substring(0,5));
    if(mensajeSMS.substring(0,5)=="*#*1@")
    {
      EEPROM_ESCRITURA(0, mensajeSMS.substring(5,17));
      telfiUsuario = EEPROM_LECTURA(0);
      enviaSMS(telefonoSMSCliente,"El nuevo numero ingresado es:" + telfiUsuario);
    }
  }
}

+CMT: "+51932333942", "Anthony", "19/11/15, 10:25:12-20"
*#*1@987654321

primeras 5 lineas:*#*1@
el número guardado es:+51932333942
```

Figura 52: Respuesta del mensaje SMS para registra un numero Celular.  
Fuente: Elaboración propia.

Si queremos saber el estado de la vivienda, si es que la puerta se encuentra cerrada o abierta, se deberá mandar un SMS con la palabra “puerta, Puerta o PUERTA”. En la figura 43 se muestra el programa que realiza dicha acción, para poder ver en pantalla si responde correctamente al SMS se agregó algunas líneas de código tal como se muestra en la figura 53.

```
Serial.print("\n");
Serial.print("Mensaje de entrada:");
Serial.println(mensajeSMS);
Serial.print("\n");

if (mensajeSMS=="puerta"||mensajeSMS=="Puerta"||mensajeSMS=="PUERTA")
{
  if (senIR==HIGH)
  {
    enviaSMS(telefonoSMSCliente,"La puerta esta CERRADA");
  }

  if (senIR==LOW)
  {
    enviaSMS(telefonoSMSCliente,"La puerta esta ABIERTA");
  }
}

+CMT: "+51932333942", "Anthony", "19/11/15, 15:56:34-20"
Puerta

Mensaje de entrada:Puerta

AT+CMGS="932333942"La puerta esta ABIERTA
> comando GSM: AT+CMGS="932333942"La puerta esta ABIERTA
>
AT+CMGS="932333942"La puerta esta ABIERTA
>
```

Figura 53: Respuesta para saber el estado de la puerta.  
Fuente: Elaboración propia.

Podemos apreciar que el SMS de entrada ha sido procesado correctamente, así mismo el sistema responde al número de estrada con un nuevo mensaje según el estado de la puerta la cual podría ser “la puerta está CERRADA o LA puerta está ABIERTA”, en la figura 54 se puede ver el mensaje que responde el prototipo.

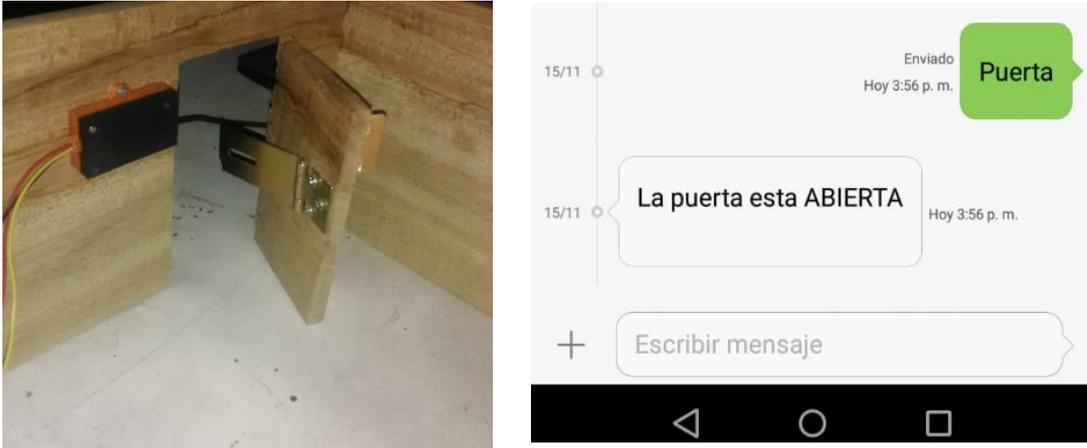


Figura 54: Mensaje enviado luego de procesar el mensaje de entrada.  
Fuente: Elaboración propia.

En la figura 55 se muestra el estado de alerta cuando la puerta se abre, el sistema manda un mensaje de texto al dueño de la vivienda alertándolo de un posible robo.

```
>comando GSM: AT+CMGS="932333942"PELIGRO!! alguien abrio la pueta de ACASH  
>  
AT+CMGS="932333942"PELIGRO!! alguien abrio la pueta de ACASH  
>
```



Figura 55: mensaje de alerta, visto en el monitor serial y el celular.  
Fuente: Elaboración propia.

Si por el contrario se ingrese a la vivienda por otra parte, el sensor PIR lo detectara enviando los mensajes de alerta al dueño de la vivienda.

### 3.6. Presupuesto del proyecto

Los módulos electrónicos fueron comprados por unidades los precios se muestran en la tabla 2 teniendo una inversión de S/. 240,00 soles si restamos la madera para la maqueta y el acrílico de la carcasa tendríamos S/. 215,00 que es el precio de todos los componentes electrónicos, dicho precio podría bajar si es que se compran los componentes electrónicos en cantidad para que le precio salga por mayor.

Tabla 2  
*Tabla de presupuesto*

Material	Cantidad	Unidad	Precio unitario	Precio total
Arduino nano	1	UND	S/. 15	15,00
Módulo SIM800L	1	UND	S/. 45	45,00
Módulo Step Up XL6009	1	UND	S/. 10	10,00
Módulo Step Down MP1584	1	UND	S/. 6	6,00
Módulo TP4056	2	UND	S/. 8	16,00
Modulo IR -FC-51	1	UND	S/. 8	8,00
Módulo PIR HC-SR501	1	UND	S/. 9	9,00
led y Resistencia de 220 ohm 1/4 w	1	UND	S/. 1	0,50
Relay 5v	1	UND	S/. 1	1,00
Diodo 1n 4007	1	UND	S/. 0,2	0,20
Transistor 2N2222A	1	UND	S/. 0,2	0,20
Borneras de 2 pines	2	UND	S/. 0,4	0,80
Jack de sonido hembra	2	UND	S/. 0,5	1,00
Cables	6	M	S/. 0,2	1,20
Batería Lipo 3.7 v 2.5A	1	UND	S/. 16	16,00
Plug DC 2.1mm	1	UND	S/. 0,5	0,50
Capacitor 1000uf 16v	2	UND	S/. 0,3	0,60
Plug macho de sonido	2	UND	S/. 1,0	2,00
Impresión 3D de las carcasas	1	UND	S/. 12	12,00
Maquinado de la PCB	1	UND	S/. 70	70,00
Carcasa de acrílico	1	UND	S/.10	10,00
Madera para Maqueta	1	UND	S/.15	15,00
<b>TOTAL</b>				<b>240,00</b>

Nota. Se muestra la moto total gastada para la elaboración del proyecto. Fuente: Elaboración propia.



## CONCLUSIONES

Al término del proyecto se llegó a las siguientes conclusiones.

- En la presente investigación, se logró desarrollar un prototipo para monitoreo del control de acceso de las viviendas en zonas rurales, utilizando módulos electrónicos y tecnología GSM, demostrando que al utilizar tecnología GSM para monitoreo del control de acceso de las viviendas rurales, no tenemos restricciones en distancia para poder monitorear la vivienda, basta con tener acceso a alguna línea telefónica.
- Se diseñó la PCB, la cual garantizó que los módulos utilizados funcionen de acuerdo a sus especificaciones, al ser un prototipo se fue regulando las tensiones de alimentación hasta llegar al funcionamiento óptimo del SIM800L.
- Se logró desarrollar un prototipo escalable, gracias a la bornera que va directamente a un relay se podría prender cualquier aparato electrónico que funciones con corriente alterna o continua, también se dejó pads sueltos para poder conectar un micrófono y realizar llamadas que se podrían grabar, el programa que rige el funcionamiento del prototipo se puede actualizar y mejorar.
- Gracias al módulo de Arduino nano y su entorno de desarrollo, se logró codificar el algoritmo de funcionamiento del sistema exitosamente, cabe resaltar que el algoritmo puede fusionar con cualquier sensor digital que tenga como parámetros lógicos de "1" como 5v y "0" como Gnd.
- En las pruebas desarrolladas del prototipo se logró, el registro un nuevo número de celular de usuario, así como los mensajes de alerta que se envían a los números de celulares pre establecidos cuando se abre la puerta, de esa manera se logra alertar al dueño de la vivienda de lo ocurrido.

## RECOMENDACIONES

- Para el funcionamiento óptimo del módulo SIM800L se recomienda utilizar una fuente de más de 2 amperio, ya que llega a picos de 2 amperios cuando realiza llamadas o envió de mensajes de texto, al no tener esta fuente el módulo tiende a reiniciarse y tarda mucho para volver a conectarse.
- Antes de la puesta en campo, se recomienda desarrollar pruebas en el lugar a monitorear, de esa manera poder validar el funcionamiento óptimo y obtener certificaciones, para que el prototipo se vuelva un producto comercial.
- En la encuesta realizada en el distrito de Molinopampa, se pudo notar que los pobladores cuentan con una vivienda de campo en diferentes lugares del distrito, lugares que no son muy cercanos, por lo que tecnologías de comunicaciones para monitoreo de viviendas con módulos de radiofrecuencia no es muy conveniente debido a su limitante de distancia; estas viviendas en su mayoría tiene acceso a una línea telefónica, lo cual es factible para el desarrollo del presente proyecto así como el uso de la red GSM para el monitoreo de las viviendas.
- Debido a que algunas viviendas se encuentra lejos del área urbana, se recomienda para futuros trabajos adicionar un módulo GSP, de esa manera poder enviar mensaje de alerta con las coordenadas y mandar drones para monitorear ya que ellos llegarían más rápido.
- Existen muchos software para desarrollo de PCB, EasyEDA es uno de ellos, ya que su funcionamiento es de forma online, se puede encontrar los diseños a medida de diversos módulos electrónicos que su comunidad proporciona, haciéndole un software muy versátil y de fácil manejo.

## BIBLIOGRAFÍA

- Adafruit. (2018). *PIR Motion Sensor*. Obtenido de <https://cdn-learn.adafruit.com/downloads/pdf/pir-passive-infrared-proximity-motion-sensor.pdf>
- Arduino. (2019). *ARDUINO.CC*. Obtenido de <https://www.arduino.cc/en/Main/software>
- Bluehack. (2005). *Comando AT*. Obtenido de <http://bluehack.elhacker.net/proyectos/comandosat/comandosat.html>
- Carlos. (2015). *Buffer en Informática*. Obtenido de <https://empiezoinformatica.wordpress.com/2015/05/09/buffers-en-informatica/>
- EasyEDA. (2019). *EasyEDA*. Obtenido de <https://easyeda.com>
- Ecured. (2018). *Microcontrolador*. Obtenido de <https://www.ecured.cu/Microcontrolador>
- Fraile, E. (2017). *Evolución de las comunicaciones móviles*. Obtenido de <https://www.interempresas.net/Electronica/Articulos/195234-Evolucion-de-las-comunicaciones-moviles.html>
- Gonzales, J. (2017). *Baudios y transmisión*. Obtenido de <https://github.com/Obijuan/open-fpga-verilog-tutorial/wiki/Cap%C3%ADtulo-21:-Baudios-y-transmisi%C3%B3n>
- GSMA. (2017). *Breve historia de GSM y GSMA*. Obtenido de <https://www.gsma.com/aboutus/history>
- Hernando, J. (2015). *Comunicaciones Móviles* (Vol. III). Madrid: Editorial Universal-Ramón Areces.
- Hernandez, A. (2015). *Abreviaturas comunes en electrónica*. Obtenido de <https://tallerelectronica.com/2015/05/25/abreviaturas-comunes-en-electronica/>
- INEI, I. N. (2019). *Encuesta Nacional de Programas Presupuestales 2011-2018*. Obtenido de [https://www.inei.gob.pe/media/MenuRecursivo/publicaciones\\_digitales/Est/Lib1662/](https://www.inei.gob.pe/media/MenuRecursivo/publicaciones_digitales/Est/Lib1662/)

- INEI, I. N. (2019). *Estadística de Seguridad Ciudadana*. Obtenido de [https://www.inei.gob.pe/media/MenuRecursivo/boletines/boletin\\_seguridad\\_ciudadana\\_departamental2013\\_2019.pdf](https://www.inei.gob.pe/media/MenuRecursivo/boletines/boletin_seguridad_ciudadana_departamental2013_2019.pdf)
- Kylinchip. (2015). *XL6009*. Obtenido de <https://www.haoyuelectronics.com/Attachment/XL6009/XL6009-DC-DC-Converter-Datasheet.pdf>
- Mendes, & Campos. (2016). *Desarrollo del Prototipo de un Sistema de Seguridad Basado en tecnología GSM para la Empresa Familiar CASTRO Y BACA S.A.C.* Obtenido de <http://repositorio.utp.edu.pe/bitstream/UTP/242/6/0920333-1210236.pdf>
- Meyrán, J. (2008). *Augustin-Jean Fresnel*. <https://www.medigraphic.com/pdfs/revmexoft/rmo-2008/rmo082n.pdf>.
- Ordoñez, E. (2010). Obtenido de <file:///C:/Users/Usuario/Downloads/UPS-CT001940.pdf>
- OSCIPTTEL. (2015). *Glosario de terminos en telecomunicaciones en el Perú*. Huancayo: Imprenta Rios S.A.C.
- OSIPTTEL. (2019). *Los servicios publicos de telecomunicaciones en los hogares peruanos*. Obtenido de <https://www.osiptel.gob.pe/repositorioaps/data/1/1/1/par/erestel-2018-servicios-telecomunicaciones-hogares/erestel-2018-servicios-telecomunicaciones-hogares.pdf>
- Padilla, J. J. (2011). *Comunicaciones Mviles*. Obtenido de <http://jpadilla.docentes.upbbga.edu.co/moviles/2%20conceptosbasicos.pdf>
- Palomino, C. (2017). *Sensores activos y pasivos*. Obtenido de <https://es.scribd.com/presentation/365248817/SENSORES-ACTIVOS-Y-PASIVOS-pptx>
- Perez, J. (2015). *Puerto Serial*. Obtenido de <https://definicion.de/puerto-serial/>
- Punto flotante S.A. (2017). *Sensor infrarrojo de movimiento PIR HC-SR501*. Obtenido de <https://puntoflotante.net/MANUAL-DEL-USUARIO-SENSOR-DE-MOVIMIENTO-PIR-HC-SR501.pdf>
- SIMCom. (2013). *SIM800L Hardware Design*. Obtenido de [https://img.filipeflop.com/files/download/Datasheet\\_SIM800L.pdf](https://img.filipeflop.com/files/download/Datasheet_SIM800L.pdf)
- Villalobos. (2008). *Fundamentos de programacion*. MacroE.I.R.L.

Zambrano, M. (2012). *Sistema de alarma para mejorar la seguridad de la Empresa AUPLATEC ubicada en el Cantón Pelileo*. Obtenido de <http://repositorio.uta.edu.ec/jspui/handle/123456789/2911>

Zuñiga Peña, J. L. (2018). Obtenido de Diseño e implementación de un timbre inteligente basado en el internet de las cosas (IOT) para fortalecer la seguridad contra robos en viviendas sociales : <http://repositorio.unprg.edu.pe/bitstream/handle/UNPRG/3272/BC-TES-TMP-2020.pdf?sequence=1&isAllowed=y>

## **ANEXOS**

ANEXO A  
ENCUESTA REALIZADA A LOS POBLADORES DEL DISTRITO DE  
MOLINOPAMPA DEPARTAMENTO DE AMAZONAS



## UNIVERSIDAD NACIONAL TECNOLÓGICA DE LIMA SUR

### ENCUESTA

Encuesta a pobladores en Molinopampa/ Amazonas, para un sondeo de seguridad en sus casas de campo.

1.- ¿Cuenta con una casa de campo?

Si  No

2.- ¿a que distancia de su casa hogar, se encuentra su casa de campo?

\_\_\_\_\_ km—nombre del lugar: \_\_\_\_\_

2.- ¿Con que frecuencia visita esa casa de campo?

Diario:  Mañanas.  Noches.  1x Semana.  C/15 días.  Otros \_\_\_\_\_

3.- ¿Las veces que va a la visita, cuantas horas se quedan?

1-2h.  2-4h.  4-6h.  6h a +.

4.- ¿Ha sido víctima de robo en su casa de campo?

Si  No

5.- ¿Cree usted que sería víctima de robo de su casa de campo?

Si  No

6.- ¿Cuenta con algún tipo de seguridad que le alerte de un posible robo de sus pertenencias en su casa de campo?

Si  No

7.- Según la zona donde se encuentra su casa de campo, ¿cree usted que es necesario algún tipo de alarma que le alerte de un posible robo?

Si  No

8.- ¿Cuenta usted con un teléfono celular?

Si  No

9.- ¿En la zona donde se encuentra su casa de campo, tiene acceso a alguna línea telefónica?

Si  No

Si responde "Si"

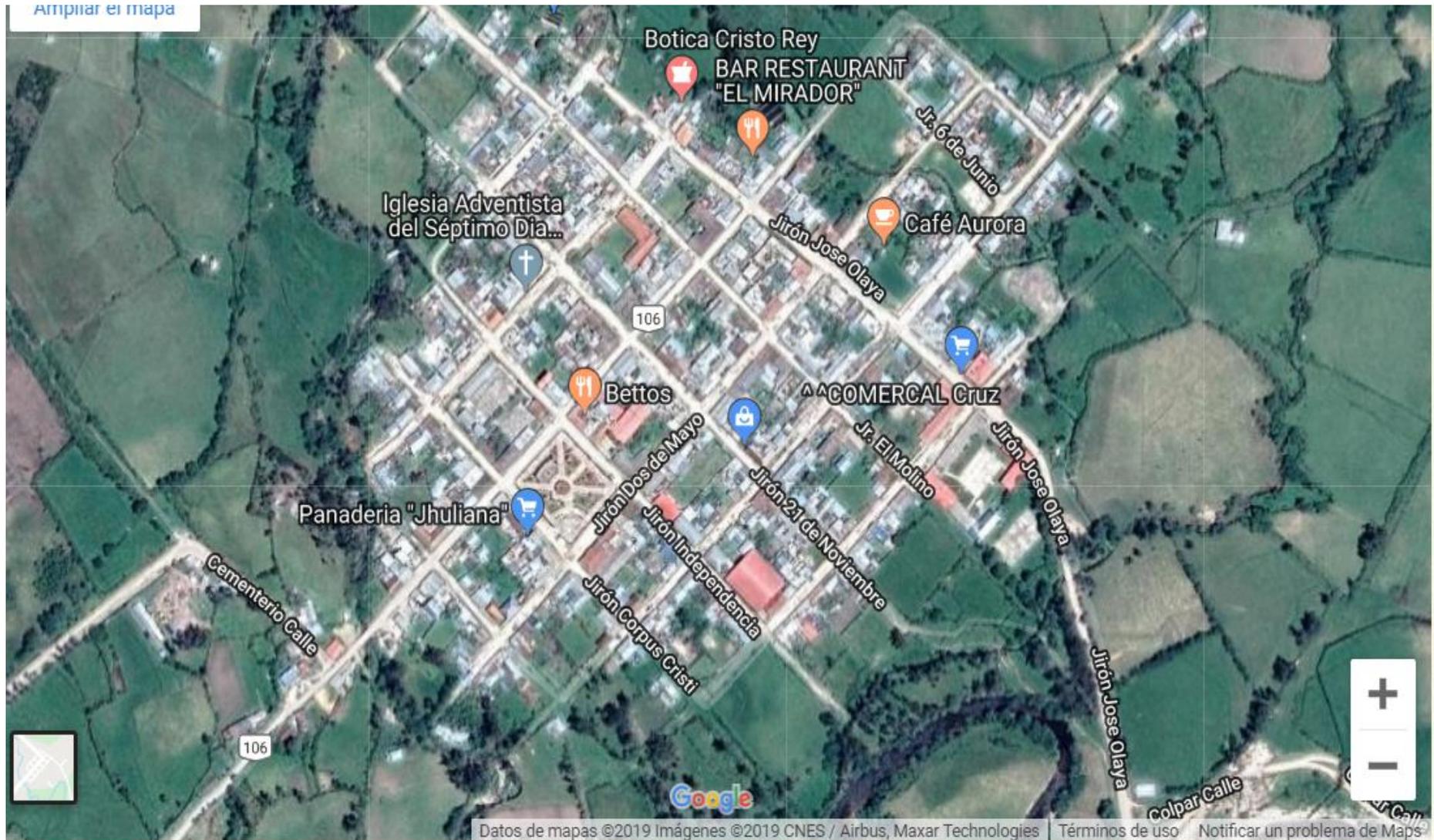
10.- ¿A qué compañía le pertenece la línea telefónica?

Claro  Movistar  Entel  Bitel  Otros \_\_\_\_\_

Fecha: / /2019

Número de encuestado	¿Cuenta con una casa de campo?	¿A qué distancia de su casa hogar, se encuentra su casa de campo?	Km	Nombre del lugar	¿Con qué frecuencia visita esa casa de campo?	¿Las veces que va a la visita, cuantas horas se queda?	¿Ha sido víctima de algún robo en su casa de campo?	¿Cree usted que sería víctima de robo en su casa de campo?	¿Cuenta con algún tipo de seguridad que le alerte de un posible robo de sus pertenencias en su casa de campo?	Según la zona donde se encuentra su casa de campo, ¿cree usted que es necesario algún tipo de alarma que le alerte de un posible robo?	¿Cuenta usted con un teléfono celular?	¿En la zona donde se encuentra su casa de campo, tiene acceso a alguna línea telefónica?	¿A qué compañía le pertenece la línea telefónica?
1	Si		0.5	Atun Pampa	Diario	1-2h	No	Si	No	Si	Si	Si	Claro
2	Si		2.5	Acash	Interdiario	1-2h	Si	Si	No	Si	Si	Si	Claro
3	Si		3	Taulia	1/mes	2-4h	No	Si	No	Si	Si	Si	Claro
4	Si		1.5	Colpar	Diario	2-4h	Si	Si	No	Si	Si	Si	Claro
5	Si		5.5	Shucsho	Diario	4-6h	No	Si	No	Si	Si	No	N/L
6	Si		1	Romompata	Diario	1-2h	Si	Si	No	Si	Si	Si	Claro
7	Si		6	Ancashuati	C/15 días	2-4h	Si	Si	No	Si	Si	Si	Claro
8	Si		6.5	Shora	1/Semana	2-4h	Si	Si	No	Si	Si	Si	Claro
9	Si		2	Ingapirca	1/Semana	2-4h	Si	Si	No	Si	Si	Si	Claro
10	Si		1.5	Estancia	Diario	1-2h	No	Si	No	Si	Si	Si	Claro
11	Si		6	Tingo	1/Semana	2-4h	Si	Si	No	Si	Si	Si	Movistar
12	Si		2.5	Cashjon	Diario	1-2h	Si	Si	No	Si	Si	Si	Claro
13	Si		0.5	Atun Pampa	1/Semana	2-4h	No	Si	No	Si	Si	Si	Claro
14	Si		1.5	Camillan	Diario	2-4h	Si	Si	No	Si	Si	Si	Claro
15	Si			Punta de peñasha	1/Semana	2-4h	Si	Si	No	Si	Si	Si	Claro
16	Si		0.5	Cruz Loma	Diario	1-2h	Si	Si	No	Si	Si	Si	Claro
17	Si		2	Chiquej	1/Semana	2-4h	Si	Si	No	Si	Si	Si	Claro
18	Si			Tres hermanos	1/mes	4-6h	Si	Si	No	Si	Si	Si	Claro
19	Si		0.5	Atun Pampa	1/mes	2-4h	Si	Si	No	Si	Si	Si	Claro
20	Si			Juana Potrero	Diario	1-2h	Si	Si	No	Si	Si	Si	Claro
21	Si			Putquero	Diario	2-4h	No	Si	No	Si	Si	Si	Claro
22	Si		2.5	Cashjon	Diario	2-4h	Si	Si	No	Si	Si	Si	Claro
23	Si		10	Zicse	C/15 días	4-6h	No	Si	No	Si	Si	Si	Movistar
24	Si		4	Lujma	Diario	2-4h	No	Si	No	Si	Si	Si	Claro
25	Si		1.3	Juana Loma	Diario	1-2h	No	Si	No	Si	Si	No	N/L
26	Si		6.5	Muños	1/Semana	4-6h	Si	Si	No	Si	Si	No	N/L
27	Si		1.5	Romompata	Diario	1-2h	No	Si	No	Si	Si	Si	Claro
28	Si			Putquero	Interdiario	2-4h	No	Si	No	Si	Si	Si	Claro
29	Si			Calo	Diario	1-2h	Si	Si	No	Si	Si	Si	Claro
30	Si			Caran	1/Semana	2-4h	Si	Si	No	Si	Si	Si	Movistar
31	Si			Tiu tiu	C/15 días	2-4h	Si	Si	No	Si	Si	Si	Movistar
32	Si			Huamazan	C/15 días	2-4h	Si	Si	No	Si	Si	Si	Claro
33	Si			Matche	1/Semana	2-4h	Si	Si	No	Si	Si	Si	Claro
34	Si			Ojo de Molino	1/Semana	4-6h	Si	Si	No	Si	Si	Si	Claro
35	Si		6	Montealegre	C/15 días	4-6h	Si	Si	No	Si	Si	Si	Claro
36	Si		1.5	Pirca	Diario	1-2h	Si	Si	No	Si	Si	Si	Claro
37	Si		10	Sizo	Interdiario	6h a+	Si	Si	No	Si	Si	No	N/L
38	Si		2.5	Cashjon	1/Semana	1-2h	No	Si	No	Si	Si	Si	Claro
39	Si			Zicse	1/mes	4-6h	No	Si	No	Si	Si	Si	Movistar
40	Si		12	Eñara	1/Semana	4-6h	Si	Si	No	Si	Si	Si	Claro
41	Si			Julian Pampa	1/Semana	2-4h	Si	Si	No	Si	Si	Si	Claro
42	Si		10	Peña Sur	C/15 días	4-6h	Si	Si	No	Si	Si	Si	Claro
43	Si		6	Putquero	1/Semana	2-4h	Si	Si	No	Si	Si	Si	Claro
44	Si		1	Molino	Diario	1-2h	Si	Si	No	Si	Si	Si	Claro
45	Si		1.5	Colpar	Diario	1-2h	Si	Si	No	Si	Si	Si	Claro
46	Si		3	Taulia	Diario	1-2h	Si	Si	No	Si	Si	Si	Claro
47	Si		2	Cedro Pampa	1/mes	1-2h	Si	Si	No	Si	Si	Si	Claro
48	Si		1	Ramada	Diario	2-4h	No	Si	No	Si	Si	Si	Claro
49	Si		3.5	Julian Potreror	1/Semana	2-4h	No	Si	No	Si	Si	No	N/L
50	Si		1.5	Corralillo	Diario	1-2h	No	Si	No	Si	Si	Si	Claro

### Mapa satelital del distrito de Molinopampa



ANEXO B  
HOJA DE DATOS DE LA PLACA ARDUINO NANO

Reference Designs ARE PROVIDED "AS IS" AND "WITH ALL FAULTS". Arduino DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Arduino may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Arduino reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.

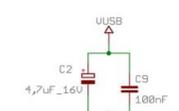
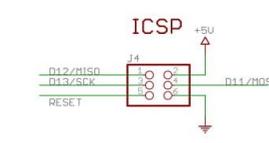
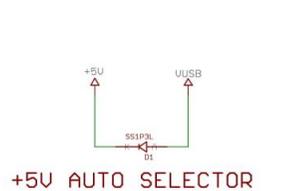
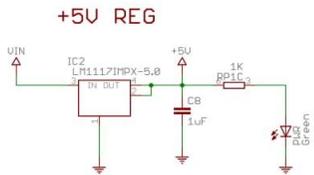
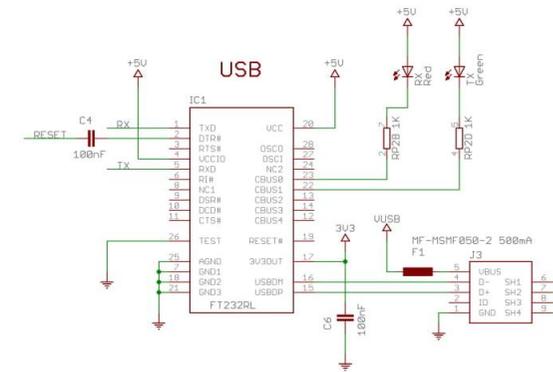
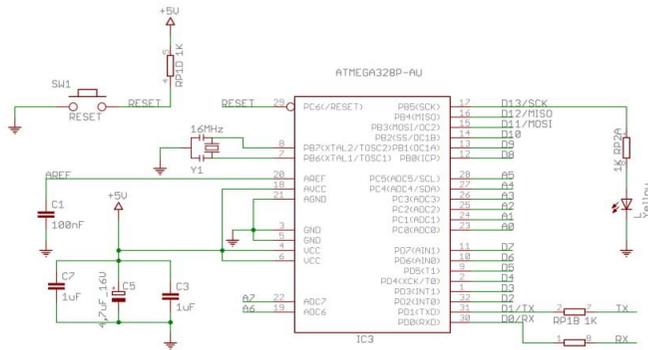
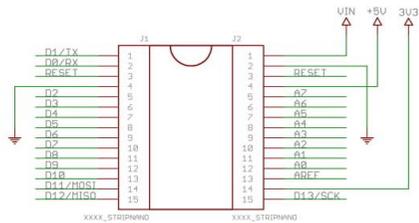
"Arduino" name and logo are trademarks registered by Arduino S.r.l. in Italy, in the European Union and in other countries of the world.

Based on design by Gravitech (gravitech.us)

# Arduino Nano

Released under Creative Commons Attribution Share Alike 3.0 Licence

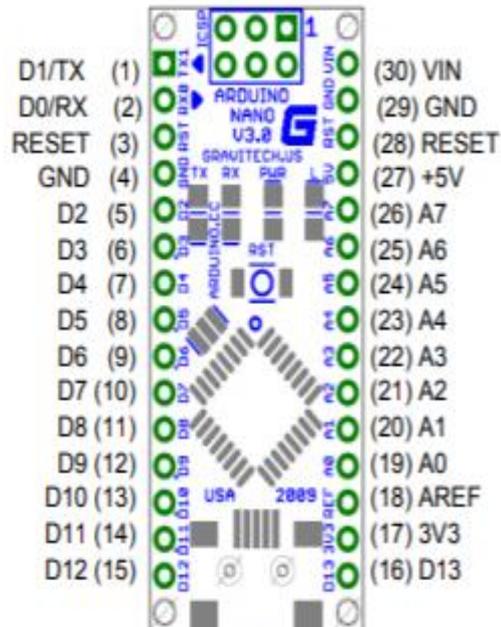
<http://creativecommons.org/licenses/by-sa/3.0/>



⊗ ⊗ ⊗ ⊗

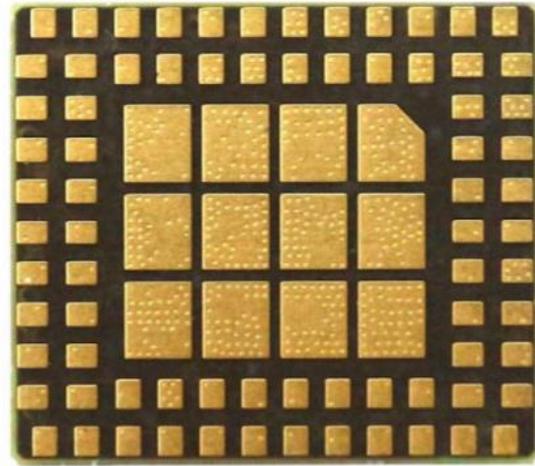
Author: E.Uita	
TITLE: Arduino Nano-Rev3.2	
Document Number:	REV: 3.2
Date: 30.12.2014 15:30:26	Sheet: 1/1

## Arduino Nano Pin Layout



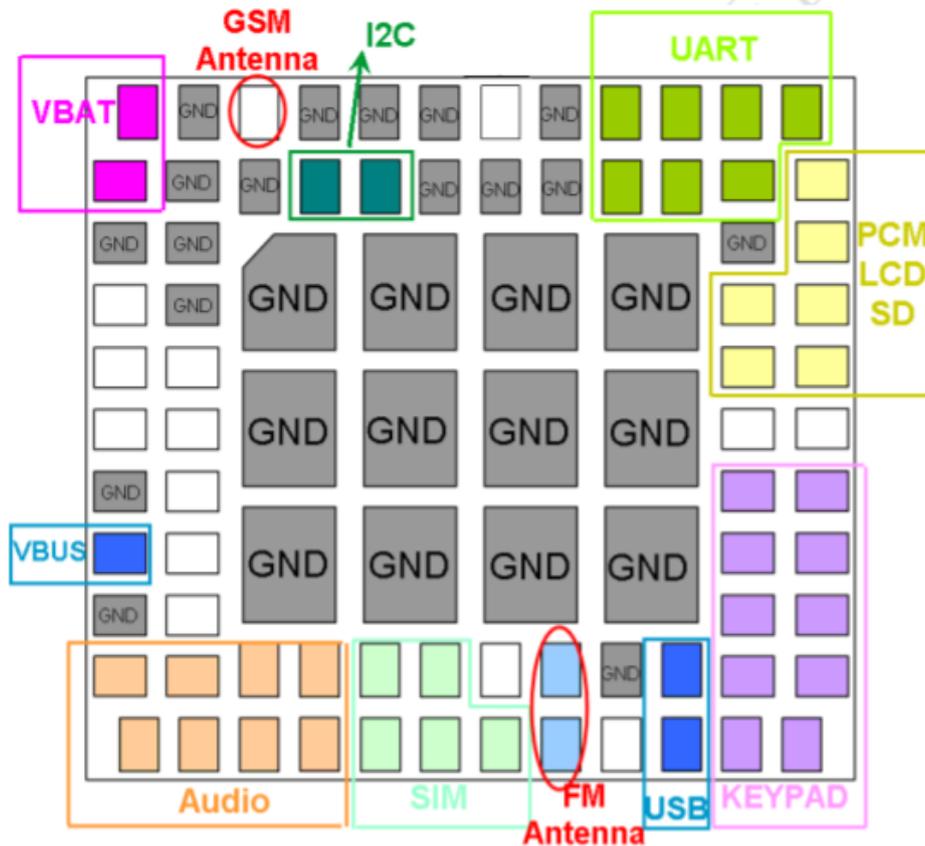
Pin No.	Name	Type	Description
1-2, 5-16	D0-D13	I/O	Digital input/output port 0 to 13
3, 28	RESET	Input	Reset (active low)
4, 29	GND	PWR	Supply ground
17	3V3	Output	+3.3V output (from FTDI)
18	AREF	Input	ADC reference
19-26	A0-A7	Input	Analog input channel 0 to 7
27	+5V	Output or Input	+5V output (from on-board regulator) or +5V (input from external power supply)
30	VIN	PWR	Supply voltage

ANEXO C  
HOJA DE DATOS DEL MODULO SIM 800L



### Pin Assignment

Before PCB layout, we should learn well about pin assignment in order to get reasonable layout with so many external components. Following figure is the overview of pin assignment of the module.



Feature	Implementation
Power supply	3.4V ~4.4V
Power saving	typical power consumption in sleep mode is 0.7mA (AT+CFUN=0 )
Frequency bands	<ul style="list-style-type: none"> <li>● Quad-band: GSM 850, EGSM 900, DCS 1800, PCS 1900. SIM800L can search the 4 frequency bands automatically. The frequency bands can also be set by AT command "AT+CBAND". For details, please refer to document [1].</li> <li>● Compliant to GSM Phase 2/2+</li> </ul>
Transmitting power	<ul style="list-style-type: none"> <li>● Class 4 (2W) at GSM 850 and EGSM 900</li> <li>● Class 1 (1W) at DCS 1800 and PCS 1900</li> </ul>
GPRS connectivity	<ul style="list-style-type: none"> <li>● GPRS multi-slot class 12 ( default )</li> <li>● GPRS multi-slot class 1~12 (option)</li> </ul>
Temperature range	<ul style="list-style-type: none"> <li>● Normal operation: -40°C ~ +85°C</li> </ul>

#### Function of Serial Port

Serial port:

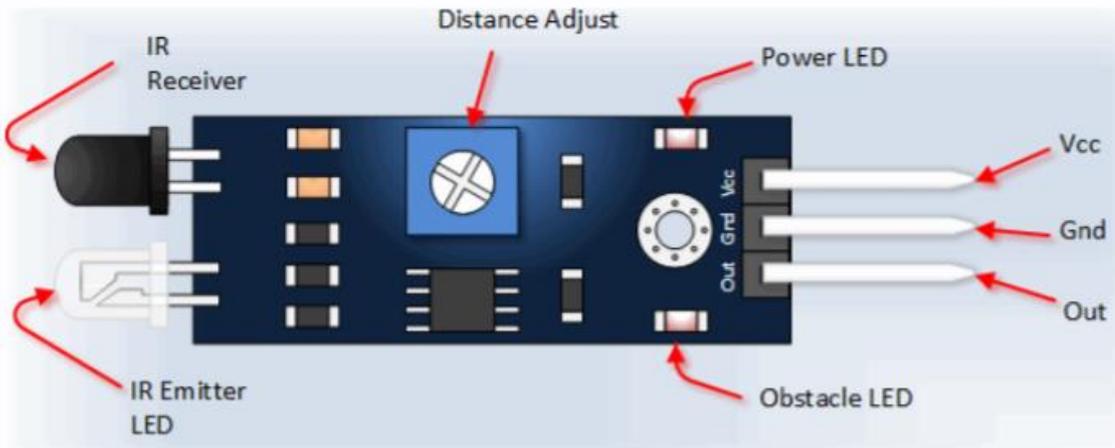
- Full modem device.
- Contains data lines TXD and RXD, hardware flow control lines RTS and CTS, status lines DTR, DCD and RI.
- Serial port can be used for CSD FAX, GPRS service and AT communication. It can also be used for multiplexing function. For details about multiplexing function, please refer to *table 11*.
- Serial port supports the following baud rates:  
1200, 2400, 4800, 9600, 19200, 38400, 57600 and 115200bps
- Autobauding only supports the following baud rates:  
1200, 2400, 4800, 9600, 19200, 38400 and 57600bps
- The default setting is autobauding.

Autobauding allows SIM800L to automatically detect the baud rate of the host device. Pay more attention to the following requirements:

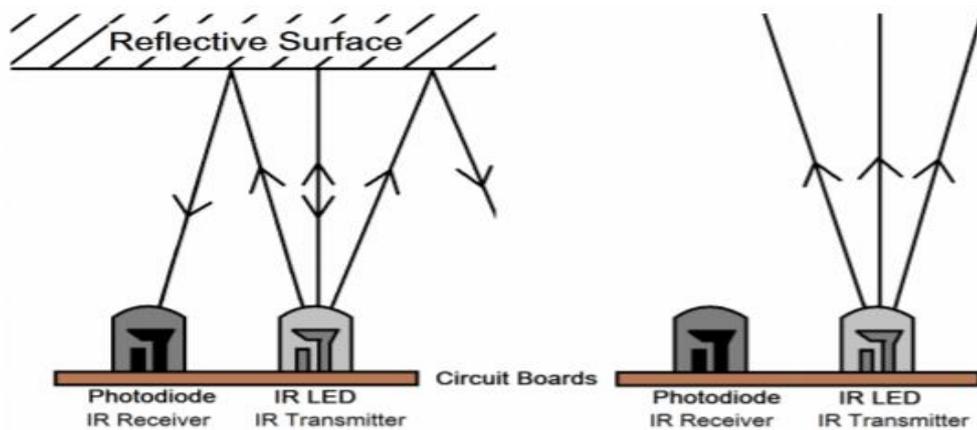
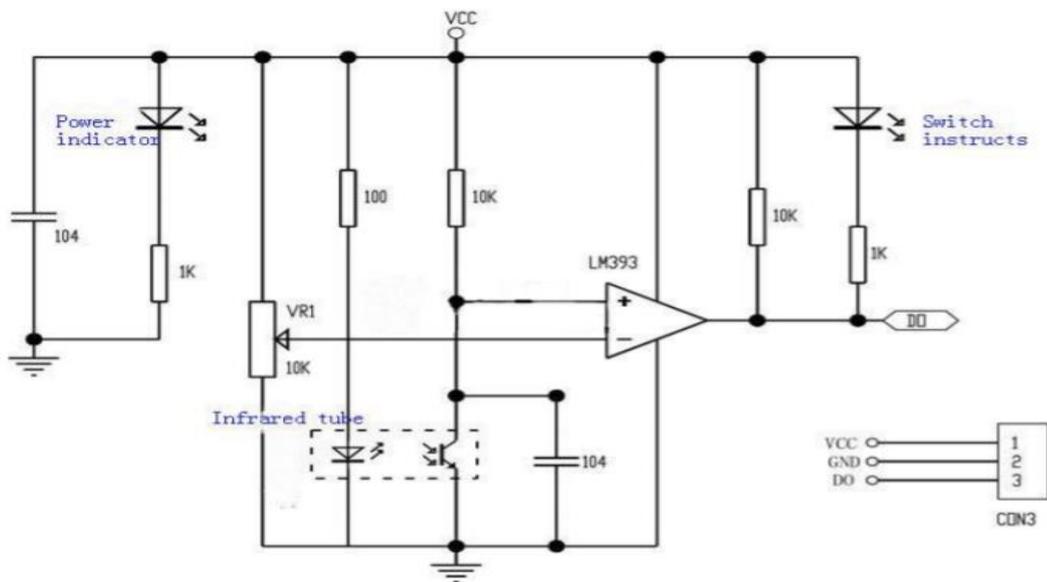
- **Synchronization between DTE and DCE:**  
When DCE powers on with autobauding enabled, it is recommended to send "AT" or "at" or "aT" or "At" to synchronize the baud rate, until DTE receives the "OK" response, which means DTE and DCE are correctly synchronized. For more information please refer to AT command "AT+IPR".
- **Restrictions of autobauding operation:**  
The DTE serial port must be set at 8 data bits, no parity and 1 stop bit.  
The URC such as "RDY", "+CFUN: 1" and "+CPIN: READY" will not be reported.

*Note: User can use AT command "AT+IPR=x" to set a fixed baud rate and the setting will be saved to non-volatile flash memory automatically. After the configuration is set as fixed baud rate, the URC such as "RDY", "+CFUN: 1" and "+CPIN: READY" will be reported when SIM800L is powered on.*

ANEXO D  
HOJA DE DATOS DEL MÓDULO FC-51.



Pin map del sensore FC-51.



Principio di funzionamento di un sensore IR con e senza un oggetto in prossimità.

## Características

- 

Voltaje de funcionamiento 3.3V~5V

Ángulo de detección 35°

Terminales de conexión

- · OUT: Señal del medio
- · GND: Tierra
- · VCC: Alimentación

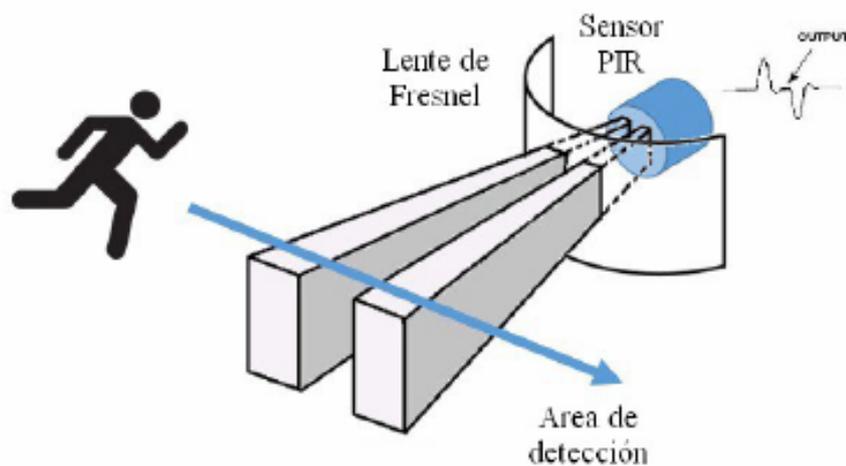
ANEXO E  
HOJA DE DATOS DEL MOLUDO PIR

### El sensor PIR infrarrojo:

1 En los sensores de movimiento, el sensor PIR consta en realidad de 2 elementos detectores separados, siendo la señal diferencial entre ambos la que permite activar la alarma de movimiento. En el caso del HC-SR501, la señal generada por el sensor ingresa al circuito integrado BISS0001, el cual contiene amplificadores operacionales e interfaces electrónicas adicionales.

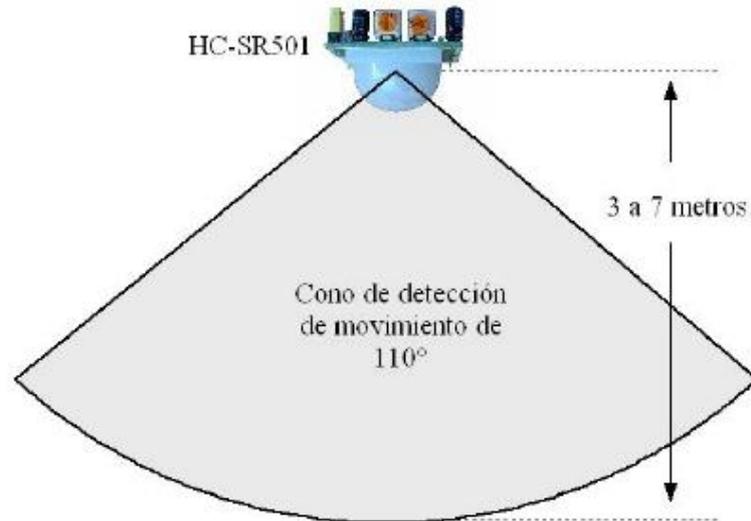
Las funciones y ajustes complementarios del sensor de movimiento son:

- Ajuste de parámetros: mediante 2 potenciómetros, el usuario puede modificar tanto la sensibilidad como la distancia de detección del PIR.
- Detección automática de luz (esta función no está disponible al adquirir el sensor de fábrica): por medio de una foto resistencia CdS (Sulfuro de Cadmio), se deshabilita la operación del sensor en caso que exista suficiente luz visible en el área. Esta función es utilizada en caso de sensores que enciendan lámparas en lugares poco iluminados durante la noche, y especialmente en corredores ó escaleras.



### Rango de detección de los sensores PIR:

Como se indicó anteriormente, el rango de detección de movimiento de los PIR es ajustable y generalmente funcionan con alcances de hasta 7 metros, y con aperturas de 90° a 110°, como se muestra en la figura. El montaje del PIR puede realizarse tanto en piso, muro ó techo, según convenga a la aplicación.



## 2. Descripción del HC-SR501

El módulo PIR modelo HC-SR501 es de bajo costo, pequeño, e incorpora la tecnología más reciente en sensores de movimiento. El sensor utiliza 2 potenciómetros y un jumper que permiten modificar sus parámetros y adaptarlo a las necesidades de la aplicación: sensibilidad de detección, tiempo de activación, y respuesta ante detecciones repetitivas.

Sus especificaciones técnicas son:

- Usa el PIR LHI778 y el controlador BISS0001
- Voltaje de alimentación: de 5 a 12 VDC
- Consumo promedio: <1 miliampere
- Rango de distancia de 3 a 7 metros ajustable.
- Angulo de detección: cono de 110°
- Ajustes: 2 potenciómetros para ajuste de rango de detección y tiempo de alarma activa.
- Jumper para configurar la salida de alarma en modo mono-disparo ó disparo repetitivo ('retriggerable')
- Salida de alarma de movimiento con ajuste de tiempo entre 3 segundos a 5 minutos.
- Salida de alarma activa Vo con nivel alto de 3.3 volts y 5 ma source, lista para conexión de un led, ó un transistor y relevador.
- Tiempo de inicialización: después de alimentar el módulo HC-SR05, debe transcurrir 1 minuto antes de que inicie su operación normal. Durante ese tiempo, es posible que el módulo active 2 ó 3 veces su salida.
- Tiempo de salida inactiva: cada vez que la salida pase de activa a inactiva, permanecerá en ese estado los siguientes 3 segundos. Cualquier evento que ocurra durante ese lapso es ignorado.
- Temperatura de operación: -15° a +70° C.
- Dimensiones: 3.2 x 2.4 x 1.8 cms

ANEXO F  
CÓDIGO DE PROGRAMACIÓN PARA EL FUNCIONAMIENTO DEL MÓDULO  
GSM SIM 800L Y LOS SENSORES DEL PROTOTIPO

```
// Trabajo de suficiencia profesional UNTELS 2019
//PROYECTO PROTOTIPO DE MONITOREO DE VIVIENDAS RURALES
```

```
#include <SoftwareSerial.h>
#include <EEPROM.h>
#define espacioEEPROM 1000
SoftwareSerial serialGSM(10, 11);
bool llegoSMS = false;
bool llamada = false;
String tlfSMS;
String tlfLlamada;
String telefonoCliente = "932333942";
String telefonoSMSCliente = "932333942";
String telefonoSMSCliente51 = "+51932333942";
String telf1Usuario;
String HoraFechaSMS;
String mensajeSMS;
String comandoGSM = "";
```

```
int IR=3;
bool senIR=LOW;
int PIR=5;
bool senPIR=LOW;
```

```
bool flanRF=HIGH;
bool flanPIR=HIGH;
```

```
void setup()
{
  Serial.begin(9600);
  serialGSM.begin(9600);
  telf1Usuario = EEPROM_LECTURA(0);

  pinMode(IR, INPUT);
  pinMode(PIR, INPUT);
  cnfGSM();
}
```

```

void loop()
{
  leerGSM();
  if (comandoGSM != "")
  {
    Serial.println(comandoGSM);
    comandoGSM = "";
  }
  mensajesSMS.trim();
  llegoSMS = false;

  //SENSOR();

  LlamadEntran();
  SMSentrante();

  tlfSMS = "";
  HoraFechaSMS = "";
  mensajesSMS = "";
}

void EEPROM_ESCRITURA(int posicion, String value)
{
  int nL;

  for (nL=0; nL < value.length(); nL++)
  {
    EEPROM.write(posicion + nL, value.charAt(nL) );

    if (nL >= 1000)
    {
      break;
    }
  }

  if (nL < 1000)
  {
    EEPROM.write(posicion + nL, (char)0 );
  }
}

```

```

String EEPROM_LLECTURA(int posicion)
{
String Num_Almacenado = "";
char Lee_Byte;
int Lee_posicion = posicion;

do {
Lee_Byte = EEPROM.read(Lee_posicion);
if(Lee_Byte != (char)0)
{
Num_Almacenado += Lee_Byte;
}
Lee_posicion++;
} while ( (Lee_Byte != (char)0) && (Lee_posicion < (posicion + 1000)) );

return Num_Almacenado;
}

void leerGSM()
{
static String textoEnt = "";
static unsigned long tmp1 = 0;
static int count=0;
static unsigned char buffer[64];
if (serialGSM.available())
{
while(serialGSM.available())
{
buffer[count++] = serialGSM.read();
if(count == 64)break;
}
textoEnt += (char*)buffer;
tmp1 = millis();
for (int i=0; i<count; i++)
{
buffer[i]=NULL;
}
count = 0;
}
}

if ( ((millis() - tmp1) > 100) && textoEnt != "" )
{
if ( textoEnt.substring(10,16) == "+CLIP:" )
{
tlfLlamada=textoEnt.substring(18,27);
}
if ( textoEnt.substring(2,7) == "+CMT:" )
{
llegoSMS = true;
}
}
}

```

```

if (llegoSMS)
{
    byte linea = 0;
    byte citas = 0;
    int nLi;
    for (int nLi=1; nLi < textoEnt.length(); nLi++)
    {
        if (textoEnt.charAt(nLi) == '')
        {
            citas++;
            continue;
        }
        if ( (linea == 1) && (citas == 1) )
        {
            tlfSMS += textoEnt.charAt(nLi);
        }
        if ( (linea == 1) && (citas == 5) )
        {
            HoraFechaSMS += textoEnt.charAt(nLi);
        }
        if ( linea == 2 )
        {
            mensajeSMS += textoEnt.charAt(nLi);
        }
        if (textoEnt.substring(nLi - 1, nLi + 1) == "\r\n")
        {
            linea++;
        }

        if (tlfSMS!="" && HoraFechaSMS!="" && mensajeSMS!="" )
        {
            Serial.print("tienes un nuevo mensaje de:");
            Serial.println(tlfSMS);
            Serial.print("a las: ");
            Serial.println(HoraFechaSMS );
            Serial.print("SMS: ");
            Serial.println(mensajeSMS);
        }
    }
    else
    {
        comandoGSM = textoEnt;
    }
    textoEnt = "";
}
}

```

```

void cnfGSM()
{
  serialGSM.print("AT+CMGF=1\n;AT+CNMI=2,2,0,0,0\n;ATX4\n;AT+COLP=1\n;AT+CLIP=1\n");
}

void LlamadEntran()
{
  if (tlfLlamada == telefonoCliente || tlfLlamada==telf1Usuario.substring(3,12))
  {
    serialGSM.print ("ATA\n");
    Serial.println("LLAMADA ENLAZADA ");
    tlfLlamada= "";
  }
  if (tlfLlamada!=" " && tlfLlamada !=telefonoCliente || tlfLlamada!=" " && tlfLlamada!=telf1Usuario.substring(3,12))
  {
    serialGSM.print ("ATH\n");
    Serial.println("NUMERO DESCONOCIDO ");
    tlfLlamada= "";
  }
}

void SENSOR()
{
  senIR=digitalRead(IR);
  senPIR=digitalRead(PIR);

  if (senIR == HIGH && flanRF == LOW)
  {
    enviaSMS(telefonosMScliente,"PELIGRO!! alguien abrio la puerta de ACASH");

    if(telf1Usuario.substring(0,3)=="+51")
    {
      String NumAlt=telf1Usuario.substring(3,12);
      delay(4000);
      enviaSMS(NumAlt,"PELIGRO!! alguien abrio la puerta de ACASH");
      Serial.println("entro el mensaje para telf alternativo");
    }
  }

  flanRF = senIR;

  if (senPIR == HIGH && flanPIR == LOW)
  {
    enviaSMS(telefonosMScliente,"Alguien esta dentro de la casa de ACASH");
    delay(500);
  }
  flanPIR = senPIR;
}

```

```

void SMSentrante()
{
  if (tlfSMS!=" " && tlfSMS==telefonosMScliente51 || tlfSMS!=" " && tlfSMS==telf1Usuario)
  {
    Serial.print("entro el mensaje de:");
    Serial.println(tlfSMS);
    Serial.print("numero registrado:");
    Serial.print(telf1Usuario);
    Serial.println(".....");

    if(mensajeSMS.substring(0,5)=="*#*1@")
    {
      EEPROM_ESCRITURA(0, mensajeSMS.substring(5,17));
      telf1Usuario = EEPROM_LECTURA(0);
      enviaSMS(telefonosMScliente,"El nuevo numero ingresado es:" + telf1Usuario);
    }
    if (mensajeSMS=="puerta"||mensajeSMS=="Puerta"||mensajeSMS=="PUERTA")
    {
      if (senIR==LOW)
      {
        enviaSMS(tlfSMS,"La puerta esta CERRADA");
      }
      if (senIR==HIGH)
      {
        enviaSMS(tlfSMS,"La puerta esta ABIERTA");
      }
    }

    if (mensajeSMS=="Informe"||mensajeSMS=="INFORME"||mensajeSMS=="informe")
    {
      enviaSMS(tlfSMS,"comunicarse con el técnico, llamando al 932999999");
    }
    if (mensajeSMS=="Numero1"||mensajeSMS=="numero1" ||mensajeSMS=="NUMERO1")
    {
      enviaSMS(tlfSMS,"El nuevo numero guardado es:" + telf1Usuario);
    }
    if (mensajeSMS=="R1"||mensajeSMS=="r1")
    {
      digitalWrite(Relay,HIGH);
    }
    if (mensajeSMS=="R0"||mensajeSMS=="r0")
    {
      digitalWrite(Relay,LOW);
    }
  }
}

void enviaSMS(String telefono, String Mensaje)
{
  serialGSM.print("AT+CMGS=\"\" + telefono + "\"\n");
  serialGSM.print(Mensaje + "\n");
  serialGSM.print((char)26);
}

```

ANEXO G  
DESARROLLO DEL PROTOTIPO Y DE LA MAQUETA PARA PRUEBAS





