



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
ESCUELA DE INGENIERÍA EN ELECTRÓNICA, TELECOMUNICACIONES
Y REDES

IMPLEMENTACIÓN DE UN PROTOTIPO DE UNA RED AUXILIAR DE
COMUNICACIÓN POR RADIOFRECUENCIA CON NODOS BASADOS EN
DISPOSITIVOS ELECTRÓNICOS Y EQUIPOS MÓVILES PARA
MENSAJERÍA INSTANTÁNEA

TRABAJO DE TITULACIÓN
Tipo: DISPOSITIVO TECNOLÓGICO

Para optar al Grado Académico de:

INGENIERA EN ELECTRÓNICA TELECOMUNICACIONES Y REDES

AUTORA: MAYRA ESTEFANIA CARRILLO MERCHAN

TUTOR: ING. JOSÉ ENRIQUE GUERRA SALAZAR.

Riobamba-Ecuador

2018

©2018, Mayra Estefanía Carrillo Merchán

Se autoriza la reproducción total o parcial de este documento, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica, siempre y cuando se reconozca el Derecho de los Autores.

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
ESCUELA DE INGENIERÍA EN ELECTRÓNICA, TELECOMUNICACIONES Y
REDES

El Tribunal del Trabajo de Titulación certifica: “IMPLEMENTACIÓN DE UN PROTOTIPO DE UNA RED AUXILIAR DE COMUNICACIÓN POR RADIOFRECUENCIA CON NODOS BASADOS EN DISPOSITIVOS ELECTRÓNICOS Y EQUIPOS MÓVILES PARA MENSAJERÍA INSTANTÁNEA”, de responsabilidad de Mayra Estefanía Carrillo Merchán ha sido minuciosamente revisado por los miembros del Tribunal del Trabajo de Titulación, quedando autorizada su presentación.

NOMBRE	FIRMA	FECHA
Dr. Julio Santillán VICEDECANO FACULTAD DE INFORMÁTICA Y ELECTRÓNICA	_____	_____
Ing. Franklin Moreno DIRECTOR DE ESCUELA DE INGENIERÍA ELECTRÓNICA EN TELECOMUNICACIONES Y REDES	_____	_____
Ing. José Guerra DIRECTOR DEL TRABAJO DE TITULACIÓN	_____	_____
Ing. Wilson Zúñiga MIEMBRO DEL TRIBUNAL	_____	_____

Yo, Mayra Estefania Carrillo Merchan soy responsable de las ideas, doctrinas y resultados expuestos en esta Tesis y el patrimonio intelectual de la Tesis de Grado pertenece a la Escuela Superior Politécnica de Chimborazo.

Mayra Estefania Carrillo Merchan.

DEDICATORIA

Esta tesis la dedico a mis padres, Ángel Jacinto y Olga Piedad por haberme apoyado en todo momento, por sus consejos, sus valores, por la motivación constante que me ha permitido ser una persona de bien y alcanzar esta meta tan anhelada. A mi esposo Juan Carlos por sus palabras y confianza, por su apoyo incondicional en los buenos y malos momentos. A mi hermano Brayan por estar presente y alegrándome cada día.

A mi querido hijo Iker Matteo por ser mi fuente de motivación e inspiración para cada día poder superarme.

Mayra

AGRADECIMIENTO

Agradezco a Dios por haberme acompañado y guiado lo largo de mi carrera y cumplir con esta meta tan importante en mi vida.

A mis padres por su esfuerzo y apoyo, siendo siempre mi impulso y mi mayor orgullo, por haberme dado la oportunidad de tener una excelente educación en el transcurso de mi vida.

A mi esposo por acompañarme en este largo camino con sus palabras de aliento y comprensión en el transcurso de mi carrera.

A mi familia, que de una u otra forma estuvieron siempre presentes y dándome esas palabras de aliento cuando más lo necesitaba.

Un agradecimiento especial al Msc. José Guerra, por la paciencia, confianza y el apoyo brindado para el desarrollo del presente trabajo de titulación.

Mayra

ÍNDICE DE CONTENIDO

ÍNDICE DE TABLAS.....	x
ÍNDICE DE FIGURAS.....	xi
ÍNDICE DE ANEXOS.....	xiii
ÍNDICE DE ABREVIATURAS.....	xiv
RESUMEN.....	xvi
SUMMARY.....	xvii
INTRODUCCIÓN.....	1
CAPÍTULO I	
1 MARCO TEÓRICO	3
1.1 Uso de la tecnología en el terremoto de Ecuador	3
1.2 Infraestructura de las comunicaciones en el terremoto de Ecuador.	3
1.3 Dispositivos móviles	4
1.4 Sistema de comunicaciones	5
1.5 Modos de transmisión	6
1.6 Medios para la transmisión de datos	8
1.7 Comunicaciones Inalámbricas	9
1.8 Tecnologías Inalámbricas	10
1.8.1 <i>Tecnología por Radio Frecuencia (RF)</i>	10
1.8.2 <i>Tecnología Zigbee</i>	10
1.8.3 <i>Tecnología Bluetooth</i>	11
1.8.4 <i>Tecnología Wi-fi</i>	11
1.8.5 <i>Comparación de tecnologías inalámbricas</i>	11
1.9 Tarjetas de desarrollo	12
1.9.1 <i>Tarjeta de desarrollo Arduino</i>	13
1.9.2 <i>Tarjeta de desarrollo Raspberry PI</i>	13
1.9.3 <i>Tarjeta de desarrollo Galileo</i>	14
1.9.4 <i>Comparación entre los diferentes tipos de tarjetas de desarrollo</i>	15

1.9.5	<i>Android Studio</i>	17
-------	-----------------------------	----

CAPÍTULO II

2	IMPLEMENTACIÓN DEL <i>HARDWARE</i> Y <i>SOFTWARE</i> DEL PROTOTIPO ..	18
2.1	Diagrama general de la red de comunicación	18
2.2	Requerimientos de la red de comunicación.	19
2.3	Requerimientos de <i>Hardware</i> de la red de comunicación	19
2.3.1	<i>Diagrama de bloques del dispositivo electrónico</i>	19
2.3.2	<i>Características de los dispositivos seleccionados</i>	20
2.3.2.1	<i>Arduino Pro micro</i>	20
2.3.2.2	<i>Módulo NRF 24L01</i>	21
2.3.3	<i>Cables de conexión</i>	22
2.3.4	<i>Esquema de conexión del dispositivo electrónico.</i>	23
2.4	Requerimientos del <i>Software</i> para la implementación del prototipo	24
2.4.1	<i>Herramientas de Software</i>	24
2.4.2	<i>Diseño de la aplicación móvil</i>	25
2.4.3	<i>Desarrollo de la programación.</i>	26
2.4.4	<i>Librerías utilizadas en el diseño de la aplicación</i>	26
2.4.5	<i>Funcionamiento de la aplicación móvil.</i>	28
2.4.6	<i>Arduino</i>	30
2.4.7	<i>Diseño de la cobertura del dispositivo</i>	32

CAPÍTULO III

3	PRUEBAS Y RESULTADOS DEL SISTEMA	34
3.1	Pruebas del <i>Hardware</i> implementado	34
3.1.1	<i>Distancia de transmisión de datos sin obstáculos.</i>	35
3.1.2	<i>Distancia de transmisión de datos con obstáculos.</i>	37
3.1.3	<i>Tiempo de transmisión de datos sin obstáculos</i>	39
3.1.4	<i>Tiempo de transmisión de datos con obstáculos.</i>	40
3.1.5	<i>Número máximo de caracteres enviados a diferentes distancias sin obstáculos.</i>	41
3.1.6	<i>Número máximo de caracteres enviados a diferentes distancias con obstáculos</i>	43

3.1.7	<i>Repetitividad del envío de mensajes</i>	44
3.1.8	<i>Consumo de energía teórico del dispositivo implementado</i>	46
3.2	Pruebas del software del prototipo implementado	48
3.2.1	<i>Aplicación Móvil</i>	48
3.2.2	<i>Evaluación del funcionamiento de la aplicación en diferentes dispositivos móviles</i>	50
3.3	Prototipo de la red de comunicación implementado.	51
3.4	Presupuesto económico del prototipo	51
	CONCLUSIONES	53
	RECOMENDACIONES	55
	BIBLIOGRAFÍA	
	ANEXOS	

ÍNDICE DE TABLAS

Tabla 1-1:	Tabla comparativa de tecnologías inalámbricas.....	12
Tabla 2-1:	Tabla de comparación de los diferentes tipos de tarjetas de desarrollo.....	15
Tabla 3-1:	Diferencias de los tipos de Arduino	16
Tabla 4-1:	Requerimientos para la instalación de Android.....	17
Tabla 1-2:	Diagrama de conexión del Hardware	23
Tabla 1-3:	Distancia de cobertura.....	36
Tabla 2-3:	Distancias en intervalos de 5m.....	38
Tabla 3-3:	Mediciones de tiempo de la transmisión de datos	39
Tabla 4-3:	Mediciones de tiempo de la transmisión de datos	40
Tabla 5-3:	Número de caracteres enviados	42
Tabla 6-3:	Envío de diferentes números de caracteres.....	43
Tabla 7-3:	Envío de caracteres.....	45
Tabla 8-3:	Consumo de energía en Transmisión de mensajes.....	46
Tabla 9-3:	Consumo de energía en la Recepción de mensajes.....	46
Tabla 10-3:	Características de teléfonos móviles.....	50
Tabla 11-3:	Presupuesto implementado.....	51

ÍNDICE DE FIGURAS

Figura 1-1:	Estado de servicio a) Servicio a 48 horas b) Servicio 1 mes después	4
Figura 2-1:	Diagrama de un sistema de comunicación	6
Figura 3-1:	Operación simplex, semidúplex, Full dúplex.....	7
Figura 4-1:	Tipos de medios de transmisión.....	8
Figura 5-1:	Comunicación Inalámbrica	9
Figura 6-1:	Placa Raspberry PI.....	14
Figura 7-1:	Tarjeta de desarrollo Galileo.....	14
Figura 1-2:	Diagrama general de la red de comunicación	18
Figura 2-2:	Diagrama de bloques del dispositivo.	20
Figura 3-2:	Estructura física de Arduino Pro Micro	21
Figura 4-2:	Módulo NRF24L01.....	21
Figura 5-2:	Regulador de Voltaje	22
Figura 6-2:	Cable y adaptador de conexión.	22
Figura 7-2:	Esquema de conexión de dispositivo electrónico.....	23
Figura 8-2:	Diagrama de Flujo de la aplicación.....	25
Figura 9-2:	a) Fragmento de la programación en java. b) Programación visual xml.....	26
Figura 10-2:	Diagrama de Flujo del funcionamiento de la aplicación móvil	29
Figura 11-2:	Diagrama de Flujo de la programación en Arduino.....	30
Figura 12-2:	Fragmento del código en Arduino a) Transmisor b) Receptor.....	31
Figura 13-2:	Segmento frontal y posterior del prototipo.	32
Figura 14-2:	Segmento portador de la antena	32
Figura 15-2:	Cubierta del prototipo	33
Figura 1-3:	Módulos implementados.	34
Figura 2-3:	Aplicación móvil Fields Area Measure PRO.....	35
Figura 3-3:	Caracteres enviados a diferentes distancias	36
Figura 4-3:	Caracteres enviados	38
Figura 5-3:	Pruebas de funcionamiento del Dispositivo sin obstáculos.	40
Figura 6-3:	Transmisión de datos con obstáculos.....	41
Figura 7-3:	Contabilización de caracteres.....	41
Figura 8-3:	Número máximo de caracteres.....	43
Figura 9-3:	Similitud de datos recibidos en los dispositivos móviles.....	44
Figura 10-3:	Proceso de prueba de la batería del Dispositivo móvil sin utilizar.	47
Figura 11-3:	Proceso de prueba de la batería del Dispositivo móvil con uso regular.....	48
Figura 12-3:	Icono de la aplicación móvil	49

Figura 13-3:	Android a) Ingreso del ID. b) Rx y Tx de mensajes	49
Figura 14-3:	Red de comunicación implementada.	51

ÍNDICE DE ANEXOS

ANEXO A: DATASHEET ARDUINO PRO MICRO

ANEXO B: DATASHEET MÓDULO NRF24L01

ANEXO C: DISEÑO DE LA PLACA

ANEXO D: DISEÑO DE LA CUBIERTA DEL DISPOSITIVO

ANEXO E: CÓDIGO ARDUINO

ANEXO F: CÓDIGO ARDUINO RECEPTOR

ANEXO G: CÓDIGO ANDROID STUDIO

ANEXO H: INTERFAZ GRÁFICA DE LA APLICACIÓN

ANEXO I: APLICACIÓN DE MEDICIÓN DE DISTANCIA

ANEXO J: PANTALLAS DE COMPROBACION DE ENVIO DE MENSAJES

ÍNDICE DE ABREVIATURAS

ESPOCH	Escuela Superior Politécnica de Chimborazo
HF	High Frequency
IEEE	Instituto de Ingenieros Eléctricos y Electrónicos
RF	Radio Frecuencia
SDK	Software Development Kit (kit de desarrollo de software)
WPAN	Red de Área Personal
ISM	Industrial, Scientific and Medical
Tx	Transmisión
Rx	Recepción
SX	Simplex
Hdx	Semi duplex
Fdx	Full duplex
F/Fdx	Duplex total
USB	Universal Serial Bus (Bus Universal en Serie).
WIFI	Wireless Fidelity (Fidelidad inalámbrica).
GND	Ground (tierra)
VCC	Voltaje de corriente continua
SDA	Serial Data Input/output (Interfaz serial para la entrada/salida de datos).
SCK	Serial Clock
MOSI	Master Out Slave
MISO	Master In Slave Out
CSN	Chip select in
CE	Chip enable in
SPI	Serial Peripheral Interface
JPEG	Joint Photographic Experts Group (Grupo Conjunto de Expertos en Fotografía).
XML	Extensible markup language (Lenguaje de Marcado Extensible).
XSL	Extensible Stylesheet Language (Lenguaje extensible de hoja de estilo).
IDE	Entorno de desarrollo integrado
URL	Uniform Resource Locator (Localizador Uniforme de Recursos).
Km	Kilometro.
Seg.	Segundo.
C	Velocidad de la luz.
HF	High Frequency

VHF	Very High Frequency
UHF	Ultra High Frequency
GSM	Global System for Mobile
CDMA	Código de División de Acceso Múltiple
PAN	Personal Area Network
NFC	Near Field Communication
MHz	Megahercio
GHz	Gigahercio
PC	Computadora personal
BT	Bluetooth
GPU	Graphics Processing Unit
RAM	Random Access Memory
SD	Secure Digital
SW	Software
GUI	Graphical user interface
Kbps	Kilobit por segundo.
Mbps	Megabit por segundo
OTG	On the go
ADC	Analog-to-Digital Converter
PWM	Pulse-width modulation
V	Voltage.
XD	Execute Disable.
GNU	Sistema Operativo tipo Unix

RESUMEN

La presente investigación, consiste en diseñar e implementar un prototipo de una red auxiliar de comunicación por radiofrecuencia con nodos basados en dispositivos electrónicos y equipos móviles para mensajería instantánea, el prototipo trabaja con Arduino Pro Micro y con el módulo NRF24L01, para la comunicación inalámbrica por radiofrecuencia, se realiza una aplicación de mensajería instantánea desarrollada en Android Studio para dispositivos móviles, la cual es de fácil manejo para cualquier tipo de usuario para el envío y recepción de mensajes de ayuda en texto. El funcionamiento de la red auxiliar de comunicación inalámbrica no requiere de infraestructura adicional para realizar la comunicación debido a que el dispositivo creado transmite y recibe los datos por radio frecuencia. De las pruebas realizadas se determinó que la comunicación entre dispositivos alcanza aproximadamente es de 238 m en línea de vista, mientras que con obstáculos es aproximadamente de 18 m, con un tiempo de transmisión y recepción de un segundo independiente de la distancia. Se puede concluir que el prototipo implementado es un dispositivo de bajo consumo, que se alimenta desde el dispositivo móvil que puede convertirse en una herramienta de ayuda principalmente en desastres naturales o provocados, que hayan generado colapso de las redes de comunicación. Se recomienda que al mejorar el prototipo se aplique transmisión y recepción de voz, datos e imágenes, además que efectúen estudios para incorporar en los dispositivos móviles la transmisión de datos por radio frecuencia.

Palabras Claves: <TECNOLOGÍA Y CIENCIAS DE LA INGENIERÍA>, <INGENIERÍA EN TELECOMUNICACIONES>, <ELECTRÓNICA>, <COMUNICACIÓN INALÁMBRICA>, <RADIOFRECUENCIA>, <ARDUINO>, <ANDROID STUDIO>.

SUMMARY

The present investigation consists of designing and implementing a prototype of an auxiliary network of communication by radiofrequency with nodes based on electronic devices and mobile equipment for instant messaging, the prototype works with Arduino Pro Micro and the NRF24L01 module, for wireless communication by radio frequency. An instant messaging application performed in Android Studio for mobile devices made, it is easy to use for any type of user to send and receive text help messages. The operation of the auxiliary network of wireless communication does not require additional infrastructure to perform the communication, due the device created transmits and receives the data by radio frequency. The tests carried out, determined the communication between devices reaches approximately 238 m. in a line of sight, while with obstacles it is approximately of 18m, with a transmission time and reception of a second independent of the distance. It concluded the implemented prototype is a low consumption device, it fed on since the mobile device that can become a helpful tool mainly in natural or provoked disasters, that they have generated collapse of communication networks. The research paper recommends that when improving the prototype transmission and reception of voice, data, and images apply, besides, to perform studies to incorporate the data transmission by radio frequency in mobile devices.

Keywords: <TECHNOLOGY AND ENGINEERING SCIENCES>, <ENGINEERING IN TELECOMMUNICATIONS>, <ELECTRONICS>, <WIRELESS COMMUNICATION>, <RADIO FREQUENCY>, <ARDUINO>, <ANDROID STUDIO>.

INTRODUCCIÓN

Debido a las últimas actividades sísmicas que se han presentado en el Ecuador y en los diferentes países, según la Organización de Naciones Unidas, en los últimos 20 años, casi tres millones de víctimas y ochocientos millones de personas han sido afectados adversamente en el mundo por causas de los desastres naturales. Desde 1976 – 1990, los terremotos fueron los responsables del casi 60 % de las muertes producidas. (ONU, 2010, <http://www.un.org/spanish/News/story.asp?newsID=17554&criteria1=ISDR#.WcwuH1v9TIU>)

El 16 de Abril del 2016, se registra en la costa ecuatoriana, un sismo de 7.8 en la escala de Richter en cantón Pedernales, de la provincia de Manabí, dejando un total de 602 víctimas fatales, 130 personas desaparecidas y más de 25.000 damnificados, provocando destrucción de viviendas y vías de acceso, averías en los servicios básicos, colapso de infraestructura telefónicas y de energía eléctrica. Su magnitud fue tal que afecto algunas ciudades del país. En cuanto al sistema de comunicación, una sola radio con cobertura provincial se mantuvo en funcionamiento reportando la situación, La madrugada el 17 de abril, La empresa de Telecomunicaciones “Movistar Ecuador” realiza acciones para reparar la red afectada, movilizandoo 190 técnicos, y 40 vehículos que disponían de teléfonos satelitales para que la ciudadanía pudiera comunicarse de forma gratuita, así como para recargar las baterías de sus dispositivos. (Encontexto, 2016, pp. 29-32).

Por otra parte las redes móviles, telecomunicaciones y grandes tecnologías los dispositivos han ido transformando al mundo, al punto que son de gran ayuda en aplicaciones militares, médicas, e inclusive en el hogar. Pero cuando estas colapsan con la constante transformación geográfica, cambios atmosféricos, constantes movimientos telúricos e inclusive drásticas erupciones volcánicas nada pueden hacer para brindar la ayuda necesaria, debido a que no existen los equipos necesarios para lograr una comunicación sin la necesidad de redes físicas.

Debido a la baja ayuda tecnológica para minimizar el número de víctimas mortales en desastres naturales se hace necesario la implementación de un mecanismo que permita la comunicación entre las personas afectadas con los equipos de rescate, en lo cual sería necesario disponer de plataformas con mensajería instantánea que ayude a la población en caso de pérdida de infraestructura de comunicaciones por desastres naturales o colapso de la red principal.

El presente trabajo de titulación se desarrollará un prototipo de una red auxiliar de comunicación por radiofrecuencia con nodos basados en dispositivos electrónicos y equipos móviles para mensajería instantánea, la cual se convertirá en una herramienta de ayuda a la población en caso de pérdida de infraestructura de comunicaciones por eventos naturales o por colapso de la red principal, la misma que será de bajo costo.

Además utilizaremos dispositivos móviles y electrónicos con aplicaciones Android para mensajería instantánea, lo que nos permitirá desarrollar nuestro prototipo de red sin la necesidad de una infraestructura. En donde la comunicación será realizada por radio frecuencia para tener una comunicación eficaz en caso de desastres naturales o colapso de las redes.

El objetivo principal del presente trabajo es implementar un prototipo de una red auxiliar de comunicación por radiofrecuencia con nodos basados en dispositivos electrónicos y equipos móviles para mensajería instantánea. Como objetivos específicos se plantea para la investigación:

- Analizar los tipos de comunicaciones inalámbricas y su desarrollo en la actualidad.
- Investigar una aplicación de mensajería instantánea de software libre para equipos móviles, que contengan el sistema operativo Android.
- Implementar el dispositivo electrónico, mediante la conexión por radiofrecuencia hacia el otro punto de conexión móvil.
- Evaluar y comprobar el funcionamiento de la mensajería instantánea a través de la red de comunicación por radiofrecuencia.

El presente documento está formado por tres capítulos que son los siguientes:

- Primer capítulo consta el marco teórico en el cual se realiza una investigación del uso de las tecnologías en desastres naturales, los dispositivos móviles, tecnologías inalámbricas, tarjetas de desarrollo, los cuales servirán para desarrollar del prototipo planteado.
- Segundo capítulo contiene los requerimientos, descripción de los dispositivos, y diseño tanto del *hardware* y *software* para realizar la implementación.
- En el tercer capítulo se obtiene las pruebas y resultados del sistema.

Y finalmente tenemos conclusiones y recomendaciones de la implementación del prototipo de una red auxiliar de comunicación por radiofrecuencia con nodos basados en dispositivos electrónicos y equipos móviles para mensajería instantánea.

CAPÍTULO I

1 MARCO TEÓRICO

En este capítulo se investigan la transmisión de datos y algunas características de las comunicaciones inalámbricas. Se revisa las tecnologías móviles y el Sistema operativo Android. Se revisa igualmente las características de la comunicación Serial, los módulos de conexión y las tarjetas de desarrollo más comunes.

1.1 Uso de la tecnología en el terremoto de Ecuador

El periodo de reacción fue corto mientras que la actuación para localizar, buscar, saber, sobre las personas desaparecidas fue inmediata. El desarrollo de la tecnología para reaccionar frente a la catástrofe, es ser el apoyo en la etapa de búsqueda de supervivientes, desaparecidos, geo localizar puntos críticos, movilización ciudadana para las donaciones y periodo de reconstrucción después del terremoto.

Las redes sociales se convirtieron en el principal instrumento de comunicación, donde Twitter se convirtió en una fuente importante de información. Los primeros actores que reaccionaron fueron las personas quienes daban las noticias, y los hashtags inmediatos donde se concentraba la conversación de lo sucedido. Facebook actuó pocos minutos después activando su función Safety Check que reportaba la situación de cientos de personas a través de los muros personales. Google también actuó rápido y activo su localizador de personas.

Las aplicaciones móviles han jugado un papel imprescindible en lo que se refiere a reportes de información para la movilización. (Martin, 2016, <https://www.eltelegrafo.com.ec/noticias/columnistas/1/el-uso-de-la-tecnologia-en-el-terremoto-de-ecuador>)

1.2 Infraestructura de las comunicaciones en el terremoto de Ecuador.

Posterior al terremoto se registró pérdida del servicio de telecomunicaciones en varias provincias, y congestión de datos 3G con un 30% de fallo de llamadas en las provincias afectadas. El cantón más afectado fue Jama (0 – 30%), seguido de. Sucre, Manta, Rocafuerte, Portoviejo, Olmedo y

Bolívar (31-60%), y Pedernales, Flavio Alfaro, San Vicente, Chone, Eloy Alfaro, Pichincha, Puerto López, Montecristi, Paján y 24 de Mayo (61- 90%). Al 10 de mayo la operatividad del servicio móvil avanzado (voz, texto, datos) es de 96%. CNT: 99%, Movistar: 98%, Claro: 94%. Los operadores trabajan con cuadrillas de alrededor de 600 técnicos. En las principales ciudades el servicio está restablecido al 100%. En los cantones Bolívar (83%) y Rocafuerte (71%) se lo está todavía mejorando. (Secretaría de Gestión de Riesgos, 2016). En la Figura 1-1 podemos observar el estado del servicio a 48 horas y el estado del servicio de 1 mes después

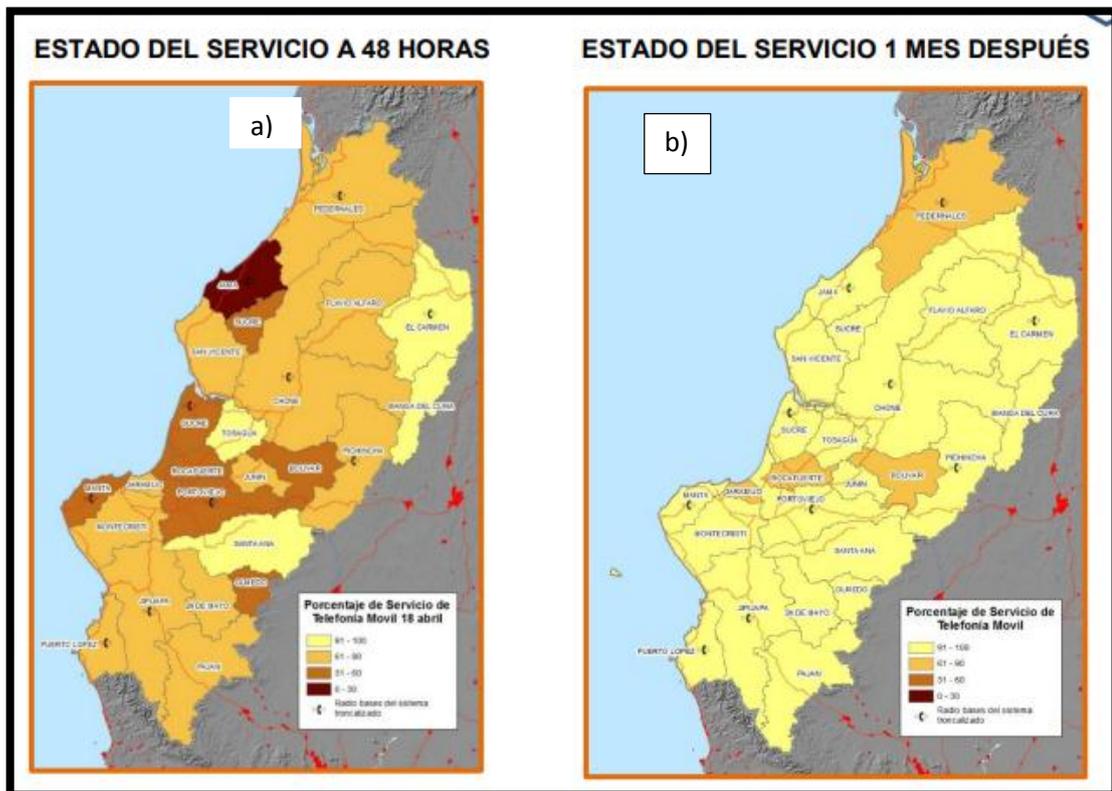


Figura 1-1: Estado de servicio a) Servicio a 48 horas b) Servicio 1 mes después

Fuente: (Secretaría de Gestión de Riesgos, 2016)

Los servicios tecnológicos implementados para la atención de emergencias, luego del terremoto de 7,8 grados en la zona costera del Ecuador, se reforzaron con el objetivo de direccionar la ayuda que requieren los afectados por este desastre natural. La red troncalizada de comunicación, articulada al Servicio Integrado de Seguridad ECU 911, es uno de los recursos fundamentales de los sistemas de coordinación que operan desde el pasado 16 de abril.

1.3 Dispositivos móviles

Son aparatos de tamaño pequeño que cuentan con características tales como:

- Capacidad especial de procesamiento.
- Conexión permanente o intermitente a una red.
- Memoria limitada.
- Diseños específicos para una función principal y versatilidad para el desarrollo de otras funciones.
- Tanto su posesión como su operación se asocia al uso individual de una persona, la cual puede configurarlos a su gusto.

Algunos de los ejemplos de estos dispositivos son los siguientes:

- Comunicadores de bolsillo
- Internet Screen Phones.
- Sistemas de navegación de automóviles.
- Sistemas de entretenimiento.
- Sistemas de televisión e Internet.
- Teléfonos móviles.
- Organizadores y asistentes personales digitales. (Soriano, 2010)

Teléfonos móviles.

Es un dispositivo inalámbrico electrónico que permite tener acceso a la red de telefonía celular o móvil. Se denomina celular debido a las antenas repetidoras que conforman la red, cada una de las cuales es una célula, si bien existen redes telefónicas móviles satelitales. Su principal característica es su portabilidad, que permite comunicarse desde casi cualquier lugar. Aunque su principal función es la comunicación de voz, como el teléfono convencional, su rápido desarrollo ha incorporado otras funciones como son cámara fotográfica, agenda, acceso a internet, reproducción de video e incluso GPS y reproductor mp3. (EcuRed, 2016, https://www.ecured.cu/Tel%C3%A9fono_celular).

1.4 Sistema de comunicaciones

Está formado por un emisor, un medio de transmisión y un receptor como se puede apreciar en la Figura 2-1.



Figura 2-1: Diagrama de un sistema de comunicación

Elaborado por: CARRILLO, Mayra 2018

Un transmisor es un conjunto de uno o más dispositivos o circuitos electrónicos que convierte la información de la fuente original en una señal que se presta más a su transmisión a través de determinado medio de transmisión. El medio de transmisión transporta las señales desde el transmisor hasta el receptor, y puede ser tan sencillo como un par de conductores de cobre que propaguen las señales en forma de flujo de corriente eléctrica. También se puede convertir la información a ondas electromagnéticas luminosas, propagarlas a través de cables de fibra óptica hechas de vidrio o de plástico, o bien se puede usar el espacio libre para transmitir ondas electromagnéticas de radio, a grandes distancias o sobre terreno donde sea difícil o costoso instalar un cable físico. Un receptor es un conjunto de dispositivos y circuitos electrónicos que acepta del medio de transmisión las señales transmitidas y las reconvierte a su forma original. (Tomasi, 2003, p. 2)

1.5 Modos de transmisión

Los sistemas electrónicos de comunicaciones se pueden diseñar para manejar la transmisión sólo en una dirección, en ambas direcciones, sólo en una a la vez, o en ambas direcciones al mismo tiempo. A éstos se les llama modos de transmisión. (Briceño, 2005, p. 155)

Hay cuatro modos de transmisión posibles: simplex, semidúplex, dúplex y dúplex/dúplex. En la Figura 3-1 se observa los diferentes modos de transmisión.

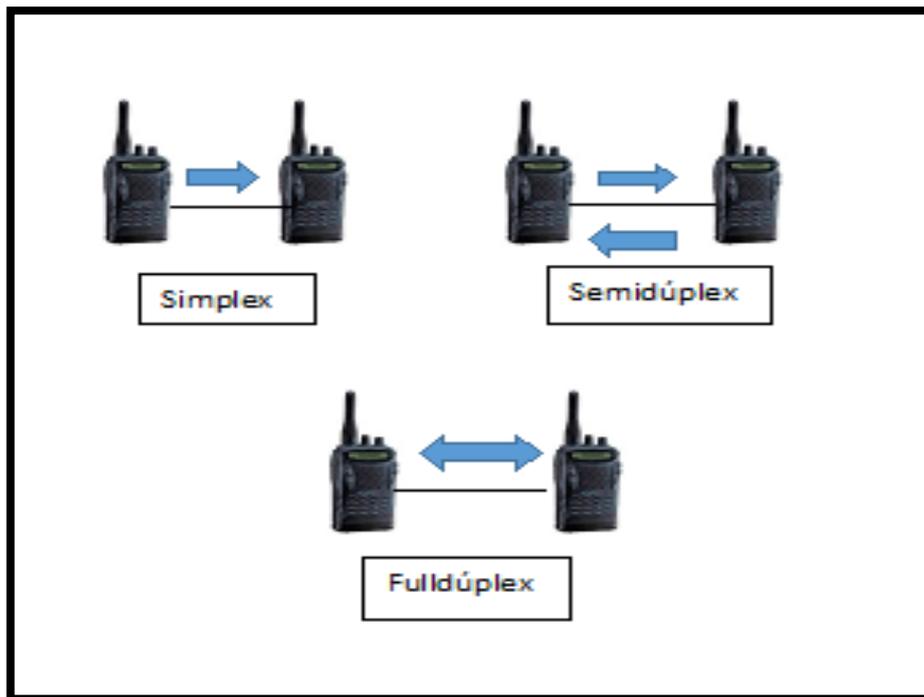


Figura 3-1: Operación simplex, semidúplex, Full dúplex

Elaborado por: CARRILLO, Mayra 2018

Simplex (SX).- Las transmisiones sólo se hacen en una dirección. A veces, a los sistemas simplex se les llama sólo en un sentido, sólo recibir o sólo transmitir. Una estación puede ser un transmisor o un receptor, pero no ambos a la vez.

Semidúplex (HDX, de half dúplex). - En el funcionamiento las transmisiones se pueden hacer en ambas direcciones, pero no al mismo tiempo. Una estación puede ser transmisora y receptora, pero no al mismo tiempo.

Dúplex (FDX, de full dúplex). - Puede haber transmisiones en ambas direcciones al mismo tiempo. Una estación puede transmitir y recibir en forma simultánea; sin embargo, la estación a la que se transmite también debe ser de la que se recibe.

Dúplex total/general (F/FDX, de full/full dúplex). - Es posible transmitir y recibir en forma simultánea, pero no necesariamente entre las mismas dos estaciones (es decir, una estación puede transmitir a una segunda estación, y recibir al mismo tiempo de una tercera estación). Las transmisiones dúplex total/general se usan casi exclusivamente en circuitos de comunicaciones de datos.

1.6 Medios para la transmisión de datos

Los avances tecnológicos han permitido aumentar progresivamente el aprovechamiento de los medios de transmisión existentes (par de cobre, coaxial, radio), así como el desarrollo de otros totalmente nuevos, como la fibra óptica. Todos los medios de transmisión tienen las siguientes características:

- Están basados en ondas electromagnéticas
- Transmiten a la velocidad de la luz en el medio considerado
- En guías de onda la velocidad disminuye en una tercera parte
- Presentan una atenuación proporcional a la distancia
- Están sujetos a interferencias y ruido
- Son limitados en el ancho de banda sobre el que pueden transmitir

Los medios de transmisión comúnmente utilizados en la transmisión de datos se pueden clasificar de la siguiente manera mostrada en la Figura 3-1. (Briceño, 2005, pp. 497-498)

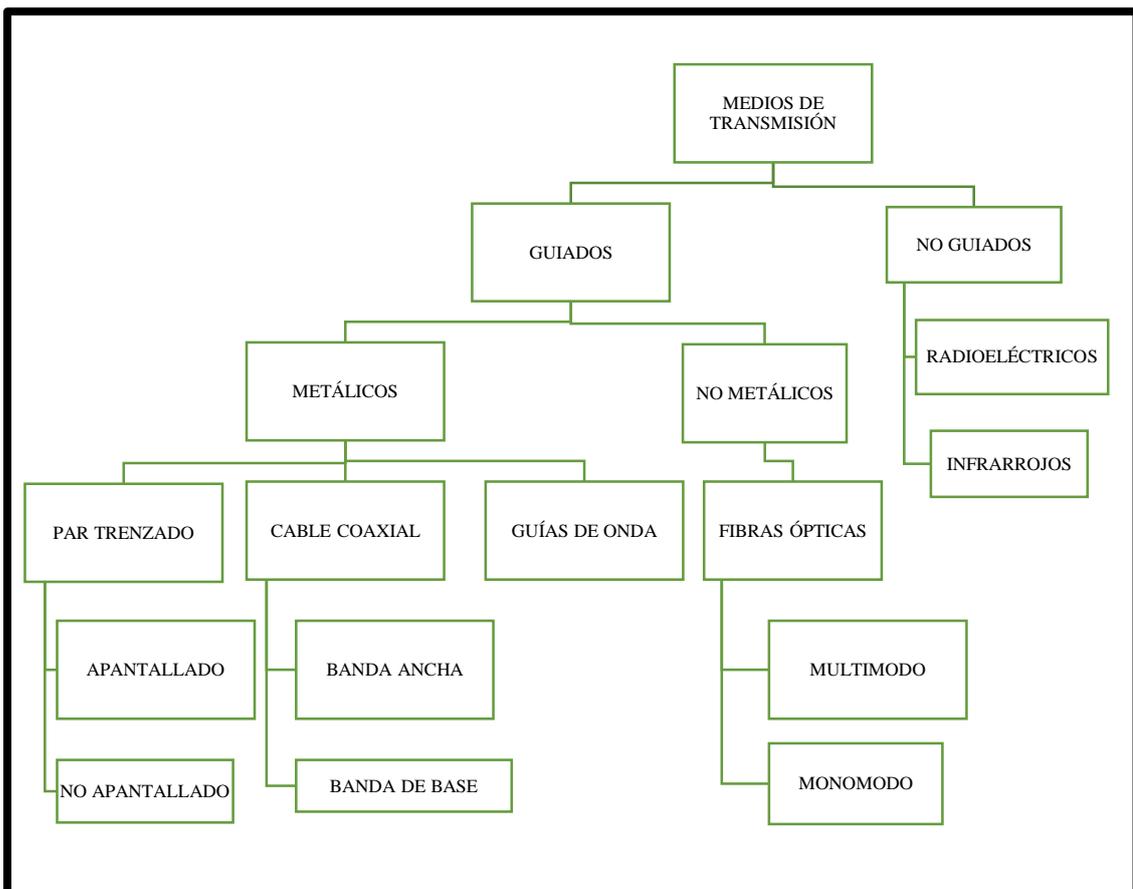


Figura 4-1: Tipos de medios de transmisión

Elaborado por: CARRILLO, Mayra.2018

Medios Guiados.- incluyen a los conductores metálicos (par trenzado, cable coaxial y guías de onda) y los conductores no metálicos como las fibras ópticas. Los medios de transmisión guiados presentan la ventaja de permitir un ancho de banda muy elevado y ser menos susceptibles a las interferencias; para distancias cortas pueden ser mucho más económicos que los medios inalámbricos.

Medios No Guiados.- transmiten por el espacio libre y comprenden las ondas radioeléctricas (HF, VHF, UHF y Microondas) los rayos o haces infrarrojos (actualmente está en desarrollo la transmisión mediante el láser. (Briceño, 2005, p. 498)

1.7 Comunicaciones Inalámbricas

En la Figura 5-1 se observa la infraestructura de las comunicaciones inalámbricas en comparación con las redes cableadas. Las características son las siguientes:

- Uso eficiente del espectro electromagnético debido a la utilización repetida de frecuencias
- Compatibilidad a nivel nacional e internacional, para que los usuarios móviles puedan utilizar sus mismos equipos en otros países o áreas.
- Prestación de servicios para aplicaciones de datos, voz y video;
- Adaptación a la densidad de tráfico; dado que la densidad de tráfico es diferente en cada punto de la zona de cobertura.
- Calidad del servicio. (Eveliux, 2002, <http://www.eveliux.com/mx/El-ABC-de-las-redes-inalambricas-WLANs.html>).

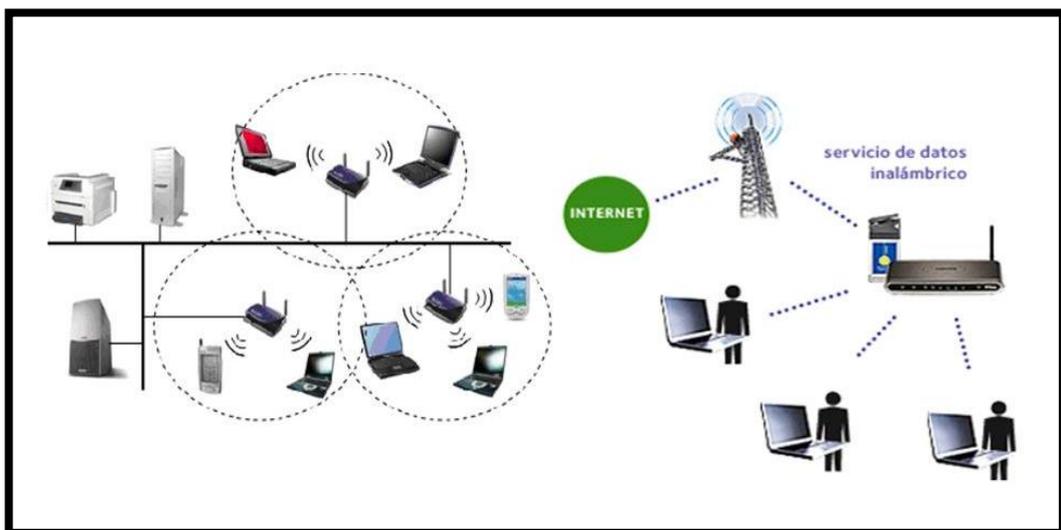


Figura 5-1: Comunicación Inalámbrica

Fuente: (Funcional, 2011, <http://elabcfuncional.blogspot.com/>)

1.8 Tecnologías Inalámbricas

La comunicación vía radio usa el espectro electromagnético para enviar información. Cuando una corriente eléctrica pasa por un cable crea un campo electromagnético que envía ondas en todas direcciones, en forma parecida a la luz, que también es parte del espectro, pero a frecuencias mucho más altas. Las tecnologías inalámbricas pueden clasificarse en cinco grandes grupos, de acuerdo con la distancia que viaja cada tipo de señal. Primero están las comunicaciones satelitales, como el sistema de posicionamiento global. Otra categoría, y con señales de dos vías, están las tecnologías de telefonía celular de cobertura amplia como GSM y CDMA. Una tercera categoría incluye señales de menor alcance utilizadas para conectar dispositivos dentro de una habitación o un edificio, como los sistemas Wi-Fi o Zigbee, En cuarto lugar están los protocolos para enlazar dispositivos en una "red de área personal" (PAN, personal area network). El último tipo de comunicaciones son las que se dan cerca de una antena transmisora (NFC, near-field communications). Podemos ver como se distribuyen las diferentes tecnologías inalámbricas dependiendo de la velocidad de transmisión y de su utilización. (Fernández, 2009, p. 28).

1.8.1 Tecnología por Radio Frecuencia (RF)

La mayoría de las comunicaciones se ejecutan sin emplear cables, es decir, son comunicaciones inalámbricas, este fenómeno de comunicación que se tuvo la necesidad de desarrollar elementos electrónicos capaces de enviar y recibir información proveniente de sitios lejanos sin estar interconectado uno con el otro a través de cables, estos elementos electrónicos son el transmisor y el receptor, instrumentos hoy en día necesarios para la mayoría de la gente en el mundo aún y cuando no se tenga conciencia de su existencia ni de su funcionamiento. (Cervantes & Delgado, 2010, p. 48).

1.8.2 Tecnología Zigbee

Utiliza la banda ISM para usos industriales, científicos y médicos; en concreto, 868 MHz en Europa, 915 en Estados Unidos y 2,4 GHz en todo el mundo. Sin embargo, a la hora de diseñar Dispositivos, las empresas optan prácticamente siempre por la banda de 2,4 GHz, por ser libre en todo el mundo. El desarrollo de la tecnología se centra en la sencillez y el bajo coste más que otras redes inalámbricas semejantes de la familia WPAN. (Fernández, 2009, p. 30)

1.8.3 Tecnología Bluetooth

Es una especificación tecnológica para redes inalámbricas que permite la transmisión de voz y datos entre distintos dispositivos mediante una radiofrecuencia segura (**2,4 GHz**). Esta tecnología, por lo tanto, permite las comunicaciones sin cables ni conectores y la posibilidad de crear redes inalámbricas domésticas para sincronizar y compartir la información que se encuentra almacenada en diversos equipos. (Pérez & Merino, 2009). Es compatible con la totalidad de los dispositivos que actualmente existen, presenta una velocidad admisible a costo bajo (Yaagoubi, 2012, p. 10).

1.8.4 Tecnología Wi-fi

Es una tecnología de área local que alcanza tasas de transmisión de hasta 54 kbps en un canal de 20 MHz en la banda de 2.4 GHz. En Wi-Fi un punto de acceso inalámbrico (Access point) transmite y recibe datos a través de ondas de radio y los equipos remotos, que cuentan con un transceptor (transmisor-receptor) en una tarjeta de acceso. Comprende una gran cantidad de estándares para redes de comunicación inalámbrica basados en las especificaciones IEEE 802.11. En sus inicios Wi-Fi fue pensado para conectar redes locales inalámbricas; sin embargo, actualmente se utiliza para el acceso a Internet. (Viloria, et al., 2009, p. 134).

1.8.5 Comparación de tecnologías inalámbricas

Realizada la investigación de las diferentes tecnologías inalámbricas se realiza una tabla comparativa con las principales características, para determinar la tecnología a utilizarse en la implementación del prototipo de la red de comunicación. En la Tabla 1-1 se aprecia una comparativa de las principales características de las tecnologías inalámbricas.

Tabla 1-1: Tabla comparativa de tecnologías inalámbricas

	Frecuencia	Alcance	Velocidad de transmisión	Características
Radiofrecuencia	2.4 GHz	100-1000 m	Velocidad de la luz	<ul style="list-style-type: none"> • Permite una comunicación inmediata. • Elimina la necesidad de emplear cables o infraestructuras. • Pueden ser configuradas en una amplia gama de topologías. • Bajo consumo y costo
Zigbee	2.4 GHz	10-100 m	250 Kbps	<ul style="list-style-type: none"> • Robustez • Alta escalabilidad • Bajo consumo
Bluetooth	2.4 GHz (ISM)	1-100 m	1 Mbps	<ul style="list-style-type: none"> • Corto alcance • Transfiere segmentos de datos (chunks) • Bajo consumo
Wifi	2.4 GHz y 5 GHz	1-50 m	150-600 Mbps	<ul style="list-style-type: none"> • Transfiere datos con rapidez. • Necesita infraestructura de red. • Proporciona tener acceso a una red o a Internet

Elaborado por: CARRILLO, Mayra 2018

De la Tabla 1-1 se obtiene que, la tecnología inalámbrica que se usa en el presente prototipo es la comunicación por radio frecuencia ya que tiene una alcance hasta de 1000 m, además no necesita de infraestructura de una red para realizar la comunicación. Es de bajo costo y consumo, puede ser configurado en diferentes topologías.

1.9 Tarjetas de desarrollo

Son placas de desarrollo de *hardware*, son útiles al realizar cualquier tipo de sistemas embebidos que cada una en particular cuenta con su lenguaje de programación o un sistema operativo. Estas placas tienen algo en común, que están compuestas por algún micro controlador de alguna empresa como Microchip, Atmel, etc. o también por un procesador de alguna plataforma

como ARM, y no tienen un Sistema Operativo. Además solo responden a un lenguaje en particular. (Quispe, <http://www.lightpath.io/tarjetas-de-desarrollo/>)

Las características que diferencia el Arduino de las demás tarjetas de desarrollo o microcontroladores son su entorno de desarrollo y lenguaje de programación. Estas placas tienen un microcontrolador ATMEL los cuales tienen entradas digitales y analógicas programables. (González, 2015, <http://panamahitek.com/que-es-arduino-y-para-que-se-utiliza/>). Las tarjetas más utilizadas en la actualidad son: Arduino, Raspberry, Galileo.

1.9.1 Tarjeta de desarrollo Arduino

Permite la creación de prototipos electrónicos rápidos y baratos. Éste convirtió lo que solían ser problemas de *hardware* bastante difíciles, en problemas de *software* mucho más simples (Hacedores, 2014, <http://hacedores.com/que-tarjeta-de-desarrollo-elegir-parte-1/#main-navigation>). Su manejo en cuanto a la compatibilidad de varios módulos de hardware adicionales y el lenguaje de programación en C, que pueden integrarse para distintos objetivos. Además la compañía Arduino pone a disposición varias librerías que pueden ser usados de acuerdo al módulo que se desee. (Quispe, <http://www.lightpath.io/tarjetas-de-desarrollo/>).

Es de código abierto cuyos elementos son contar con *software* y *hardware* fáciles de usar. Es decir, se pueden realizar proyectos interactivos de una forma sencilla y con diversas aplicaciones. (Doutel, 2015, <https://www.xataka.com/especiales/guia-del-arduinomaniaco-todo-lo-que-necesitas-saber-sobre-arduino>)

1.9.2 Tarjeta de desarrollo Raspberry PI

Es una pequeña tarjeta del tamaño de una tarjeta de crédito pero con las virtudes de una PC, fue diseñada en Reino Unido por una fundación cuyo nombre es el mismo de la placa. Los primeros diseños de Raspberry Pi se basaban en el microcontrolador Atmel ATmega644. El diseño contiene un chip Broadcom BCM2835, tiene un procesador central ARM1176JZF-S a 700 MHz. Tiene un procesador gráfico (GPU) VideoCore IV, y 512 MiB de memoria RAM. El diseño no incluye un disco duro ni unidad de estado sólido, usa una tarjeta SD. (Chalá, 2014)

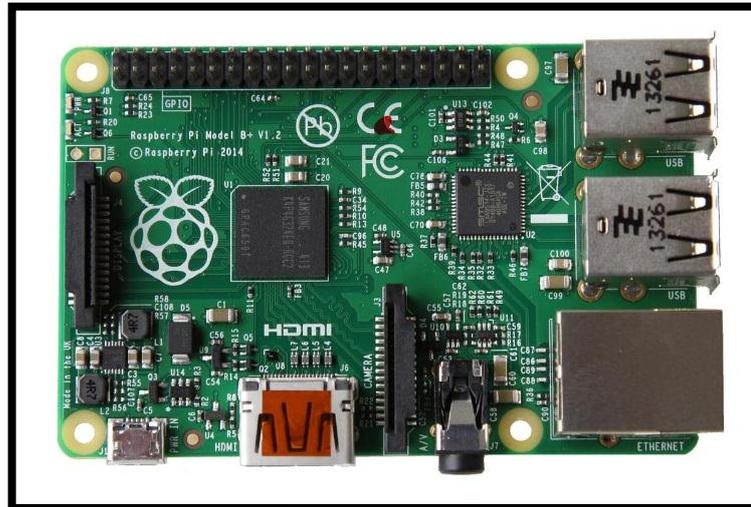


Figura 6-1: Placa Raspberry PI

Fuente: (ZDNet)

1.9.3 Tarjeta de desarrollo Galileo

Se trata de una familia de placas de desarrollo compatibles con Arduino que trabajan bajo arquitectura Intel. Esta placa de desarrollo ejecuta un sistema operativo Linux libre que contiene las librerías de software de Arduino, lo que le permite ofrecer una mayor escalabilidad y reutilizar el software ya existente. Los usuarios podrán programar sus Intel Galileo desde los sistemas operativos Mac OS, Microsoft Windows y Linux de sus equipos. Además, estas placas también han sido diseñadas para ser compatibles, a nivel de software y de hardware. (Maturana, 2013, <https://www.xataka.com/makers/intel-galileo-placa-de-desarrollo-fruto-de-la-colaboracion-entre-intel-y-arduino>).



Figura 7-1: Tarjeta de desarrollo Galileo

Fuente: (Maturana, 2013)

1.9.4 Comparación entre los diferentes tipos de tarjetas de desarrollo

Analizados los diferentes tipos de tarjetas de desarrollo se realizó una tabla comparativa de las mismas como se lo puede apreciar en la Tabla 2-1.

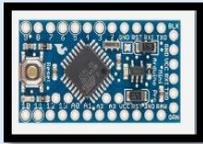
Tabla 2-1: Tabla de comparación de los diferentes tipos de tarjetas de desarrollo.

	Arduino	Raspberry Pi	Galileo
Procesador	ATMega 328	Broadcom BCM2836 ARM Cortex-A7	SoC Quark X100
Sistema Operativo	Ninguno	Distribuciones de Linux	Linux, Windows
Velocidad	16 MHz	900 MHz	400 MHz
Entorno de desarrollo	Arduino	Linux, IDLE, QEMU, Eclipse, Windows.	Arduino
Ethernet	10/100	10/100	Ninguno
USB	1	4	2
Voltaje	5 V	3.3 V	3.3 V
Memoria RAM	2KB	-	8 MB
Precio	\$25	\$55	\$95

Elaborado por: CARRILLO, Mayra 2018

En la Tabla 2-1 se aprecia que Arduino tiene un entorno de desarrollo amigable, es de bajo consumo además cuenta con una amplia gama de tutoriales los cuales enseñan sus diferentes aplicaciones, consumo y modelos de Arduino, para una mejor selección de la clase de Arduino para nuestro prototipo en la Tabla 3-1 se podrá apreciar las características de las distintas clases para un óptimo funcionamiento de la implementación del prototipo.

Tabla 3-1: Diferencias de los tipos de Arduino

	Arduino Pro Mini	Arduino Pro Micro	Arduino MEGA 2560	Arduino UNO
Microprocesador	ATmega168	Atmega 32U4	ATmega2560	ATmega 328 P
Pines digitales	8	12	16	14
Pines Analógicos	14	4	54	6
Voltaje de entrada	5-12 V	7-12 V	7-12 V	7 -12 V
Voltaje de salida	5V	5 V	5 V	5 V
Memoria	32 KB	32 KB	256 KB	32 KB
Conector	Por medio de una tarjeta o un cable FTDI	Micro-USB	Conector USB, Power Jack ICSP Botón de Reset	Conector USB, Power Jack ICSP Botón de Reset
Dimensiones	18 x 33 mm	32 x 17 mm	100 x 50 mm	68.6 x 53.4 mm
Imagen				

Elaborado por: CARRILLO, Mayra 2018.

De la tabla 3-1 se obtiene que la tarjeta de desarrollo Pro-micro es óptima para el uso en la implementación de este prototipo, debido a su tamaño que es muy reducido a diferencia de los otros modelos de Arduino, además tiene un bajo consumo de energía y se le puede alimentar directamente desde el celular.

1.9.5 Android Studio

Proporciona las herramientas más rápidas para crear apps en todas las clases de dispositivos Android. La edición de códigos de primer nivel, la depuración, las herramientas de rendimiento, un sistema de compilación flexible y un sistema instantáneo de compilación e implementación te permiten la creación de aplicaciones únicas y de alta calidad. (Android Studio, <https://developer.android.com/studio/index.html?hl=es-419>)

1.9.5.1 Requerimientos para la instalación de Android Studio

En la Tabla 4-1 se describen los requerimientos para la instalación en los diferentes sistemas operativos.

Tabla 4-1: Requerimientos para la instalación de Android

WINDOWS	MAC	LINUX
<ul style="list-style-type: none">• 3GB de memoria RAM como mínimo, más 1 GB para el emulador de Android.• 2GB de espacio en disco disponible como mínimo; 500 MB para el IDE + 1,5 GB para Android SDK y la imagen de sistema del emulador.• Para el emulador acelerado: Sistema operativo de 64 bits y procesador Intel® compatible con Intel® VT-x, Intel® EM64T y la funcionalidad Execute Disable (XD) Bit.	<ul style="list-style-type: none">• 3GB de memoria RAM como mínimo, más 1 GB para el emulador de Android.• 2GB de espacio en disco disponible como mínimo; 500 MB para el IDE + 1,5 GB para Android SDK y la imagen de sistema del emulador.	<ul style="list-style-type: none">• GNOME o KDE de escritorio.• Distribución de 64 bits capaz de ejecutar aplicaciones de 32 bits.• GNU C Library 2.19 o versiones posteriores.• 3GB de memoria RAM como mínimo, más 1 GB para el emulador de Android.• 2GB de espacio en disco disponible como mínimo; 500 MB para el IDE + 1,5 GB para Android SDK y la imagen de sistema del emulador.

Elaborado por: CARRILLO, Mayra 20018

CAPÍTULO II

2 IMPLEMENTACIÓN DEL *HARDWARE* Y *SOFTWARE* DEL PROTOTIPO

En este capítulo se presenta el diagrama general de la red auxiliar de comunicación que se implementó, se describe los requerimientos de *hardware* y *software*, los esquemas de conexión, los dispositivos que lo conforman y el *software* utilizado e implementado en el prototipo.

2.1 Diagrama general de la red de comunicación

En la Figura 1-2 se aprecia la descripción general de la red de comunicación; la misma que está formada por dos nodos, el nodo uno está conectado con dos usuarios el mismo que se conecta con otro usuario y continua la conexión hasta llegar al nodo dos, de esta manera se va realizando la red de comunicación inalámbrica, la cual permite comunicarse mediante mensajes de texto, para lo cual se creó una aplicación para dispositivos móviles que tengan el sistema operativo Android, al mismo que se conecta un dispositivo electrónico el cual permite la transmisión y recepción de mensajes sin la necesidad de infraestructura, la comunicación se lo realiza por radio frecuencia.

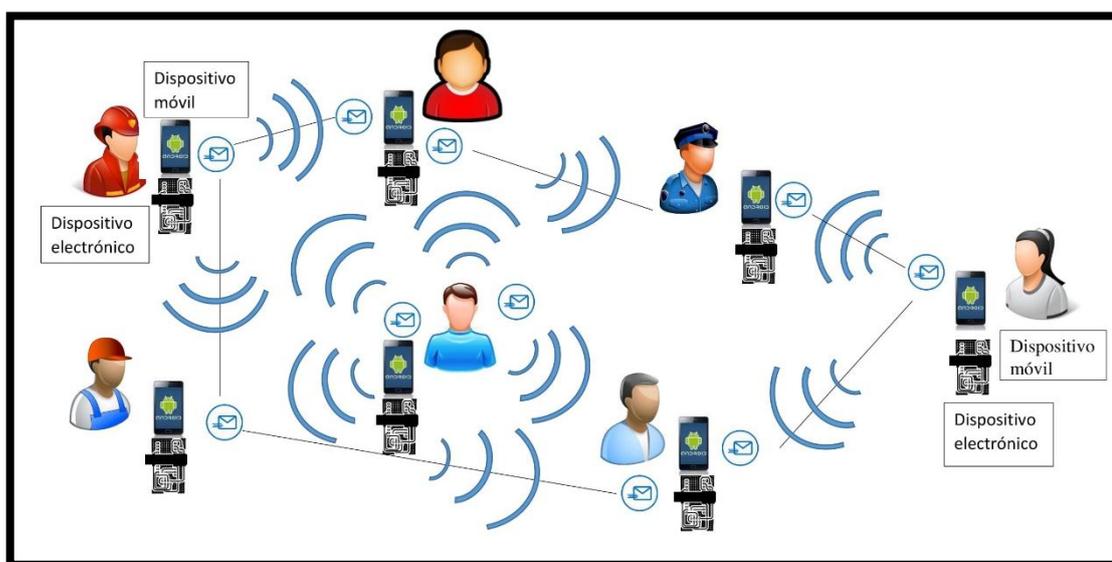


Figura 1-2: Diagrama general de la red de comunicación

Elaborado por: CARRILLO Mayra, 2018

2.2 Requerimientos de la red de comunicación.

Realizados los estudios en el capítulo anterior se puede definir los requerimientos que satisfacen para la implementación del prototipo que son los siguientes:

- La red de comunicación inalámbrica no requiere de infraestructura para realizar la emisión y recepción de datos en caso de desastres naturales o colapso de las redes.
- El dispositivo electrónico sea lo más pequeño posible
- Se pueda alimentar directamente con el celular móvil y sea compatible.
- La aplicación de mensajería instantánea debe ser de fácil manejo e instalación para dispositivos móviles que tengan el sistema operativo Android.
- El prototipo de la red auxiliar de comunicación a implementarse sea de bajo consumo de energía.

2.3 Requerimientos de *Hardware* de la red de comunicación.

Una vez realizado el estudio en el capítulo anterior se pueden definir los requerimientos de diseño que debe satisfacer el prototipo de una red auxiliar de comunicación por radiofrecuencia con nodos basados en dispositivos electrónicos y equipos móviles para mensajería instantánea. Son las siguientes:

- Su tamaño sea lo más reducido posible.
- El dispositivo electrónico sea de bajo consumo de energía.
- Su alimentación sea directa del dispositivo móvil.

2.3.1 *Diagrama de bloques del dispositivo electrónico*

En la en la Figura 2-2, se presenta el esquema del diseño del módulo el cual está formado por 5 bloques, donde:

- Los datos son tomados por el bloque de adquisición de datos y transmitidos al bloque de codificación que están formado por un módulo NRF 24L01, conjuntamente con una aplicación móvil.
- El bloque de alimentación está constituido por un regulador de voltaje y la batería propia del dispositivo móvil.

- El bloque de transmisión permite la comunicación entre el dispositivo móvil y el prototipo implementado, por medio de un conector OTG a micro USB y un conector USB a micro USB. Finalmente es bloque de procesamiento lo conforma un Arduino PRO-micro que interactúa con todos los bloques.

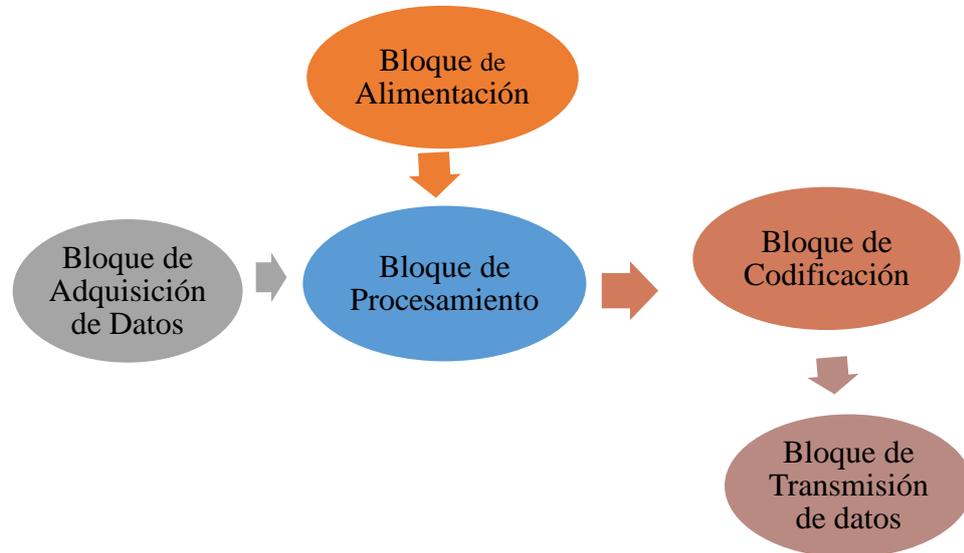


Figura 2-2: Diagrama de bloques del dispositivo.

Elaborado por: CARRILLO Mayra, 2018

2.3.2 Características de los dispositivos seleccionados

2.3.2.1 Arduino Pro micro

Cuenta con un micro controlador ATMEGA32U4, un transceptor USB dentro del 32U4 el cual nos permite añadir conectividad USB a bordo y acabar con voluminosos interfaz USB externo. Esta pequeña tarjeta hace todas las funciones Arduino con las que se está familiarizado con: 4 canales de 10-bit ADC, 5 terminales PWM, 12 DIO, así como hardware conexiones serie Rx y Tx. Funcionando a 16 MHz y 5 V. Este pequeño micro controlador puede ir en cualquier lugar. Hay un regulador de tensión a bordo para que pueda aceptar una tensión de hasta 12V. (Labs, 2016, <http://saber.patagoniatec.com/arduino-pro-micro-atmega32u4-arduino-argentina-ptec-leonardo/>). En la Figura 3-2 se observa la estructura física del Arduino Pro Micro.

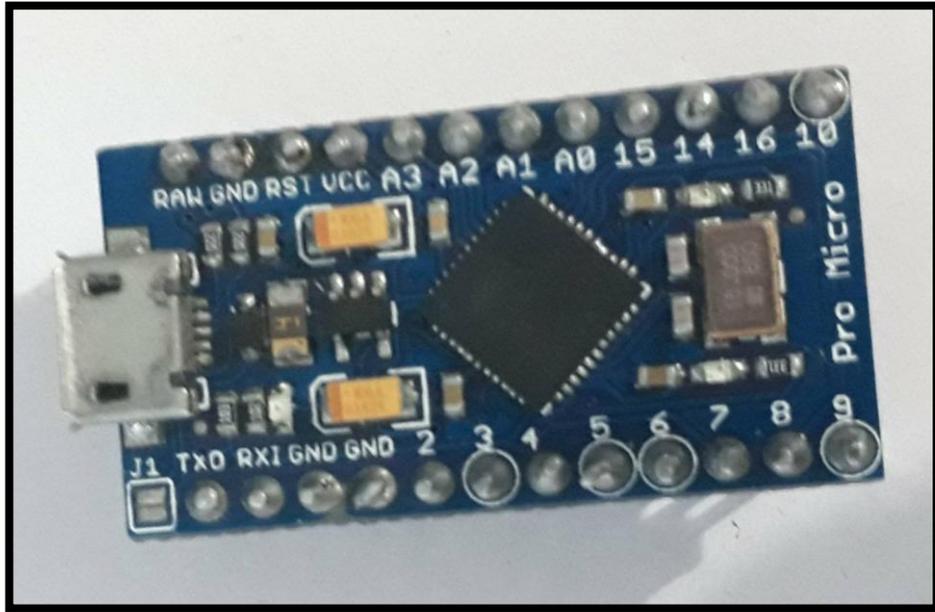


Figura 3-2: Estructura física de Arduino Pro Micro

Elaborado por: CARRILLO, Mayra 2018

2.3.2.2 Módulo NRF 24L01

Está integrado por un transceptor RF (transmisor + receptor) a una frecuencia entre 2.4GHz a 2.5GHz, una banda libre para uso gratuito. La velocidad de transmisión es configurable entre 250 Kbps, 1Mbps, y 2 Mbps y permite la conexión simultánea con hasta 6 dispositivos. El control del módulo se realiza a través de bus SPI. Se recomienda usar las frecuencias de 2501 a 2525 MHz para evitar interferencias con las redes Wifi. (LLamas, 2016, <https://www.luisllamas.es/comunicacion-inalambrica-a-2-4ghz-con-arduino-y-nrf24l01/>). En la Figura 4-2 se aprecia la parte física del módulo NRF24L01 con su antena, el que permitirá la conexión entre los dispositivos móviles.



Figura 4-2: Módulo NRF24L01

Elaborado por: CARRILLO, Mayra 2018.

2.3.2.3 Regulador de Voltaje

Está basado en el circuito integrado AMS1117-5.0V. Este dispositivo permite alimentar el circuito desde una amplia gama de tensiones de entrada, tiene un voltaje de entrada: 4.75V hasta 12V, voltaje de salida: 3.3V y corriente de 800mA máx. (Electronilab, 2015). En la Figura 5-2 se observa el dispositivo físico.

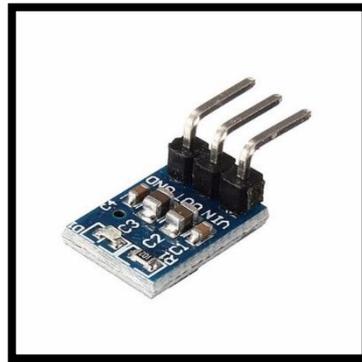


Figura 5-2: Regulador de Voltaje

Elaborado por: CARRILLO, Mayra 2018

2.3.3 Cables de conexión.

Para realizar la conexión entre el dispositivo móvil y el dispositivo electrónico utilizamos un cable de datos USB a micro USB el cual conectamos al dispositivo electrónico, y un adaptador OTG para conectar al equipo móvil. En la Figura 6-2 se aprecia el cable y adaptador de conexión.



Figura 6-2: Cable y adaptador de conexión.

Elaborado por: CARRILLO, Mayra 2018.

2.3.4 Esquema de conexión del dispositivo electrónico.

El Arduino Pro-Micro tiene sus terminales conectadas al módulo NRF24L01 como se aprecia en la Tabla 1-2. El terminal de salida del Arduino, que entrega los 5V se encuentra unido al regulador de tensión ASM1117.

Tabla 1-2: Diagrama de conexión del Hardware

TERMINAL ARDUINO PRO MICRO	TERMINAL MODULO NRF24L01
D8	CE
D9	CSN
D15	SCK
D14	MISC
D16	MOSI
GND2	GND

Elaborado por: CARRILLO, Mayra 2018

- El terminal de salida del regulador de voltaje se conecta al módulo NRF24L01 al terminal de 3,3 V, y el terminal de entrada se conecta a VCC del Arduino Pro micro

En la Figura 7-2 se observa el esquema del prototipo del módulo de comunicación inalámbrica para mensajería instantánea realizado en el simulador en línea Fritzing (Fritzing, 2015).

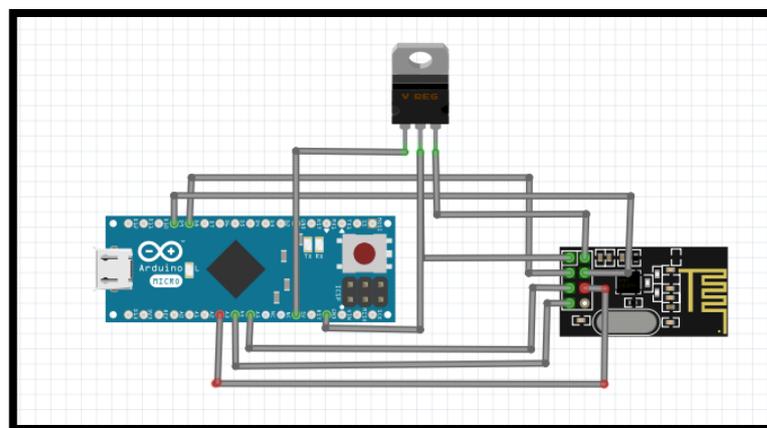


Figura 7-2: Esquema de conexión de dispositivo electrónico.

Elaborado por: CARRILLO, Mayra.2018

2.4 Requerimientos del *Software* para la implementación del prototipo.

Para la implementación del *software* del prototipo de la red se desarrolla una aplicación para los dispositivos móviles que tenga el sistema operativo Android, a continuación se detallan los requerimientos:

- La aplicación deber ser de fácil manejo e instalación para cualquier tipo de dispositivo móvil.
- Debe enviar y recibir mensajes sin infraestructura adicional.
- La transmisión y recepción de datos debe ser inmediata.

2.4.1 *Herramientas de Software*

Las herramientas utilizadas para el desarrollo de *software* se detallan a continuación:

2.4.1.1 *Android Studio versión 2.3.1*

Entre las principales características tenemos:

- Tiene entorno de desarrollo claro y robusto
- Facilidad para testear el funcionamiento en otros tipos de dispositivos.
- Asistentes y plantillas para los elementos comunes de programación en Android.
- Un completo editor con muchas herramientas extra para agilizar el desarrollo de nuestras aplicaciones. (Rosso, <https://android-studio.uptodown.com/windows>)

2.4.1.2 *Arduino versión 1.8.5*

El software proporciona soporte para una amplia gama de placas Arduino, incluidas Arduino Uno, Nano, Mega, Esplora, Ethernet, Fio, Pro o Pro Mini, así como LilyPad Arduino. El entorno de desarrollo integrado de Arduino es de fácil manejo. En él se desarrollan los programas denominados sketch, conjunto de instrucciones de código fuente muy parecido al lenguaje C. (Arduino Genuino). Además El programa incluye una amplia gama de bibliotecas integradas, como EEPROM, Firmata, GSM, Servo, TFT, WiFi, etc. Los diseños se pueden verificar y compilar, con un registro de errores en la parte inferior de la IU que le permite revisar el código. (Matei, 2017, <http://www.softpedia.com/get/Programming/Other-Programming-Files/Arduino.shtml>).

2.4.2 Diseño de la aplicación móvil

2.4.2.1 Diagrama de Flujo de la aplicación móvil.

En la Figura 8-2 se observa el diagrama de flujo el que se detalla a continuación:

- Inclusión de librerías.
- Definición de variables visuales.
- Procesos para la programación de los diferentes botones en java.

El diseño y programación de la aplicación de mensajería instantánea se realiza en el *software* Android Studio versión 2.3.1.

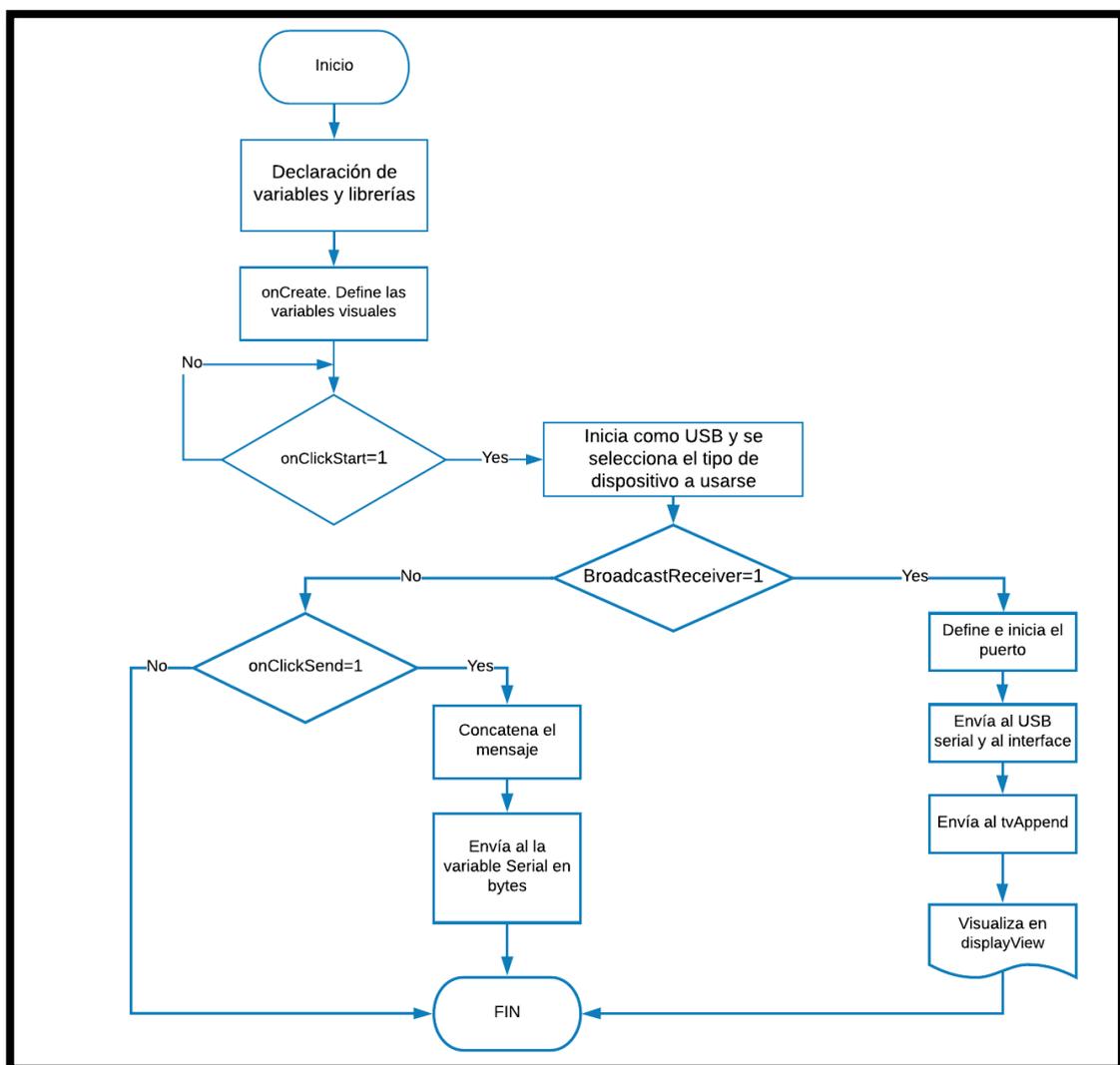


Figura 8-2: Diagrama de Flujo de la aplicación

Elaborado por: CARRILLO, Mayra 2018

2.4.3 Desarrollo de la programación.

Permite enviar y recibir mensajes instantáneos. Para su desarrollo se utilizó Android Studio 2.3.1 el cual tiene dos ventanas básicas:

- ActivityMain.java permite realizar la programación de los botones que conforman la aplicación; el lenguaje de programación q se utiliza es java en la Figura 9-2 a) se aprecia un fragmento de la programación en java.
- ActivityMain.xml permite realizar la parte visual de la aplicación la programación está basada en xml. En la Figura 9-2 b) se observa la ventana visual de la aplicación móvil.

Las dos programaciones son distintas pero se complementa entre sí; debido a que en la programación java se define los objetos de la programación visual en xml como son el textview, button.

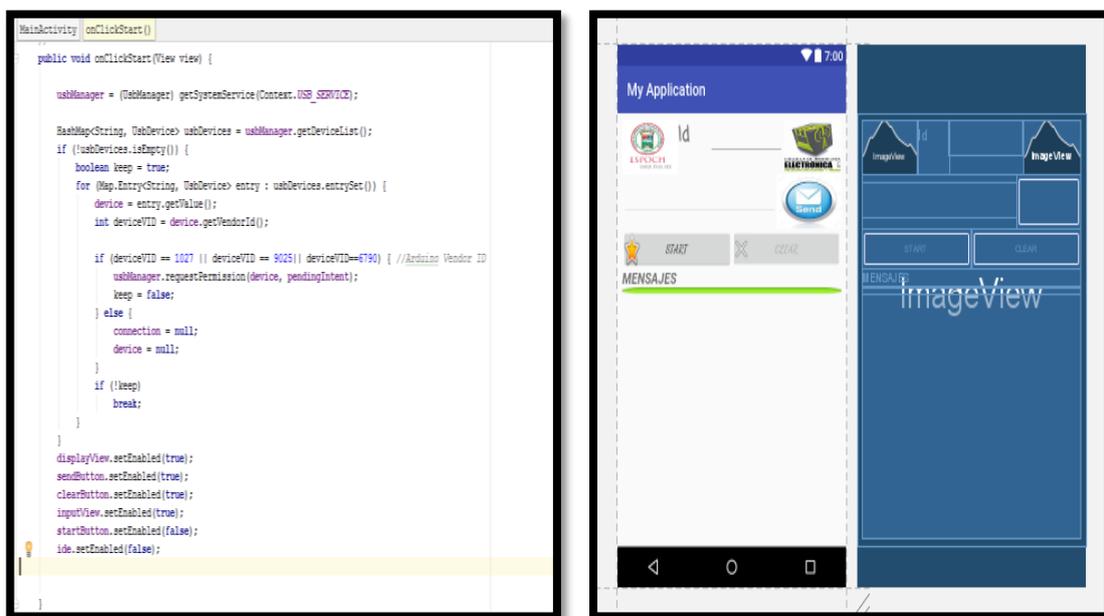


Figura 9-2: a) Fragmento de la programación en java. b) Programación visual xml

Elaborado por: CARRILLO, Mayra 2018

2.4.4 Librerías utilizadas en el diseño de la aplicación.

Las librerías que se utilizan en la aplicación son propias del *software* de Android Studio las que se utilizan son:

- **Android.os.Bundle.-** Se usa para pasar información entre actividades y otros componentes de la aplicación. (Stackoverflow, 2016, <https://stackoverflow.com/4999991/what-is-a-bundle-in-an-android-application>)
- **Android.app.PendingIntent.-** Las instancias de esta clase se crean con `getActivity(Context, int, Intent, int)` , `getActivities(Context, int, Intent[], int)` , `getBroadcast(Context, int, Intent, int)` y `getService(Context, int, Intent, int)`. (Android Developers, <https://developer.android.com/reference/android/app/PendingIntent.html>)
- **Android.content.BroadcastReceiver.** Un Broadcast Receiver es el componente que está destinado a recibir y responder ante eventos globales generados por el sistema, como un aviso de batería baja, un SMS recibido, un SMS enviado, una llamada, un aviso de de la tajea SD, etc. y también a eventos producidos por otras aplicaciones. (Salas, 2014, <http://programandoointentandolo.com/2014/08/broadcast-receiver-android.html>)
- **Android.content.Intent.-** Es un objeto que proporciona enlace de tiempo de ejecución entre componentes separados. (Alcalde, 2017, <https://elbauldelprogramador.com/programacion-android-intents-conceptos/>)
- **Android.content.IntentFilter.-** Se crean en XML como parte del archivo `AndroidManifest.xml`.
- **Android.hardware.usb.UsbDevice.-** Proporciona soporte para comunicarse con periféricos de hardware USB que están conectados a dispositivos con Android. (GitHub, 2015, <https://github.com/search?q=org%3Agithub+libreria+android>)
- **Android.hardware.usb.UsbDeviceConnection.-** Esta clase se utiliza para enviar y recibir mensajes de datos y control a un dispositivo USB. (grepcode, http://grepcode.com/file/repo1.maven.org/maven2/org.robolectric/android-all/4.1.2_r1-robolectric-0/android/hardware/usb/UsbDeviceConnection.java?av=h)
- **Android.media.MediaPlayer.-** Se puede usar para controlar la reproducción de archivos de audio / video y transmisiones. (Xamarin, <https://developer.xamarin.com/api/type/Android.Media.MediaPlayer/>)
- **Android.text.method.ScrollingMovementMethod.-** Un método de movimiento que interpreta las teclas de movimiento desplazando el búfer de texto.
- **Android.view.View.-** Es una estructura de datos cuyas propiedades contienen los datos de la capa, la información específica del área rectangular de la pantalla y permite establecer el layout.
- **Android.content.Context.-** Interfaz con información global sobre un entorno de aplicación. (Carmona, 2013, pp. 176-178)
- **Android.hardware.usb.UsbManager.-** Esta clase le permite acceder al estado de USB y comunicarse con dispositivos USB. Actualmente, solo el modo de host es compatible con la API pública. (Regupathy, 2014, p. 16)

- **Android.util.Log.-** Esta clase define los métodos familiares de información, advertencia y error que puede filtrar dentro del Log Can. (Hashimi, et al., 2011, p. 54)
- **Android.widget.Button.-** Es un elemento de interfaz de usuario que el usuario puede tocar o hacer clic para realizar una acción. (Jackson, pp. 77-78)
- **Android.widget.EditText.-**Es una subclase de TextView con operaciones de edición de texto. (AbhiAndroid, <http://abhiandroid.com/ui/edittext>).
- **Android.widget.TextView.-** Un elemento de interfaz de usuario que muestra texto al usuario.

Además de las librerías citadas anteriormente se instala una librería externa llamada com.felhr la que sirve para la comunicación USB (Felhr, 2014, <https://felhr85.net/2014/11/11/usbserial-a-serial-port-driver-library-for-android-v2-0/>).

- felhr.usbserial.UsbSerialInterface.
- felhr.usbserial.UsbSerialDevice.

Para el correcto funcionamiento de la librería se debe definir algunas funciones como:

- **On click.-** En esta función se configura los Baudios, ID del dispositivo y otras características necesarias para la aplicación.
- **BroadcastReceiver.-** Lee los datos que detecta el puerto USB, los mismos que son enviados a la siguiente función.
- **UsbInterface.UsbRealCallback.-** Esta función prepara el texto para mostrar en la pantalla.
- **On send.-** Envía el texto q se coloca en el edit.text mediante el puerto serial.

2.4.5 Funcionamiento de la aplicación móvil.

Es una aplicación de fácil manejo que se encuentra al alcance de cualquier usuario independientemente de su edad. . En la Figura 10-2 se observa el diagrama de flujo del funcionamiento de la aplicación. Los pasos a seguir para la utilización de la aplicación son

- Iniciar la aplicación.
- Ingresar el nombre del usuario en el ID.
- Dar clic en Start.
- Escribir el mensaje de texto que va a ser enviado, finalmente
- Dar clic en Send y el mensaje será enviado inmediatamente.

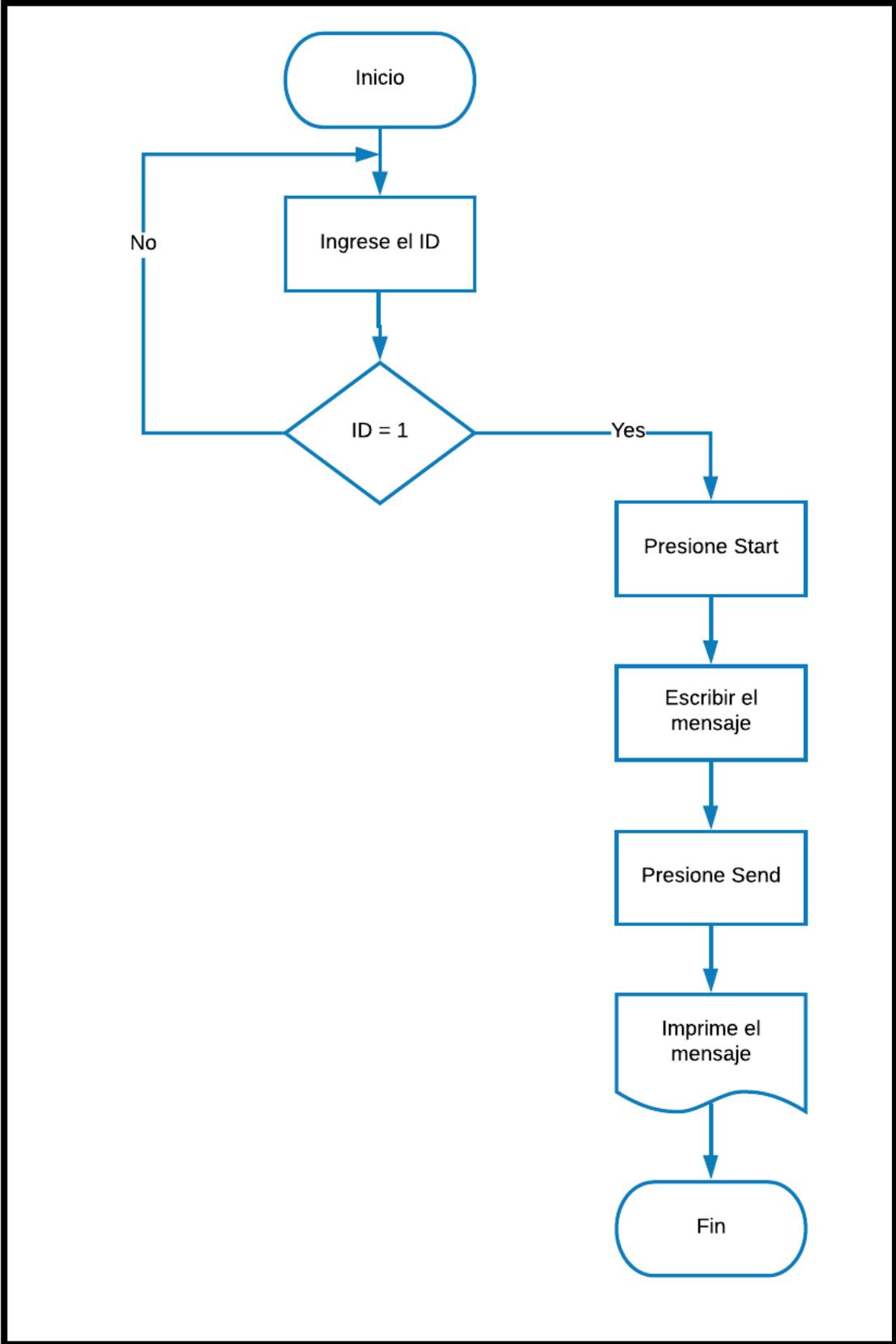


Figura 10-2: Diagrama de Flujo del funcionamiento de la aplicación móvil

Elaborado por: CARRILLO, Mayra 20018

2.4.6 Arduino

Es una plataforma de *hardware* y *software* de código abierto, basada en una sencilla placa con entradas y salidas, analógicas y digitales, en un entorno de desarrollo que está basado en un lenguaje de programación Processing. Es decir, una plataforma de código abierto para prototipos electrónicos. (Amangandi, 2012, <http://jamangandi2012.blogspot.com/2012/10/que-es-arduino-te-lo-mostramos-en-un.html>). En la Figura 11-2 se aprecia el diagrama de flujo de la programación a realizarse en el IDE de Arduino.

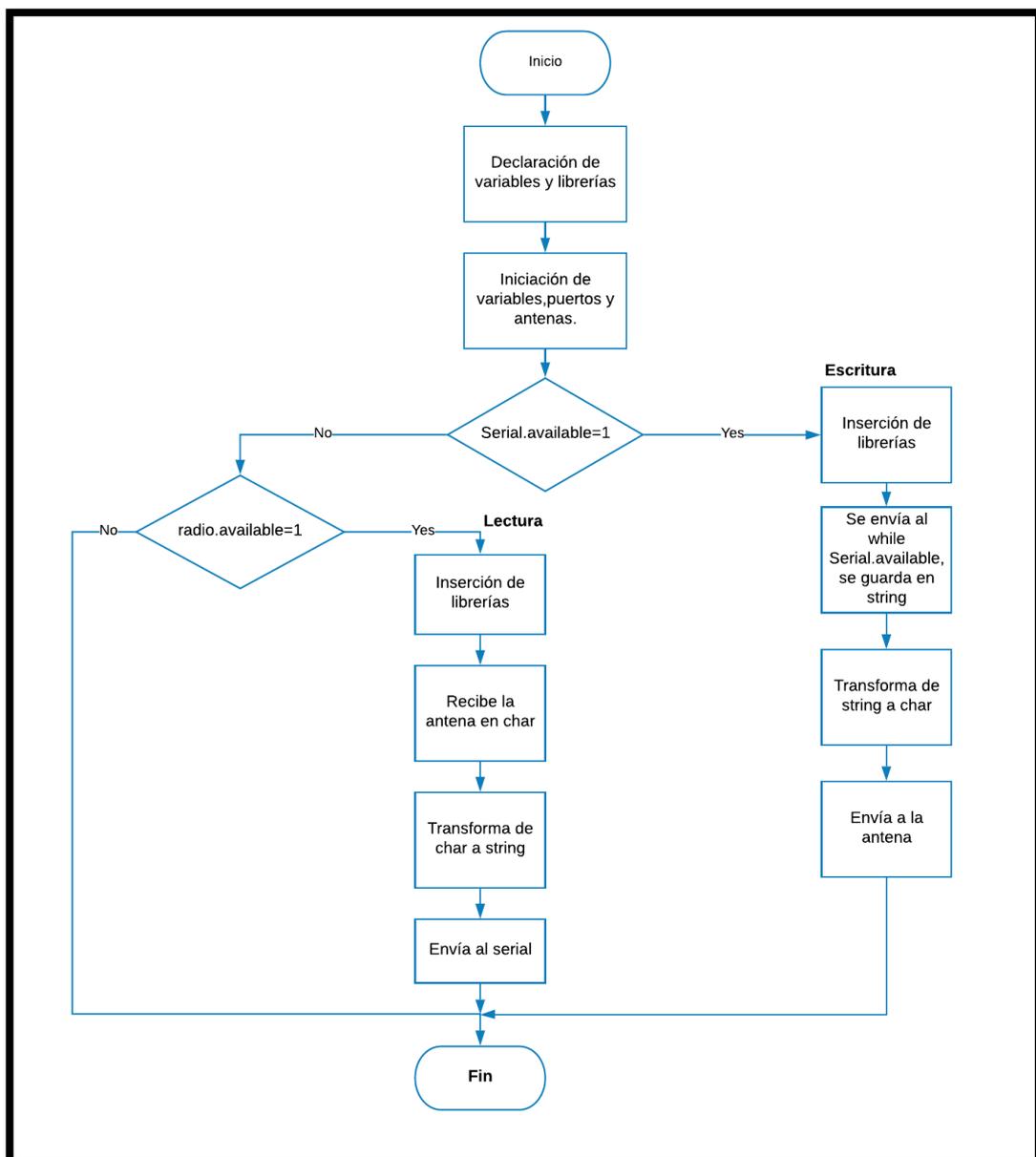


Figura 11-2: Diagrama de Flujo de la programación en Arduino

Elaborado por: CARRILLO, Mayra 2018.

En la figura 12a-2 se observa un fragmento del código de la programación del transmisor realizada en el Arduino en la versión 1.8.5, mientras que en la Figura 12b-2 se aprecia un fragmento del código del receptor.

```

emisor
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>

RF24 radio(8, 9); // CE, CNS,
const byte addresses[][6] = {"00001", "00002"};

String dato="";
String mensa="";
boolean h=false;
boolean t=false;
boolean m=false;
void setup()
{
  Serial.begin(9600);
  radio.begin();
  radio.openWritingPipe(addresses[1]); // 00001
  radio.openReadingPipe(1, addresses[0]); // 00002
  radio.setPALevel(RF24_PA_MIN);
}
void loop()
{
  delay(5);
  dato="";
  while (Serial.available())
  {
    char inChar = (char)Serial.read();
    dato=dato+inChar;
  }
}
receptor
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>

RF24 radio(8, 9); // CE, CSN
const byte addresses[][6] = {"00001", "00002"};

String dato="";
String mensa="";
boolean h=false;
boolean t=false;
boolean m=false;
void setup()
{
  Serial.begin(9600);
  radio.begin();
  radio.openWritingPipe(addresses[0]); // 00002
  radio.openReadingPipe(1, addresses[1]); // 00001
  radio.setPALevel(RF24_PA_MIN);
}
void loop()
{
  char mess[100]="";
  delay(5);
  radio.startListening();
  if ( radio.available())
  {
    while (radio.available())
    {

```

Figura 12-2: Fragmento del código en Arduino a) Transmisor b) Receptor

Fuente: CARRILLO, Mayra 2018

2.4.6.1 Librerías utilizadas en el Arduino

Las librerías utilizadas para la programación del emisor y receptor de mensajes son las siguientes

- **SPI.h.-** Esta biblioteca le permite comunicarse con dispositivos SPI, con el Arduino como dispositivo maestro. (Arduino, <https://www.arduino.cc/en/Reference/SPI>)
- **RF24.h.-** Driver para tranceptor inalámbrico nRF24L01. (Github, 2012, <https://maniacbug.github.io/RF24/classRF24.html>)
- **nRF24L01.h.-** Utiliza las capacidades de la radio en todo su potencial a través de Arduino, es confiable, receptivo, libre de errores y rico en funciones; además es compatible con la operación prevista del chip.

2.4.7 Diseño de la cobertura del dispositivo

Para realizar el diseño de la cubierta del dispositivo se utiliza el software Solidworks, que es un software de diseño mecánico en tercera dimensión 3D que utiliza un entorno gráfico basado en Microsoft Windows. (EcuRed, <https://www.ecured.cu/SolidWork>).

La cobertura del dispositivo está formada por el: segmento frontal, posterior y portador de la antena. En la Figura 13-2 se observa el segmento frontal y posterior, con medidas de: 37 mm x 56 mm x 12 mm de cada uno.

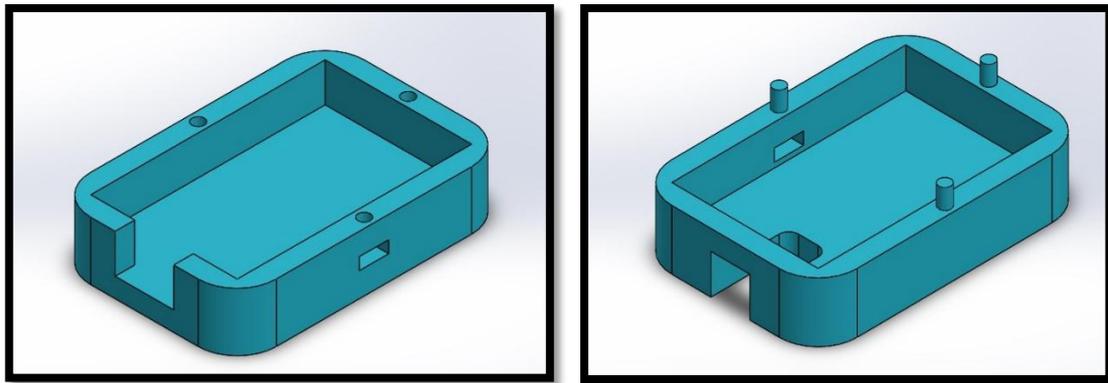


Figura 13-2: Segmento frontal y posterior del prototipo.

Elaborado por: CARRILLO, Mayra 2018

En la Figura 14-2 se observa el segmento portador de la antena mismo que posee un radio de 4mm.

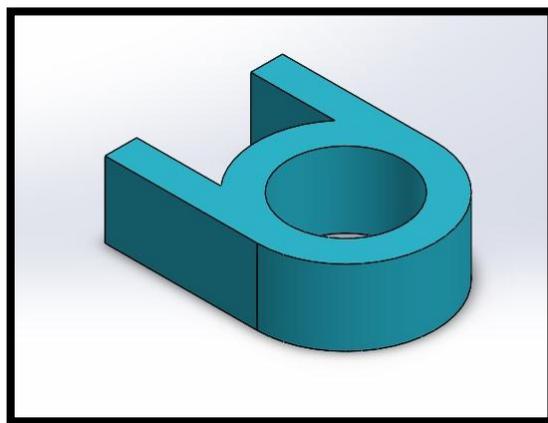


Figura 14-2: Segmento portador de la antena

Elaborado por: CARRILLO, Mayra 2018

Finalmente en la Figura 15-2 se aprecia la unión de los segmentos que forman la cubierta del prototipo, que mide 37 mm x 56 mm x 25mm. Dimensiones que cumplen con los requerimientos establecidos en esta investigación.

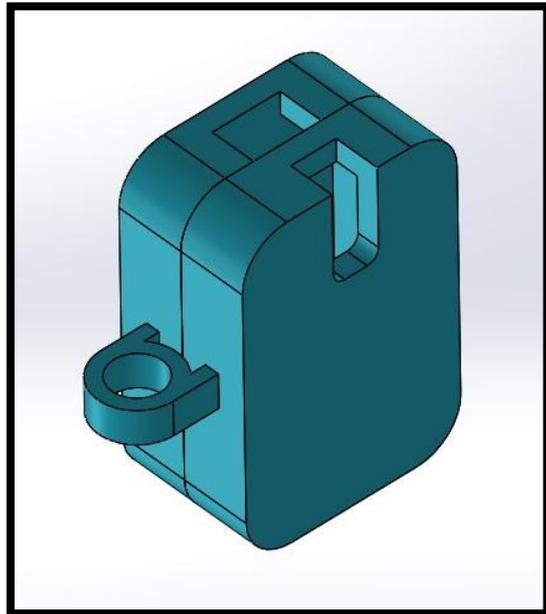


Figura 15-2: Cubierta del prototipo

Elaborado por: CARRILLO, Mayra 2018.

CAPÍTULO III

3 PRUEBAS Y RESULTADOS DEL SISTEMA

En el presente capítulo se muestra los resultados obtenidos en las pruebas realizadas con el prototipo de la red comunicación, tanto en *hardware*, como en *software*. La pruebas de funcionamiento de *hardware* se ejecuta tomando en cuenta tiempos de respuesta, distancia de operación con obstáculos y sin obstáculos, en las pruebas de *software* se realiza la evaluación del funcionamiento de la aplicación móvil. Se presenta también el análisis de costos de la implementación del prototipo.

3.1 Pruebas del *Hardware* implementado

Las dimensiones de los módulos implementados del sistema de comunicación inalámbrica es de: 37 mm x 56 mm x 25mm En la Figura 1-3 se observa la parte frontal y posterior del dispositivo electrónico implementado.

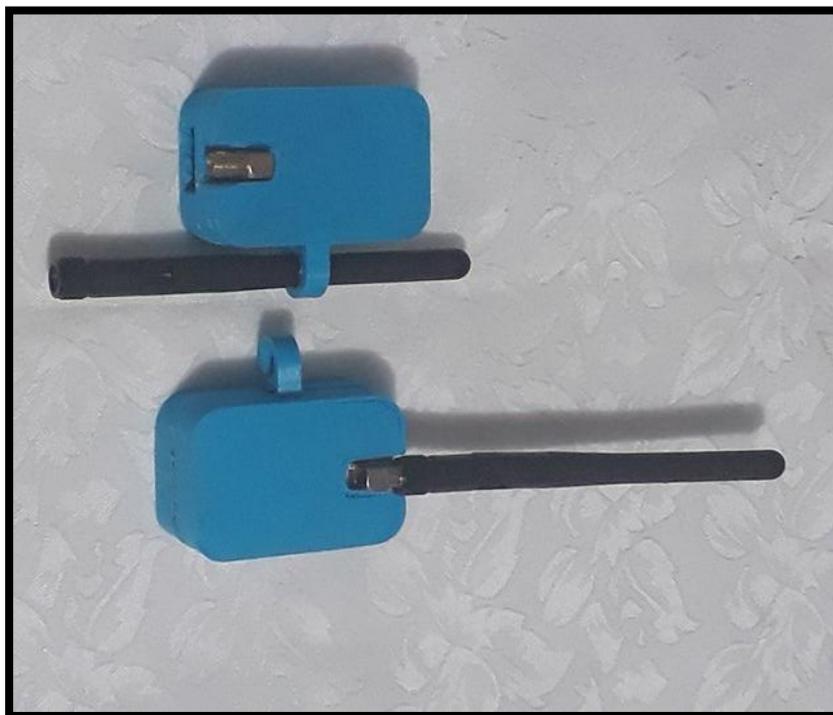


Figura 1-3: Módulos implementados.

Elaborado por: CARRILLO, Mayra 2018.

3.1.1 Distancia de transmisión de datos sin obstáculos.

Para determinar la distancia las pruebas se realiza en el Barrio San José de Tapi en las calles El Oro y Pedro Donoso tomando como punto de inicio, la medición se lo realiza manualmente y mediante una aplicación móvil Fields Area Measure PRO, como se observa en la Figura 2-3.

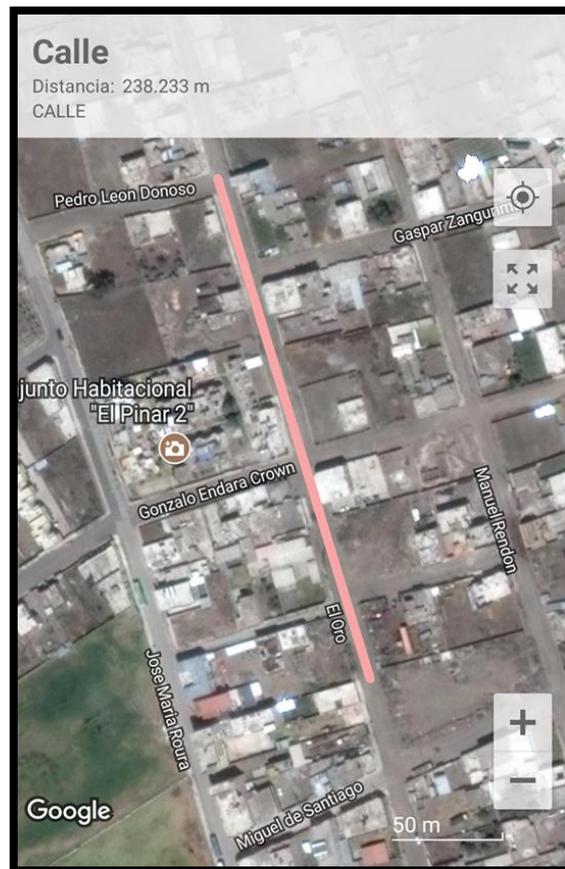


Figura 2-3: Aplicación móvil Fields Area Measure PRO

Elaborado por: CARRILLO, Mayra 2018

Se toma diferentes distancias en distintos rangos de 5 y 10 metros, se envía el mensaje SOS, para determinar el estado del mensaje y la distancia de cobertura que alcanza el dispositivo implementado, en la Figura 3-3 se observa el envío del mensaje SOS utilizando el dispositivo electrónico y el equipo móvil.



Figura 3-3: Caracteres enviados.

Elaborado por: CARRILLO, Mayra 2018

En la Tabla 1-3 se observa los intervalos de distancia tomados para realizar la prueba y el estado del mensaje SOS enviado, se determina que la distancia total obtenida de la medición manual es de 238 m y mediante la aplicación móvil Fields Area Measure PRO es de 238.233 m como se observa en la Figura 2.3.

Tabla 1-3: Distancias de cobertura.

N°	DISTANCIA (m)	Caracteres enviados	ESTADO
1	2.5 m	SOS	Sin Pérdidas
2	5 m	SOS	Sin Pérdidas
3	7.5 m	SOS	Sin Pérdidas
4	10 m	SOS	Sin Pérdidas
5	15 m	SOS	Sin Pérdidas
6	20 m	SOS	Sin Pérdidas
7	25 m	SOS	Sin Pérdidas
8	30 m	SOS	Sin Pérdidas

9	40 m	SOS	Sin Pérdidas
10	55 m	SOS	Sin Perdidas
11	65 m	SOS	Sin Pérdidas
12	75 m	SOS	Sin Pérdidas
13	85 m	SOS	Sin Pérdidas
14	95 m	SOS	Sin Pérdidas
15	100 m	SOS	Sin Pérdidas
16	115 m	SOS	Sin Pérdidas
17	125 m	SOS	Sin Pérdidas
18	130 m	SOS	Sin Pérdidas
19	145 m	SOS	Sin Pérdidas
20	155 m	SOS	Sin Pérdidas
21	165 m	SOS	Sin Pérdidas
22	175 m	SOS	Sin Pérdidas
23	185 m	SOS	Sin Pérdidas
24	200 m	SOS	Sin Pérdidas
25	210 m	SOS	Sin Pérdidas
26	220 m	SOS	Sin Pérdidas
27	238 m	SOS	Sin Pérdidas
28	238.5 m	SOS	Pérdidas

Elaborado por: CARRILLO, Mayra 2018

De la toma de 28 datos de la Tabla 1-3 se observa que en los 238 m si recibe el mensaje, mientras que en los 238.5 m no recibe el mensaje, por lo que se determina que la máxima distancia de cobertura del prototipo es de 238 m en línea de vista sin obstáculos.

3.1.2 *Distancia de transmisión de datos con obstáculos.*

Esta prueba consiste en determinar la distancia de cobertura que alcanza el dispositivo implementado con obstáculos, se envía 7 caracteres de la palabra Auxilio que se muestra en la Figura 4-3.

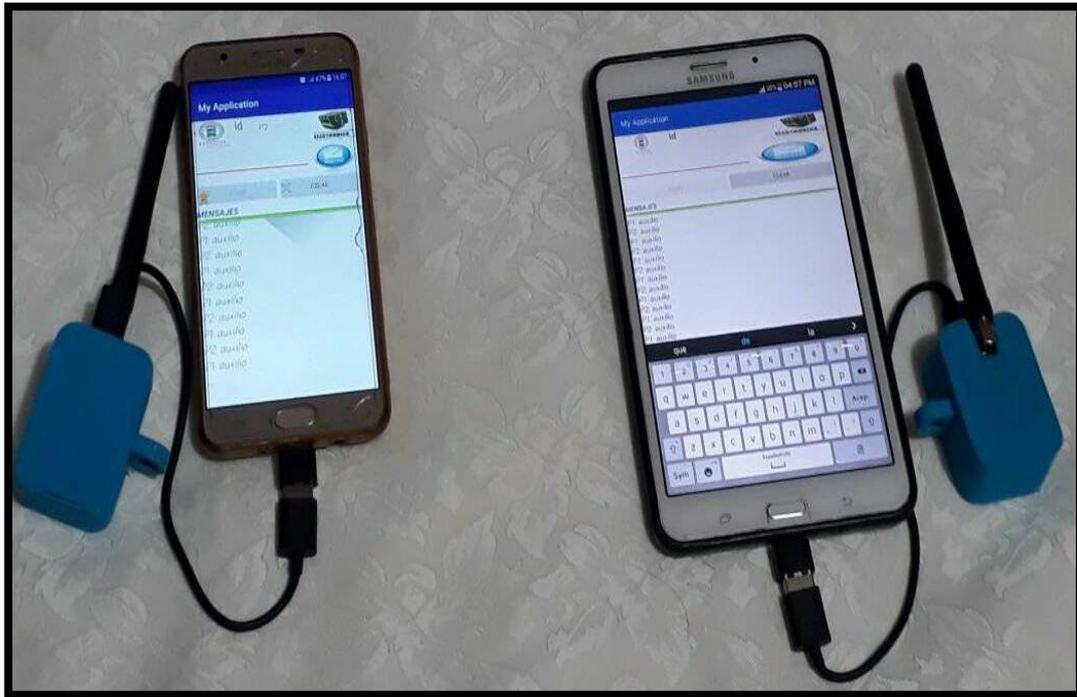


Figura 4-3: Caracteres enviados

Elaborado por: CARRILLO, Mayra 2018

Se toma distancias con rangos de 5 metros, para determinar la distancia de cobertura con obstáculos. En la Tabla 2-3 se observa los intervalos de la distancia tomados.

Tabla 2-3: Distancias en intervalos de 5 metros

N°	Distancia (m)	Caracteres	Estado
1	5 m	Auxilio	Sin pérdidas
2	10 m	Auxilio	Sin pérdidas
3	15 m	Auxilio	Sin pérdidas
4	18 m	Auxilio	Sin pérdidas
5	18.5 m	Auxilio	Pérdidas

Elaborado por: CARRILLO, Mayra 2018

De la toma de 5 muestras variando las distancias se obtiene como resultado que la distancia de cobertura con obstáculos y sin pérdida de los 7 caracteres enviaos es de 18 metros.

3.1.3 *Tiempo de transmisión de datos del módulo de comunicación sin obstáculos.*

Para realizar las mediciones de tiempo de transmisión de datos se lo hace en unidades de tiempo tomando en cuenta la distancia y sin obstáculos. En la Tabla 3-3 se muestra las mediciones del tiempo de la transmisión de datos.

Tabla 3-3: Mediciones de tiempo de la transmisión de datos

N°	DISTANCIA (m)	TIEMPO TX (s)	TIEMPO RX (s)
1	2.5 m	1s	1s
2	5 m	1s	1s
3	7.5 m	1s	1s
4	10 m	1s	1s
5	15 m	1s	1s
6	20 m	1s	1s
7	25 m	1s	1s
8	30 m	1s	1s
9	40 m	1s	1s
10	55 m	1s	1s
11	65 m	1s	1s
12	75 m	1s	1s
13	85 m	1s	1s
14	95 m	1s	1s
15	100 m	1s	1s
16	115 m	1s	1s
17	125 m	1s	1s
18	130 m	1s	1s
19	145 m	1s	1s
20	155 m	1s	1s
21	165 m	1s	1s
22	175 m	1s	1s
23	185 m	1s	1s
24	200 m	1s	1s
25	210 m	1s	1s
26	220 m	1s	1s
27	238 m	1s	1s

Elaborado por: CARRILLO, Mayra. 2018

Como se puede apreciar en la Tabla 1-3 de la toma de 27 datos, se obtiene como resultado que el tiempo de transmisión y recepción de datos es de un segundo el cual no varía con la distancia.

En la Figura 5-3 se aprecia una de las pruebas realizadas con dos personas estableciendo conversación con el prototipo implementado, a una distancia de 2 metros.

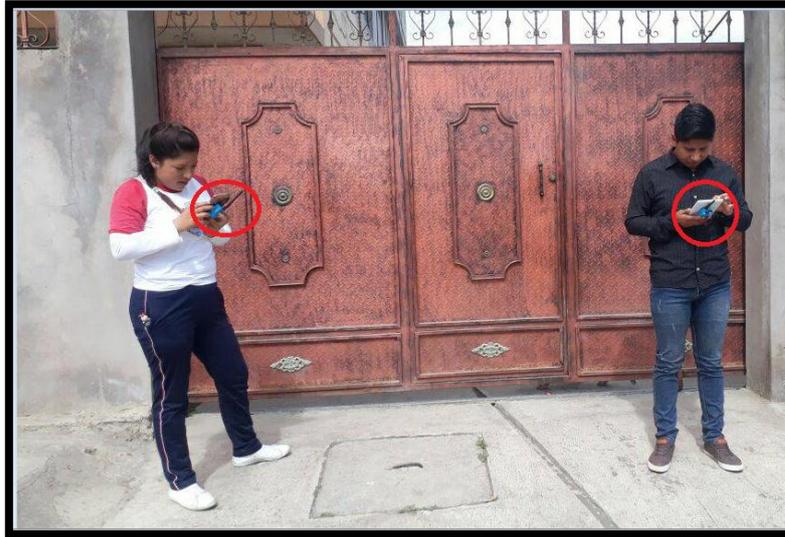


Figura 5-3: Pruebas de funcionamiento del Dispositivo sin obstáculos.

Elaborado por: CARRILLO, Mayra 2018

3.1.4 *Tiempo de transmisión de datos del módulo de comunicación con obstáculos.*

Las mediciones de tiempo de la comunicación inalámbrica se lo realizaron en segundos, tomando en cuenta la distancia y los obstáculos (paredes). En la Tabla 4-3 se muestra el tiempo de transmisión y recepción de datos, empleado por el prototipo implementado.

Tabla 4-3: Mediciones de tiempo de la transmisión de datos

N°	Distancia (m)	Tiempo Tx (s)	Tiempo Rx (s)
1	5 m	1 s	1 s
2	10 m	1 s	1 s
3	15 m	1 s	1 s
4	18 m	1 s	1 s

Elaborado por: CARRILLO, Mayra 2018

En la Tabla 4-3 se aprecia que de un total de 4 datos tomados con diferentes distancias, la comunicación con obstáculos alcanza 18 m con un tiempo de transmisión y recepción de mensajes de 1 segundo sin influir la distancia entre dispositivos.

En la Figura 6-3 se observa parte de las pruebas de comunicación con obstáculos realizadas utilizando el dispositivo a una distancia de 5 m.



Figura 6-3: Transmisión de datos con obstáculos

Elaborado por: CARRILLO, Mayra 2018

3.1.5 *Número máximo de caracteres enviados a diferentes distancias sin obstáculos.*

Se toma distintos intervalos de distancias para realizar la contabilización de caracteres que se pueden enviar teniendo en cuenta el tiempo y la distancia sin obstáculos. En la Figura 7-3 se observa el envío de caracteres de las palabras ayuda, ayuda por favor y auxilio ayúdame por favor.

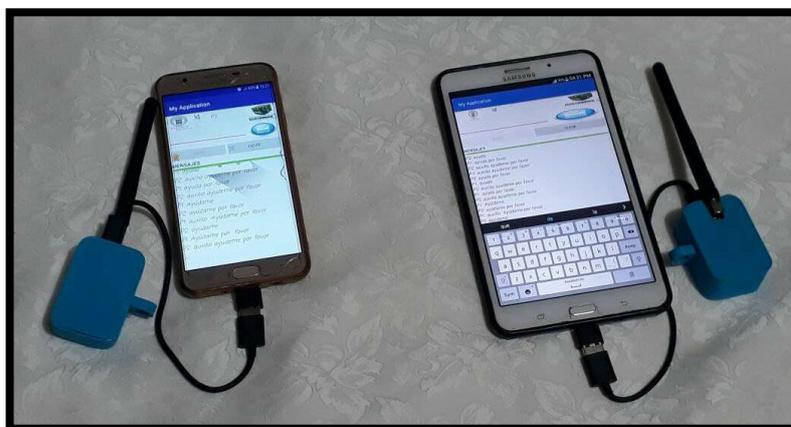


Figura 7-3: Contabilización de caracteres

Elaborado por: CARRILLO, Mayra 2018

En la Tabla 5-3 de una muestra de 27 datos se aprecia que se pueden enviar máximo 21 caracteres en todas las distancias tomadas desde 1 m hasta 238m en diferentes intervalos de distancia, con un tiempo de transmisión y recepción de un segundo teniendo una respuesta inmediata.

Tabla 5-3: Número de caracteres enviados

N°	Distancia	5 Caracteres		10 Caracteres		21 Caracteres	
		Tx	Rx	Tx	Rx	Tx	Rx
1	2.5 m	1s	1s	1s	1s	1s	1s
2	5 m	1s	1s	1s	1s	1s	1s
3	7.5 m	1s	1s	1s	1s	1s	1s
4	10 m	1s	1s	1s	1s	1s	1s
5	15 m	1s	1s	1s	1s	1s	1s
6	20 m	1s	1s	1s	1s	1s	1s
7	25 m	1s	1s	1s	1s	1s	1s
8	30 m	1s	1s	1s	1s	1s	1s
9	40 m	1s	1s	1s	1s	1s	1s
10	55 m	1s	1s	1s	1s	1s	1s
11	65 m	1s	1s	1s	1s	1s	1s
12	75 m	1s	1s	1s	1s	1s	1s
13	85 m	1s	1s	1s	1s	1s	1s
14	95 m	1s	1s	1s	1s	1s	1s
15	100 m	1s	1s	1s	1s	1s	1s
16	115 m	1s	1s	1s	1s	1s	1s
17	125 m	1s	1s	1s	1s	1s	1s
18	130 m	1s	1s	1s	1s	1s	1s
19	145 m	1s	1s	1s	1s	1s	1s
20	155 m	1s	1s	1s	1s	1s	1s
21	165 m	1s	1s	1s	1s	1s	1s
22	175 m	1s	1s	1s	1s	1s	1s
23	185 m	1s	1s	1s	1s	1s	1s
24	200 m	1s	1s	1s	1s	1s	1s
25	210 m	1s	1s	1s	1s	1s	1s
26	220 m	1s	1s	1s	1s	1s	1s
27	238 m	1s	1s	1s	1s	1s	1s

Elaborado por: CARRILLO, Mayra 2018

En la Figura 8-3 se aprecia 21 caracteres máximos enviados y recibidos en un tiempo de emisión y recepción de un segundo.

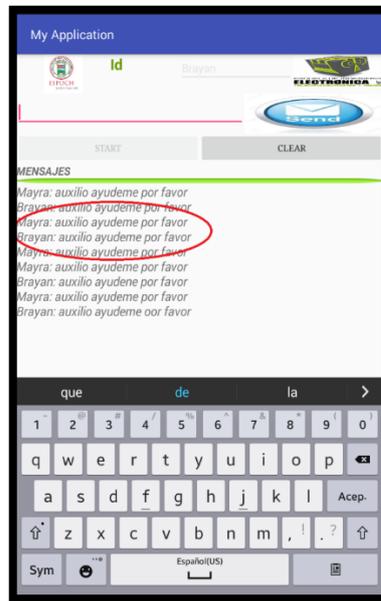


Figura 8-3: Número máximo de caracteres

Elaborado por: CARRILLO, Mayra 2018

3.1.6 Número máximo de caracteres enviados a diferentes distancias con obstáculos

Se envía diferentes números de caracteres tomando en cuenta el tiempo de transmisión, recepción y la distancia con obstáculos. En la Tabla 6-3 se muestra los tiempos de retardo en función al tamaño de la cadena transmitida.

Tabla 6-3: Envío de diferentes números de caracteres.

N°	Distancia	5 Caracteres		10 Caracteres		21 Caracteres	
		Tx	Rx	Tx	Rx	Tx	Rx
1	5 m	1 s	1 s	1 s	1 s	1 s	1 s
2	10 m	1 s	1 s	1 s	1 s	1 s	1 s
3	15 m	1 s	1 s	1 s	1 s	1 s	1 s
4	18 m	1 s	1 s	1 s	1 s	1 s	1 s

Elaborado por: CARRILLO, Mayra 2018

De la Tabla 6-3 se obtiene como resultado que se puede enviar 21 caracteres hasta una distancia de 18m, con un tiempo de respuesta de un segundo tanto en transmisión y recepción de caracteres cuando se presenta obstáculos entre el transmisor y el receptor.

3.1.7 Repetitividad del envío de mensajes

Esta prueba consiste en enviar mensajes en un lapso de tiempo de unos 10 minutos en intervalos de 10 segundos. El envío del mensaje en dos dispositivos móviles de diferentes características: un celular Samsung J7 Prime y una Tablet Samsung Galaxy. En la Figura 9 -3 se aprecia los datos recibidos en los dispositivos móviles (celular y Tablet) apreciándose similitud en los mismos.

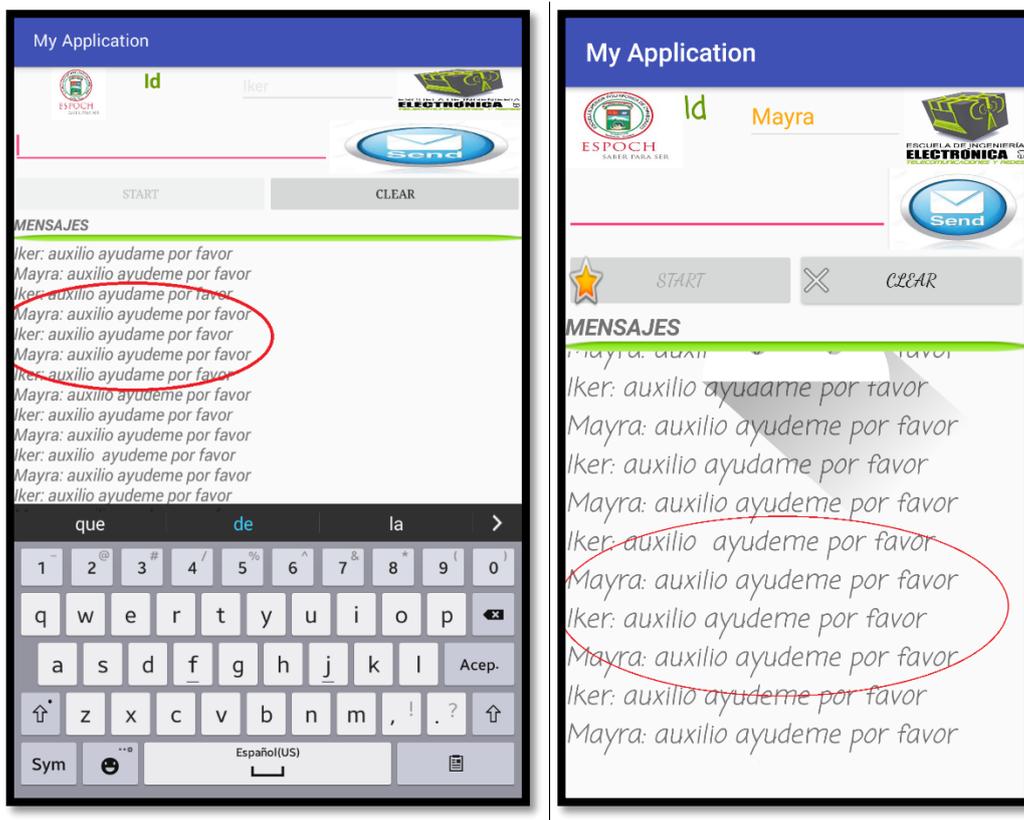


Figura 9-3: Similitud de datos recibidos en los dispositivos móviles.

Elaborado por: CARRILLO; Mayra 2018

En la Tabla 7-3 se observa que de la toma de 28 mensajes enviados, llegan 21 caracteres del celular a la Tablet y viceversa en un intervalo de tiempo de un segundo tanto en emisión como en la recepción.

Tabla 7-3: Envío de caracteres

N°	Tiempo de Repetitividad	N° de caracteres enviados	N° de caracteres recibidos	Tiempo	
				Tx	Rx
1	10 s	21 caracteres	21 caracteres	1s	1s
2	20 s	21 caracteres	21 caracteres	1s	1s
3	30 s	21 caracteres	21 caracteres	1s	1s
4	40 s	21 caracteres	21 caracteres	1s	1s
5	50 s	21 caracteres	21 caracteres	1s	1s
6	60 s	21 caracteres	21 caracteres	1s	1s
7	70 s	21 caracteres	21 caracteres	1s	1s
8	80 s	21 caracteres	21 caracteres	1s	1s
9	90 s	21 caracteres	21 caracteres	1s	1s
10	100 s	21 caracteres	21 caracteres	1s	1s
11	110 s	21 caracteres	21 caracteres	1s	1s
12	120 s	21 caracteres	21 caracteres	1s	1s
13	130 s	21 caracteres	21 caracteres	1s	1s
14	140 s	21 caracteres	21 caracteres	1s	1s
15	150 s	21 caracteres	21 caracteres	1s	1s
16	160 s	21 caracteres	21 caracteres	1s	1s
17	170 s	21 caracteres	21 caracteres	1s	1s
18	180 s	21 caracteres	21 caracteres	1s	1s
19	190 s	21 caracteres	21 caracteres	1s	1s
20	200 s	21 caracteres	21 caracteres	1s	1s
21	210 s	21 caracteres	21 caracteres	1s	1s
22	220 s	21 caracteres	21 caracteres	1s	1s
23	230 s	21 caracteres	21 caracteres	1s	1s
24	240 s	21 caracteres	21 caracteres	1s	1s
25	250 s	21 caracteres	21 caracteres	1s	1s
26	260 s	21 caracteres	21 caracteres	1s	1s
27	270 s	21 caracteres	21 caracteres	1s	1s
28	280 s	21 caracteres	21 caracteres	1s	1s

Elaborado por: CARRILLO, Mayra 2018

3.1.8 Consumo de energía teórico del dispositivo implementado

La tarjeta de desarrollo utilizada Arduino pro Micro tiene un consumo de energía de 150mA (Patagoniatec, 2015), el módulo NRF24L01 tiene un consumo de energía 11.3mA transmitiendo a 0dBm de potencia de salida y 12,13 mA recibiendo a una velocidad de 2Mbps (Nomada, <https://nomada-e.com/store/modulos-de-comunicacion-e-iot/136-modulo>) y el módulo asm 1177 consume 800mA (Banggood, 2016, <https://www.banggood.com/DC-DC-33V-AMS1117-33V-Power-Module-Voltage-Regulator>). En la Tabla 7-3 se observa el consumo de energía máxima en la transmisión de mensajes.

Tabla 7-3: Consumo de energía Transmitiendo mensajes

Consumo de energía	
Arduino Pro micro	150 mA
Módulo NRF24L01	11.3 mA
Módulo ASM1177	800 mA
TOTAL CONSUMO	961.3 mA

Elaborado por: CARRILLO, Mayra 2018

Como resultado del consumo de energía total del prototipo implementado es de 961.3 mA. En la Tabla 8-3 se aprecia el consumo de energía en la recepción de mensajes.

Tabla 8-3: Consumo de energía en la recepción de mensajes.

Consumo de energía	
Arduino Pro micro	150 mA
Módulo NRF24L01	12.13 mA
Módulo ASM1177	800 mA
TOTAL CONSUMO	962.13 mA

Elaborado por: CARRILLO, Mayra 2018

Para obtener el resultado del consumo total de energía del receptor, se suman las tres energías máximas de cada módulo, obteniendo un consumo total de 962.13 mA en la recepción de mensajes.

3.1.8.1 *Tiempo del consumo de energía conectado al dispositivo móvil en standby.*

Las pruebas se realizan con un celular Samsung J7 Prime el cual se le carga el 100% de la batería, usándolo habitualmente, se tiene como resultado un tiempo de duración de 25 horas. Después se conecta el dispositivo electrónico al celular móvil se obtiene como resultado que la batería se descarga a las 24 horas. En la Figura 10-3 apreciar el estado de la batería de dispositivo móvil en el proceso de pruebas realizado.

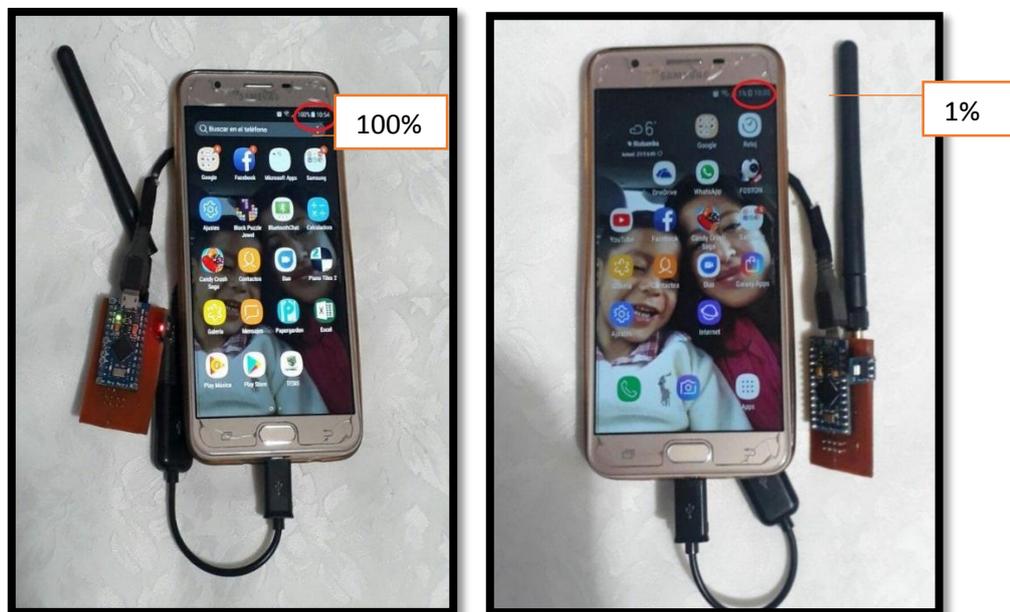


Figura 10-3: Proceso de prueba de la batería del Dispositivo móvil sin utilizar.

Elaborado por: CARRILLO, Mayra 2018

3.1.8.2 *Tiempo de consumo de energía utilizando el dispositivo electrónico regularmente*

Esta prueba consiste en utilizar la aplicación creada para mensajería instantánea junto con el dispositivo electrónico conectado al celular móvil, para determinar la duración de la batería del celular. En la Figura 11-3 se aprecia el proceso de prueba realizado con la batería completamente cargada, manteniendo un uso regular, se observó que cuando está en el 1% de la batería la aplicación se cierra automáticamente



Figura 11-3: Prueba de la batería del Dispositivo móvil con uso regular.

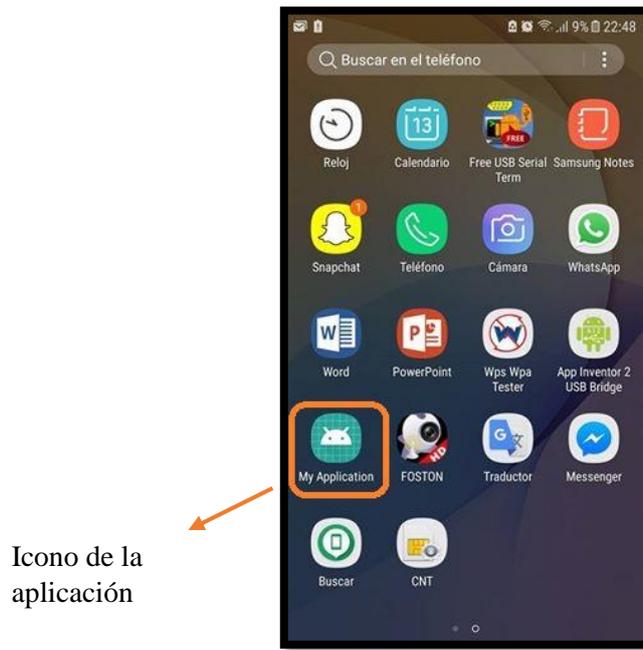
Elaborado por: CARRILLO, Mayra 2018

De la prueba realizada se concluye que apenas el 1% de la batería es consumida en 2 horas 30 min cuando se utiliza la aplicación regularmente, se determinó que el dispositivo móvil utilizado puede mantener un funcionamiento continuo de 23 horas con el prototipo implementado, una hora menos de lo que se obtiene conectado el dispositivo en standby.

3.2 Pruebas del *software* del prototipo implementado

3.2.1 *Aplicación Móvil*

La interfaz gráfica y funcionamiento de la aplicación es realizada en Android Studio 2.3.1 en la Figura 12-3 podemos observar el ícono del logo “Android” el cual permite ejecutar la aplicación desde un dispositivo móvil.

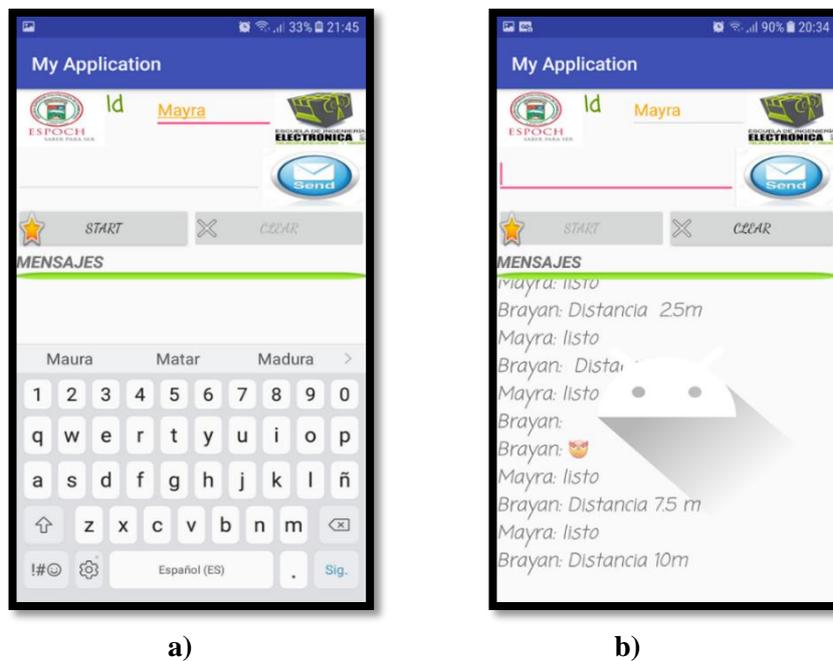


Icono de la aplicación

Figura 12-3: Icono de la aplicación móvil

Elaborado por: CARRILLO, Mayra 2018

Una vez que se abre la aplicación, se ingresa la identificación o nombre de usuario como observamos en la Figura 13a-3. Una vez ingresado el ID se da clic en el botón STAR para activar la aplicación para envío y recepción de mensajes como se aprecia en la Figura 13b-3.



a)

b)

Figura 13-3: Android a) Ingreso del ID. b) Rx y Tx de mensajes

Elaborado por: CARRILLO, Mayra 2018

3.2.2 Evaluación del funcionamiento de la aplicación en diferentes dispositivos móviles.

La instalación de la aplicación para transmisión y recepción de datos se realiza en diferentes teléfonos móviles, los cuales presentan distintas características que se detallan a continuación en la Tabla 9-3.

Tabla 9-3: Características de teléfonos móviles

	SAMSUNG GALAXY J3	SAMSUNG GALAXY PRIME J7	TABLET SAMSUNG GALAXY	HTC ONE M9
Pantalla	5"	5,5" FullHD	7"	5"
Procesador	Quad-core a 1.5GHz	Exynos 7890 de ocho núcleos	Quad-Core de 1.2 GHz.	Qualcomm Sanapdragon 810
Sistema operativo Android	5.1.1	7.0	4.4.4	5.0.2
RAM	1.5GB	3GB	1.5GB.	3GB
Memoria	8GB	16GB	8GB	32GB
Cámara	8 y 5 mpx	13 y 5 mpx	3 y 1.3 mpx	20.7 mpx
Batería	2600mAh	3300 mAh	4000mAh	2840mAh
Horas aprox. en standby	94 h	114 h	127 h	99 h

Elaborado por: CARRILLO, Mayra 2018

Como resultado de la prueba se obtiene que la aplicación no presenta problemas en su ejecución al variar las características de los dispositivos móviles en la que se ejecutara, se destaca su ejecución en diferentes versiones de Sistema operativo Android. Se restringe como elemento necesario para su funcionamiento la necesidad de permitir una comunicación OTG (Gonzáles, 2017)

3.3 Prototipo de la red de comunicación implementado.

En la Figura 14-3 se muestran las partes que conforman una red auxiliar de comunicación por radiofrecuencia. Se probó su funcionamiento y se comprueba que la red auxiliar se desempeña correctamente y cumple con los requerimientos establecidos para la implementación.

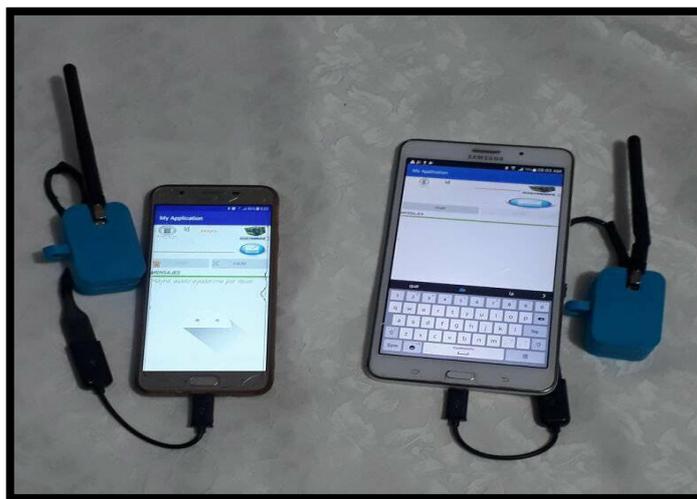


Figura 14-3: Red de comunicación implementada.

Elaborado por: CARRILLO, Mayra 2018

3.4 Presupuesto económico del prototipo

El presupuesto económico total del prototipo implementado tanto de hardware y software se tiene en la siguiente Tabla 10-3

Tabla 10-3: Costo del prototipo implementado.

	DISPOSITIVO	CANT.	COSTO UNITARIO	COSTO TOTAL
Hardware	Tarjeta Arduino Pro Micro	2	13,00	26,00
	Módulo NRF24L01	2	18,00	36,00
	Regulador de voltaje de 3.3 V	2	2,00	4,00
	Cables	2	1,50	3,00
	Adaptador USB macho a micro USB	2	1,50	3,00
	Varios	1	40,00	40,00
Total				112,00

Elaborado por: CARRILLO, Mayra 2018

En la Tabla 10-3 se puede apreciar el costo del *hardware* del prototipo implementado que alcanza 112,00 dólares norteamericanos. Se determina que la implementación del prototipo es de bajo costo, ya que constituye un 60% del valor que presenta un dispositivo móvil de gama media. Valor que puede disminuir si se realiza una producción en serie y se encuentra acorde con los requerimientos planteados al inicio de la investigación.

CONCLUSIONES

1. Se implemento un prototipo de una red auxiliar de comunicación por radiofrecuencia basado en dispositivos electrónicos y equipos móviles para mensajería instantánea incidió positivamente ya que no utilizamos una infraestructura para la comunicación inalámbrica, ni redes Wifi
2. Después de la comparativa y análisis de las tecnologías existentes para la implementación de la red auxiliar de comunicación, se eligió la comunicación por radiofrecuencia ya que trabaja en la frecuencia de 3Hz hasta los 300 GHz la cual no necesita infraestructura para realizar la comunicación y se ajusta a los requerimientos de la implementación del prototipo.
3. Como elemento de procesamiento del módulo se seleccionó la tarjeta de desarrollo Arduino Pro-micro por su tamaño reducido y facilidad de manejo mientras que para la comunicación inalámbrica se escogió un transceptor NRF24L01 debido a que tiene un consumo de energía de 11.3 en transmisión y 12.13 en recepción.
4. La aplicación diseñada se lo realizó en Android Studio 2.3.1, la cual permite la transmisión y recepción de mensajes, por medio de radiofrecuencia al ser conectado por el puerto serie del dispositivo móvil al prototipo implementado, por medio de un cable OTG.
5. Las pruebas de comunicación permiten determinar que la transmisión y recepción de mensajes sin pérdida entre dispositivos móviles conectados al prototipo implementado es 238 m a línea de vista, mientras que con obstáculos alcanza una distancia de 18 m.
6. De las pruebas se determinó que el tiempo de transmisión y recepción entre los dispositivos móviles conectados con el prototipo implementado es de un segundo independientemente del número de caracteres que tenga el mensaje. Constituyéndose un tiempo de respuesta óptimo para cumplir el principal objetivo planteado en la investigación.
7. De las pruebas de comunicación realizadas se obtiene que se puede enviar hasta 21 caracteres a una distancia de 238 m en línea de vista y a 18 metros cuando existen obstáculos. Distancias de comunicación aceptable en caso de requerir localizar a una persona atrapada en eventos de emergencia.

8. La alimentación del prototipo se lo realiza con la batería del dispositivo móvil al cual se incorpora, mediante la conexión serial. De las pruebas se determinó que el Prototipo incorporado al dispositivo móvil puede mantener hasta 24h de funcionamiento en reposo y 23 h de uso constante, antes de agotar los recursos de la batería y que al llegar al 1% de su capacidad la aplicación implementada se cierra y apaga el dispositivo.

RECOMENDACIONES

1. Se recomienda realizar estudios que permitan incorporar en los dispositivos móviles la transmisión y recepción de datos por radio frecuencia y a distancias mayores a las obtenidas en la presente investigación.
2. Es importante estudiar la posibilidad de ampliar la funcionalidad de la aplicación móvil desarrollada, como el envío y recepción de mensajes de voz, imágenes, signos vitales y coordenadas de posicionamiento.
3. Se recomienda estudiar la posibilidad de convertir a los dispositivos móviles que integren la red, como repetidoras de señal para ampliar los rangos de cobertura y comunicación de la red.
4. Se recomienda estudiar la posibilidad de buscar el financiamiento y auspicio para implementar el sistema propuesto a nivel del país, considerando la vulnerabilidad que existe en nuestro medio frente a desastres naturales.

BIBLIOGRAFÍA

ALCALDE, A. *Programación Android*. 2017. [En línea]. Disponible en: <https://elbauldelprogramador.com/programacion-android-intents-conceptos/>. [Último acceso: 3 Marzo 2018].

AMANGANDI, J. *Arduino*. 2012. [En línea]. Disponible en: <http://jamangandi2012.blogspot.com/2012/10/que-es-arduino-te-lo-mostramos-en-un.html> [Último acceso: 11 Enero 2018].

AMS1117-3.3 3.3V DC-DC Step-Down Power Module Buck Module LDO 800MA. 2016. [En línea]. Disponible en: <https://md.bayshop.com/en/ebay-de/product-381599529100-5pcs-ams1117-3-3-3-3v-dc-dc-step-down-power-module-buck-module-ldo-800ma>. [Último acceso: 10 Enero 2018].

ANDROID. *Características*. 2013. [En línea]. Disponible en: <https://stuff.mit.edu/afs/sipb/project/android/docs/reference/android/text/method/ScrollingMovementMethod.html>. [Último acceso: 3 Marzo 2018].

ANDROID STUDIO. *Android Studio*. 2016. [En línea]. Disponible en: <https://developer.android.com/studio/index.html?hl=es-419>. [Último acceso: 06 12 2017].

Antenas WiFi (blog). 2011 [En línea] Disponible en: <http://elabcfuncional.blogspot.com/>. [Último acceso: 26 Marzo 2011].

Android.hardware.usb.UsbDeviceConnection. [En línea] Disponible en: http://greppcode.com/file/repo1.maven.org/maven2/org.roboelectric/android-all/4.1.2_r1-roboelectric-0/android/hardware/usb/UsbDeviceConnection.java?av=h. [Último acceso: 3 Marzo 2018].

Android.Content.IntentFilter Class. [En línea]. Disponible en: <https://developer.xamarin.com/api/type/Android.Content.IntentFilter/>. [Último acceso: 3 Marzo 2018].

ARDUINO GENUINO. *Arduino Software (IDE)*. [En línea]. Disponible: www.arduino.cc/en/Guide/Environment. [Último acceso: 01 Marzo 2018].

ARDUINO. *Biblioteca SPI*. [En línea] .Disponible en: <https://www.arduino.cc/en/Reference/SPI> [Último acceso: 4 Marzo 2018].

Arduino Pro Micro. 2015. [En línea]
Disponible en: <http://saber.patagoniatec.com/arduino-pro-micro-atmega32u4-arduino-argentina-ptec-leonardo/>. [Último acceso: 4 Marzo 2018].

Arduino Pro Micro (Atmega32u4). 2016. [En línea]
Available at: <http://saber.patagoniatec.com/arduino-pro-micro-atmega32u4-arduino-argentina-ptec-leonardo/> [Último acceso: 05 Enero 2018].

BRICEÑO, J. *Transmisión de datos*. Tercera ed. Mérida, Venezuela. Edición Digital 2005, pp 155-160.

CARMONA, A. M. *Programación Android "Paso a Paso para principiantes"*. [En línea]. RTA 2013. Disponible en: <http://online.fliphtml5.com/afhl/yhzj/#p=1> [Último acceso: 10 Diciembre 2017]

CERVANTES, J. & DELGADO, E. *Transmisión de datos por RF*. 2010. Volumen IV. pp 178-179

CHALÁ, D. X. G. “*Diseño y construcción de un Sistema de control de luces para una casa utilizando la placa raspberry.*” 2014.

Dc/dc 3.3v ams1117-3.3v regulador del voltaje del módulo del suministro de energía. 2016. [En línea] Disponible en: <https://www.banggood.com/es/DC-DC-33V-AMS1117-33V-Power-Supply-Module-Voltage-Regulator-p-87945.html>. [Último acceso: 4 Marzo 2018].

ECURED. *Teléfono Celular*. 2016. [En línea]. Disponible en: https://www.ecured.cu/Tel%C3%A9fono_celular. [Último acceso: 26 Enero 2018].

ECURED, *Solidwork*. [En línea]. Disponible en: <https://www.ecured.cu/SolidWork> [Último acceso: 4 Marzo 2018].

EditText Tutorial With Example In Android Studio: Input Field. [En línea] Disponible en: <http://abhiandroid.com/ui/edittext>[Último acceso: 4 Marzo 2018].

El ABC de la redes inalámbricas. [En línea]. Disponible en: <http://www.eveliux.com/mx/El-ABC-de-las-redes-inalambricas-WLANs.html> [Último acceso: 24 de Octubre 2017].

ELECTRONILAB. *AMS1117-3.3 – Regulador SMD de 3.3 V SOT-223*. 2015. [En línea]. Disponible en: <https://electronilab.co/tienda/ams1117-3-3-regulador-smd-de-3-3-v-sot-223/>. [Último acceso: 10 Enero 2018].

ENCONTEXTO. Terremoto: la semana que sumió en la tristeza a todo el Ecuador. *Encontexto*, Volumen 68. 2016. pp. 29-32.

Felhr85. [En línea]. Disponible en: <https://felhr85.net/2014/11/11/usbserial-a-serial-port-driver-library-for-android-v2-0/>. [Último acceso: 2 Marzo 2018].

FERNÁNDEZ, R. *Redes inalámbricas de sensores: teoría y aplicación práctica*. España: Universidad de la Rioja. 2009

FRITZING. *Fritzing*. 2015. [En línea]. Disponible en : <http://fritzing.org/home/>. [Último acceso: 27 Febrero 2018].

GITHUB. *Arduino driver for nRF24L01 2.4GHz Wireless Transceiver*. 2013 [En línea] Disponible en: <https://github.com/maniacbug/RF24> [Último acceso: 4 Marzo 2018].

GONZÁLES, J. C. *Android*. 2017. [En línea]. Disponible en: <https://www.xatakandroid.com/moviles-android/que-es-usb-otg-y-como-sacarle-todo-el-partido> [Último acceso: 05 Marzo 2018].

HASHIMI, S., KOMATINENI, S. & MACLEAN, D. *Pro Android 3*. 2011. [En línea]. Ilustrada ed. s.l.:Apress, pp 160-168. [Último acceso: 19 Diciembre 2017]. Disponible en: <https://books.google.com.ec/books?id=cU3l9z0tTa4C>

JACKSON, W., *Pro Android UI*:Apress, pp 77-78. [Último acceso: 29 Noviembre 2017] .Disponible en: <https://books.google.com.ec/books?id=OzaAgAAQBAJ&printsec=frontcover&dq=pro+android+3&hl=es-419&sa=X&ved=0ahUKEwjkwMLJt9nZAhUKZd8KHf4aDxoQ6AEIdTAJ#v=onepage&q=pro%20android%203&f=false>

LLAMAS, L. *Ingeniería, informática y diseño*. 2016. [En línea] Disponible en: <https://www.luisllamas.es/comunicacion-inalambrica-a-2-4ghz-con-arduino-y-nrf24l01/>. [Último acceso: 10 Enero 2018].

MARTÍN, Á. P., 2016. *El uso de la tecnología en el terremoto de ecuador* .[En línea] . Disponible en: <https://www.eltelegrafo.com.ec/noticias/columnistas/1/el-uso-de-la-tecnologia-en-el-terremoto-de-ecuador>. [Último acceso: 23 Enero 2018].

MATEI, A., 2017. *Arduino*. [En línea]. Disponible en: <http://www.softpedia.com/get/Programming/Other-Programming-Files/Arduino.shtml>. [Último acceso: 27 Febrero 2018].

MATURANA, J., 2013. Xataka. [En línea]. Disponible en: <https://www.xataka.com/makers/intel-galileo-placa-de-desarrollo-fruto-de-la-colaboracion-entre-intel-y-arduino> [Último acceso: 3 Octubre 2013].

MAYNÉ, J., 2009. *Estado actual de las Comunicaciones por Radio Frecuencia*. Silica

Módulo de Radiofrecuencia nRF24L01. [En línea] Disponible en: <https://nomada-e.com/store/modulos-de-comunicacion-e-iot/136-modulo-de-radiofrecuencia-nrf24l01.html>. [Último acceso: 4 Marzo 2018].

ONU. Organización de Naciones Unidas. *Centro de Noticias ONU.* 2010. [En línea]. Disponible en: <http://www.un.org/spanish/News/story.asp?newsID=17554&criteria1=ISDR#.WcwuH1v9TIU>. [Último acceso: 19 Octubre 2017].

PÉREZ, J. & MERINO, M., 2009. *Definición de bluetooth.* [En línea] Disponible en: <https://definicion.de/bluetooth/>. [Último acceso: 4 Octubre 2017].

¿Qué tarjeta de desarrollo elegir?. 2014. [En línea]. Disponible en: <http://hacedores.com/que-tarjeta-de-desarrollo-elegir-parte-1/#main-navigation>. [Último acceso: 05 Enero 2018].

Quispe, O., *Tarjetas Para Desarrollo De Hardware.* [En línea]. Disponible en: <http://www.lightpath.io/tarjetas-de-desarrollo/> [Último acceso: 16 Abril 2017].

RF24. 2012. [En línea]. Disponible en: <https://maniacbug.github.io/RF24/classRF24.html>. [Último acceso: 4 Marzo 2018].

REGUPATHY, R., *Unboxing Android USB.* 2014 [En línea]. Apress, pp 67-68. [Último acceso: 13 Noviembre 2017]. Disponible en: https://books.google.com.ec/books?id=0cuQAwAAQBAJ&pg=PA13&dq=android.hardware.usb.UsbManager.&hl=es419&sa=X&ved=0ahUKEwiyhtOK3tHZAhVBvVMKHWNNB_QQ6AEIKTAA#v=onepage&q=android.hardware.usb.UsbManager.&f=false

ROSSO, R. *Uptodown.* [En línea]. Disponible en: <https://android-studio.uptodown.com/windows>. [Último acceso: 27 Febrero 2018].

SALAS, I. *Broadcast-receiver-android.*[blog]. [En línea] Disponible en : <http://programandointentandolo.com/2014/08/broadcast-receiver-android.html> [Último acceso: 3 Marzo 2018].

SECRETRÍA DE GESTIÓN DE RIESGOS. *Terremoto 7.8°, Pedernales.* 2016.

Se usa para pasar información entre actividades y otros componentes de la aplicación. 2016. [En línea]. Disponible en: <https://stackoverflow.com/questions/4999991/what-is-a-bundle-in-an-android-application>. [Último acceso: 3 Marzo 2018].

SORIANO, A. G. *Dispositivos móviles. Revista Seguridad.* 2010,pp. 1 251 478, 1 251 477.

Transceptor NRF24L01+ Transmisor / Receptor Inalambrico de 2.4GHz c/Antena Externa Largo Alcance. [En línea]. Disponible en: <https://banshee.mx/etiquetas/rf>. [Último acceso: 10 Enero 2018].

TOMASI, W., 2003. *Sistemas de comunicaciones electrónicas*. Cuarta ed. México: Prentice hall.

VILORIA, C., CARDONA, J. & LOZANO, C. *Ingeniería y Desarrollo*. 2009. Colombia

XAMARIN. Android MediaPlayer. [En línea] Disponible en: <https://developer.xamarin.com/api/type/Android.Media.MediaPlayer/>. [Último acceso: 3 Marzo 2018].

YAAGOUBI, M. *Acceso a Internet vía WiFi WiMax...*: Leganés : Universidad Carlos III de Madrid. 2012.

ZDNet. [En línea]. Disponible en: http://www.zdnet.com/pictures/six-clicks-more_linux_single-board-computers/ [Último acceso: 2018 Marzo 05].

ANEXO B: DATASHEET MÓDULO NRF24L01

RB-Ita-47

2.4G nRF24L01 Wireless Module w / PA and LNA



Overview

The nRF24L01 module is the latest in RF modules from SparkFun. This module uses the 2.4GHz transceiver from Nordic Semiconductor, the nRF24L01+. This transceiver IC operates in the 2.4GHz band and has many new features! Take all the coolness of the nRF24L01A and add some extra pipelines, buffers, and an auto-retransmit feature - very nice!

This board features a reverse polarized SMA connector for maximum RF range. And there is the PA and LNA circuit on board, with the external antenna it can reach long distance than the one without these parts.

This module comes with the 2.4G antenna (2DB), with 250Kbps transmission rate on open air it can reach the 800-1K meters communication distance.

Note: Manufacturer now populate these boards with the nRF24L01+. The '+' version of the IC has improved range, sensitivity, and data rates. The command set is backward compatible with the original nRF24L01.

Features

- Voltage: 3-3.6V (recommended 3.3V) V
- Maximum output power: +20dBm
- Emission mode current(peak): 115mA
- Receive Mode Current(peak): 45mA
- Power-down mode current: 4.2uA
- Sensitivity 2Mbps mode in received: -92dBm

- Sensitivity 1Mbps mode in received: -95dBm
- Sensitivity 250kbps mode in received: -104dBm
- PA gain: 20DbLNA gain: 10Db
- LNA Noise figure: 2.6Db
- Antenna Gain (peak): 2Dbi
- 2MB rate (Open area): 520m
- 1MB rate (Open area): 750m
- 250kb rate (Open area): 1000m

Specification

- Module Type: Wireless
- Weight: 25.00g
- Board Size: 4.6 x 1.7 x 1.2cm
- Version: 1
- Operation Level: Digital 5V
- Power Supply: External 5V

Single chip 2.4 GHz Transceiver **nRF24L01**

FEATURES

- True single chip GFSK transceiver
- Complete OSI Link Layer in hardware
- Enhanced ShockBurst™
- Auto ACK & retransmit
- Address and CRC computation
- On the air data rate 1 or 2Mbps
- Digital interface (SPI) speed 0-8 Mbps
- 125 RF channel operation
- Short switching time enable frequency hopping
- Fully RF compatible with nRF24XX
- 5V tolerant signal input pads
- 20-pin package (QFN20 4x4mm)
- Uses ultra low cost +/- 60 ppm crystal
- Uses low cost chip inductors and 2-layer PCB
- Power supply range: 1.9 to 3.6 V

APPLICATIONS

- Wireless mouse, keyboard, joystick
- Keyless entry
- Wireless data communication
- Alarm and security systems
- Home automation
- Surveillance
- Automotive
- Telemetry
- Intelligent sports equipment
- Industrial sensors
- Toys

GENERAL DESCRIPTION

nRF24L01 is a single chip radio transceiver for the world wide 2.4 - 2.5 GHz ISM band. The transceiver consists of a fully integrated frequency synthesizer, a power amplifier, a crystal oscillator, a demodulator, modulator and Enhanced ShockBurst™ protocol engine. Output power, frequency channels, and protocol setup are easily programmable through a SPI interface. Current consumption is very low, only 9.0mA at an output power of -6dBm and 12.3mA in RX mode. Built-in Power Down and Standby modes makes power saving easily realizable.

QUICK REFERENCE DATA

Parameter	Value	Unit
Minimum supply voltage	1.9	V
Maximum output power	0	dBm
Maximum data rate	2000	kbps
Supply current in TX mode @ 0dBm output power	11.3	mA
Supply current in RX mode @ 2000 kbps	12.3	mA
Temperature range	-40 to +85	°C
Sensitivity @ 1000 kbps	-85	dBm
Supply current in Power Down mode	900	nA

Table 1 nRF24L01 quick reference data

PRELIMINARY PRODUCT SPECIFICATION

nRF24L01 Single Chip 2.4 GHz Radio Transceiver

Type Number	Description	Version
nRF24L01	20 pin QFN 4x4, RoHS & SS-00259 compliant	D
nRF24L01 IC	Bare Dice	D
nRF24L01-EVKIT	Evaluation kit (2 test PCB, 2 configuration PCB, SW)	1.0

Table 2 nRF24L01 ordering information

BLOCK DIAGRAM

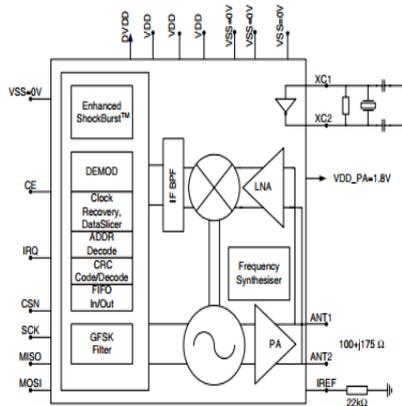


Figure 1 nRF24L01 with external components.

PIN FUNCTIONS

Pin	Name	Pin function	Description
1	CE	Digital Input	Chip Enable Activates RX or TX mode
2	CSN	Digital Input	SPI Chip Select
3	SCK	Digital Input	SPI Clock
4	MOSI	Digital Input	SPI Slave Data Input
5	MISO	Digital Output	SPI Slave Data Output, with tri-state option
6	IRQ	Digital Output	Maskable interrupt pin
7	VDD	Power	Power Supply (+3V DC)
8	VSS	Power	Ground (0V)
9	XC2	Analog Output	Crystal Pin 2
10	XC1	Analog Input	Crystal Pin 1
11	VDD_PA	Power Output	Power Supply (+1.8V) to Power Amplifier
12	ANT1	RF	Antenna interface 1
13	ANT2	RF	Antenna interface 2
14	VSS	Power	Ground (0V)
15	VDD	Power	Power Supply (+3V DC)
16	IREF	Analog Input	Reference current
17	VSS	Power	Ground (0V)
18	VDD	Power	Power Supply (+3V DC)
19	DVDD	Power Output	Positive Digital Supply output for de-coupling purposes
20	VSS	Power	Ground (0V)

Table 3 nRF24L01 pin function

PIN ASSIGNMENT

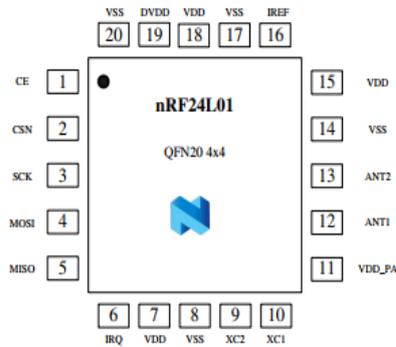


Figure 2 nRF24L01 pin assignment (top view) for a QFN20 4x4 package.

PRELIMINARY PRODUCT SPECIFICATION



nRF24L01 Single Chip 2.4 GHz Radio Transceiver

ELECTRICAL SPECIFICATIONS

Conditions: VDD = +3V, VSS = 0V, T_A = -40°C to +85°C

Symbol	Parameter (condition)	Notes	Min.	Typ.	Max.	Units
Operating conditions						
VDD	Supply voltage		1.9	3.0	3.6	V
TEMP	Operating Temperature		-40	+27	+85	°C
Digital input pin						
V _{IH}	HIGH level input voltage	¹	0.7VDD		5.25	V
V _{IL}	LOW level input voltage		VSS		0.3VDD	V
Digital output pin						
V _{OIH}	HIGH level output voltage (I _{OL} =0.25mA)		VDD-0.3		VDD	V
V _{OIL}	LOW level output voltage (I _{OL} =0.25mA)		VSS		0.3	V
General RF conditions						
f _{OP}	Operating frequency	²	2400		2525	MHz
f _{CRYSTAL}	Crystal frequency			16		MHz
M _{FM}	Frequency deviation @ 1000kbps			±160		kHz
M _{FM}	Frequency deviation @ 2000kbps			±320		kHz
R _{PACK}	Data rate ShockBurst™		>0		2000	kbps
F _{CHANNEL}	Channel spacing @ 1000kbps			1		MHz
F _{CHANNEL}	Channel spacing @ 2000kbps			2		MHz
Transmitter operation						
P _{RE}	Maximum Output Power	³		0	+4	dBm
P _{ACC}	RF Power Control Range		16	18	20	dB
P _{ACC}	RF Power Accuracy				±4	dB
P _{BW}	20dB Bandwidth for Modulated Carrier (2000kbps)			1800	2000	kHz
P _{AC1}	1 st Adjacent Channel Transmit Power 2MHz				-20	dBm
P _{AC2}	2 nd Adjacent Channel Transmit Power 4MHz				-50	dBm
I _{SD0}	Supply current @ 0dBm output power	⁴		11.3		mA
I _{SD0}	Supply current @ -18dBm output power			7.0		mA
I _{SD0}	Average Supply current @ -6dBm output power, Enhanced ShockBurst™	⁵		0.05		mA
I _{SD0}	Supply current in Standby-1 mode	⁶		32		µA
I _{SD0}	Supply current in power down			900		nA

Advanced Monolithic Systems

AMS1117

1A LOW DROPOUT VOLTAGE REGULATOR

FEATURES

- Three Terminal Adjustable or Fixed Voltages*
1.5V, 1.8V, 2.5V, 2.85V, 3.3V and 5.0V
- Output Current of 1A
- Operates Down to 1V Dropout
- Line Regulation: 0.2% Max.
- Load Regulation: 0.4% Max.
- SOT-223, TO-252 and SO-8 package available

APPLICATIONS

- High Efficiency Linear Regulators
- Post Regulators for Switching Supplies
- 5V to 3.3V Linear Regulator
- Battery Chargers
- Active SCSI Terminators
- Power Management for Notebook
- Battery Powered Instrumentation

GENERAL DESCRIPTION

The AMS1117 series of adjustable and fixed voltage regulators are designed to provide 1A output current and to operate down to 1V input-to-output differential. The dropout voltage of the device is guaranteed maximum 1.3V at maximum output current, decreasing at lower load currents.

On-chip trimming adjusts the reference voltage to 1%. Current limit is also trimmed, minimizing the stress under overload conditions on both the regulator and power source circuitry.

The AMS1117 devices are pin compatible with other three-terminal SCSI regulators and are offered in the low profile surface mount SOT-223 package, in the 8L SOIC package and in the TO-252 (DPAK) plastic package.

ORDERING INFORMATION:

PACKAGE TYPE			OPERATING JUNCTION TEMPERATURE RANGE
TO-252	SOT-223	8L SOIC	
AMS1117CD	AMS1117	AMS1117CS	-40 to 125° C
AMS1117CD-1.5	AMS1117-1.5	AMS1117CS-1.5	-40 to 125° C
AMS1117CD-1.8	AMS1117-1.8	AMS1117CS-1.8	-40 to 125° C
AMS1117CD-2.5	AMS1117-2.5	AMS1117CS-2.5	-40 to 125° C
AMS1117CD-2.85	AMS1117-2.85	AMS1117CS-2.85	-40 to 125° C
AMS1117CD-3.3	AMS1117-3.3	AMS1117CS-3.3	-40 to 125° C
AMS1117CD-5.0	AMS1117-5.0	AMS1117CS-5.0	-40 to 125° C

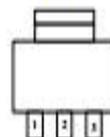
*For additional available fixed voltages contact factory.

PIN CONNECTIONS

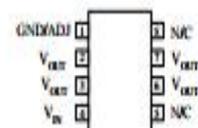
3 PIN FIXED/ADJUSTABLE
VERSION

- 1- Ground/Adjust
- 2- V_{OUT}
- 3- V_{IN}

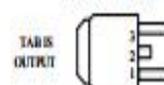
SOT-223 Top View



8L SOIC Top View



TO-252 FRONT VIEW



APPLICATION HINTS

The AMS1117 series of adjustable and fixed regulators are easy to use and are protected against short circuit and thermal overloads. Thermal protection circuitry will shut-down the regulator should the junction temperature exceed 165°C at the sense point.

Pin compatible with older three terminal adjustable regulators, these devices offer the advantage of a lower dropout voltage, more precise reference tolerance and improved reference stability with temperature.

Stability

The circuit design used in the AMS1117 series requires the use of an output capacitor as part of the device frequency compensation. The addition of 22µF solid tantalum on the output will ensure stability for all operating conditions.

When the adjustment terminal is bypassed with a capacitor to improve the ripple rejection, the requirement for an output capacitor increases. The value of 22µF tantalum covers all cases of bypassing the adjustment terminal. Without bypassing the adjustment terminal smaller capacitors can be used with equally good results.

To further improve stability and transient response of these devices larger values of output capacitor can be used.

Protection Diodes

Unlike older regulators, the AMS1117 family does not need any protection diodes between the adjustment pin and the output and from the output to the input to prevent over-stressing the die. Internal resistors are limiting the internal current paths on the AMS1117 adjustment pin, therefore even with capacitors on the adjustment pin no protection diode is needed to ensure device safety under short-circuit conditions.

Diodes between the input and output are not usually needed. Microsecond surge currents of 50A to 100A can be handled by the internal diode between the input and output pins of the device. In normal operations it is difficult to get those values of surge currents even with the use of large output capacitances. If high value output capacitors are used, such as 1000µF to 5000µF and the input pin is instantaneously shorted to ground, damage can occur. A diode from output to input is recommended, when a crowbar circuit at the input of the AMS1117 is used (Figure 1).

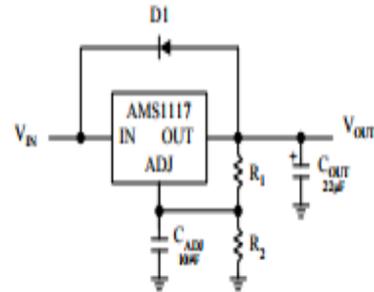


Figure 1.

Output Voltage

The AMS1117 series develops a 1.25V reference voltage between the output and the adjust terminal. Placing a resistor between these two terminals causes a constant current to flow through R1 and down through R2 to set the overall output voltage. This current is normally the specified minimum load current of 10mA. Because I_{ADJ} is very small and constant it represents a small error and it can usually be ignored.

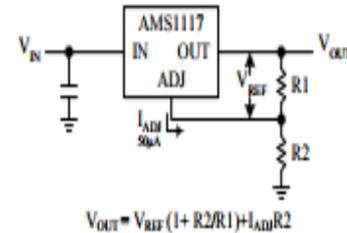


Figure 2. Basic Adjustable Regulator

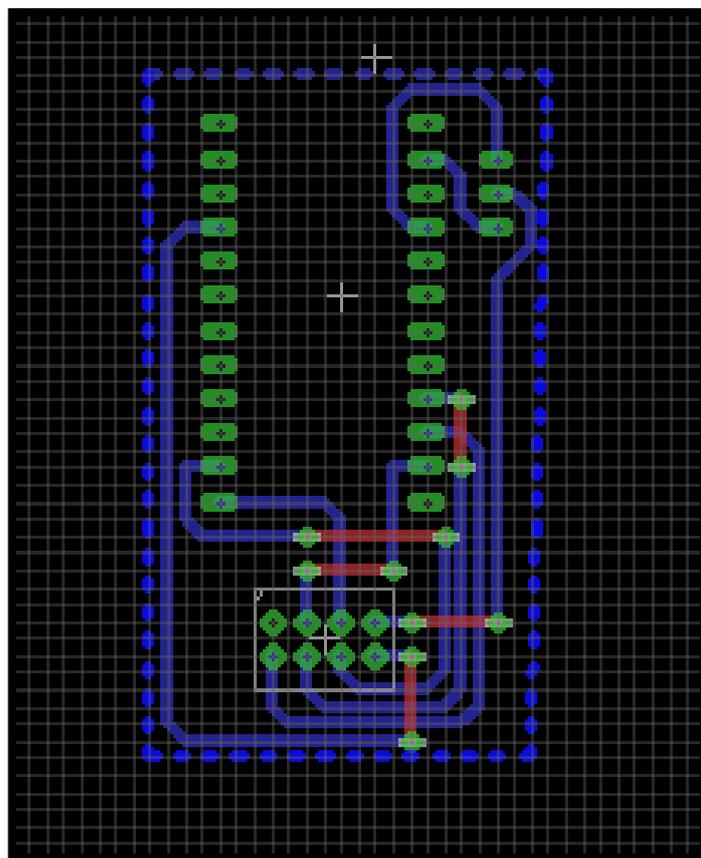
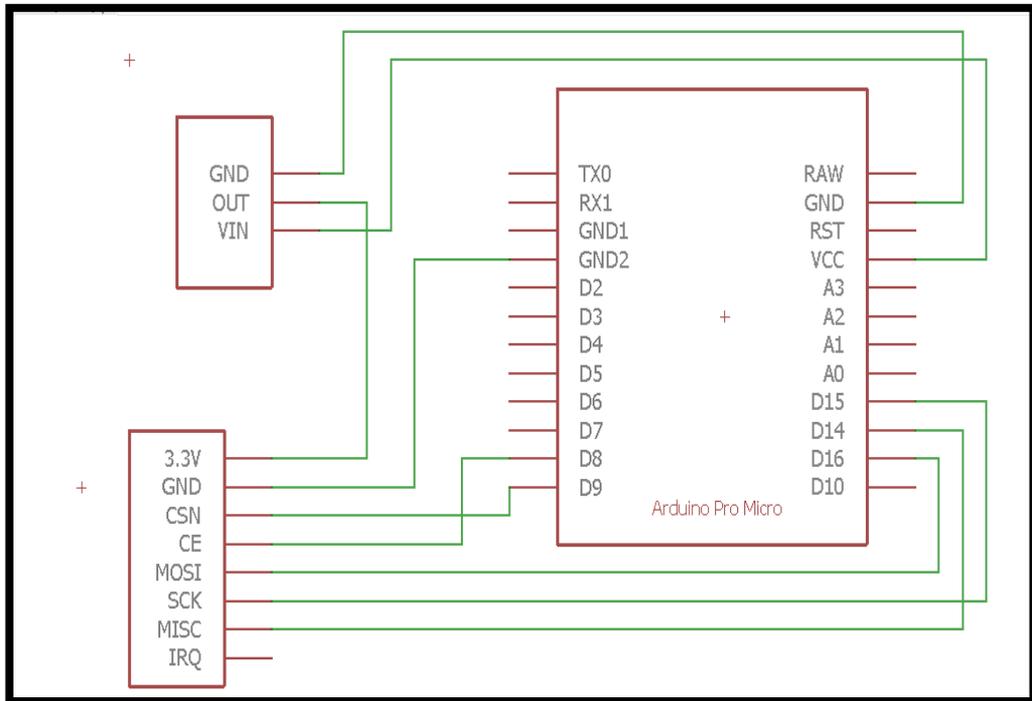
Load Regulation

True remote load sensing it is not possible to provide, because the AMS1117 is a three terminal device. The resistance of the wire connecting the regulator to the load will limit the load regulation. The data sheet specification for load regulation is measured at the bottom of the package. Negative side sensing is a true Kelvin connection, with the bottom of the output divider returned to the negative side of the load.

The best load regulation is obtained when the top of the resistor divider R1 is connected directly to the case not to the load. If R1 were connected to the load, the effective resistance between the regulator and the load would be:

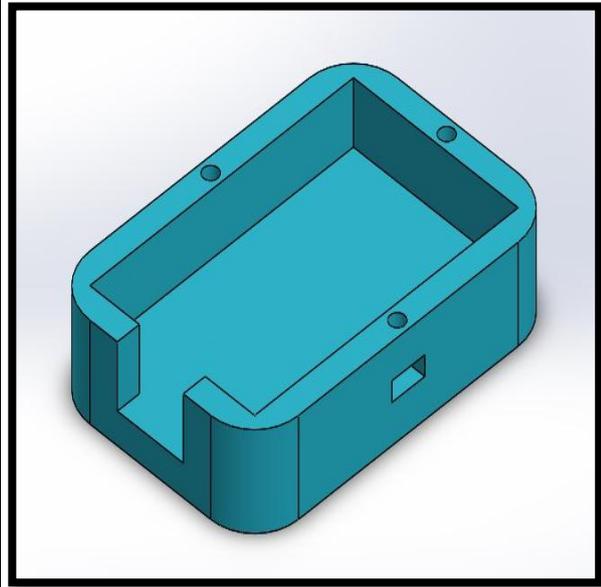
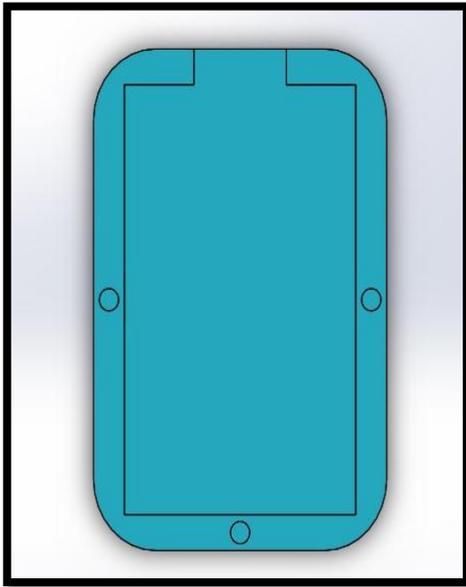
$$R_p \times \left(\frac{R2+R1}{R1} \right), \quad R_p = \text{Parasitic Line Resistance}$$

ANEXO C: DISEÑO DE LA PLACA

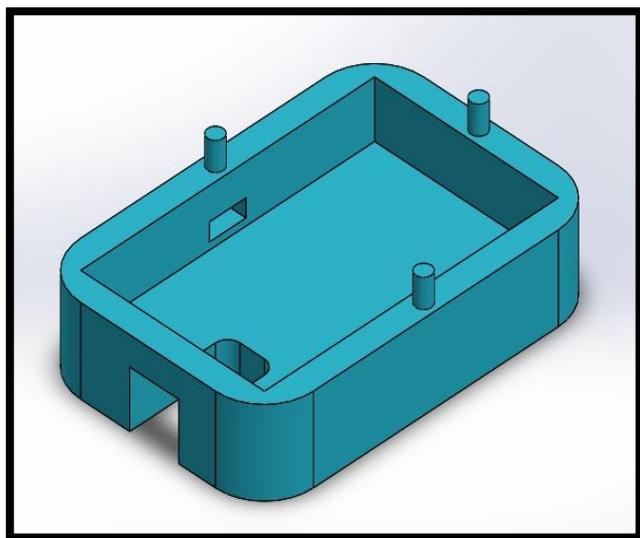
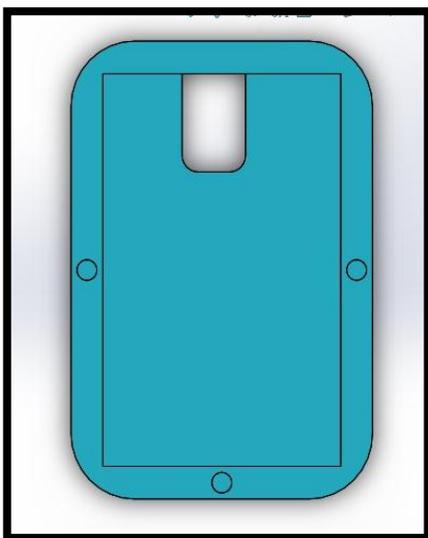


ANEXO D: DISEÑO DE LA CUBIERTA DEL DISPOSITIVO

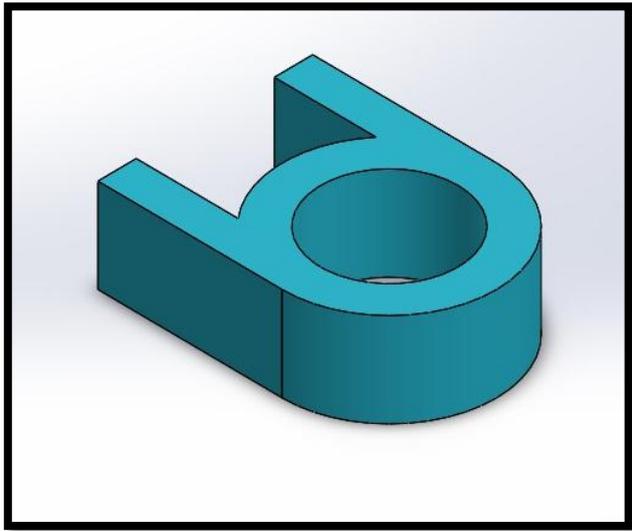
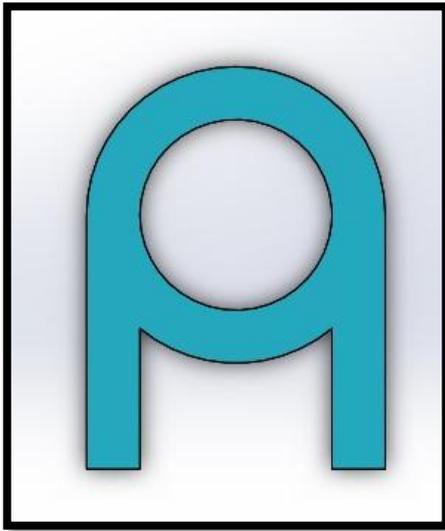
SEGMENTO 1



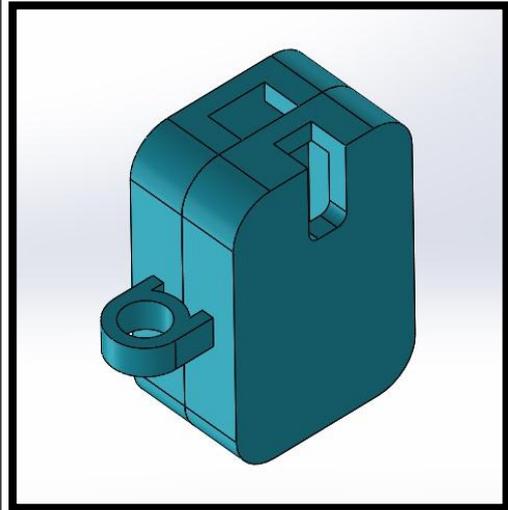
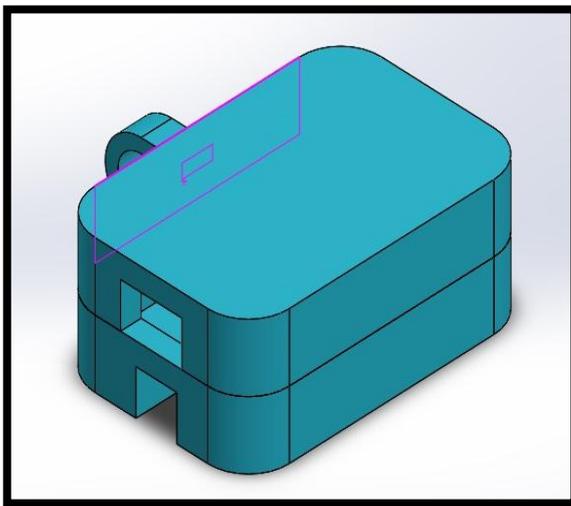
SEGMENTO 2



SEGMENTO 3



ENSAMBLAJE



ANEXO E: CÓDIGO ARDUINO EMISOR

```
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>

RF24 radio(8, 9); // CE, CNS,
const byte addresses[][6] = {"00001", "00002"};

String dato="";
String mensa="";
boolean h=false;
boolean t=false;
boolean m=false;
void setup()
{
  Serial.begin(9600);
  radio.begin();
  radio.openWritingPipe(addresses[1]); // 00001
  radio.openReadingPipe(1, addresses[0]); // 00002
  radio.setPALevel(RF24_PA_MIN);
}
void loop()
{
  delay(5);
  dato="";
  while(Serial.available())
  {
    char inChar = (char)Serial.read();
    dato=dato+inChar;

    h=true;
```

```

    }
    if(h==true)
    {
        h=false;
        radio.stopListening();
        char text[300]=""; // longitud de los caracteres 300
        dato.toCharArray(text, dato.length()+1);
        radio.write(&text, sizeof(text));
        delay(5);
    }

// recibiendo datos ////
char mess[300]="";
delay(5);
radio.startListening();
if ( radio.available())
{
    while (radio.available())
    {
        radio.read(&mess, sizeof(mess));
        t=true;
    }
    delay(5);
}
if(t==true)
{
    t=false;
    mensa = String(mess);
    Serial.println(mensa);
}
}

```

ANEXO F: CÓDIGO ARDUINO RECEPTOR

```
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>

RF24 radio(8, 9); // CE, CSN
const byte addresses[][6] = {"00001", "00002"};

String dato="";
String mensa="";
boolean h=false;
boolean t=false;
boolean m=false;

void setup()
{
  Serial.begin(9600);
  radio.begin();
  radio.openWritingPipe(addresses[0]); // 00002
  radio.openReadingPipe(1, addresses[1]); // 00001
  radio.setPALevel(RF24_PA_MIN);
}

void loop()
{
  char mess[300]="";
  delay(5);
  radio.startListening();
  if ( radio.available())
  {
    while (radio.available())
    {
      radio.read(&mess, sizeof(mess));
      t=true;
    }
  }
}
```

```
    }
    delay(5);
}

if(t==true)
{
    t=false;
    mensa = String(mess);
    Serial.println(mensa);
}
////////////////// enviando //////////////////

delay(5);
dato="";
while(Serial.available())
{
    char inChar = (char)Serial.read();
    dato=dato+inChar;

    h=true;
}

if(h==true )
{
    h=false;
    radio.stopListening();
    char text[300]="";// longitud de los caracteres 300
    dato.toCharArray(text, dato.length()+1);
    radio.write(&text, sizeof(text));
    delay(5);
}
```

ANEXO G: CÓDIGO ANDROID STUDIO

MainActivity.java

```
import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;
import android.app.PendingIntent;
import android.content.BroadcastReceiver;
import android.content.Intent;
import android.content.IntentFilter;
import android.hardware.usb.UsbDevice;
import android.hardware.usb.UsbDeviceConnection;
import android.media.MediaPlayer;
import android.text.method.ScrollingMovementMethod;
import android.view.View;
import android.content.Context;
import android.graphics.Color;
import android.hardware.usb.UsbManager;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import java.io.ByteArrayInputStream;
import java.io.IOException;
import java.io.UnsupportedEncodingException;
import java.util.HashMap;
import java.util.Map;

import com.felhr.usbserial.UsbSerialInterface;
import com.felhr.usbserial.UsbSerialDevice;

public class MainActivity extends AppCompatActivity {

    private static final String ACTION_USB_PERMISSION =
"com.android.example.USB_PERMISSION";
    UsbDevice device;
    UsbDeviceConnection connection;
    UsbManager usbManager;
    UsbSerialDevice serialPort;
    PendingIntent pendingIntent;
    EditText inputView;
    Button startButton;
    Button clearButton;
    Button sendButton;
    TextView displayView;
    EditText ide;
    MediaPlayer son;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        inputView=(EditText) findViewById(R.id.editText);
        startButton=(Button) findViewById(R.id.btn_Start);
        displayView =(TextView) findViewById(R.id.textView);
        clearButton= (Button) findViewById(R.id.btn_Clear);
        sendButton=(Button) findViewById(R.id.btn_Send);
        ide=(EditText) findViewById(R.id.txt_id);
    }
}
```

```

        son=MediaPlayer.create(this,R.raw.uu);
        displayView.setMovementMethod(new ScrollingMovementMethod());
        pendingIntent = PendingIntent.getBroadcast(this, 0, new
Intent(ACTION_USB_PERMISSION), 0);
        IntentFilter filter = new IntentFilter(ACTION_USB_PERMISSION);
        registerReceiver(broadcastReceiver, filter);
    }

//=====
=====
    public void onClickStart(View view) {

        usbManager = (UsbManager)
getSystemService(Context.USB_SERVICE);

        HashMap<String, UsbDevice> usbDevices =
usbManager.getDeviceList();
        if (!usbDevices.isEmpty()) {
            boolean keep = true;
            for (Map.Entry<String, UsbDevice> entry :
http://www.softpedia.com/get/Programming/Other-Programming-Files/Arduino.shtml usbDevices.entrySet()) {
                device = entry.getValue();
                int deviceVID = device.getVendorId();

                if (deviceVID == 1027 || deviceVID == 9025||
deviceVID==6790) { //Arduino Vendor ID
                    usbManager.requestPermission(device,
pendingIntent);
                    keep = false;
                } else {
                    connection = null;
                    device = null;
                }
                if (!keep)
                    break;
            }
        }
        displayView.setEnabled(true);
        sendButton.setEnabled(true);
        clearButton.setEnabled(true);
        inputView.setEnabled(true);
        startButton.setEnabled(false);
        ide.setEnabled(false);

    }

    public void onClickSend(View view)
    {
        String textInput = inputView.getText().toString();
        textInput=ide.getText().toString()+": "+textInput;
        serialPort.write(textInput.getBytes());

        displayView.setText(displayView.getText().toString()+textInput+"\n");
        inputView.setText("");
    }
//=====
    private final BroadcastReceiver broadcastReceiver = new
BroadcastReceiver() {
        @Override

```

```

        public void onReceive(Context context, Intent intent) {
            if (intent.getAction().equals(ACTION_USB_PERMISSION)) {
                boolean granted =
                intent.getExtras().getBoolean(UsbManager.EXTRA_PERMISSION_GRANTED);
                if (granted) {
                    connection = usbManager.openDevice(device);
                    serialPort =
                UsbSerialDevice.createUsbSerialDevice(device, connection);
                    if (serialPort != null) {
                        if (serialPort.open()) {
                            serialPort.setBaudRate(9600);

                serialPort.setDataBits(UsbSerialInterface.DATA_BITS_8);

                serialPort.setStopBits(UsbSerialInterface.STOP_BITS_1);

                serialPort.setParity(UsbSerialInterface.PARITY_NONE);

                serialPort.setFlowControl(UsbSerialInterface.FLOW_CONTROL_OFF);
                            serialPort.read(mCallback);
                        } else {
                            Log.d("SERIAL", "PORT NOT OPEN");
                        }
                    } else {
                        Log.d("SERIAL", "PORT IS NULL");
                    }
                } else {
                    Log.d("SERIAL", "PERMISSION NOT GRANTED");
                }
            } else if
            (intent.getAction().equals(UsbManager.ACTION_USB_DEVICE_ATTACHED)) {
                onClickStart(startButton);
            } else if
            (intent.getAction().equals(UsbManager.ACTION_USB_DEVICE_DETACHED)) {
                //can add something to close the connection
            }
        }
    };

    //=====
    private UsbSerialInterface.UsbReadCallback mCallback = new
    UsbSerialInterface.UsbReadCallback() {
        //Defining a Callback which triggers whenever data is read.
        @Override
        public void onReceivedData(byte[] arg0) {
            String data = null;
            try {
                son.start();
                data = new String(arg0, "UTF-8");
                data.concat("\n");
                tvAppend(displayView, data);
            } catch (UnsupportedEncodingException e) {
                e.printStackTrace();
            }
        }
    };

    private void tvAppend(final TextView tv, final CharSequence text)
    {
        runOnUiThread(new Runnable() {
            @Override public void run() {
                if (text != null) {

```

ACTIVITY_MAIN.XML

```
<?xml version="1.0" encoding="utf-8"?>

<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.carlos.myapplication.MainActivity">

    <RelativeLayout

        android:layout_width="368dp"
        android:layout_height="495dp"
        tools:layout_editor_absoluteY="8dp"
        tools:layout_editor_absoluteX="8dp">

        <LinearLayout

            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:id="@+id/linearLayout3">

            <ImageView

                android:id="@+id/imageView2"
                android:layout_width="61dp"

                android:layout_height="@android:dimen/notification_large_icon_height"
                android:layout_weight="1"
                android:background="@drawable/photo"
                android:scaleType="centerInside" />

            />

        />

    />
```

```
<TextView
    android:id="@+id/textView6"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:fontFamily="casual"
    android:foreground="@android:color/transparent"
    android:foregroundTintMode="src_over"
    android:text="Id"
    android:textSize="24sp"
    android:textStyle="bold"
    tools:background="@android:color/transparent" />
```

```
<EditText
    android:id="@+id/txt_id"
    android:layout_width="92dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:ems="10"
    android:inputType="textPersonName"

    android:textColor="?android:attr/colorPressedHighlight"
    android:textSize="18sp" />
```

```
<ImageView

    android:layout_width="@android:dimen/notification_large_icon_width"

    android:layout_height="@android:dimen/notification_large_icon_width"
    android:layout_weight="1"
    android:background="@drawable/p1"
    android:scaleType="centerInside" />
```

```
</LinearLayout>
```

```
<LinearLayout
```

```
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_alignParentStart="true"  
    android:layout_below="@+id/linearLayout3"  
    android:orientation="horizontal"  
    android:id="@+id/linearLayout4">
```

```
<EditText
```

```
    android:id="@+id/editText"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_weight="1"  
    android:ems="10"  
    android:enabled="false"  
    android:inputType="textPersonName"  
    android:textStyle="bold" />
```

```
<Button
```

```
    android:id="@+id/btn_Send"  
  
    android:layout_width="@android:dimen/notification_large_icon_height"  
    android:layout_height="@android:dimen/notification_large_icon_height"  
    android:layout_weight="1"  
    android:background="@drawable/send"  
    android:enabled="false"  
    android:fontFamily="casual"  
    android:foregroundGravity="fill"
```

```

        android:onClick="onClickSend" />
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
    android:layout_below="@+id/linearLayout4"
    android:orientation="horizontal"
    android:id="@+id/linearLayout5">

    <Button
        android:id="@+id/btn_Start"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:fontFamily="cursive"
        android:foreground="@android:drawable/btn_star_big_on"
        android:foregroundGravity="top|bottom|left|fill_vertical|center_hori
zontal|clip_vertical|clip_horizontal"
        android:onClick="onClickStart"
        android:text="Start" />

    <Button
        android:id="@+id/btn_Clear"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:enabled="false"
        android:fontFamily="cursive"

```

```
        android:foreground="@android:drawable/ic_menu_close_clear_cancel"
            android:foregroundGravity="center_vertical"
            android:onClick="onClickClear"
            android:text="Clear" />
    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_alignParentStart="true"
        android:layout_below="@+id/linearLayout5"
        android:orientation="vertical">

        <TextView
            android:id="@+id/textView10"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="MENSAJES"
            android:textSize="18sp"
            android:textStyle="bold|italic" />

        <ImageView
            android:id="@+id/imageView"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:scaleType="fitXY"

            app:srcCompat="@android:drawable/button_onoff_indicator_on" />
```

LISTA DE DISPOSITIVOS

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <!-- 0x0403 / 0x6001: FTDI FT232R UART -->
  <usb-device vendor-id="1027" product-id="24577" />

  <!-- 0x0403 / 0x6015: FTDI FT231X -->
  <usb-device vendor-id="1027" product-id="24597" />

  <!-- 0x2341 / Arduino -->
  <usb-device vendor-id="9025" />

  <!-- 0x16C0 / 0x0483: Teensyduino -->
  <usb-device vendor-id="5824" product-id="1155" />

  <!-- 0x10C4 / 0xEA60: CP210x UART Bridge -->
  <usb-device vendor-id="4292" product-id="60000" />

  <!-- 0x067B / 0x2303: Prolific PL2303 -->
  <usb-device vendor-id="1659" product-id="8963" />

  <!-- 0x1a86 / 0x7523: Qinheng CH340 -->
  <usb-device vendor-id="6790" product-id="29987" />
</resources>
```

ANDROID MANIFEST

```
<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

    package="com.example.carlos.myapplication">

    <application

        android:allowBackup="true"

        android:icon="@mipmap/ic_launcher"

        android:label="@string/app_name"

        android:roundIcon="@mipmap/ic_launcher_round"

        android:supportsRtl="true"

        android:theme="@style/AppTheme">

        <activity android:name=".MainActivity">

            <intent-filter>

                <action android:name="android.intent.action.MAIN" />

                <category

                    android:name="android.intent.category.LAUNCHER" />

            </intent-filter>

            <intent-filter>
```

```
        <action
android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED" />

    </intent-filter>

    <meta-data

android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED"

        android:resource="@xml/device_filter" />

    <intent-filter>

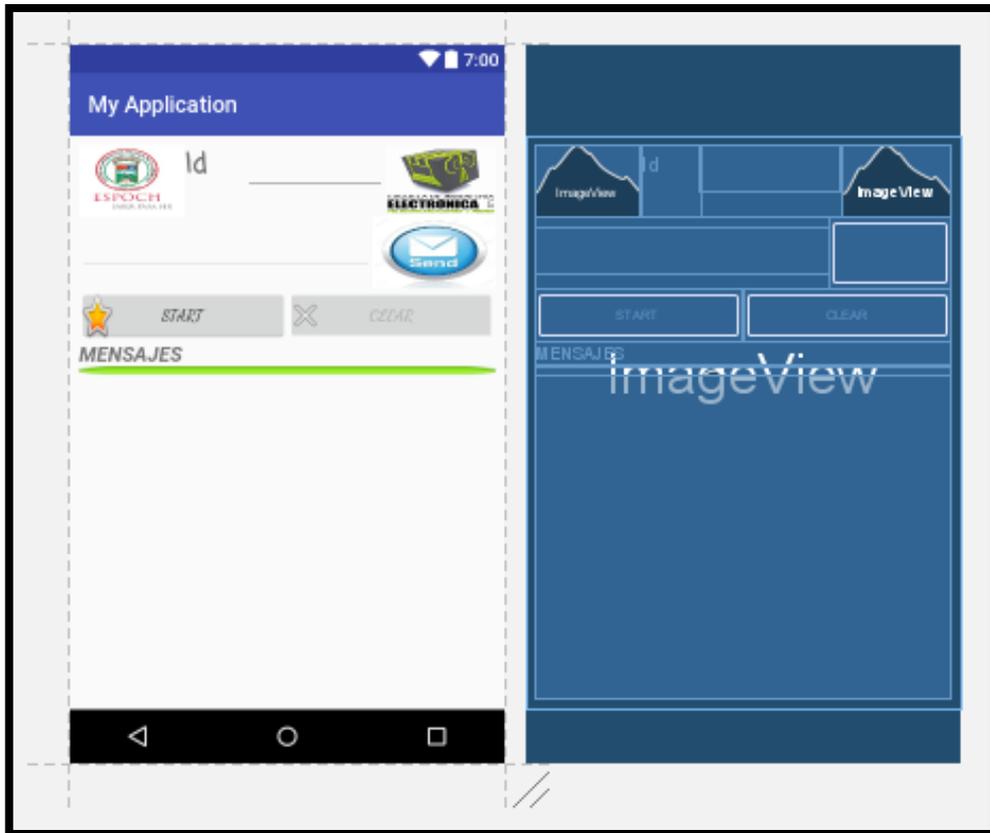
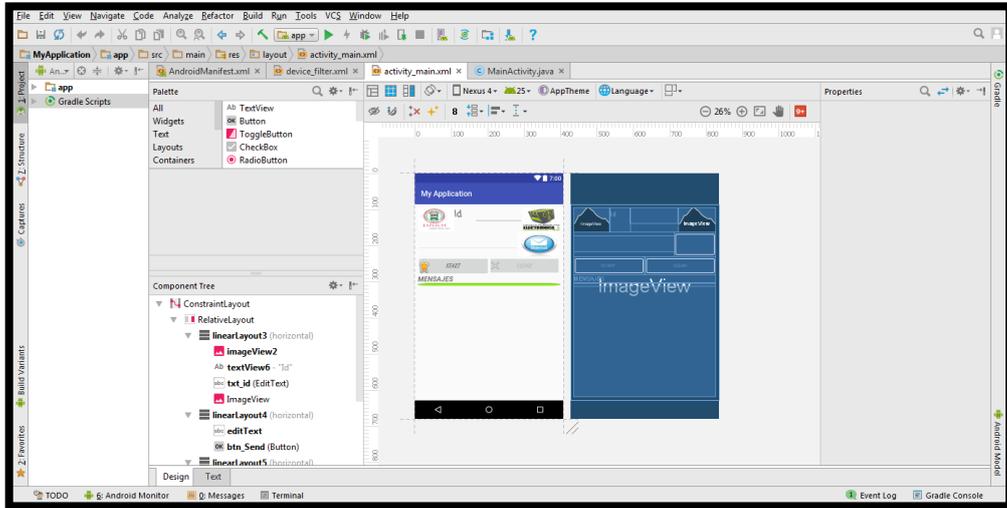
        <action
android:name="android.hardware.usb.action.USB_DEVICE_DETACHED" />

    </intent-filter>

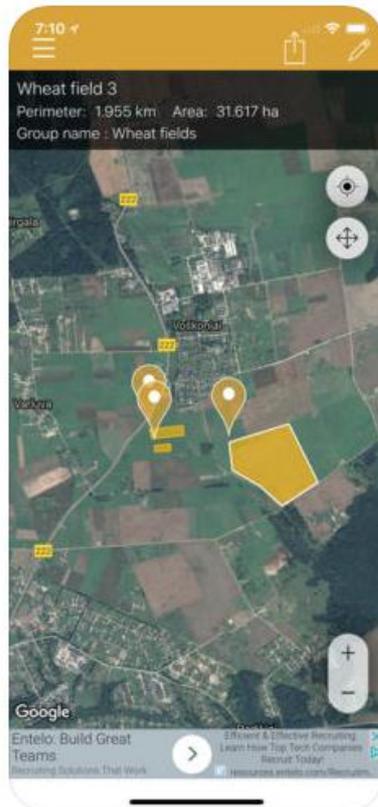
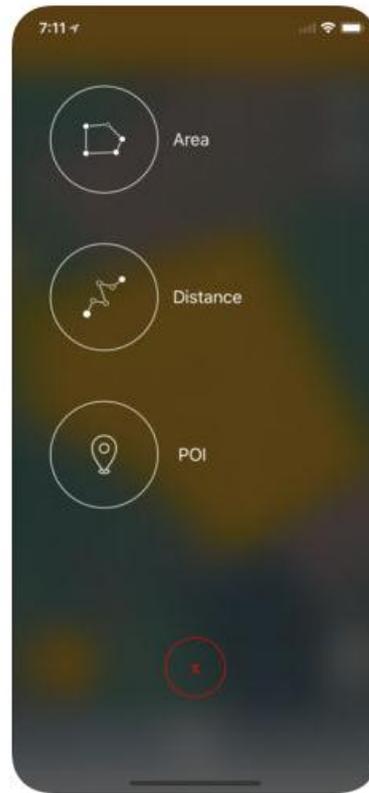
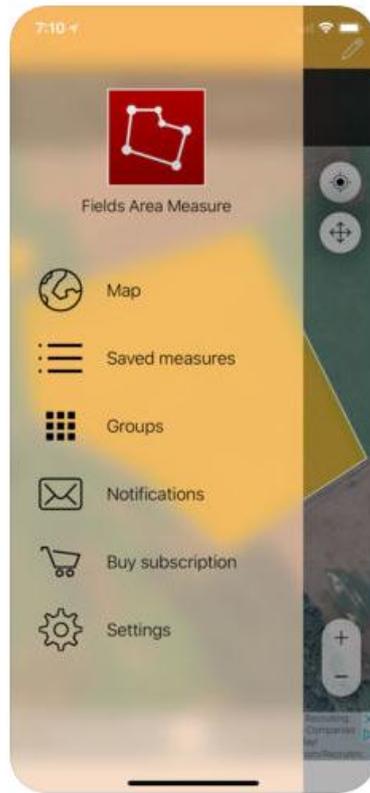
</activity>

</application>
```

ANEXO H: INTERFAZ GRÁFICA DE LA APLICACIÓN



ANEXO I: APLICACIÓN DE MEDICIÓN DE DISTANCIA



ANEXO J: PATALLAS DE COMPROBACIÓN DE ENVIO DE MENSAJES

